
# Statistical Arabic Grammar Analyzer Based on Rules Mining Approach Using Naïve Bayesian Algorithm

# محلل نحوي عربي إحصائي مبني على منهج التنقيب عن القواعد باستخدام خوارزمية النايف بييزين

Prepared by

**Ahmad Wasef Alfares**

Supervisor

**Dr. Ahmad Adel Abu-shareha**

**Thesis Submitted in Partial Fulfillment of the Requirements**

**for the Degree of Master of Computer Science**

**Department of Computer Science**

**Faculty of Information Technology**

**Middle East University**

**January, 2017**

# AUTHORIZATION STATEMENT

I, Ahmad Wasef Alfares, authorize the Middle East University to provide hard copies or soft copies of my Thesis to libraries, institutions or individuals upon their request.

**Name: Ahmad Wasef Alfares**

**Date: 18/01 /2017**

**Signature:** 

## اقرار تفويض

انا أحمد واصف الفارس افوض جامعة الشرق الاوسط بتزويد نسخ من رسالتي للمكتبات او المؤسسات او الهيئات او الافراد عند طلبها.

الاسم : أحمد واصف الفارس

التوقيع : 

التاريخ : 2017/ 01/ 18

Middle East University

Examination Committee Decision

**This is to certify that the thesis entitled "Statistical Arabic Grammar Analyzer Based on Rules Mining Approach Using Naïve Bayesian Algorithm" was successfully defended and approved in 18 / 01 / 2017.**

**Examination Committee MembersSignature**

**Dr. Ahmad Abu-Shareha (Supervisor)**

Assistant Professor, Department of Computer Information Systems

Middle East University (MEU)

**Dr. Fayez Alshrouf(Internal Member)**

Associate Professor, Department of Computer Science

Middle East University (MEU)

**Dr. Bassam Hammo (External Member)** Bassam Hammo

Professor, Department of Computer Information Systems

Jordan University (JU)

# ACKNOWLEDGMENT

I would like to thank to almighty God for blesses which enabled me to achieve this thesis.

This thesis would not have been possible without the support of many people.

I would like to thank my soul mother and the spirit of my father asked God's mercy and forgive him and freed his neck from the fire, since the continuous moral support.

I would like to express my sincere appreciation and great thanks to my supervisor Dr. Ahmad Adel Abu-shareha for his guidance, who read my numerous, helping and encouraging my efforts during this research, supporting during writing and motivation throughout Master's Thesis.

Continuously, Many thanks to my supervisor Dr.Ahmad Adel Abu-shareha.

I would like to thank deeply Prof. Bassam Hammo (Faculty King Abdullah II School for IT, Department of Computer Information Systems – Jordan

University - Jordan), for helping and support in Arabic Natural Language Processing, additionally, a big thanks for reading and revising my thesis, since he is one of Examination Committee Members of my thesis.

I would like to thank my best friends. They were always standing with me through the good times and bad also, specifically to my dear friend and honest brother Mr. Said Abd Alrabei'a Alsaaidah.

I would like to thank my friends. They were give me the Arabic corpus which I use it in my thesis, Mr. Michael Nawar Ibrahim, Mr., Mahmoud N. Mahmoud, and Ms. Dina A. El-Reedy (all of them are master's degree from the Faculty of Engineering, Cairo University-Egypt).

I would like to thank Prof. Majdi Shaker Sawalha (Faculty King Abdullah II School for IT, Department of Computer Information Systems – Jordan University - Jordan), for his support in consultation in my thesis.

# DEDICATION

This Thesis is dedicated to the people who gave me everything and waited from me nothing in return… my parents Wasef Alfares and Fawzeia Khader, My beloved wