Deanship of Graduate Studies

Al- Quds University

Al-Farāhīdī Arabic Diacrizer System

Iyad Ahmad Abusamrah

M.Sc. Thesis

Jerusalem / Palestine

February, 2015

(Al-Farāhīdī Arabic Diacrizer System)

Prepared By:

Iyad Ahmad Abusamrah

B.Sc.: Computer Engineering, Al-Quds University, Palestine

Supervisor: Dr. Labib Arafeh

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Electronic and Computer Engineering, Faculty of Engineering.

Al- Quds University

2015

Al-Quds University

Faculty of Engineering

Electronic and Computer Engineering Master Program

Thesis Approval

(Al-Farāhīdī Arabic Diacrizer System)

Prepared by:Iyad Ahmad M. Abusamrah

Reg. No: 21111189

Supervisor: Dr. Labib Arafeh

Master thesis submitted and accepted, Date: ----------------

The names and signatures of the examining committee members are as follows:

1-Dr. Labib arafeh : Head of Committee Signature ………………..

2- Dr. Ahmad Alqutob : Internal Examiner Signature ………………..

3- Dr. Radwan Tahboub : External Examiner Signature ………………..

Jerusalem/Palestine

2015

Dedication

I dedicate this work to my parents,

my wife,

my daughters,

and finally to my brother, sister, and their families

Iyad Abusamrah

Declaration

I hereby Certify that this thesis submitted for the Degree of Master is the result of my own research, except where otherwise acknowledged, and that this thesis (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: ....

Iyad Ahmad Abusamrah

Date: …………………….

Acknowledgments

I would like to thank and express my sincere gratitude to Dr. Labib Arafeh for his guidance, encouragement, advice, incentive and insightful comments during my work. A special thanks to him for offering related references and reviewing my reports until the end.

I would like to thank all the professionals who were willing to reserve some time to discuss my graduation topic, my doctors and colleagues in the Engineering Faculty at Al-Quds University, for sharing their knowledge and experience. Furthermore, Special thanks to my friends Dr. Nizar Habbash and Dr. Sadiq Dabbas for his advices and useful help.

I would like to thank my partners in the IT department Colombia University and Mr. Ayman Albayya and also a special thanks to Dr. Ahmad Abusamrah for his useful help and patience during my study.

I would like to thank with a deep gratitude my Examiner Dr. Radwan Tahboub and Dr. Ahmad Alqutob.

I would like to thank with a deep gratitude my mother and my father for their endless support, patience and encouragement during my whole life.

I express my deeply sincere gratitude to my wife for her great incentive, encouragement and spiritual support throughout my study.

Last but not least, my deep gratitude to my family.

Abstract

Automatic words diacritization is one of the NLP challenges with languages having diacritics unveiling the phonetic transcription of their words. Arabic is an example of such languages where different diacritics over for the same spelling produce different words with maybe different meanings

Text-to-speech (TTS), Part-of-Speech (PoS) tagging, Word Sense Disambiguation (WSD), and Machine Translation can be enumerated among a longer list of applications that vitally benefit from automatic diacritization. One major challenge with Arabic is its rich derivative and inflective nature, so it is very difficult to build a complete vocabulary that cover all (or even most of) the Arabic generable words. In fact, while Arabic is on the extreme of richness as per its vocabulary when regarded as full- form words, this language is also on the extreme of compactness of atomic building entities due to its very systematic and rich derivative and inflective nature.

This thesis introduces a proposed diacrize system Al-Farāhīdī Arabic Diacrizer System (AADS) which is a hybrid system to automatically diacritize raw Arabic text that is known to be quite a tough problem. The first part tries to decide about the most likely diacritics by choosing the sequence of full-form Arabic word diacritizations with maximum marginal probability via probability estimation. When full-form words happen to be out-of-vocabulary, the second part is resorted. This second part factorizes each Arabic word into its possible morphological constituents (prefix, root, pattern and suffix), then uses rule based to find the proper diacritics. While the second part has the advantage of excellent coverage over the Arabic language, the first part enjoys a better disambiguation for the same size of training corpora especially for inferring syntactical (case-based) diacritics. The presented hybrid system enjoys the advantages of both parts.

The AADS work on the level of letter, word and sentence, while the other system work

on the level of word and some of them neglect the end letter because it cant decide the grammar rule.

In this study, two kinds of modules have been developed, namely the morphological sub model and statistical sub model. These two approaches developed by using Microsoft studio 2010, a pre-processing stage has been accomplished for the collected tool to decide which tool that can be used in every sub model and what kind of modification need to satisfy our aim, after testing many tool we decide to use Alkhali morpho system as part of morphological sub model, and MADAMIRA tool as part of statically sub model.

In order to develop the sub models, we have divided the system into stages, the first stage developing morphological sub model, in this stage we found the alkhalil system cant perform our aim exactly, to mange this issue we found there is a need to re-programming Alkhalil by adding another class and method and re-programming existing class, another modification need by creation new database in order to perform our aim.

The major finding in the AADS it can work in any level (letter, word and sentence), and it can represent status syntactic (الحالة الاعرابية) for each word and suggest other form root conjugation (التصريف), and AADS is adaptive where the user can add any word that not found in its Database.

In comparison with other tools that work in the same filed, we can notice from the result that AADS has performed much better than its counterpart’s tool in the level of letter, word and sentence with mixed result depend of level of testing.

These preliminary and promised result indicate the sustainable and adequacy of the developed system to diacrize Arabic text. Further investigation and work still needed to be applied to furtherlly investigation model with more rules, and enhance the developed system.

الفرابي المشكل الالي للغة العربية

ملخص الرسالة

للغة قيمة اجتماعية ، فهي تحقق التواصل بين الناس ، فمن خلال اللغة يتواصل الناس فيما بينهم ، ويحققون أغراضهم ومآربهم ، فيتناقلون الأفكار ، ويطلبون تحقيق مصالحهم من بعضهم البعض. وهذا التواصل غرض أساسي بالنسبة للإنسان ؛ لأن الإنسان مدني بطبعه ، يحب الاجتماع والمدنية ، ويزعجه الانفراد والوحدة ، ولهذا كان السجن عبارة عن عقوبة للإنسان .

وللغة أيضًا قيمة إنسانية اختص الله بها الإنسان ، فهي تتيح للإنسان التعبير عن أفكاره ومشاعره وآماله وآلامه ، وهذه نعمة خاصة بالإنسان من بين سائر الكائنات ،وللغة قيمه حضارية فهي وعاء الفكر وكل حضارات الأمم مرتبطة بلغتهم ، وإذا أردت أن تعرف حضارة أمة ما فتعرف على لغتهم.

تعتبر اللغة العربية أكبر لغات المجموعة السامية من حيث عدد المتحدثين، وإحدى أكثر اللغات انتشارًا في العالم، يتحدثها أكثر من 422 مليون نسمة، ويتوزع متحدثوها في المنطقة المعروفة باسم الوطن العربي، بالإضافة إلى العديد من المناطق الأخرى المجاورة كالأهواز وتركيا وتشاد ومالي والسنغال وإرتيريا وللغة العربية أهمية قصوى لدى المسلمين، فهي لغة القرآن الكريم، ولا تتم الصلاة (وعبادات أخرى) في الإسلام إلا بإتقان بعض من كلمات هذه اللغة.كما ان العربية لغة رسمية في كل دول العالم العربي. وقد اعتمدت العربية كإحدى لغات منظمة الأمم المتحدة الرسمية الست.

تظهر الكتابة العربية المعاصرة على شكل حروف دون علامات التشكيل، هذا ما يقرأه العربي في الكتب والصحف والإعلانات وعلى شبكات الإنترنت . وهذا يعني أن الخط العربي المعاصر يفتقد للرموز التي تمثل الصوائت . وقد لا يبرز ذلك إلا عند المقارنة بكلمات من لغات أخرى .

إلا أن عدم وجود علامات التشكيل على الحروف العربية يسبب معضلة ليست باليسيرة لكثير من البرمجيات المعاصرة ومنها الناطق الآلي . فلا يمكن لأ ي نظام نطق آلي أن يقوم بعملية النطق دون تشكيل للحروف . فعند ورود أي ة كلمة عربية لا يمكن لنظام الناطق الآلي معرفة علامات التشكيل التي تستوجب وضعها على الحروف المكونة لها . لهذا لا بد من توفر مشكل آلي ليقوم الناطق الآلي بمهمته.

وتمثل علامات التشكيل فيما تمثله الصوائت التي لها دور أساسي في الخصائص الفيزيائية لموجات الكلام المنطوق . ولهذا تصبح عملية النطق الآلي عملية مستحيلة دون التشكيل . كما أن للتشكيل أهمية في عمليات حاسوبية أخرى كما في محركات البحث والنقل الكتابي للأسماء من وإلى اللغة العربية والتعرف الآلي على الكلام العربي وغيرها.

يهدف هذا المشروع الى بناء نظام تشكيل الي لنصوص اللغة العربية، يستخدم فيه النظام الاحصائي والنظام الصرفي، الجزء الأول من هذا النظام يحدد التشكيل على الأرجح عن طريق اختيار شكل كامل للنصوص المحركة في الجملة الفرعية مع أعلى وزن وتقدير الاحتمالات، والجزء الثاني يعتمد على استنباط الحركات الخاصة بالنص بالرجوع الى قواعد اللغة وتصريفاتها بالاعتماد على قاعدة بيانات تم تطويرها لتخدم هذا الغرض.

ومن اهم النتائج ان النظام يعمل على مستوى الحرف والكلمة والجملة، ويوصف الحالة الاعرابية لكل كلمة مع بيان الجذر والتصريف للكلمة، كما ان النظام قادر على التكيف حيث أن المستخدم يمكنك إضافة أي كلمة غير موجودة في قاعدة بياناتها.

وبالمقارنة مع الادوات التي تعمل في نفس المجال يمكن ان نلاحظ ان اداء النظام افضل من غيرة بنسبة تقدر بمعدل 20% من نظراءه، هذة النتائج الاولية تشير الى استدامة النظام ويمكن تطويرة باضافة المزيد من القواعد وتعزيز كفاءته باضافة المزيد من المفردات.

List of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction the state-of-art

1.1. Introduction

Arabic language belongs to the group of the Semitic alphabetical scripts that use letters to represent 26 consonants, and on an optional basis, diacritics to indicate vowels. The class of writing systems considered is known as Abjads, in which each symbol always stands for a consonant. In Abjads, the reader must supply the appropriate vowel to indicate inflectional or derived forms. The issue of diacritics is not simply that of the script, but rather a combination with the spelling rules of Arabic. Similarly, one may write English without vowels. The issue is not the script, but it is the spelling system and orthography. Diacritics are added according to their position in the sentence to help clarify the meaning of words and disambiguate any vague spellings or pronunciations. In order to facilitate learning Arabic for foreigners and children, diacritic symbols as known today were introduced in the 11th century by Al Farāhídi' to provide enough information about the correct pronunciation. But these are usually omitted, except in the Holy Qur'an, dictionaries, pedagogic books, and others as decided by the writer. In modern times, Text-to-Speech (TTS) systems request a diacritization algorithm to be added to the Natural Language Processing (NLP) system [1]. Moreover, they particularly request discourse analysis, Part-of Speech-Tagging, Named Entity Recognition, Sentence Breaking, and

Word Sense Disambiguation. Diacritization is perceived as the first stage of an Arabic NLP system Appendix B represent Arabic Letter and Appendix C represent Arabic diacrtis.

Research on Arabic computational morphology has increased considerably in recent years. Indeed, research on Arabic morphology has always been not extraordinarily prolific due to the complexity of the subject. However, despite the reasonable amount of computational models that have been proposed, the different approaches have not been completely explored and a vast amount of continued work is needed [2].

Several researchers have addressed the diacritization problem. The various reported approaches in the available literature on the problem include [2], Center for Computational Learning Systems, Columbia University, a diacritization system for written Arabic which is based on a lexical resource. It combines a tagger and lexeme language model. Rayan and his coauthors [3] have investigated the tasks of general morphological tagging, diacritization, and lemmatization for Arabic. They reported that for all tasks they consider, both modeling the lexeme explicitly, and retuning the weights of individual classifiers for the specific task, improve the performance. Zitouni and colleagues [4] have used a maximum entropy classifier to assign diacritics to the letters of each word. Rashwan [5] has also introduced a two-layer stochastic system to automatically diacritize raw Arabic text that is known to be quite a tough problem. The first layer attempts to decide the most likely diacritics by choosing the sequence of full form Arabic word diacritizations with maximum marginal probability via long A* lattice search and m-gram probability estimation. They reported a reduction in word error rate in comparison with the first three approaches. Furthermore, Shaalan [6] has followed the rule-based approach in developing his Arabic natural processing tools and systems. Shaalan concluded that rapid development of rule-based systems is feasible, especially in the absence of linguistic resources and the difficulties faced in adapting tools from other languages due to peculiarities and the nature of Arabic language.

In spite of the fact that Arabic is an intensively diacritized language, however, Modern Standard Arabic, (MSA), is typically written by existing natives without diactrics. Thus, the major task of the NLP is to correctly infer all missing diactrics of the input MSA text and to amend those diactrics so as to account for the mutual phonetic effects among adjacent works upon their continuous pronunciation. In addition, MSA script normally includes many common mistakes. Furthermore, 7.5% of open domain Arabic text includes transliterated words that lack any Arabic constraining model. Many of these words are confusingly analyzable as normal Arabic words [7].

From an industry viewpoint, the most representative commercial Arabic morphological processors include Sakhr's, Xerox's, and RDI's [5]. Sakhr's system is a factorizing one based on the standard Arabic dictionaries. Sakhr's declares 97% accuracy. Xerox's system is also a factorizing system based on the standard Arabic dictionaries [7]. RDI's system is a factorizing system where each regular derivative root is allowed to combine with any form as long as this combination is morphologically allowed. Although it reaches an accuracy of 96%, it suffers from a high processing time [8].

1.2. Research motivations and problem statements

Natural Language Processing is the field of computer science concentrated on the development of technology that assists computers in dealing with human language as input and output. NLP was founded to create computational model that equals human performance. Jurafsky and Martin [8] describe NLP as “Computational techniques that process spoken and written human language as language”. Besides that, Microsoft researcher explained the goal of the NLP as “to design and build software that will analyze, understand and generate language that human use naturally, so that eventually one will be able to address their computer like addressing another person” [10].

Arabic language contains complexities that present a considerable challenge to the computational linguistics, these challenges include, Firstly, it is different from other Latin characters which contain 60 unique characters for letters, numbers, punctuation marks, and diacritics, as the platform represents Arabic from right to left but with spelt character and the user can’t read it. Secondly, Arabic language has free word order, which makes the morphological analysis complicated because Arabic is built from roots rather than stems. Besides that, the subject can be omitted leaving any parser with the challenge of deciding whether or not it is most appropriate for it to be omitted [11].

The main aim of the research is to develop novel approach to deal with automatic Arabic diacrization. In order to achieve this aim, the following objectives were identified:

1. To investigate the retrieval performance of the following methods: word, stem, root, and diacrizer.
2. To investigate and implement a novel method and system of a diacrizer for the Arabic language based on proposed hypered method and then link the statistical method with the morphological method in order to compare the performance of the various methods and systems.
3. Design and implement an Arabic diacrizer that contains a morphological component to support information retrieval.

4. To investigate the effectiveness of the hybrid system that we design.

This study also aims to answer the following questions:

1. Does the hybrid Method improve the Arabic diacrizer and information retrieval performance in terms of recall?
2. Does the hybrid Method improve retrieval performance of the root method in terms of precision?
3. Does the hybrid Method have an effect on dicrization and retrieval performance in Arabic?
4. Why do some methods diacrize more than others?

Motivation:

- Arabic is the largest Languages Commissioner group in terms of number of speakers: >422 million people;
- Automatic words diacritization is one of the NLP challenges with languages having diacritics
- One major challenge with Arabic is its rich derivatives and inflective nature,
- Research on Arabic computational morphology has increased considerably in recent years.
- To build an Arabic diacritizer is quite complex and not easy
- Arabic is a very rich language
- Serve Arabic language
- To learn Arabic for foreigners to be able learn about Arabic culture.

1.3. Justification

Arabic is a language were written word cannot be completely determined by its standard where the diacritics are omitted. Out of context and in the absence of diacritics, a word can have several meaning. Diacritics are therefore very useful for understanding. While native speakers are able to disambiguate the intended meaning and choose the right diacritization from the surrounding context with minimal difficulty, automatic language processing of an Arabic text often suffer from the lack of diacritics. The various potential applications of TTS vary from machine translation to facilitating learning and helping visually impaired individuals. It can be used in Arabic Text message for emails, caller ID, mobile messaging, and so on.

1.4. Automatic Arabic text diacritization problems and importance

Arabic is one of a class of languages where the intended pronunciation of a written word cannot be completely determined by its standard. Rather, a set of special diacritics is needed to indicate its meaning. Different diacritics over for the same spelling produce different words [12].

These diacritics, however, are typically omitted in most written Arabic, which lead to widespread ambiguities in meaning. While native speakers are able to disambiguate the intended meaning from the surrounding context with minimal difficulty, automatic processing of Arabic is often suffer by the lack of diacritics. Text-to-speech (TTS), Part-of-

Speech (POS) Tagging, Word Sense Disambiguation (WSD), and Machine Translation can be of applications that vitally benefit from automatic diacritization [13].

One major challenge with Arabic is its rich derivative, so it is very difficult to build a complete vocabulary that covers all the Arabic generable words [12].

1.5. Challenges and innovation points

To build an Arabic diacritizer is quite complex and not easy because modern Arabic text is written without any diacritics, in addition to the fact that modern Arabic text contains many mistakes. Arabic is a very rich language containing innumerable derivatives so it is nearly impossible to build a complete vocabulary spanning all words in the Arabic language. Finally, if we attempt the other methods and use morphological diacrization we cannot count all rules for the Arabic language and we are likely to find some irregular words or statements that cannot fit in this rule.

Innovation point in Al-Frahidi work

1. Autonomy software, partially or completely restricted: The new system is completely independent and can be programmed in any programming language research and distributed commercially.
2. Deal directly with Arabic script: This is unlike other systems as it allows users to solve the issue of converting Arabic letters to Roman, choose the appropriate accents, and convert the Roman numerals to an Arabic characters.
3. High percentage of correct configuration: The rate of correct composition ratio, including all letters and phrases including last words.

4. The hybrid system of the factorizing system, depending on the morphological analyzer, and the unfactorizing system, depending on the statistical analyzer: This will contain the advantage of the two methods and will increase overall accuracy.

The adaptive system: This system can learn any new words that are absent in the corpus of terms.

The goal of this system is to develop novel approach to deal with automatic Arabic diacrization, we will explain this tool in detail in chapter three.

1.6. Thesis organization

The present thesis is organized into run chapters entitled: introduction literature review; morphological system of Arabic; system architecture; system evaluation; and conclusion and future work.

Chapter One of the thesis is mainly an introduction to the study which includes a problem statement and the aims of the study, in addition to the research questions the significance of the study, the scope and limitation of the study, and finally a summary of the chapters.

Chapter Two covers the close-related literature. The chapter starts with a brief background of the structure of the Arabic language. It is then followed by survey of current morphological analyzer. Chapter two ends with a discussion.

Chapter Three is a detailed description of the system architecture and implementation. The chapter describes the prototype and its components. The chapter starts with a brief background of the structure of the system. It is then followed by detailed of each sub model. Chapter three ends with a discussion.

Chapter Four is a detailed description of the experimental research design and methodology. The parameter used in this study. Three parameters are used to evaluate the retrieval performance of each method, and the result are discussed in this chapter.

Chapter Five is the last chapter of the thesis. It is a summary of the work which has been carried out in the current study. It also shows the main findings of the system evaluation and attempts to answer the research questions. The chapter presents several recommendations to those involved in natural language processing. The chapter ends with some suggestions for future work to be done in this area.

Chapter 2

2. Literature review

2.1. Introduction to the Arabic language

The Arabic language related to a group of languages spoken nowadays, mainly in the North of Africa, the Middle East and the Arabian Gulf. These languages are commonly known as Arabic dialects [14]. Furthermore, Modern Standard Arabic (MSA) is derived from the spoken Arabic and classical Arabic, a literary language, which dates back to the seventh century. Classical Arabic is the liturgical language of the Islamic religion, as the Qur'an, or Islamic holy book, is a representation of this language and it plays an important role in Arab society today. It was known to be the mother tongue of the Prophet Muhammad [15].

Genetically, Arabic belongs to the Semitic Language Family, which itself is part of the Afroasiatic Phylum [16]. The Afroasiatic phylum is divided into five families:

1. Tamazight (Berber): Spoken in North Africa
2. Chad languages: Spoken in the Northwest of Africa
3. Ancient Egyptian and Coptic
4. The Cushitic languages: Spoken in the Northeast
5. The Semitic Family: Includes extinct languages such as Akkadian

Typologically, MSA is usually classified in the literature as a fusion type of synthetic language. It exhibits morphological system, Each morpheme generally compiles more morphological characteristics and it cannot be segmented. Their preferred sentence consist of a verb, subject, and object and, in modern usage, the subject-verb-object carry the most weight [17]. Following is the geolinguistic division of the Arab world proposed by Habash [17] there are seven main groups of colloquial Arabic varieties:

1. Egypt and Sudan
2. Levantine: Lebanon, Syria, Jordan, Palestine
3. Gulf countries: Kuwait, UAE, Bahrain, Qatar, Saudi Arabic, Oman
4. North African (*maghrebi*): Morocco, Algeria, Tunisia, Mauritania, Libya
5. Iraqi Arabic: has elements of both Levantine and Gulf
6. Yemeni Arabic: is often considered in its own class
7. Maltese Arabic.

Owens defines Classical Arabic as "the endpoint of a development within the complex of varieties of Old Arabic" [18] Classical Arabic was the standardized and written version of a group of spoken languages used by the Arabian in the pre-Islamic times. These Arabic varieties spoken before the rise of Islam [18].

Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field in computing that aims at finding suitable formal representation of natural language to enable smooth interaction between human and machine. Most linguistic theories attempting to define how to represent natural language were effectively represented by means of computational systems [19].

2.2. Review of the current statistical approaches

The analysis of natural language by means of formal operations has influenced the different paths taken by modern linguistic studies. In the words of Cooper [20], “The view of natural languages as formal languages was a tremendously productive abstraction which enabled researcher to apply twentieth century logical techniques to the characterization of human ability” [20].

The implemented system for the Arabic diacritizer exists in two categories. One of these categories is implemented as a part of research by an individual group, which presents a good idea theoretically. However, they attempt to improve the idea from a research perspective, as it is partially completed, but not necessarily produce a complete system that can deal with real Arabic text.

We review six approaches that are directly relevant to these researches:

- In Diacritization as a Machine Translation Problem and as a Sequence Labeling Problem research done by Tim Schlippe, ThuyLinh Nguyen, and Stephan Vogel[21], they describe and compare the tow technique for automatic diacrization of Arabic text. It is studied with regard to diacrization as a monotone machine translation as model, which uses statistical machine translation, while the other uses a sequence labeling problem and proposes a solution using condition random field. They then use both sets of data resources to further discuss the diacrization LDCs Arabic tree bank data along with the data provided by AppTek. In their system, they found that the experimental result reduced the word error rate by 0.2% and 0.1% in lexical weight [21].

- In the Hybrid System for Automatic Arabic Diacritization research done by Mohsen A. A. Rashwan, Mohammad Al-Badrashiny, Mohamed Attia, Sherif M. Abdou, they introduce a two-layer stochastic system to automatically diacritize raw Arabic text by first choosing the sequence of full form of Arabic words with the maximum marginal probability. They then factorize each Arabic word into its possible morphological constituents, and they use annotated DB to train their system via two packages. The first is a standard Arabic text corpus collected from numerous domains and the second is from classical Islamic literature. After completing their experiment, they found an 11.5% morphological error in factorization diacritizer and 9.2% in hybrid diacrization. With regards to syntactical errors, they found it to be 26.1% for factorizing diacritizeris and 21% for hybrid diacrization, when the using 128K training corpus size [22].

- In the Arabic Diacritization Using Weighted Finite-State Transducers research done by Rani Nelken and Stuart M. Shieber, they presented an algorithm for restoring these symbols using cascade probabilistic finite state transducers on the Arabic tree bank by integrating a word based language model, a letter-based language model, and morpholigcal model. Their model was expressed as a finite state model and they use viterbi decoding, with their basic model consisting of the following transducers (language model, spelling, diacritic drop and unknowns). They used a random sample of news articles in their experiment and found that when they use baseline model and without case, that the word error rate was 15.48% and diacrtization error rate is 17.33%. With the case, the word error rate was 30.39% and diacrtization error rate was 24.03% [23].

- In the Arabic Diacritization in the Context of Statistical Machine Translation done by Mona Diab, Mahmoud Ghoneim, Nizar Habash, they investigated the impact of diacrization on statistical machine translation and their research suggested that the effect of Statistical Machine Translation performance is positively correlated with the increase in the number of tokens that are correctly used in their experiment. This included an Arabic-English parallel news of 5 million words [24].

- In the Maximum Entropy Based Restoration of Arabic Diacritics research done by Imed Zitouni, Je_rey S. Sorensen, Ruhi Sarikaya, they proposed a maximum entropy approach for restoring diactric in document, and they achieve an error rate of 5.1%, a segment error rate of 8.5%, and a word error rate of 17.3% [25].

- In the Automatic Restoration of Arabic Diacrtics: A Simple, Purely Statistical Approach research by Mansour Alghamdi, Zeeshan Muzaffar and Hazim Alhakami, their system uses a statistical method that relies on quad-gram probabilities. Its accuracy rate was relatively high when compared to previous systems that are based only on statistics. Their technique used in the present system has two major steps. The first step was to create a very rich list of frequently used Arabic quad-grams, or pattern of 4 consecutive diacritized letters. The second step was to use this list in diacritizing almost any Arabic text. Their findings indicate that their system was less than 3 MB in the size, with a speed of more than 500 words per second using a 533-MHz processor [26].

- In the Statistical Methods for Automatic diacritization of Arabic text research done by Moustafa Elshafei, Husni Al-Muhtaseb, and Mansour Alghamdi, the researchers use statistical methods for language modeling. Their approach required a large array of fully diacrtized text, which they then used to search algorithms and find the best probable sequence of diacritized words. Their system training was based on a domain of knowledge [27],

Due to the rising interest rates in Diacritization and in Semitic languages, there has been an increase in importance of other NLP related tasks and diacritization at large. Based on the resources that they need, as they are the basic units of analysis, existing methods regarding their target language can efficiently be classified. Probabilistic systems generate and conditions can be further divided.

The existing methods can be placed on a continuum quantity as they reflect the resources needed on the basis of its cost. The use of resources include the morphological analyzer[28] [29] [30], When these resources are utilized for a particular language, they usually improve performance. For example, the Habash [28] method Zitouni [2] reduced error rate by up to 30% by using the morphological analyzer [31].

Table 1: Summary of given published statistical approaches

Paper title	Author	Approach	Data used	Result	Advantage
Arabic Diacritization in the Context of Statistical Machine Translation(2007)	Mona Diab, Mahmoud Ghoneim, Nizar Habash	Statistical machine translation	An Arabic-English parallel news wire corpus Of about 5 million words	97%	
Maximum Entropy Based Restoration of Arabic Diacritics(2006)	Imed Zitouni, Je_rey S. Sorensen, Ruhi Sarikaya	Statistical	Arabic Treebank Of diacritized news stories –- (including Case -endings).	94.5%	Model has the advantage Of successfully combining diverse sources Of information ranging from lexical, segment-based
Automatic restoration of arabic diacritics: A simple, purely statistical approach(2010)	Mansour Alghamdi* Zeeshan Muzaffar And Hazim Alhakami	Statistical	.	94%	
Statistical Methods for Automatic diacritization Of Arabic text(2006)	Moustafa Elshafei1, Husni Al-Muhtaseb, and Mansour Alghamdi	Using statistical methods for Language modeling Use of HMM	Based on domains of knowledge, e. G. Sports, Weather, local news, international news, business, economics, religion, etc. .	96.9%	The approach requires a large corpus of fully diacritized text for extracting the Language monograms, bigrams, and trigrams for words and letters
Automatic Diacritization for Low-Resource Languages Using a Hybrid Word and Consonant CMM	Robbie A. Haertel, Peter mcclanahan, and Eric K. Ringger	Conditional Markov Models (CMM)		94.5%	

We can notice that from Table 1 that major of the researcher concentrate on the statistical approach and use HMM which cost more time and need huge data for training in malty disciplinary in order to get good result.

Moreover, there are many other systems founded in recently that can perform Diacritization, A large company to utilize in their applications, as systems in this category are enhancing the previous category to produce complete systems that can deal with real Arabic text, has implemented the second system. It was founded as a part of bigger system like text to speech system

For this category, the Sakhrs, Xeorox and RDI Arabic morphological analyzer are the best representatives:

1. Sakhrs: An Arabic diacrtizer that is produced by Sakhr Company and used in office tools. This system has some shortcomings; One of them is the restriction of possible Arabic words are not registered in the dictionary that the system used because this dictionary does not list all the used Arabic vocabulary. The second shortcoming is that the system uses statistical disambiguation, which means that the frequency of single word in the text corpus does not concentrate on the sequence of the sentences [32].
2. Xerox: An Arabic diacritizer that is produced by Xerox Company. The company developed this system by a non-native Arabic speaker so this system also has some shortcomings such as the fact that the developer used statistical approach with a selected dictionary. Additionally, the developer omitted the morphological entity completely (i.e. root, prefix, etc.) [33].
3. RDI: RDI stands for “Research & Development International”, and whose official name is The Engineering Company for Digital Systems Development. It was

established in 1993 and located in Egypt with a spectrum of products and activities including the following:

a. Written Human Language Technologies (HLT) and written Arabic HLT; namely Lexical Analyzer, PoS Tagger, Automatic Diacritizer (Vowelizer), Semantic Analyzer, Search Engine, and Written Language Resources (LR)
b. Spoken HLT and spoken Arabic HLT; namely Speech Verification for the interactive self learning of spoken language and the recitation of the Holy Qur’an (Tajweed), Text-To-Speech (TTS), Text-Concept-To-Speech (TCTS) which is a novel technology for Very Low Bit Rate speech compression, small vocabulary Automatic Speech Recognition for voice commanding, and Speech Language Resources (LR) [34].

The shortcoming of this system is that it is expensive and time consuming to build and validate.

2.3. Review of the current morphological analyzers

In this section, we are going to provide a brief description of the most relevant computational models carried out in the field of Arabic language processing. We will provide a comparison table at the end of the section.

- **Xerox Arabic morphological analyzer (Beesley)**

Xerox: Kenneth Beesley developed it beginning in 1996 at the Xerox Research Centre Europe, which is a finite-state morphological analyzer for Arabic-based Diacritization on full-vocalized lexicon and rules [33].

- **A Lexeme-Based model (Cavalli-Sforza et al.)**

Cavalli-Sforza and Soudi proposed a lexeme-based model for Arabic morphology-based diacritization. The model is built on fully vocalized words. It is motivated by practical concerns as root-based models [35].

- **Standard Arabic morphological analyzer (Buckwalter)**

The Standard Arabic Morphological Analyzer (SAMA), formerly known as Buckwalter Arabic Morphological Analyzer (BAMA) up to version 3 was created by Tim Buckwalter in 2002 [36].

- **MAGEAD (Habash et al.)**

MAGEAD is a morphological analyzer and generator for MSA and the spoken dialects. It relates a lexeme and a set of linguistic features to a surface word form through a sequence of transformations [37].

- **MADA+TOKAN and ALMORGEANA (Habash et. al)**

MADA+TOKA is a toolkit which contains different NLP tools for processing Arabic language. The MADA system performs morphological analysis and disambiguation, and Diacritization. TOKAN, in turn, performs the tokenizing task. The package offers a wide range of tasks, which can be used in many applications [28].

- **AraComLex (Mohammed Attia)**

AraComLex is a large-scale finite-state morphological analyzer toolkit for MSA developed principally by Mohammed Attia. It is based on the lemma as the basic lexical entry for the morphological analyzer [38] [39] [40].

- **Khoja stemmer:**

Khoja stemmer is rule based stemming, and its heavy stemmer, it work in two phase , start by removing the longest suffix and the longest prefix and then matches the remaining word with the verbal and noun patterns, to extract the root

- **AlKhalil morpho system**:

AlKhalil Morpho System could be considered as the best Arabic morphological system. Actually, AlKhalil won the first position, among 13 Arabic morphological systems round the world, at a competition held by The Arab League Educational, Cultural and Scientific Organization (ALECSO)

- **PurePos**

PurePos is an open-source HMM-based automatic morphological annotation tool.

It can perform tagging and lemmatization at the same time, it is very fast to train, with the possibility of easy integration of symbolic rule-based components into the annotation process that can be used to boost the accuracy of the tool.

The hybrid approach implemented in PurePos is especially beneficial in the case of rich morphology, highly detailed annotation schemes and if a small amount of training data is available.

Comparative summary of the various covered technology are tabulated in table 2

Table 2: Comparison between current morphological Analyzers

	Xerox (Beesley 1998b, 2001)	(Cavalli-Sforza, 2000)	SAMA 3.1 (Buckwalter, 2004)	Magead (Habash, 2005; 2010)	Cahill 2007, 2010	Aracomlex (Attia, 2011a; 2011b)	Khoja stemmer	AlKhalil morpho system	PurePos 2007
Technology	Two-level morph.	Morphe			DATR polilex		Morphe	Morphe	automatic morphological
Programming language	Perl, lexc, twolc	Lisp	Perl			Lexc		java	java
Linguistic model	Root-and-pattern	Lexeme-based	Concatenative stem-based	Root-and-pattern representation)		Exeme-based	rule based stemming	Root-and-pattern representation	symbolic rule-based
Input lexicons	Prefixes, roots, patterns and suffixes	Lexeme and features	Prefixes, suffixes , stems, compatibility tables	Root, pattern, vocalism, affixes	Root, pattern and vowel inflections	Lemmas, patterns root-lemma lookup database	Root-and-pattern representation	Prefixes, roots patterns and suffixes	
Grammatical coverage	Overgeneration (but removed in a following step)		Large-scale	Large-scale	Partial	130 alteration rules)	heavy stemmer	Large-scale	heavy stemmer
Transliteration			Buckwalter (2002)		SAMPA				HMM
Evaluation		Partial			Small evaluation	Compared to SAMA on a corpus			
Availability	Propriety software		Open source	Free for research		Open source			

2.4. Review of the current diacrizer systems

1. Mishkal tool: is arbic diacrizer found as desktop or web, The most important feature of this tool that automatically suggests formation of the diacrization,
2. RDI tool: this tool generated in RDI labatory, they used the morphological diacrization method of ArabMorpho ver4 that depend on the morphological analysis, and for syntactical diacrrization the use syntax analyzer by the statistical method that depend on POS tags of the word.
3. MADAMIRA tool: its tool immolated in Colombia university, a system for morphological analysis and disambiguation of Arabic that combines some of the best aspects of two previously commonly used systems for Arabic processing, MADA (Habash and Rambow, 2005; Habash et al., 2009; Habash et al., 2013) and AMIRA (Diab et al., 2007). MADAMIRA improves upon the two systems with a more streamlined Java implementation that is more robust, portable, extensible, and is faster than its ancestors by more than an order of magnitude.

2.5. Summary

This chapter studied existing morphological and statistical analysis systems for text in four dimensions, First, systems that implemented as a part of research bu individual group. Second, the systems that implemented in a large company to produce complete systems such as Sakhrs, Xerox and RDI Company. Third, examine and description of the most relevant computational models carried out in the field of Arabic language processing, Forth, it studied the current diacrizer systems concentrating on methodologies, challenges.

Several morphological analyses for Arabic text exist. Morphological analysis is an important preprosing step for many text analytics applications. The aim of the morphological analysis is to define the information of the word. It was found originally in Xerox .

Chapter 3

3. System architecture and implementation:

3.1. Introduction

In this chapter we will deal with prototype architecture, components such as database, search engine method in addition to the statistical analysis and morpho analysis component, the prototype was developed to implement the hyper system theory as statistical and morphological sub model, this chapter start with overview of the system followed by statistical sub model as MADAMIRA tool used in the system then morphological sub model as AlKalil morpho and the addition to this part including functionality of each part.

In the system design and implementation, the important steps are to identify structure and parameters of the system based on available data. The structure identification itself can be considered as two types, identification of the input of the the model and input-output relation.

The system consists of three main components:

- Input text (undiacrized text)
- Processing
    - Morphological sub model

- Statistical sub model

- Output (diacrized text)

The input of the system is a undiacrized text that could be provided by user from the interface. The output is the diacrized text of each word of the input text and some attribute like suffix, prefix, root, grammar (verb, noun, letter,…).

3.2. Overview of the Al-Farāhīdī system

A prototype of Arabic Diacrization system was developing using Visual Basic.NET. Figure 1 show the main blocks of the system, later in this chapter we will discuss each sub model, the main aim of developing the current system is to investigate the retrieval performance of the novel hyper method against other methods used in Arabic diacrizatin systems (i.e Mishkal, RDI, MADIMAR)

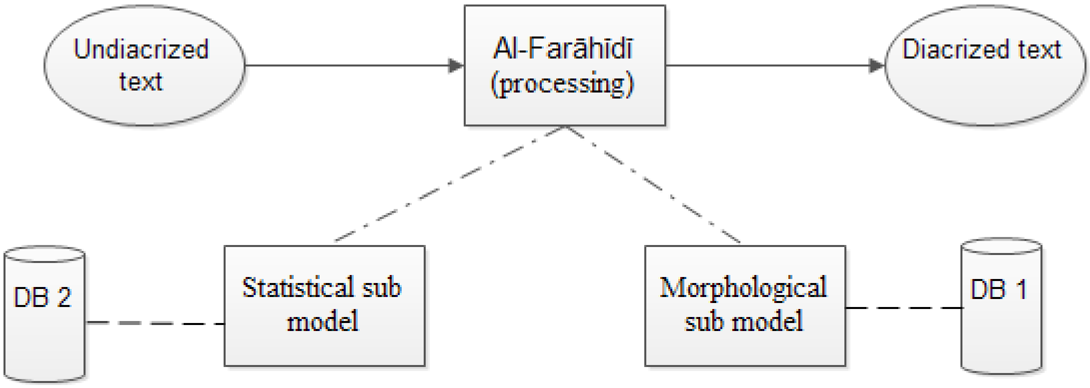


Figure 1: Al-Farāhīdī Block Diagram

The proposed system develops a hybrid system that combines the statistical machine translation based diacritizer with another diacritizer that is based on morpho-syntactical knowledge. Each of these approaches has its own advantages and disadvantages. This hybrid system will take the advantages of both approaches to increase the accuracy of the

Arabic diacritizer and to remove large extent ambiguities to enhance the performance of the diacritizer of Arabic text. Figure 2a and Figure 2b shows the architecture of this hybrid Arabic diacritizer system.

The proposed system consists of two sub models. First sub model is the statistical Arabic diacritizer and the second sub model is the morpho- syntactical Arabic diacritizer. Statistical Arabic diacritizer analyzes the undiacritized Arabic text as one sentence set and generates subsets of words from the original sentence to find the highest probability in the statistical language model and to diacritize these sub sentences. These statistically diacritized sentences are sent to the morpho-syntactic diacritizer. Statistical language model also determines the probability of words sequence in the sentence. The proposed system constructs a general model from translation relations and acquires special rules automatically. These rules are coarse and statistical probabilistic. Morpho-syntactic diacritizer will identify the functional morphemes to merge them into meaning-bearing stems or to remove them from statistical probabilities. Morphemes functions belong to prefixes and suffixes. This procedure checks the statistically diacritized text by applying grammatical rules from the morpho-syntactic language model.

Appendix G show system requirements to install in any environment.

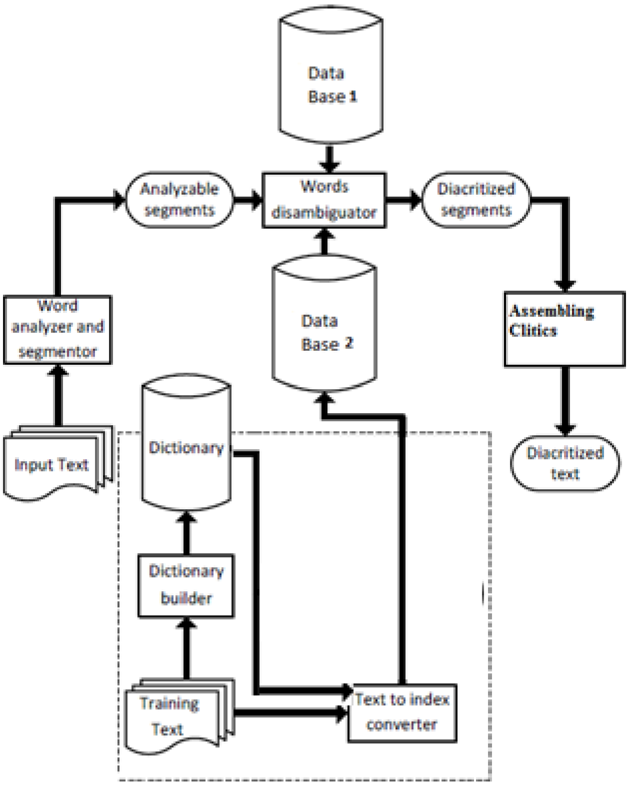


Figure 2: Hybrid Arabic diacritizer Architecture

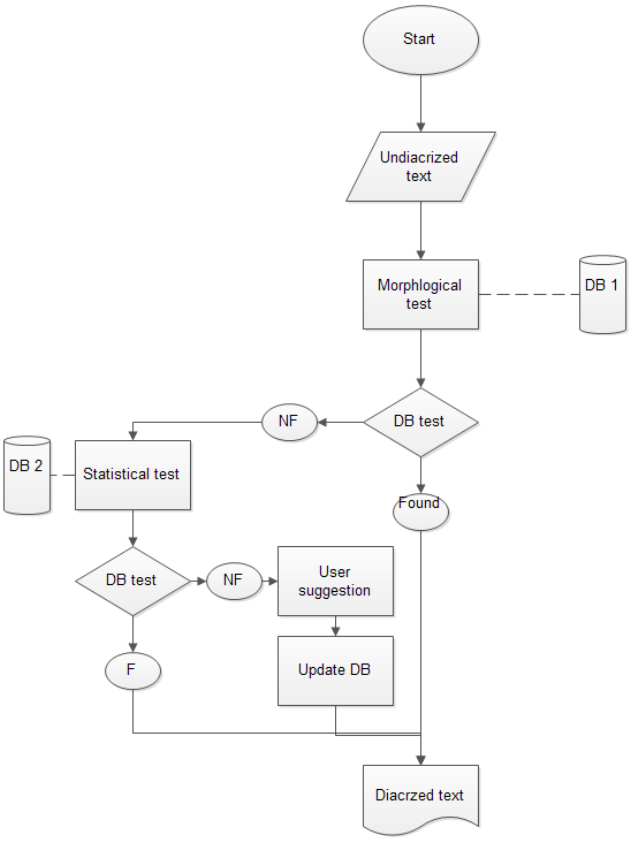


Figure 3: Hybrid Arabic diacritizer Architecture

The algorithm that represent Figure 2 program as follow:

Algorithm:

- Start
- Input Undiacrized text
- Processing
    - Perform morphological test
    - Check available DB 1
    - If found produce diacrized text
    - Else check statistical test
    - If found produce diacrized test
    - Else user suggestion
    - Update DB 1
- Produce diacrized text

  Appendix F give more system flowchart

3.2.1. Statistical Arabic diacritizer sub model

The first sub-model proposed a statistical approach that relies heavily on the training data are available system. Typically, the use of more data to estimate the model parameters diacritization better it can bring real possibilities diacritization. Sub-model statistical language consists of three main steps. The first step is to create a list of commonly used phrases in Arabic diacritized well. The second step is to create a copy diacritized is to build a training model. In the third step, you use the list created in Step 1 to diacritize Arabic text. Also, the secondary language model training process diacritized statistical corpus Arabic text (books, articles and documents) to manually placed by the experts of the Arabic language applies [41].

According to Edmundson there are two broad class of mathematical model in statistical approach: deterministic and stochastic. Mathematical approach in statistical method is saied to be deterministic if it does not involve the concept of probability, otherwise it said to be stochastic [42].

Statistical Natural Language Possessing aim to perform statistical interface for the field of Natural Language Possessing, statistical interface consist of tacking data with some unknown probability and making interface. The importance of Statistical approach is to provide the flexibility required for making the modeling of language more accurate.

Statistical modeling can classified as:

- Primitive acoustic features
- Quantization
- Maximum likelihood and related rules
- Class conditional density function
- Hidden Markov Model Methodology

Where **Primitive acoustic features** are used to estimate the speech spectrum on the basis of its statistical properties. And by means of **quantization** a typical speech signal can be represented as a sequence of symbols and can be mapped using statistical decision rules into a multidimensional acoustic feature space, thus classifying the signal.

Although there is no direct method for computing the probability of a phonetic unit given its acoustic features, we can use Bayes rule to estimate the probability of a phonetic class given its features from the likelihood of the features given the class. This method leads to the maximum likelihood classifier which assigns an unknown vector to that class whose probability density function conditioned on the class has the maximum value[40].

A Hidden Markov Model, is a set of states (lexical categories in our case) with directed edges labeled with **transition probabilities** that indicate the probability of moving to the state at the end of the directed edge, given that one is now in the state at the start of the edge. The states are also labeled with a function which indicates the probabilities of outputting different symbols if in that state (while in a state, one outputs a single symbol before moving to the next state). In our case, the symbol output from a state/lexical category is a word belonging to that lexical category[42].

3.2.2. Morphological sub model:

The second sub model of the proposed system is morpho-syntactical Arabic diacritizer. This sub model uses a mature functional Arabic morphology analyzer called AlKhalil Morpho System to develop a computational model of the morph-syntactical analysis.

Morphology is the study of the internal structure of words. It is the, analysis of the structure of morphemes and other units of meaning in a language. The four morphological processes is:

- Derivation
- Inflection
- Cliticization
- Compounding

The derivation process produces nous (nous in Arabic includes adverbs, adjectives, pronouns, proper nouns and many others) and verbs from the roots (first stem of verbs).

So, the roots, which are verbs consist of three (most cases), four and five letters (rare roots), are the origin of all the Arabic words.

Inflection: happened by adding some well known affixes (prefixes, suffixes and infixes) in order to give some attributes to the word. For example, adding the suffix "ات" (At) to the noun will make it give the meaning of feminine plural.

Cliticization is the use of Clitics; where Clitic in morphology: is a word which is written or pronounced as part of another word [43]. This word is an independent unit at the parsing phase but sticks to other word just like an affix. When it comes before the other word it is called proclitics and when it comes after the other word it is called enclitics. For example, in English the word "ve" in "I've" is enclitics. And the Compounding Some words are formed from a combination of two words and its actually one word this phenomenon is not common in Arabic, but still counts.

In the computational morphology there some method used for morphological analysis which is :

- Root and pattern
- Stem-Based Arabic Lexicon with Grammar

In the root and pattern method it reflect the nature of the Arabic morphology sine all derivations are done according to (الاوزان) Measures, but the Stem-Based Arabic Lexicon with Grammar approach based on stem lexical database and entries associated with grammar and lexis.

There is some application that is known in morphological analysis domain like **Khoja Stemmer and** AlKhalil Morpho System, that mention in previous chapter, Appendix E show AlKhalil Descreption.

3.3. Al-Farāhīdī system detail:

Development Process:

We have been followed Software Engineering in the development Al-Farāhīdī system, we have used Visual studio platform as tool of programming and development, the tow sub modeled named morphological and statistical sub model are described in the following sub section.

Figure 2 and Figure 3 above illustrate a general developing block diagram of our system. It consist three main stages. These stages can be summarized as follow:

- The first stage is pre-processing the input text for the system. These pre-processing contain divide the sentence into word.
- The second stage is concerned with main part of the system, these stage start with morphological sub model and check the form of sentence in database if found the system apply main opration which include Analyzing, Isolation, Lookup at Closed Lists, Un-diacritized Pattern Matching and Root Extraction. If the system not found the form of the sentence or not found any of the word in its database the system start statistical sub model by calling MADAMIRA with JNbridge component.
- The third stage check the diacritics which passing from morphological sub model and statistically sub model to perform Assembling Clitics with Matches if found. Else the system ask the user suggestion.

The Input undiacrized text passed to system from interface, then it forward to the Morphological sub model, the system convert the text to chunks (words) and search for every word and return the root and add the dialects to the original word after search in the rule form, if the system not found the word or rule then it passed it to the statistical sub model and add it to the morphological database.

Al-Farāhīdī give the option to the user to correct every word that suggest from list that appear to the user in the interface, and give a power to the user to enter a new word that not include in the database and his dialects, our constraint in the AADS that we have used in our database ten grammar rule and the sentence can be any number of word that cover these rule.

3.3.1 Development morphological sub model:

After we test three tools that work as morphological analyzer (Khoja Stemmer, The Aramorph System (Buckwalter), and AlKhalil Morpho Sys) we decied to use AlKhalil Morpho Sys because its won the first position, among 13 Arabic morphological systems round the world, at a competition held by The Arab League Educational, Cultural and Scientific Organization (ALECSO) and its open source and easier to include and reuse in our development environment.

Because of the sophistication of the Arabic language, and because this is the first version of AlKhalil, lots of things could be addressed in the database shortage. However, we will address some important issues that we could enrich:

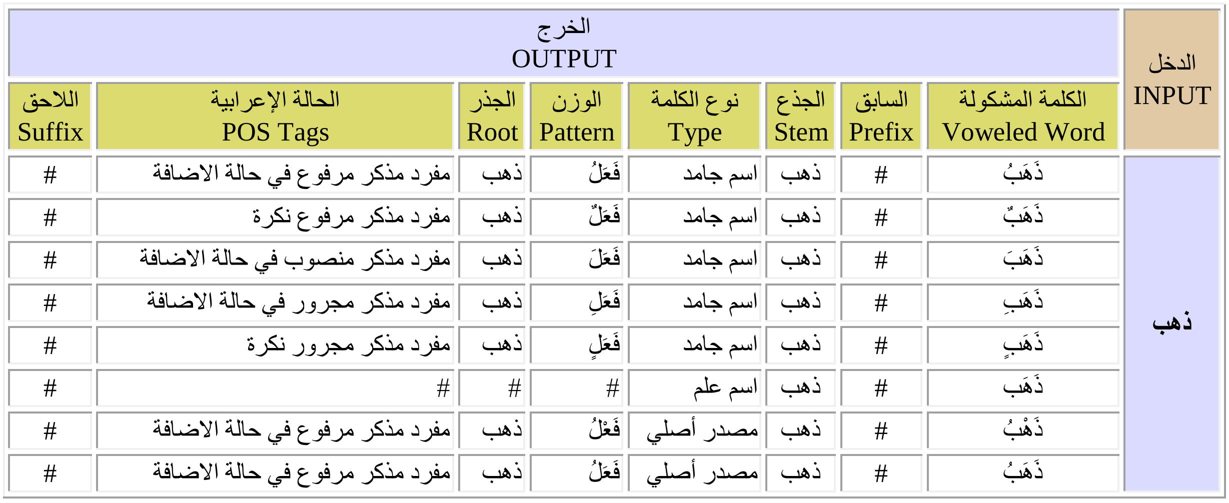
- All nouns, of closed classes, and particles are provided as tool words without classifying them as nouns or particles.
- The database does not contain information about the closed nouns except their fully diacritized form and their Arabic class name, along with the allowed proclitics and enclitics
- There are some missing nouns in the list of tool words.
- There are some missing particles at the list of tool words.

After start working with AlKhalil we found that it’s easer to create Database that reflect the structure of the data used in AlKhalil with some modification and we decide to add some field that we want to complete our process, and the final database contain this tables:

- Table: GrammarRules
- Table: Prefixes
- Table: ProperNouns
- Table: Roots
- Table: Suffixes
- Table: WordTemplates

Annex (D) describe database in detail.

In the second step after creating and testing database we have started programming this block, we found that we need remodification AlKhalil system because AlKhalil build as morph system not diacrizer the output of the AlKhalil a list of word with some specifications, every word with all form like Figure 4



الخرج OUTPUT								الدخل INPUT
اللاحق Suffix	الحالة الإعرابية POS Tags	الجذر Root	الوزن Pattern	نوع الكلمة Type	الجذع Stem	السابق Prefix	الكلمة المشكولة Voweled Word	
#	مفرد مذكر مرفوع في حالة الاضافة	ذهب	فَعَلُ	اسم جامد	ذهب	#	ذَهَبُ	ذهب
#	مفرد مذكر مرفوع نكرة	ذهب	فَعَلٌ	اسم جامد	ذهب	#	ذَهَبٌ	
#	مفرد مذكر منصوب في حالة الاضافة	ذهب	فَعَلَ	اسم جامد	ذهب	#	ذَهَبَ	
#	مفرد مذكر مجرور في حالة الاضافة	ذهب	فَعَلِ	اسم جامد	ذهب	#	ذَهَبِ	
#	مفرد مذكر مجرور نكرة	ذهب	فَعَلٍ	اسم جامد	ذهب	#	ذَهَبٍ	
#	#	#	#	اسم علم	ذهب	#	ذَهَب	
#	مفرد مذكر مرفوع في حالة الاضافة	ذهب	فَعْلُ	مصدر أصلي	ذهب	#	ذَهْبُ	
#	مفرد مذكر مرفوع في حالة الاضافة	ذهب	فَعَلُ	مصدر أصلي	ذهب	#	ذَهَبُ	

Figure 4:the output of the original AlKhalil system(without modification)

To serve our aim in this research we have decided to recoding AlKhalil to serve our aim and the final version of the code contain this classes:

- Analyzer
- ForeignWord
- Interpreter
- Morphology
- Tashkeel
- WordInfo
- Prefix
- Suffix

Analyzer class:

Contains basic functions to start the process of analysis and require functions of other items. Methods knowledge is static,

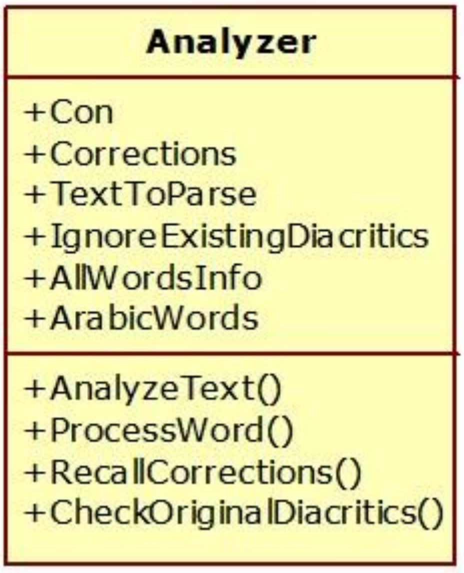


Figure 5: Analyzer class

Morphology class:

Contains morphological analysis of words procedures

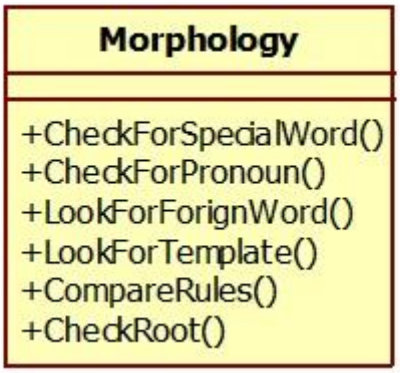


Figure 6: Morphology class

WordInfo class:

Represents the result of the analysis of the word, and every word may have more than one object number of possibilities morphological analysis and contain this field:

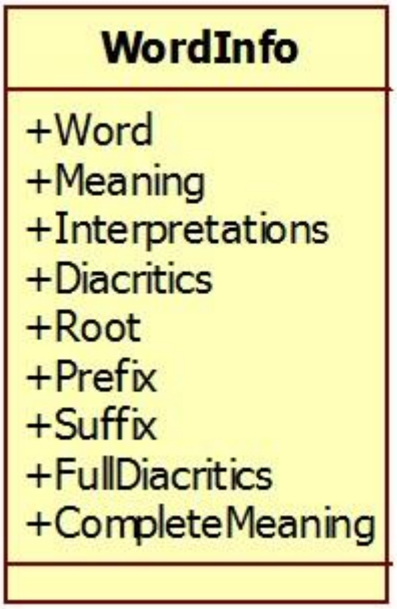


Figure 7: **WordInfo class**

Prefix class:

represent the prefix for every word and contain these filed:

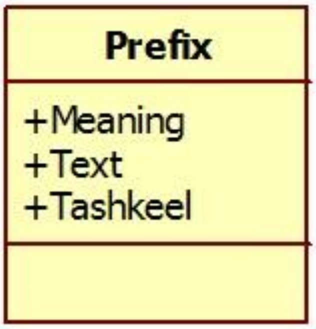


Figure 8: **Prefix class**

Suffix class:

Represent the suffix for every word and contain this filed:

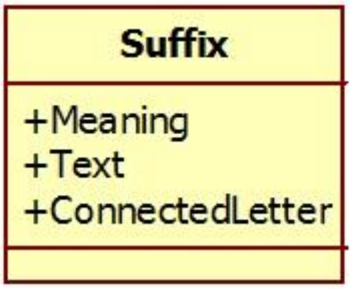


Figure 9: **Suffix class**

Tashkeel class:

Contain the method to return proper dialects for every word and contain this method:

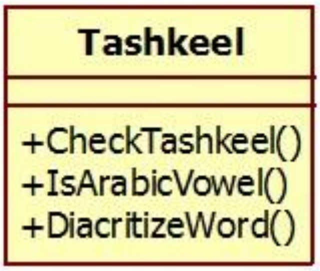


Figure 10: Tashkeel class

The processing of morphology analysis sub model consists several steps as illustrated in figure 10 the step include:

- Analyzing اكتشاف التراكيب
- Isolation عزل السوابق واللواحق
- Lookup at Closed Lists
- Un-diacritized Pattern Matching مطابقة الاوزان
- Root Extraction تطبيق التجذير
- Assembling Clitics with Matches

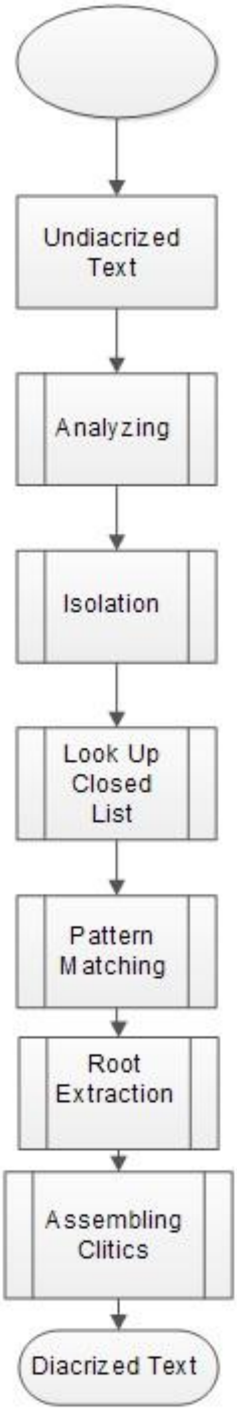


Figure 11: Morphology analysis steps

Analyzing اكتشاف التراكيب

In this phase each compound words are recombined together in one word and the related morphological attributes are assigned.

- **Isolation عزل السوابق واللواحق**

In this part is implemented using two sets of automates. The first set is for proclitics, while the second is for enclitics. For that, the first set scans the letters of the word from right to left, while the second scans the letters from left to right.

- **Lookup at closed lists**

In this part a search for cliticless word, that is the main POS of the word segments, is performed. And the related matches are selected and passed to the next processing steps.

- **Un-diacritized pattern matching مطابقة الاوزان**

In this phase the words are expected to be provided to the system without diacritics, while just a few words would be fully or semi-diacritized.

- **Root extractionتطبيق التجذير**

All possible root(s) of each matched pattern is extracted according to rules. Two processes of verification for the extracted roots are performed. First, the roots which do not exist in the roots database are neglected.

- **Assembling clitics with matches**

Clitics (proclitics and enclitics) are assembled with the compatible cliticless words according to defined rules.

3.3.2Development statitistical sub model:

After we examine several statistical tool that used in Natural language Processing (purepos, Standford, irstlm, nltk-3.0.0, jahmm-0.6.1, MADAMIRA-release-20140825-1.0) and after contact Colombia University (Dr. Nizar Habbash) we decide to use MADAMIRA tool and we get a free license from university to use it in AlFrahidi tool, MADAMIRA tool contain systems, MADA and AMIRA.

MADA uses a morphological analyzer to produce, for each input word, a list of analyses specifying every possible morphological interpretation of that word, covering all morphological features of the word (diacritization, POS, lemma, and 13 in- flectional and clitic features). MADA then applies a set of models – Support Vector Machines (SVMs) and N-gram language models – to produce a prediction, per word in- context, for different morphological features, such as POS, lemma, gender, number or person. A ranking component scores the analyses produced by the morphological analyzer using a tuned weighted sum of matches with the predicted features. The top-scoring analysis is chosen as the predicted interpretation for that word in context; this analysis can then be used to deduce a proper tokenization for the word.

The AMIRA toolkit includes a tokenizer, a part of speech tagger (POS), and a base phrase chunker (BPC), also known as a shallow syntactic parser. The technology of AMIRA is based on supervised learning with no explicit dependence on knowledge of deep morphology; hence, in contrast to MADA, it relies on surface data to learn generalizations.

Input text enters the Preprocessor, which cleans the text and converts it to the Buckwalter representation used within MADAMIRA. The text is then passed to the Morphological Analysis component, which develops a list of all possible analyses for each word. The text and analyses are then passed to a Feature Modeling component, which applies SVM and language models to derive predictions for the word's morphological features. SVMs are used for closed-class features, while language models predict open- class features such as lemma and diacritic forms. An Anal- ysis Ranking component then scores each word's analy- sis list based on how well each analysis agrees with the model predictions, and then sorts the analyses based on that score. The top-scoring analysis of each word can then be passed to the Tokenization component to generate a customized tokenization (or several) for the word, according to the schemes requested by the user. Users can request specifically what information they would like to receive; in addition to tokenization, base phrase chunks and named entities, the diacritic forms, lemmas, glosses, morphological features, parts-of-speech, and stems are all directly provided by the chosen analysis.

3.4. Summary

The present chapter showed the process of constructing the alfrahidi system, the chapter started by giving some introduction about the system focused on its two parts statistically and morphologically, this was followed by some surveying especially for arabic language focusing on the challenges and drawbacks of these parts.

The alfrahidi diacrizer is constructed by combining morphological and statically component, the processing steps in the constructing the alfrahidi involve; first, extracting the input text to token, then analyzing every token separately to find the root, suffix and prefix, and if the system didn't found any token in morphological part the system try to find it in statistical part. After that the system decide every token by its root by refereeing to the role stored in his DB and then combine every word with his dialects.

The morphological analyzer was developed to analyze the word and specify its morphological feature, the morphological analyzer uses the tokenization scheme of Arabic word that distinguishes between parts of word morphemes (prefix, suffix, root).

Chapter 4

4. Experimental results

4.1. Introduction

In this experimental research, the implementation of diacritics through automatic tool will be monitored, and acquisition about Arabic text and motivation (dependent variables) will be measured. The grammar points in this research are Arabic text and its diacritics

4.2. Experimental setup

In this research we have used several error counting metrics to measure the AADS performance, the metrics include [38]:

1. The sentence is said to have a "Syntactical error" if the syntactical diacritic (Case- ending) of the all word in the sentence is wrong.
2. The word is said to have a "Syntactical error" if the syntactical diacritic (Case-ending) of the word is wrong.
3. The letter is said to have a "Syntactical error" if the syntactical diacritic (Case-ending) of every letter in the word is wrong.
4. So at any time the sum of the errors is the summation of the morphological errors and the syntactical errors.

4.3. A Comparison with the recent related work

Among the other recent attempts on the tough problem of Arabic text diacritization, these systems will compare with them:

1. Mishkal tool: is arbic diacrizer found as desktop or web , The most important feature of this tool that automatically suggests formation of the diacrization,
2. RDI tool: this tool generated in RDI labatory, they used the morphological diacrization method of ArabMorpho ver4 that depend on the morphological analysis, and for syntactical diacrrization the use syntax analyzer by the statistical method that depend on POS tags of the word.
3. MADAMIRA tool: its tool immolated in Colombia University, a system for morphological analysis and disambiguation of Arabic.
4. Al-Farāhīdī tool: its tool immolated in Alquds University, which is a hybrid system to automatically diacritize raw Arabic text that is known to be quite a tough problem.
5. Al-Farāhīdī (alkhalil): its tool immolated in Alquds University, which is a a system to automatically diacritize raw Arabic text that is depend on alkhalil system in its database.

   Appendix A show screen shot for The systems

In order to allow a fair comparison with the work of mishkal, RDi and MADAMIRA, we used the same testing sentences; and also we adopt their metric.

- Counting all words, including numbers and punctuation. Each letter (or digit)
- A word is a potential host for a set of diacritics [15].

- Counting all diacritics on a single letter as a single binary choice. So, for example, if we correctly predict a “Fatha” but get the vowel wrong, it counts as a wrong choice [15].
- We approximate non-variant diacritization by removing all diacritics from the final letter (Ignore Last), while counting that letter in the evaluation [15].
- We give diacritic error rate (DER) which tells us for how many letters we incorrectly restored its and word error rate (WER), which tells us how many words had at least one DER and sentence error rate (SER) which tells us how many words had at least one DER [15].

In our experiment we try to cover ten Arabic grammar rule as mention below in Table 3

Table 3: Arabic grammar rule as mention below

ملاحظات	شكل الجملة	الرقم
	مبتدأ+ خبر	1
	فعل ماضي+فاعل+مفعول به	2
	فعل ماضي +فاعل+حرف جر+اسم مجرور	3
	ان + اسم+اسم	4
	كان+اسم+اسم	5
	فعل مضارع+فاعل+مفعول به	6
	فعل مضارع+فاعل+حرف جر + اسم مجرور	7
	فعل امر +مفعول به	9
حرف جر ملتصق(ب)	فعل ماضي +فاعل+حرف جر+اسم مجرور	10

And we used 40 sentences to compare between our work and other work and use the same sentences every time in order to allow a fair comparison with the work.

Table 4 display the sentence that use with the sentence form :

Table 4: Sentence Form

الجملة	الرقم	الجملة	الرقم
يَنْزِل المطَر مِن السَّماء	21.	الجو جميل	1.
تَسِير السُّفُن في الْبحار	22.	درس الولد الدرس	2.
ركب إبراهيم الحصان	23.	ذهب الولد إلى المدرسة	3.
تأكل الشاة فولا وشعيرا	24.	ان الجو جميل	4.
يحصد الفلاح القمح	25.	كان الجو جميل	5.
الكلب ينام في البستان	26.	يدرس الولد الدرس	6.
يسبح الأولاد في البحر	27.	يذهب الولد إلى المدرسة	7.
الثعلب يأكل الدجاج	28.	يأكل الولد التفاحة	8.
العصفور يغرد على الشجرة	29.	أحمد مجتهد	9.
يذهب العمال إلى المصنع	30.	محمد طويل	10.
ينبح الكلب	31.	ادم نائم	11.
نلعب بالكرة	32.	شرب أحمد الدواء	12.
العب بالكرة	33.	أكل ادم التفاحة	13.
أطعم قطك	34.	لعب أدم في الملعب	14.
جنى الفلاح قطنه وباعه ثم اشترى ببعض ثمنه ما يحتاج إليه	35.	لعب أدم و محمد	15.
نظر الطفل إلى الطائر وهو يحلق في السماء، فأحب أن يطير مثله	36.	نام الولد و اخاه	16.
لا تكثر من الكلام، ولا تنطق بما لا تعلم	37.	أكل ادم في المطعم	17.
الطّائر فَوْق الشَّجرَة	38.	قطف محمد زهرة	18.
الْحَديقة جميلة	39.	يعيش السمك في الماء	19.
يكثر النخيل في مصر	40.	الشمس طالعة	20.

4.4. Results analysis

This experiment compares the diacritization accuracy of the four architectures. The change of diacritization accuracy sensed. All these measure are registered in table 5 below.

For each approach, we report the Word Error Rate (WER) (i.e., the percentage of words that were incorrectly diacritized), along with the Diacritic Error Rate (DER) (i.e., the percentage of diacritics, including the null diacritic, that were incorrectly predicted) and sentence error rate and Sentence Error Rate (SER)

Table 5: The Output of RDI, MISHKAL, MADAMIRA and Al-Farāhīdī

Al-Farāhīdī (alkhalil)	Al-Farāhīdī	MADAMIRA	MiSHKAL	RDi	الجملة
الْجَوُّ جَمِيلٌ	الْجَوُّ جَمِيلٌ	الجَوُّ جَمِيل	الْجَوُّ جَمِيلَ	اَلْجَوّ جَمِيل	الجو جميل
دَرَسَ الْوَلْدُ الدَّرْسَ	دَرَسَ الْوَلْدُ الدَّرْسَ	دَرْسُ الوَلَدِ الدَّرْسِ	دِرْسُ الْوَلَدِ الدِّرْسِ	دَرَسَ اَلْوَلَد اَلدَّرْس	درس الولد الدرس
ذَهَبَ الْوَلْدُ إِلَى الْمُدَرَّسَةِ	ذَهَبَ الْوَلْدُ إِلَى الْمُدَرَّسَةِ	ذَهَبَ الوَلَدُ إِلَى المَدْرَسَةِ	ذَهَبُ الْوَلَدِ إِلَى الْمَدْرَسَةِ	ذَهَبَ اَلْوَلَد إِلَى اَلْمَدْرَسَة	ذهب الولد إلى المدرسة
إِنَّ الْجَوَّ جَمِيلٌ	إِنَّ الْجَوَّ جَمِيلٌ	إِنَّ الجَوَّ جَمِيل	ان الْجَوَّ جَمِيلَ	إِنَّ اَلْجَوّ جَمِيل	ان الجو جميل
كَانَ الْجَوُّ جَمِيلً	كَانَ الْجَوُّ جَمِيلً	كانَ الجَوَّ جَمِيل	كَانَ الْجَوُّ جَمِيلَ	كَانَ اَلْجَوّ جَمِيل	كان الجو جميل
يَدْرَسُ الْوَلْدُ الدَّرْسَ	يَدْرَسُ الْوَلْدُ الدَّرْسَ	يَدْرُسُ الوَلَدُ الدَّرْسِ	يُدَرِّسُ الْوَلَدُ الدِّرْسُ	يَدْرُس اَلْوَلَد اَلدَّرْس	يدرس الولد الدرس
يَذْهَبُ الْوَلْدُ إِلَى الْمُدَرَّسَةِ	يَذْهَبُ الْوَلْدُ إِلَى الْمُدَرَّسَةِ	يَذْهَبُ الوَلَدُ إِلَى المَدْرَسَةِ	يُذْهِبُ الْوَلَدُ إِلَى الْمَدْرَسَةِ	يَذْهَب اَلْوَلَد إِلَى اَلْمَدْرَسَة	يذهب الولد إلى المدرسة
يُأكَلُ الْوَلْدُ التَّفَّاحَةَ	يُأكَلُ الْوَلْدُ التَّفَّاحَةَ	يَأكُلُ الوَلَدُ التَّفّاحَةِ	يَأكُلُ الْوَلَدُ التَّفَّاحَةُ	يَأكُل اَلْوَلَد اَلتَّفَاحَة	يأكل الولد التفاحة
أَحْمَدُ مُجْتَهِدٌ	أَحْمَدُ مُجْتَهِدٌ	أَحْمَد مُجْتَهِد	أَحْمَدُ مجتهد	أَحْمَد مُجْتَهِد	أحمد مجتهد
مُحَمَّدُ طَوِيلٌ	مُحَمَّدُ طَوِيلٌ	مُحَمَّد طَوِيل	مُحَمَّدُ طَوِيلِ	م مُحَمَّد طَوِيل	ﷴ طويل
ادْمُ نَائِمٌ	ادْمُ نَائِمٌ	أَدُمْ نائِمٌ	اِدْمَ نَائِمَ	أَدَمّ نَائِم	ادم نائم
شَرِبَ أَحْمَدُ الدُّوَاءَ	شَرِبَ أَحْمَدُ الدُّوَاءَ	شَرِبَ أَحْمَد الدَّواءُ	شَرَبُ أَحْمَدَ الدَّواءِ	شِرْب أَحْمَد اَلدَّوَاء	شرب أحمد الدواء
أَكَلَ ادْمٌ التَّفَّاحَةَ	أَكَلَ ادْمٌ التَّفَّاحَةَ	أَكَلَ آدَم التَّفّاحَةِ	أَكُلَّ اِدْمَ التَّفَّاحَةَ	أَكْل أَدَمّ اَلتَّفَاحَة	أكل ادم التفاحة
لَعَبَ أَدْمٌ فِي الْمُلَعَّبِ	لَعَبَ أَدْمٌ فِي الْمُلَعَّبِ	لَعِبَ آدَم فِي المَلْعَبِ	لُعَبُ أَدَمَ فِي الْمَلْعَبِ	لَعِب أَدْم فِي اَلْمُلَعَّب	لعب أدم في الملعب

لعب أدم و محمد	لَعِب أَدْم وَ مُحَمَّد	لَعَبُ أَدَمَ و مُحَمَّدَ	لَعِبَ آدَم و مُحَمَّد	لَعَبَ أَدْمٌ وَ مُحَمَّدُ	لَعَبَ أَدْمٌ وَ مُحَمَّدُ
نام الولد و اخاه	نَامَ اَلْوَلَد وَ أَخَاهُ	نَامَ الْوَلَدُ و اخاه	نامَ الوَلَدَ وَ أَخاهُ	نَامَ الْوَلْدُ وَ اخَاهَ	نَامَ الْوَلْدُ وَ اخَاهَ
أكل ادم في المطعم	أَكْل أَدَمّ فِي اَلْمَطْعَم	أَكُلَّ اِدْمَ فِي الْمُطْعِمِ	أَكَلَ آدَم فِي المَطْعَمِ	أَكَلَ ادْمٌ فِي الْمُطّعِمِ	أَكَلَ ادْمٌ فِي الْمُطّعِمِ
قطف محمد زهرة	قَطَف مُحَمَّد زَهْرَة	قِطْفُ مُحَمَّدِ زَهْرَةٍ	قَطْفِ مُحَمَّد زُهْرَة	قَطَفَ مُحَمَّدُ زَهْرَةً	قَطَفَ مُحَمَّدُ زَهْرَةً
يعيش السمك في الماء	يَعِيش اَلسَّمَك فِي اَلْمَاء	يَعِيشُ السَّمَكُ فِي الْمَاءِ	يَعِيشُ السَّمَكُ فِي الماءِ	يُعِيشُ السَّمْكُ فِي الْمَاءِ	يُعِيشُ السَّمْكُ فِي الْمَاءِ
الشمس طالعة	اَلشَّمْس طَالِعَة	الشَّمْسُ طَالِعَةَ	الشَّمْسُ طالِعَةٌ	الشَّمْسُ طَالِعَةٌ	الشَّمْسُ طَالِعَةٌ
يكثر النخيل في مصر	يَكْثُر اَلنَّخِيل فِي مِصْر	يُكْثِرُ النَّخِيلُ فِي مُصِرّ	يُكْثِرُ النَّخِيلِ فِي مِصْرَ	يَكْثَرُ النَّخِيلُ فِي مُصِرٍّ	يَكْثَرُ النَّخِيلُ فِي مُصِرٍّ
الطّائر فَوْق الشَّجرَة	اَلطَّائِر فَوْق اَلشَّجَرَة	الطَّائِرُ فَوْقَ الشَّجَرَةِ	الطَّائِرُ فَوْقَ الشَّجَرَةِ	الطَّائِرُ فَوْقَ الشَّجْرَةِ	الطَّائِرُ فَوْقَ الشَّجْرَةِ
الْحَديقة جميلة	اَلْحَدِيقَة جَمِيلَة	الْحَديقَةُ جَمِيلَةَ	الحَدِيقَةُ جَمِيلَةٌ	الْحَدِيقَةُ جَمِيلَةٌ	الْحَدِيقَةُ جَمِيلَةٌ
يَنْزِل المطَر مِن السَّماء	يُنَزِّل اَلْمَطَر مِنْ اَلسَّمَاء	يَنْزِلُ الْمَطَرُ مِنْ السَّمَاءِ	يَنْزِلُ المَطَرِ مِن السَّماءِ	يَنْزِلُ الْمُطَرٌّ مِنْ السَّمَاءِ	يَنْزِلُ الْمُطَرٌّ مِنْ السَّمَاءِ
تَسِير السُّفُن في الْبحار	تَسِير اَلسُّفُن فِي اَلْبِحَار	تَسِيرُ السُّفُنُ فِي الْبِحارِ	تَسِيرُ السُّفُنِ فِي البِحارِ	تَسِيرُ السُّفُنُ فِي الْبُحَّارِ	تَسِيرُ السُّفُنُ فِي الْبُحَّارِ
ركب إبراهيم الحصان	رَكَّبَ إِبْرَاهِيم اَلْحَصَان	رُكْبُ إبراهيم الْحِصَانَ	رَكِبَ إِبْراهِيم الحِصان	رَكَبَ إِبْرَاهِيمُ الْحُصَّانَ	رَكَبَ إِبْرَاهِيمُ الْحُصَّانَ
تأكل الشاة فولا وشعيرا	تَأَكُل أَلْشَاهُ فَوِلَا وَشَعِيرَا	تَأَكَّلَ الشَّاةُ فُولَا وَشَعِيرَا	تَأَكُلُ الشّاةُ فُولاً وَشَعِيراً	تَأَكَّلْ الْشَاةَ فَوُلا وَشَعِيرًا	تَأَكَّلْ الْشَاةَ فَوُلا وَشَعِيرًا
يحصد الفلاح القمح	يَحْصُد اَلْفَلَّاح اَلْقَمْح	يَحْصِدُ الْفَلاَحُ الْقَمْحُ	يَحْصِدُ الفَلّاحُ القَمْحُ	يَحْصَدُ الْفُلّاحُ الْقَمْحَ	يَحْصَدُ الْفُلّاحُ الْقَمْحَ
الكلب ينام في البستان	اَلْكَلْب يُنَامِ فِي اَلْبُسْتَان	الْكَلِبُ يَنَامُ فِي البستان	الكَلْبُ يَنامُ فِي البُسْتانِ	الْكَلْبُ يُنَامُ فِي الْبَسَّتَانِ	الْكَلْبُ يُنَامُ فِي الْبَسَّتَانِ
يسبح الأولاد في	يَسْبَح اَلْأَوْلَاد فِي اَلْبَحْر	يُسْبِحُ الأولاد فِي الْبَحْرِ	يُسَبِّحُ الأَوْلادِ فِي	يَسْبَحُ الْأَوْلَادُ فِي	يَسْبَحُ الْأَوْلَادُ فِي الْبَحْرِ

البحر			البَحْرِ	الْبَحْرِ	
الثعلب يأكل الدجاج	اَلثّعْلَب يَأكُل اَلدَّجَاج	الثّعْلَبُ يَأكُلُ الدَّجَاجُ	الثّعْلَبُ يَأكُلُ الدَّجاجَ	أَلْثَعْلَبْ يُأكَلُ الدُّجَّاجُ	أَلْثَعْلَبْ يُأكَلُ الدُّجَّاجُ
العصفور يغرد على الشجرة	اَلْعُصْفُور يُغَرَّد عَلَى اَلشَّجَرَة	الْعَصْفُورُ يُغْرِدُ عَلَى الشَّجَرَةِ	العُصْفُورُ يَغْرَدُ عَلَى الشَّجَرَةِ	الْعَصْفُورَ يَغْرَدُ عَلَى الشَّجْرَةِ	الْعَصْفُورَ يَغْرَدُ عَلَى الشَّجْرَةِ
يذهب العمال إلى المصنع	يَذْهَب اَلْعَمَال إِلَى اَلْمَصْنَع	يُذْهِبُ الْعُمَّالُ إِلَى الْمُصَنِّعِ	يَذْهَبُ العُمّالُ إِلَى المَصْنَع	يَذْهَبُ الْعُمَّالُ إِلَى الْمُصَنَّع	يَذْهَبُ الْعُمَّال إِلَى الْمُصَنَّع
ينبح الكلب	يُنَبّح اَلْكَلْب	يَنْبَحُ الْكَلِبُ	يَنْبَحُ الكَلْبُ	يَنْبَحُّ الْكَلْبُ	يَنْبَحُّ الْكَلْبُ
نلعب بالكرة	نَلْعَب بِالْكَرَّةِ	نُلْعِبُ بِالْكُرَةِ	نَلْعَبُ بِالْكُرَةِ	نَلْعَبُ بِالْكُرَةِ	نَلْعَبُ بِالْكُرَةِ
العب بالكرة	اَللّعِب بِالْكَرَّةِ	الْعُبُّ بِالْكُرَةِ	أَلْعَبُ بِالْكُرَةِ	الْعَبْ بِالْكُرَةِ	الْعَبْ بِالْكُرَةِ
أطعم قطك	أَطْعَمَ قَطّكَ	أَطْعَمُ قِطِّكَ	أَطْعَمَ قِطّكِ	أَطْعِمْ قَطّكَ	أَطْعِم قَطّك
جنى الفلاح قطنه وباعه ثم اشترى ببعض ثمنه ما يحتاج إليه		جَنى الْفَلاَحِ قُطْنَهُ وَبَاعَهُ ثَمَّ اِشْتَرَى بِبَعْضُ ثَمَنَهُ مَا يَحْتَاجُ إِلَيه	جَنْيِ الفَلّاحُ قُطْنُهُ وَباعِهِ ثُمَّ اِشْتَرَى بِبَعْضِ ثَمَنِهِ ما يَحْتاجُ إِلَيهِ	جَنَى الْفُلّاحُ قَطْنَهُ وَبَاعَهَ ثُمَّ اشْتَرَى بِبَعْضٍ ثَمْنُهُ مَا يَحْتَاجُ إِلَيْهِ	جَنَى الْفُلّاحُ قَطْنَهُ وَبَاعَهَ ثُمَّ اشْتَرَى بِبَعْضٍ ثَمْنُهُ مَا يَحْتَاجُ إِلَيْهِ
نظر الطفل إلى الطائر وهو يحلق في السماء، فأحب أن يطير مثله		نَظَرُ الطِّفْلِ إِلَى الطّائِرِ وَهُوَ يُحَلِّقُ ' فِي السَّمَاءِ '، فَأُحِبُّ أَنْ يُطَيِّرَ مِثْلُه	نَظَرِ الطِّفْلِ إِلَى الطّائِرُ وَهُوَ يُحَلّقُ فِي السَّماءِ فَأُحِبُّ أَنْ يَطِيرَ مِثْلَهُ	نَظَرَ الطِّفْلُ إِلَى الطَّائِرِ وَهُوُ يَحْلَقُ فِي السُّمَاءِ، فَأَحُبْ أَنْ يُطِيرُ مُثِلُّهُ	نَظَرَ الطِّفْلُ إِلَى الطّائِرِ وَهُوُ يَحْلَقُ فِي السُّمَاءِ، فَأَحُبْ أَنْ يُطِيرُ مُثِلُّهُ
لا تكثر من الكلام، ولا تنطق بما لا تعلم		لَا تَكَثّرَ مِنْ الْكَلاَمِ ، وَلَا تَنَطّقَ بِمَا لَا تَعَلُّمَ	لا تُكْثِرُ مِن الكَلامِ ,وَلا تَنْطُقُ بِما لا تَعْلَمُ	لَا تَكَثّرَ مِنْ الْكُلّامِ، وَلَا تَنْطَقُ بَمَّا لَا تَعلُّمِ	لَا تَكَثّرَ مِنْ الْكُلّامِ، وَلَا تَنْطَقُ بَمَّا لَا تَعَلُّمِ

After we test all the output we found that:

SER:

At SER =(number of corrected sentence/40) X 100% …………………………..(1)

Table 6 shows the result of the error measure SER for all the systems. These results are shown graphically in Figure 12, its clear from table and the figure that the Al-Farāhīdī diacritizer outperforms the all. While the difference between the Al-Farāhīdī diacritization error rates is clearly wide, the difference between the MADAMIRA and MISHKAL error rates is much closer, and we can notice that the difrence between Al-Farāhīdī, Al-Farāhīdī alkhalil and MADAMIRA

Table 6: SER Resualt

NAME	SENTENCE
MADAMIRA	35%
Al-Farāhīdī	82%
MISHKAL	32.50%
RDI	0%
Al-Farāhīdī alkhalil	77%

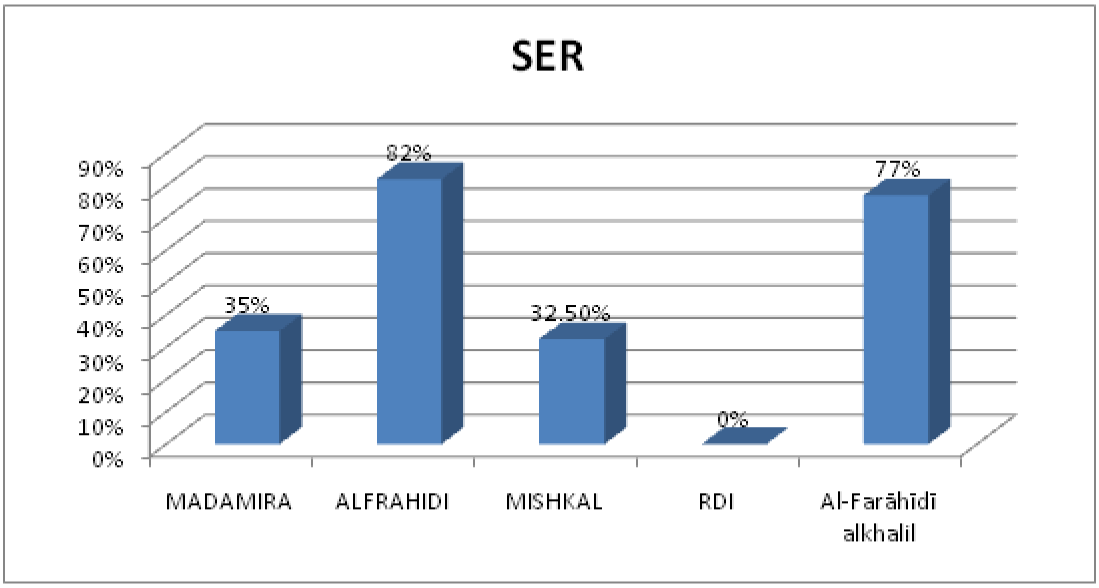


Figure 12: SER Column Chart

A graphical representation for the average error SER measure are shown in Figure 12, the four measures are combined together in the same graph to take a clear look to the behavior of these measure. A relation can be concluded from the graph which is: increase the number of diacrized character in the sentence leads to increase SER, as an example for this Al-Farāhīdī system has performed much better than counterpart tools, because RDi tool not diacrize the end of word, and we can notice that the difrence between Al-Farāhīdī, Al-Farāhīdī alkhalil and MADAMIRA

WER :

WER=((∑(#corrected word/#word))/40)X100%. ………………………………………………………………(2)

Table 7 shows the result of the error measure WER for all the systems. These result are shown graphically in Figure 1, its clear from table and the figure that the Al-Farāhīdī diacritizer outperforms the all. While the difference between the Al-Farāhīdī diacritization error rates is clearly wide, the difference between the MADAMIRA and MISHKAL error rates is much closer and we can notice that the success rate for MADAMIRA and MISHKAL are increased because in this case we don't concentrate at the end of the word.

Table 7: WER result

NAME	WORD
MADAMIRA	62%
Al-Farāhīdī	92%
MISHKAL	60.40%
RDI	6.40%
Al-Farāhīdī alkhalil	88%

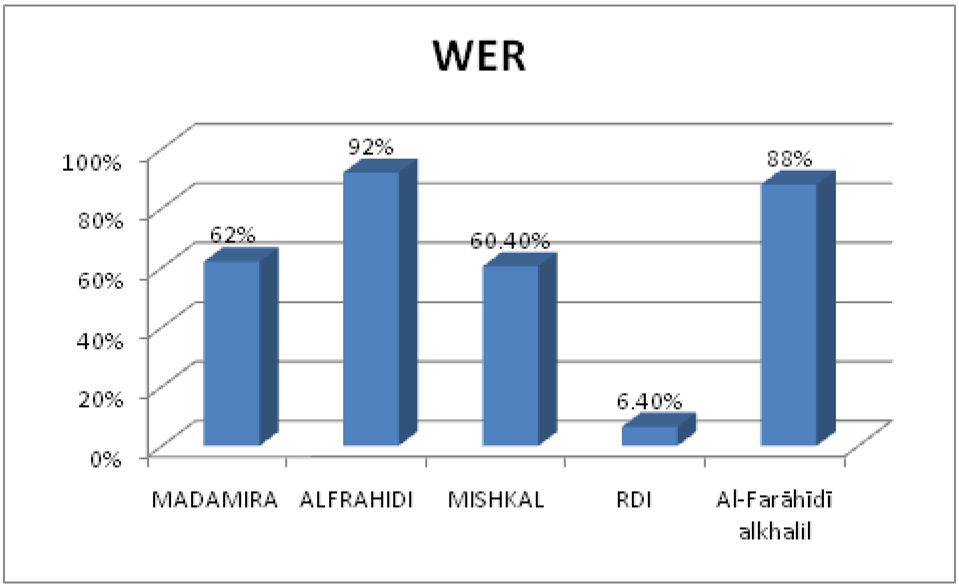


Figure 13: WER Column Chart

A graphical representation for the average error WER measure are shown in Figure 13, the four measures are combined together in the same graph to take a clear look to the behavior of these measure. A relation can be concluded from the graph which is: increase the number of diacrized dialects in the word leads to increase WER. And we can notice that the difference between the Al-Farāhīdī diacritization error rates is clearly wide, the difference between the MADAMIRA and MISHKAL error rates is much closer, and we can notice that the difrence between Al-Farāhīdī, Al-Farāhīdī alkhalil and MADAMIRA

DER:

DER=((∑(#corrected diacritic/# diacritic))/40)X100%. ……………………………………………………….(3)

Table 8 shows the result of the error measure DER for all the systems. These results are shown graphically in Figure 14, its clear from table and the figure that the Al-Farāhīdī diacritizer outperforms the all. While the difference between the Al-Farāhīdī diacritization error rates is clearly close, the difference between the MADAMIRA and MISHKAL error rates is much closer and we can notice that the success rate for RDI are increased.

Table 8: DER Resault

NAME	letter
MADAMIRA	80%
Al-Farāhīdī	90%
MISHKAL	60.00%
RDI	69.00%
Al-Farāhīdī alkhalil	86%

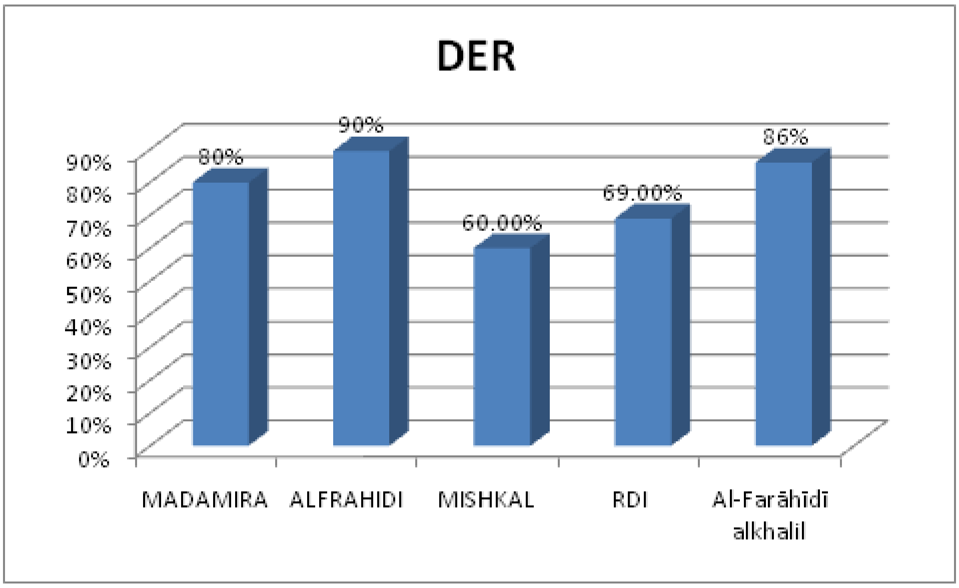


Figure 14: DER Column Chart

A graphical representation for the average error WER measure are shown in Figure 13, the four measures are combined together in the same graph to take a clear look to the behavior of these measure. A relation can be concluded from the graph which is: increase the number of diacrized dialects in the word leads to increase DER. And we can notice that Al-Farāhīdī have the best correction rate at DER and MADAMIRA, Mishkal and RDI tool get correction rate more than SER and WER.

Figure 12, 13,14 can be summarized by the flowing points:

- In RDI and MADAMIRA the SER and WER are high because may our sentences are not include in their corpus.
- DER are less than WER and SER in RDI and MADAMIRA because this system not focus on the end of the word dialect
- Al-Farāhīdī gives highest percentage of correction in all DER, WER, and SER.
- Al-Farāhīdī give better result more than Al-Farāhīdī-Alkhalel and MADAMIRA which mean that the hybrid method is better than morphological and statistical method

4.5. Summary

This chapter discussed the evaluation of the alfrahidi Arabic diacrizer, the first part of the chapter discussed the development agreed standard for evaluating diacrizer for Arabic text, the evaluation specifications and procedure and evaluation metrics were reused to generate standard for evaluating Arabic diacrization.

The developed evaluation standard was constructed to evaluate various diacrization system for Arabic text and allow comparisons between different diacrizer. The detailed information is: the input word, its root and dialicts at the level of word, letter and syntance.

The metrics was used to evaluate the systems, the evaluation focused on measuring the predictin accuracy of 40 sentences, the result AlFrahidi give highest percentage of correction in all DER, WER, and SER and DER are less than WER and SER in RDI and MADAMIRA because this system not focus on the end of the word dialect

Chapter 5

5. Conclusion and future work

5.1.Conclusion

Initially, this study provided a broad theoretical and technical explanation about different types of Arabic text diacrization, statistical approach and morphological approach with their particularities. Many advances in information technology during the last two decades have made Natural Language possessing but with some challenges remaining. With all information and communication available these days, it is possible to provide tool that can be used in different activities and experiences for Language manipulation.

We presented in this research four tool to diacrization Arabic text, in the first tool we examine RDI tool which developed in RDI laboratory and used MORPHO V4 as processor, the second approach and tool is MADAMIRA tool which developed in Colombia university laboratory and use statistically approach to diacrize Arabic text, the third tool is Mishkal which developed to help researcher in this filed and its open source, the last tool which developed during this research and called Al-Farāhīdī.

It has got clear after extensive research and experimentation on the problem of Natural Language Processing in the Arabic language and examines statistical approach verse morphological approach that:

1. The morphological systems are faster to learn but suffer from out of vocabulary.
2. Morphological systems has low preprocessing need reflect low cost of these systems; since by using a small corpus size with a small preprocessing, good results can be obtained.
3. Although the statistical systems need a large training corpus size, but the problem of the out of vocabulary does not exist.
4. Furthermore, it is suggested to add some syntactical linguistic rules and to add a semantic layer to increase the accuracy of both morphological and syntactical diacritizations.

For the problem of Arabic text diacritization; the best strategy to realize usable results is to marry statistical methods with morphological approach ones. Morphologically methods working on full-form words are faster to learn but suffer from poor coverage (OOV) which can be complemented by statistically approach, Moreover The presented hybrid system shows competent error margins with other systems that to manipulate the same problem.

A state of arts about NLP and language manipulation has been presented in this thesis, the research approaches about Arabic diacrization systems can mainly divided to two categories: statistically approach and morphologically approach.

In our literature review of Arabic diacrization systems we have been found that various analyzier attribute could be considered for the systems such as : Technology ,Programming language, Linguistic model , Input lexicons , Grammatical coverage, Transliteration and Evaluation.

On the whole, this thesis is composed of three parts: introduction and literature review, system implementation and experimental and result.

Developing the system for Al-Farāhīdī diacrizer system has been applied firstly using the available data. Two kinds of sub modules have been developed, morphological sub model and statistical sub model, one main input on the system which is undiacrized text, and three outputs (diacreized text, suggestion for every word and grammatical statues).

The adequacy of the developed system has been checked using error rate at the level of sentence, word and dialects level to measure the agreement between the actual and the output of the system. While testing this system with referencing to other three diacrized system which is RDI, Mishkal and MADAMIRA the best correction rate found in Al-Farāhīdī system at SER, WER and DER level.

It was noticed that from result tables and the figures that the Al-Farāhīdī diacritizer outperforms the all. While the difference between the Al-Farāhīdī diacritization error rates is clearly wide, the difference between the MADAMIRA and MISHKAL error rates is much closer and we can notice that the success rate for MADAMIRA and MISHKAL and RDI are increased from SER to DER.

Finally, different works in the field of Arabic diacrization using different technique accomplished by other researchers have been compared with our development prototype, these works show the ability of the hyper approach to represent the diacrization technique, and agree with our results that the rule based technique produce the lowest error rate.

5.2.Future work

There is a huge potential for future research to go deeply and deeply, improving and developing tools in lattice Nature Language Processing. This research at hand indeed deserves further research. Natural language processing is a revolutionary IT trend that will change a lot of companies' IT infrastructure and make the usage of their IT more efficient. This technology topic can attract many potential companies that want to invest more in this topic.

From my opinion, we need to take a step forward in develop our tool with new infrastructure technologies.

Although, we have obtained a preliminary and promising results, but still the following recommendations may help to further contributions in this area

For the problem of statistically approach sub model we need to increase the size of the training data and try to use another tool to study the effect of this increase on the statistical approach sub model.

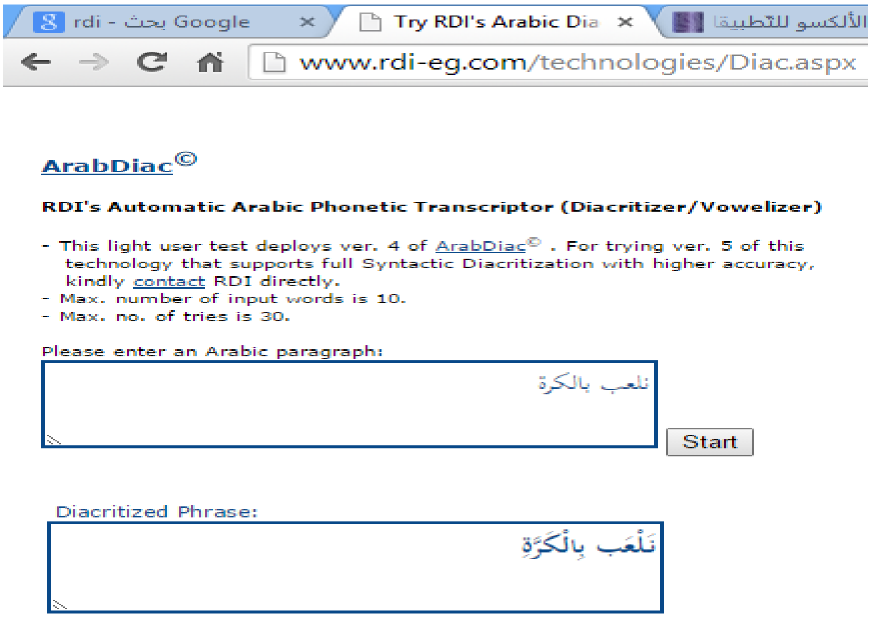
For the problem of automatic Arabic text diacritization:

- We need to increase the number of grammars rule that used in our prototype
- We need to cover irregular grammars rule
- We need to manipulate the diacrize which need to add letter at end of word
- We need to increase the adaptively of the system.
- We need to improve the data used in morphological sub model to cover more and more

Appendixes

Appendix A

A screenshot for the online RDI Diacrizer➔



ArabDiac©

RDI's Automatic Arabic Phonetic Transcriptor (Diacritizer/Vowelizer)

- This light user test deploys ver. 4 of ArabDiac© . For trying ver. 5 of this technology that supports full Syntactic Diacritization with higher accuracy, kindly contact RDI directly.
- Max. number of input words is 10.
- Max. no. of tries is 30.

Please enter an Arabic paragraph:

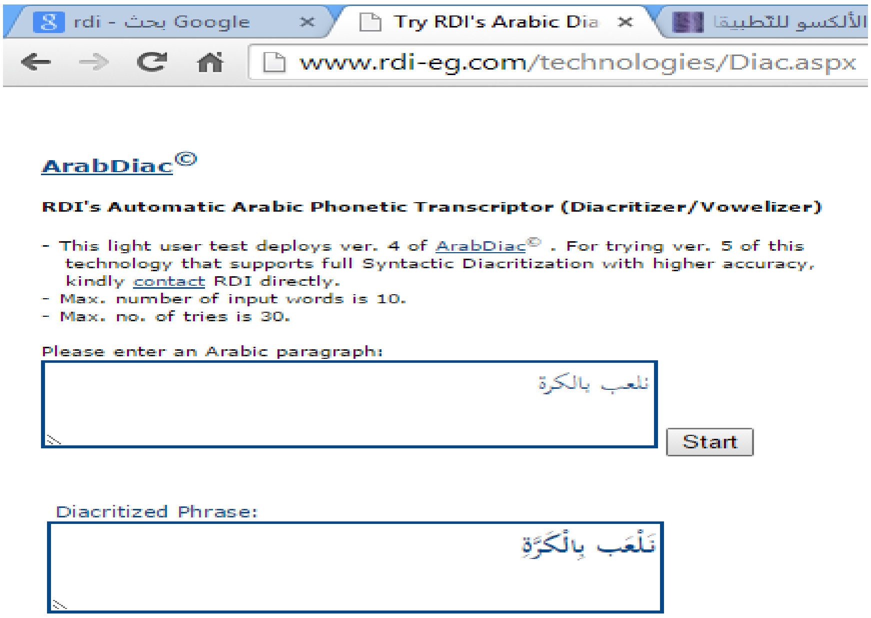
نلعب بالكرة

Diacritized Phrase:

نَلْعَب بِالْكُرَةِ

Figure 15: SER A screenshot for the online RDI Diacrizer

A screenshot for the responses sheet ➔



ArabDiac©

RDI's Automatic Arabic Phonetic Transcriptor (Diacritizer/Vowelizer)

- This light user test deploys ver. 4 of ArabDiac© . For trying ver. 5 of this technology that supports full Syntactic Diacritization with higher accuracy, kindly contact RDI directly.
- Max. number of input words is 10.
- Max. no. of tries is 30.

Please enter an Arabic paragraph:

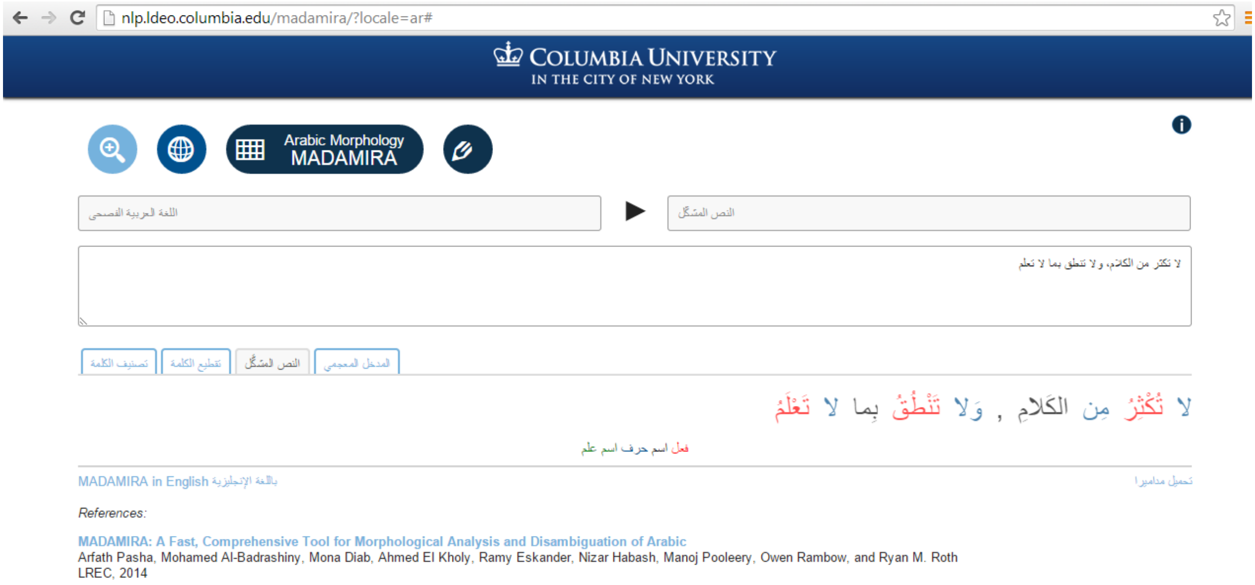
نلعب بالكرة

Diacritized Phrase:

نَلْعَب بِالْكُرَة

Figure 16: SER A screenshot for the responses sheet

A screenshot for the online MADMIAR Diacrizer➔



لا تكثر من الكلام، و لا تنطق بما لا تعلم

لا تُكْثِرْ مِن الكَلامِ , وَلا تَنْطُقْ بِما لا تَعْلَمْ

فعل اسم حرف اسم علم

MADAMIRA in English باللغة الإنجليزية

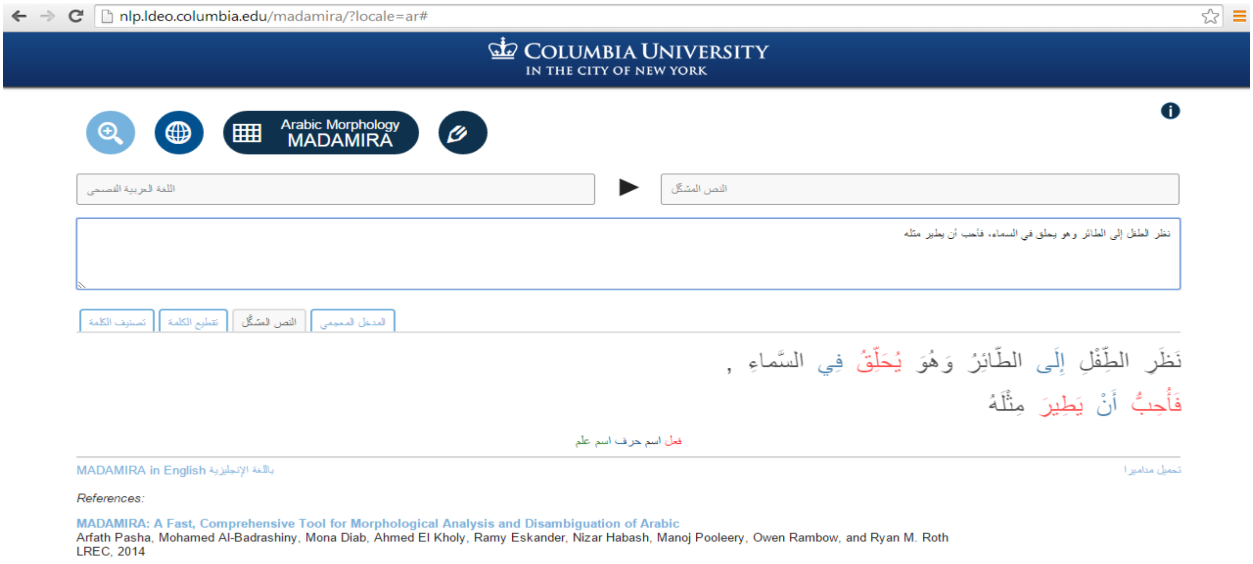
تحميل مداميرا

References:

MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic
Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M. Roth
LREC, 2014

Figure 17: A screenshot for the online MADMIAR Diacrizer

A screenshot for the responses sheet ➔



نظر الطفل إلى الطائر وهو يحلق في السماء، فأحب أن يطير مثله

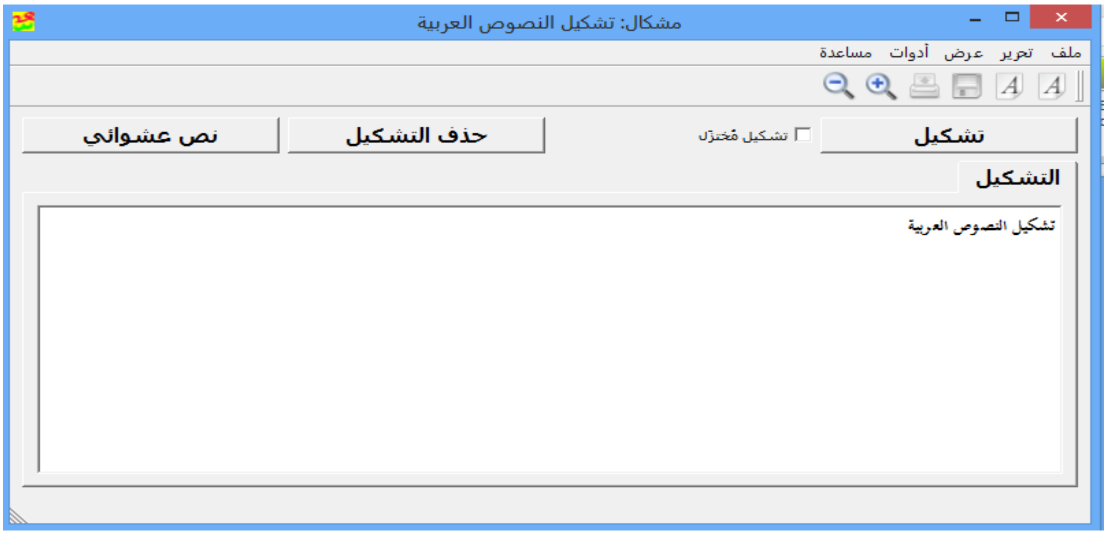
نَظَرَ الطِّفْلِ إِلَى الطّائِرُ وَهُوَ يُحَلِّقُ فِي السَّماءِ ,
فَأُحِبُّ أَنْ يَطِيرَ مِثْلَهُ

References:

MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic
Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M. Roth
LREC, 2014

Figure 18: A screenshot for the responses sheet

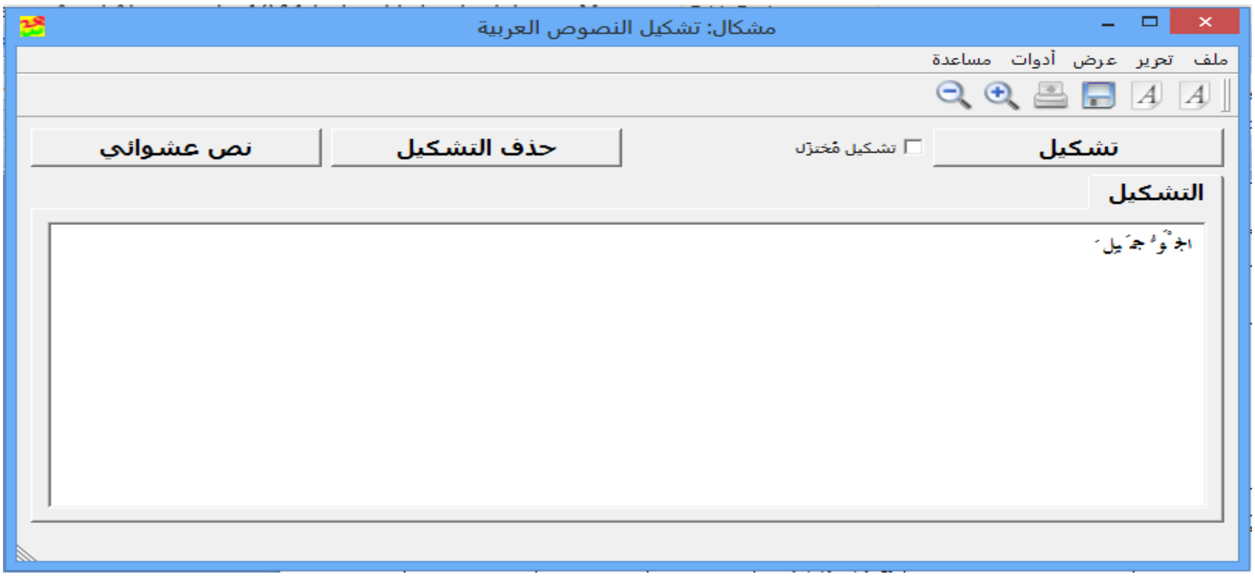
A screenshot for the Mishkal Diacrizer➔



تشكيل النصوص العربية

Figure 19: A screenshot for the Mishkal Diacrizer

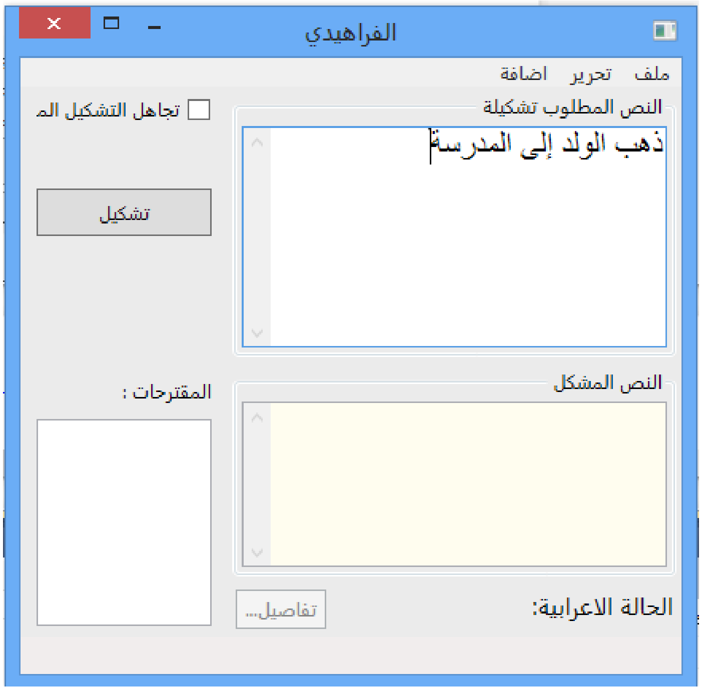
A screenshot for the responses sheet ➔



اجْوُّ جَيلٌ

Figure 20: A screenshot for the Mishkal Diacrizer 2

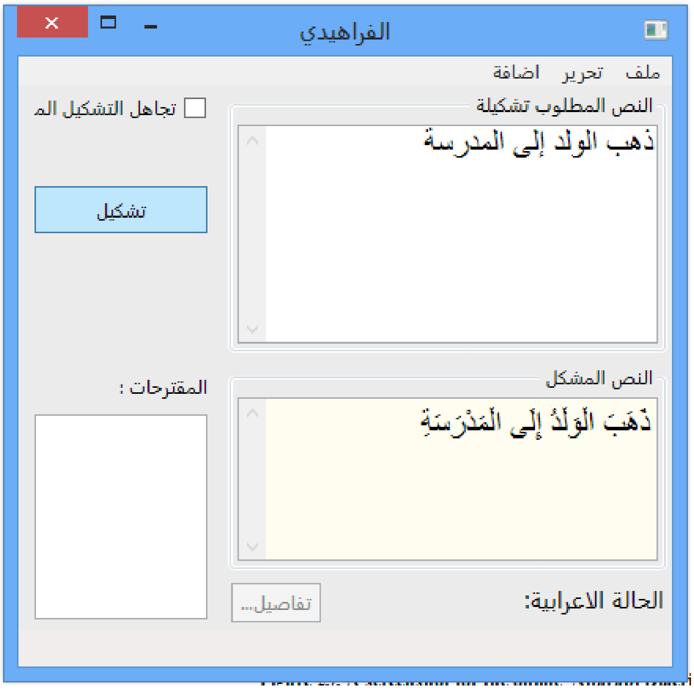
A screenshot for the   Al-Farāhīdī Diacrizer➔



ذهب الولد إلى المدرسة

Figure 21: A screenshot for the Al-Farāhīdī Diacrizer

A screenshot for the responses sheet ➔



النص المطلوب تشكيلة

ذهب الولد إلى المدرسة

النص المشكل

ذَهَبَ الْوَلَدُ إلَى الْمَدْرَسَةِ

الحالة الاعرابية:

Figure 22: A screenshot for the Al-Farāhīdī Diacrizer 2

Appendix B (Arabic Letter)

Table 9: Arabic Letter

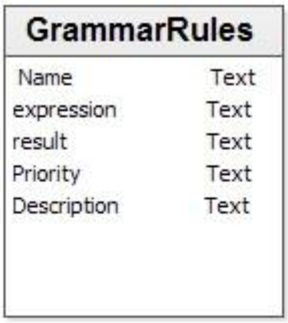
Cat.	Characters	ASCII Range	Mapping key	New IDs
Arabic letters come at the start of the words	ـي	-19	+19	0
	م ن ه و	-29 →-26	+30	1→4
	ل	-31	+36	5
	ن, ق, ف	-35 → -33	+41	6→8
	غ, ع, ظ, ط	-40 → -37	+49	9→12
	ض, ص, ش, ض, ش, ز, ذ, د, خ, ح, ج, ث, خ	-54→-42	+67	13→25
	ب, ا	-57→ -56	+83	26→27
	إ	-59	+87	28
	أ, آ	-62→ -61	+91	29→30
Arabic letters does not come at the start of the words	ي	-20	+51	31
	ج	-55	+87	32
	ئ	-58	+91	33
	ؤ	-60	+94	34
	ء	-63	+98	35
Diacritics	◌ْ	-6	+42	36
	◌ّ	-8	+45	37
	◌ِ ,	-11→-10	+49	38→39
	◌َ, ◌ٍ, ◌ٌ ,	-16 →-13	+56	40→43
Arabic signs	÷	-9	+53	44
	–	-36	+81	45
	×	-41	+87	46
	؟	-65	+112	47
	؛	-70	+118	48
	،	-95	+144	49
	’ ’	-111→ -110	+161	50→51
Numbers	0,1,2,3,4,5,6,7,8,9	48→ 57	+4	52→61
Delimiters	Tab, New line	9→ 10	+53	62→63
	Enter	13	+51	64
	Space	32	+33	65
Arabic and English signs	!,", #, $, %, &, ', (,), *, +, ,, -, ., /	33→47	+33	66→80
	:, ;, <, =, >, ?, @	58→ 64	+23	81→87
	[, \,], ^, _, `	91→ 96	-3	88→93
	{, |, }, ~	123→ 126	-29	94→97

Appendix C (Arabic diacritics set)

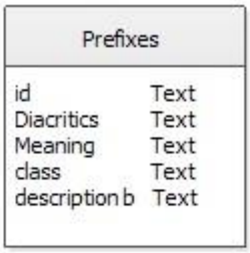
Table 10: Arabic diacritics set

Diacritic's type	**Diacritic**	**Example on a letter**	**Pronunciation**
Short vowel	Fatha	◌َ	/b//a/
	Kasra	◌ِ	/b//i/
	Damma	◌ُ	/b//u/
Doubled case ending (Tanween)	Tanween Fatha	◌ًا	/b//an/
	Tanween Kasra	◌ٍ	/b//in/
	Tanween Damma	◌ْب	/b//un/
Syllabification marks	Sukuun	◌ِ	No vowel: /b/
	Shadda	◌ٓ ب	Consonant doubling: /b//b/

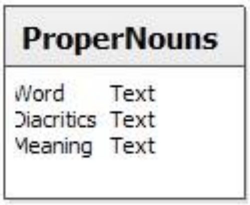
Appendix D (Database Descriptions)



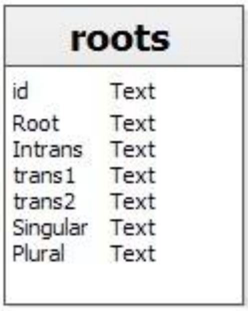
GrammarRules	
Name	Text
expression	Text
result	Text
Priority	Text
Description	Text



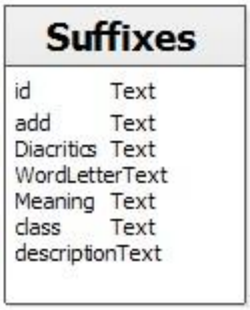
Prefixes	
id	Text
Diacritics	Text
Meaning	Text
class	Text
description b	Text



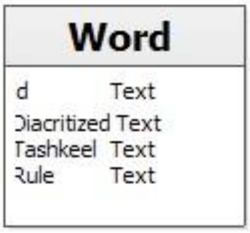
ProperNouns	
Word	Text
Diacritics	Text
Meaning	Text



roots	
id	Text
Root	Text
Intrans	Text
trans1	Text
trans2	Text
Singular	Text
Plural	Text



Suffixes	
id	Text
add	Text
Diacritics	Text
WordLetter	Text
Meaning	Text
class	Text
description	Text



Word	
d	Text
Diacritized	Text
Tashkeel	Text
Rule	Text

Table: GrammarRules

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorIn	-1
DatasheetGridlinesThemeCol	-1	DateCreated:	12/7/2011 2:45:46 AM
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {1F88904C-5A27-4769-8B31-0F632A12F35E}}
HideNewField:	False	LastUpdated:	1/6/2012 10:03:53 PM
NameMap:	Long binary data	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Left-to-Right
RecordCount:	18	ThemeFontIndex:	-1
TotalsRow:	False	Updatable:	True

Columns

Name	Type	Size
expression	Text	255
result	Text	255
Priority	Text	255
Description	Text	255

Table: Prefixes Page: 2

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1
AlternateBackTint:	100	BackShade:	100

BackTint:	100	DatasheetFontHeight:	14	
DatasheetFontItalic:	False	DatasheetFontName:	Calibri	
DatasheetFontUnderline:	False	DatasheetFontWeight:	Normal	
DatasheetForeColor:	0	DatasheetForeColor12:	0	
DatasheetForeThemeColorIn		-1	DatasheetGridlinesThemeCol	-1
DateCreated:	8/15/2010 2:11:19 AM	DefaultView:	2	
DisplayViewsOnSharePointSit		1	FilterOnLoad:	False
GUID:	{guid {6800D9E7-09A1-4F35-91C0-77F52B1CBD64}}	HideNewField:	False	
LastUpdated:	12/30/2011 2:42:37 AM	NameMap:	Long binary data	
OrderByOn:	False	OrderByOnLoad:	True	
Orientation:	Left-to-Right	RecordCount:	117	
TabularCharSet:	0	TabularFamily:	34	
ThemeFontIndex:	-1	TotalsRow:	False	
Updatable:	True			

Columns

Name	Type	Size
add	Text	12
Diacritics	Text	11
Meaning	Text	12
class	Text	50
description	Text	255

Table: Prefixes

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1	
AlternateBackTint:	100	BackShade:	100	
BackTint:	100	DatasheetFontHeight:	14	
DatasheetFontItalic:	False	DatasheetFontName:	Calibri	
DatasheetFontUnderline:	False	DatasheetFontWeight:	Normal	
DatasheetForeColor:	0	DatasheetForeColor12:	0	
DatasheetForeThemeColorIn		-1	DatasheetGridlinesThemeCol	-1
DateCreated:	8/15/2010 2:11:19 AM	DefaultView:	2	
DisplayViewsOnSharePointSit		1	FilterOnLoad:	False
GUID:	{guid {6800D9E7-09A1-4F35-91C0-77F52B1CBD64}}	HideNewField:	False	
LastUpdated:	12/30/2011 2:42:37 AM	NameMap:	Long binary data	
OrderByOn:	False	OrderByOnLoad:	True	
Orientation:	Left-to-Right	RecordCount:	117	
TabularCharSet:	0	TabularFamily:	34	
ThemeFontIndex:	-1	TotalsRow:	False	
Updatable:	True			

Columns

Name	Type	Size
add	Text	12
Diacritics	Text	11
Meaning	Text	12
class	Text	50
description	Text	255

Table: ProperNouns

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetForeThemeColorIn	-1
DatasheetGridlinesThemeCol	-1	DateCreated:	12/25/2011 12:06:42 AM
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {7708A113-F653-46BD-9795-DE2CD6C56277}}
HideNewField:	False	LastUpdated:	12/25/2011 9:13:12 PM
NameMap:	Long binary data	OrderByOn:	False
OrderByOnLoad:	True	Orientation:	Left-to-Right
RecordCount:	36	ThemeFontIndex:	-1
TotalsRow:	False	Updatable:	True

Columns

Name	Type	Size
Word	Text	50
Diacritics	Text	50
Meaning	Text	50

Table Indexes

Name	Number of Fields
Word	1
Fields:	
Word	Ascending

Table: roots

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetFontHeight:	16
DatasheetFontItalic:	False	DatasheetFontName:	Arial
DatasheetFontUnderline:	False	DatasheetFontWeight:	Normal
DatasheetForeColor:	0	DatasheetForeColor12:	0
DatasheetForeThemeColorIn	-1	DatasheetGridlinesThemeCol	-1
DateCreated:	7/24/2010 6:57:07 PM	DefaultView:	2
DisplayViewsOnSharePointSit	1	FilterOnLoad:	False
GUID:	{guid {13A7F0F4-B932-4BB3-94BD-04DCF0E0C3CA}}	HideNewField:	False
LastUpdated:	1/2/2012 8:48:58 PM	NameMap:	Long binary data
OrderByOn:	False	OrderByOnLoad:	True
Orientation:	Left-to-Right	RecordCount:	3440
TabularCharSet:	0	TabularFamily:	34
ThemeFontIndex:	-1	TotalsRow:	False
Updatable:	True		

Columns

Name	Type	Size
Root	Text	255
Intrans	Text	255
trans1	Text	255
trans2	Text	255
Singular	Text	255
Plural	Text	255

Table Indexes

Name			Number of Fields
PrimaryKey			1
	Fields:		
	Root	Ascending	
Root			1
	Fields:		
	Root	Ascending	

Table: Suffixes

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetFontHeight:	14
DatasheetFontItalic:	False	DatasheetFontName:	Calibri
DatasheetFontUnderline:	False	DatasheetFontWeight:	Normal
DatasheetForeColor:	0	DatasheetForeColor12:	0
DatasheetForeThemeColorIn		-1	DatasheetGridlinesThemeCol -1
DateCreated:	12/19/2011 2:17:52 PM	DefaultView:	2
DisplayViewsOnSharePointSit		1	FilterOnLoad: False
GUID:	{guid {2B710233-8353-400E-BCA1-62EC1463AD7A}}	HideNewField:	False
LastUpdated:	1/6/2012 10:01:40 PM	NameMap:	Long binary data
OrderBy:	[Suffixes].[add]	OrderByOn:	True
OrderByOnLoad:	True	Orientation:	Left-to-Right
RecordCount:	65	TabularCharSet:	0
TabularFamily:	34	ThemeFontIndex:	-1
TotalsRow:	False	Updatable:	True

Columns

Name	Type	Size
add	Text	12
Diacritics	Text	12
WordLetter	Text	1
Meaning	Text	12
class	Text	100
description	Text	255

Table: WordTemplates

Properties

AlternateBackShade:	100	AlternateBackThemeColorInd	-1
AlternateBackTint:	100	BackShade:	100
BackTint:	100	DatasheetFontHeight:	16
DatasheetFontItalic:	False	DatasheetFontName:	Arial Narrow
DatasheetFontUnderline:	False	DatasheetFontWeight:	Normal
DatasheetForeColor:	0	DatasheetForeColor12:	0
DatasheetForeThemeColorIn		-1	DatasheetGridlinesBehavior: Vertical
DatasheetGridlinesThemeCol		-1	DateCreated: 7/23/2010 2:53:17 PM
DefaultView:	2	DisplayViewsOnSharePointSit	1
FilterOnLoad:	False	GUID:	{guid {A800EAAA-75A0-4036-93F2-6225ADD1B97D}}
HideNewField:	False	LastUpdated:	1/29/2012 1:34:08 PM
NameMap:	Long binary data	OrderBy:	[WordTemplates].[Diacritized], [WordTemplates].[Rule]
OrderByOn:	True	OrderByOnLoad:	True
Orientation:	Left-to-Right	RecordCount:	420
RowHeight:	510	TabularCharSet:	0

```
TabularFamily:          34                      ThemeFontIndex:       -1
TotalsRow:              False                   Updatable:            True
```

Columns

```
    Name                                    Type              Size
    Diacritized                             Text                    12
    Tashkeel                                Text                    12
    Template                                Text                    12
    Mask                                    Text                    50
    Class                                   Text                    10
    ف                                       Text                     3
    ع                                       Text                     3
    ل                                       Text                     3
    Rule                                    Text                   255
```

Table Indexes

```
    Name                              Number of Fields
    Mask                              1
         Fields:
         Template                  Ascending
    Mask1                             1
         Fields:
         Mask                      Ascending
    WordTemplatestemplate             1
         Fields:
         Diacritized               Ascending
```

Appendix E (Al-Khalil Database Descriptions)

1- db/prefixes.xml

This file contains a list of precedents used in the program where it was attached to each former with the following information:

أ – السابق دون علامات التشكيل (unvoweledform)

ب –السابق مشكولا (voweledform)

ج–التوصيف (desc)

د-) الصنف classe :(Where were these precedents be classified into three categories:

- Class N and we mean precedents that do not fall only on the names
- الصنف N ونعني به السوابق التي لا تدخل إلا على الأسماء

Class V and symbolizes the precedents that specially for verbs

- الصنف V ويرمز للسوابق الخاصة بالأفعال
- Class C and symbolizes the common history between nouns and verbs
- الصنف C ويرمز للسوابق المشتركة بين الأسماء والأفعال

As each of those items are divided into the following subsections:

N1: ويضم هذا القسم ال التعريف وتفريعاتها

  - N2: يضم همزة الاستفهام + ال التعريف وتفريعاتها
  - N3: يضم لام التوكيد+ ال التعريف وتفريعاتها
  - N4: يضم حروف الجر وتفريعاتها
  - N5: يضم حروف الجر + ال التعريف وتفريعاتها
  - V1: يضم سين المضارعة وتفريعاتها
  - V2: يضم لام النصب وتفريعاتها
  - V3: يضم لام الجزم وتفريعاتها
  - C1: يضم الواو والفاء و ‘‘ الذي يعني غياب السابق
  - C2: يضم همزة الاستفهام وتفريعاتها
  - C3: يضم همزة الاستفهام + ال التعريف وتفريعاتها

اللواحقdb/suffixs.xml:

This file contains a list of suffixes used in the program and as is the case for the record, is attached to each of the subsequent three Information

أ - اللاحق مشكولا (voweledform)

ب -اللاحق دون علامات التشكيل (unvoweledform)

ج-التوصيف (desc)

د- الصنف (classe): Where were classified into three categories suffixes:

- الصنف N ونعني به اللواحق التي لا تدخل إلا على الأسماء
- الصنف V ويرمز للواحق الخاصة بالأفعال

- الصنف C ويرمز للواحق المشتركة بين الأسماء والأفعال

وقد تم تقسيم كل صنف من هذه الأصناف إلى الأقسام الفرعية التالية

- C1: يضم ‘‘ الذي يعني غياب اللاحق
- C2: يضم الضمائر البسيطة المسندة إلى المخاطب
- C3: يضم الضمائر البسيطة المسندة إلى الغائب
- N1: ويضم ياء النسبة
- V1: يضم نون الوقاية ياء المتكلم
- V2: يضم نون الوقاية + باقي ما يلحق بها من الضمائر
- V3: يضم الضمائر المركبة من اللواحق من القسم C2 و C3
- V4: يضم واو الجماعة+ ما يلحق بها من الضمائر

1. الأدوات **db/specialwords/toolwords.xml**

وقد أرفقت كل أداة بالمعلومات التالية:

أ – الأداة دون علامات التشكيل (unvoweledform)

ب –الأداة مشكولة (voweledform)

ج– النوع (type):

د– أقسام السوابق التي تدخل عليها prefixeclass

هـ– أقسام اللواحق التي تدخل عليها suffixeclass

و – **أسماء الأعلام db/specialwords/propernouns.xml**

ويحتوي هذا الملف على 2040 اسما علما تم ضبطها بعلامات التشكيل كما في المثال التالي:

```
<propernoun unvoweledform="محمد"  voweledform= "مُحَمَّد"/>
```

2. الأوزان

Has been placed in folders db / nouns / patterns your weights names and db / verbs / patterns weights private acts. And each comprising two volumes, the first containing the weights is Almhkolh db / nouns / patterns / Unvoweled) for the names and db / verbs /

patterns / Unvoweled for acts (and the second to the formations of these weights db / nouns /) patterns / Voweled) for the names and db / verbs / patterns / Voweled for acts

الأوزان غير المشكولة db/nouns/patterns/Unvoweled و db/verbs/patterns/Unvoweled

These two volumes contain XML files containing the formula weights of nouns and verbs is Almhkolh. Has been divided these weights according to their length, for example includes UnvoweledNominalPatterns2.xml nominal weights Almhkolh is composed of two characters View while featuring UnvoweledVerbalPatterns7.xml actual weights Almhkolh is composed of .seven letters file

تم إرفاق كل وزن بالمعلومات التالية

- value: وترمز للوزن السماعي (مثلا فع هو وزن لكلمة مَرَّ)
- rules: وترمز لموقع الحروف الأصلية للجذر داخل الوزن مما يمكن من استخلاص الجذر
- ids: ترمز للتشكيلات الممكنة لهذا الوزن

الأوزان المشكولة db/nouns/patterns/Voweled و db/verbs/patterns/Voweled

These two volumes contain XML files containing the formula weights of nouns and verbs Almhkolh. As is the case with its counterpart is Almhkolh has been split weights according to their length, for example includes VoweledNominalPatterns3.xml nominal weights Almhkolh consisting of three characters View while featuring VoweledVerbalPatterns4.xml actual weights Almhkolh consisting of four characters file.

Attached to each different weight depending on the type of information the floor where we encodes this information as shown in Tables 1 and 2 assigned to the interpretation of these symbols for the name and act in order.

For the names were attached weights with the following information

- id: ونعني به الرقم التسلسلي للوزن المشكول في قاعدة المعطيات
- diac: ويمثل الوزن السماعي المضبوط بعلامات التشكيل

- canonic: ويمثل الوزن القياسي المضبوط بعلامات التشكيل
- type: ويمثل نوع الاسم التي يشتق منه هذا الوزن
- cas: ويحدد حالة تعريف الاسم من تنكيره
- ncg: ويحدد عدد الاسم وجنسه وحركته الإعرابية

وبالنسبة للأفعال تم إرفاق الوزن بالمعلومات التالية

- id: ونعني به الرقم التسلسلي للوزن المشكول في قاعدة المعطيات
- diac: ويمثل الوزن السماعي المضبوط بعلامات التشكيل
- canonic: ويمثل الوزن القياسي المضبوط بعلامات التشكيل
- type: ويمثل زمن الفعل التي يشتق منه هذا الوزن
- aug: ويمثل حالة الفعل من حيث التجرد والزيادة
- cas: ويحدد حركة الفعل المضارع
- ncg: ويحدد إسناد الفعل
- trans: ويحدد إن كان الفعل لازما أم متعديا

3. الجذور

Been used in the development of the program roots attached Berqamha serial in two files: the first file db / Allroots1.txt contains extended base of the roots of 7502 root contains the second file db / Allroots2.txt the brief database contains 2,900 root. Has also been attached to these roots weights derivatives have been placed in the following four folders:

•Vols db / nouns / roots1 and db / verbs / roots1 Covenants extended to al-Qaeda and the actual nominal roots, which includes. We point out that the roots of these two volumes.

•Vols db / nouns / roots2 and db / verbs / roots2 Covenants shortcut par for the roots of al Qaeda and the actual common use, which includes the 2900 root.

These folders contain files dedicated each one of the roots of that share in the first letter XML format. Has been attached to the root of all the following information:

- val: وترمز للجذر
- vect: ويضم الأرقام التسلسلية لجميع الأوزان المشكولة لمشتقات هذا الجذر

Appendix F (System Flowchart)

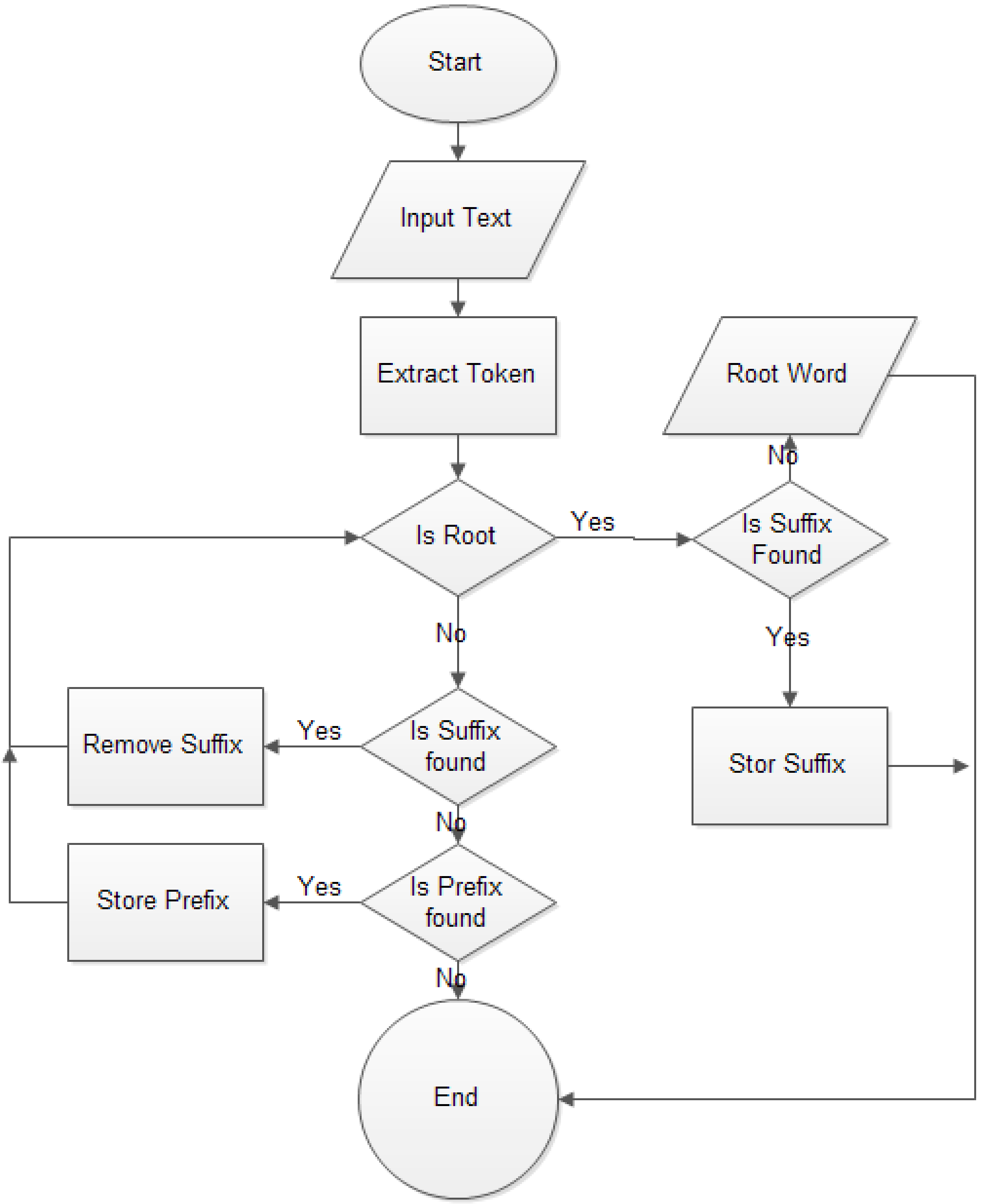


Figure 23: A flowchart for the analyzer

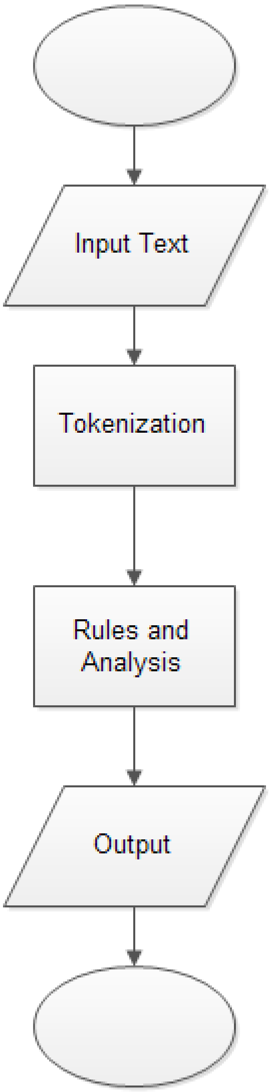


Figure 24: A flowchart for main class

Figure 25: A flowchart for morphological

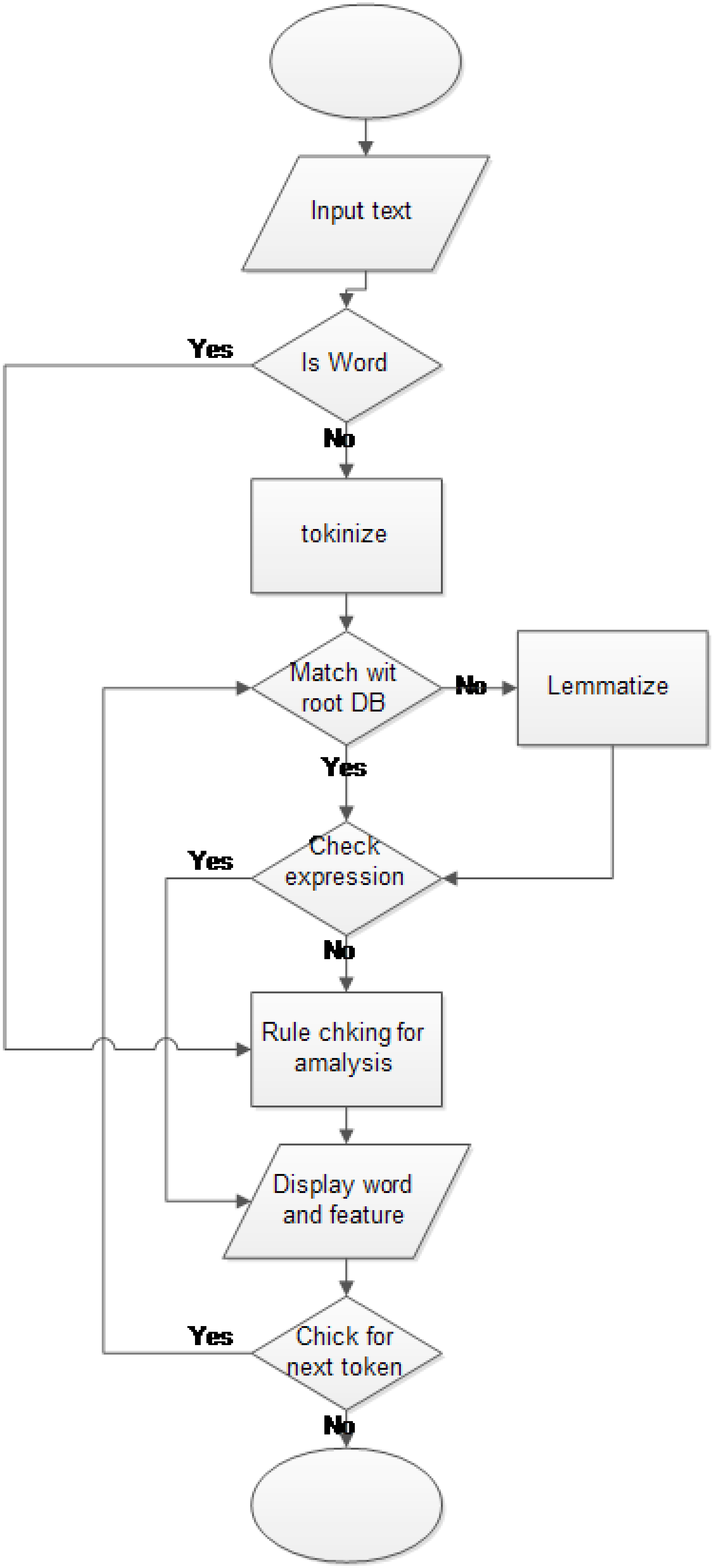


Figure 26: A flowchart for morphological

Appendix G (System Requerment)

Basic Requirements

Like all programming languages, VB.NET is nothing more than text that you type into a text editor. Computers convert that text into instructions that they can execute. Therefore, the only system requirement for doing that is a computer that can open Notepad. However, if you want a computer to convert your text into an application, it will need a copy of Microsoft's .NET Framework. Since Microsoft has been upgrading that Framework for years, different versions exist and many people have various versions installed on their computers.

.NET Framework Requirements

The latest .NET Framework version was 4.5 as of January 2013. This framework gives you the ability to create more powerful VB.NET applications than those you might build using older versions. Your computer needs a processor that runs at a speed of at least 1 GHz and it should have at least 512 MB of free RAM. The .NET 4.5 framework runs on all 32-bit and 64-bit operating systems later than Windows XP. You can download this framework from Microsoft's Microsoft .NET Framework 4.5 Web page (see Resources).

Microsoft Visual Studio Requirements

While you could use Notepad to create VB.NET apps, you'll have a much easier time using an Integrated Development Environment such as Visual Studio. This application integrates seamlessly with the .NET framework and has a toolbox containing controls you drag and drop onto forms. Visual Studio also makes it easier to debug your code, compile it into an application and distribute that application. To run Visual Studio 2012, your computer needs a 1.6 GHz processor or faster and 1GB of RAM. If you plan to run it on a virtual machine, you'll need 1.5GB of RAM. Your computer should also have a 5400-RPM hard drive and 10 GB (NTSF) of free hard drive space. In addition, the computer needs a video card capable of running DirectX 9 at screen resolutions of 1024 by 768 or higher. Visual Studio 2012 works on 32-bit and 64-bit Windows operating systems newer than Vista.

Deployment Considerations

When you deploy a Visual Basic.NET Web application, any Web surfer can view it using any browser. Desktop applications are different because they will not run on computers that do not have the .NET framework installed. Because you have the ability to target different versions of the .NET framework when you build your desktop application, your users must have the corresponding framework version installed too. For instance, if you create a desktop app that targets the .NET 3.0 Framework, people must have that framework version or higher to run the app. When advertising your application, include a message that tells people the version of the framework they'll need to make your app work.

6. References

[1] Almosallam, Ibrahim, et al. "SASSC: A Standard Arabic Single Speaker Corpus." *proceedings of 8th ISCA Speech Synthesis Workshop*. 2013

[2] I. Zitouni, J. S. Sorensen & R. Sarikaya, “Maximum Entropy Based Restoration of Arabic Diacritics”, Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL); Workshop on Computational Approaches to Semitic Languages; SydneyAustralia, July 2006;

[3] Rashwan, M., et al. "A hybrid system for automatic arabic diacritization." *The 2nd International Conference on Arabic Language Resources and Tools*. 2009.

[4] Habash, Nizar, and Owen Rambow. "Arabic diacritization through full morphological tagging." *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Association for Computational Linguistics, 2007.

[5] Roth, Ryan, et al. "Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking." *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008.

[6] Shaalan, Khaled. "Rule-based approach in Arabic natural language processing."*The International Journal on Information and Communication Technologies (IJICT)* 3.3 (2010): 11-19..

[7] Attia, Mohammed, Lamia Tounsi, and Josef van Genabith. "Automatic Lexical Resource Acquisition for Constructing an LMF-Compatible Lexicon of Modern Standard Arabic." Technical report, The NCLT Seminar Series, DCU, Dublin, Ireland, 2010.

[8] Mohamed Attia Mohamed Elaraby Ahmed,” A LARGE-SCALE COMPUTATIONAL PROCESSOR OF THE ARABIC MORPHOLOGY, AND APPLICATIONS”,2000.

[9] Daniel Jurafsky and James H. Martin” An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition” , 1999.

[10] Microsoft website, http://research.microsoft.com/en-us/groups/nlp/ , June 2014.

[11] Attia, Mohammed A. *Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation*. Diss. University of Manchester, 2008..

[12] Ali, EL-Desoky, Marwa Fayz, and Doaa Samir. "A smart Dictionary for the Arabic Full-Form Words." International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-5, November 2012.

[13] Mohammad Ahmed Sayed Ahmed Ahmed Al Badrashiny, “AUTOMATIC DIACRITIZER FOR ARABIC TEXTS”, Thesis, Publication date: June 2009.

[14] Encyclopedia_of_Lingguistics_Volume_1 2005.

[15] Ryding, K. C. A reference grammar of modern standard Arabic. Cambridge: Cambridge University Press 2005

[16] Watson, Janet CE. The phonology and morphology of Arabic. Oxford university press, 2007.

[17] Habash, N. Y. . Introduction to Arabic Natural Language Processing. UK: Morgan & Claypool Publishers. 2010

[18] Owens, Jonathan. A linguistic history of Arabic. Oxford: Oxford University Press, 2006.

[19] Owens, Jonathan. *The foundations of grammar: An introduction to medieval Arabic grammatical theory*. Vol. 45. John Benjamins Publishing, 1988

[20] Cooper, Robin, and Aarne Ranta. "Natural languages as collections of resources." *Language in Flux: Relating Dialogue Coordination to Language Variation, Change and Evolution. College Publications, London* (2008).

[21] Schlippe, Tim, ThuyLinh Nguyen, and Stephan Vogel. "Diacritization as a machine translation problem and as a sequence labeling problem." *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas (AMTA), Hawai'i, USA*. 2008.

[22] Rashwan, M., et al. "A hybrid system for automatic arabic diacritization." *The 2nd International Conference on Arabic Language Resources and Tools*. 2009.

[23] Nelken, Rani, and Stuart M. Shieber. "Arabic diacritization using weighted finite-state transducers." *Proceedings of the ACL Workshop on Computational*

Approaches to Semitic Languages. Association for Computational Linguistics, 2005.

[24] Diab, Mona, Mahmoud Ghoneim, and Nizar Habash. "Arabic diacritization in the context of statistical machine translation." *Proceedings of MT-Summit*. 2007.

[25] Zitouni, Imed, Jeffrey S. Sorensen, and Ruhi Sarikaya. "Maximum entropy based restoration of Arabic diacritics." *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006.

[26] Alghamdi, Mansour, Zeeshan Muzaffar, and Hazim Alhakami. "Automatic restoration of arabic diacritics: a simple, purely statistical approach." *Arabian Journal for Science and Engineering* 35.2 (2010): 125.

[27] Elshafei, Moustafa, Husni Al-Muhtaseb, and Mansour Alghamdi. "Statistical methods for automatic diacritization of Arabic text." *The Saudi 18th National Computer Conference. Riyadh*. Vol. 18. 2006.

[28] Habash, Nizar, Owen Rambow, and Ryan Roth. "Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization." *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt*. 2009.

[29] Vergyri, Dimitra, and Katrin Kirchhoff. "Automatic diacritization of Arabic for acoustic modeling in speech recognition." *Proceedings of the workshop on computational approaches to Arabic script-based languages*. Association for Computational Linguistics, 2004.

[30] Ramanathan, Ananthakrishnan, et al. "Case markers and morphology: addressing the crux of the fluency problem in English-Hindi SMT." *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009.

[31] Haertel, Robbie A., Peter McClanahan, and Eric K. Ringger. "Automatic diacritization for low-resource languages using a hybrid word and consonant CMM." *Human Language Technologies: The 2010 Annual Conference of the*

North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010.

[32] ww.sakhr.com/nlp.aspx ,December 2014

[33] https://open.xerox.com/Services/arabic-morphology, December 2014.

[34] http://www.rdi.eg.com/ , December 2014

[35] http://www.facstaff.bucknell.edu/RBEARD/lexbase.html, December 2014

[36] https://catalog.ldc.upenn.edu/LDC2004L02 , December 2014

[37] Habash, Nizar, and Owen Rambow. "MAGEAD: a morphological analyzer and generator for the Arabic dialects." *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006.

[38] Attia, Mohammed A. "Developing a robust Arabic morphological transducer using finite state technology." *8th Annual CLUK Research Colloquium, Manchester, UK*. 2005.

[39] Attia, Mohammed, et al. "An open-source finite state morphological transducer for modern standard Arabic." *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*. Association for Computational Linguistics, 2011.

[40] Attia, Mohammed, et al. "A lexical database for modern standard Arabic interoperable with a finite state morphological transducer." *Systems and Frameworks for Computational Morphology*. Springer Berlin Heidelberg, 2011. 98-118.

[41] Hattab, Abdullah Mamoun, and Abdulameer Khalaf Hussain. "HYBRID STATISTICAL AND MORPHO-SYNTACTICAL ARABIC LANGUAGE DIACRITIZING SYSTEM." *International Journal of Academic Research* 4.4 (2012).

[42] Microsoft Encarta Dictionary, 2007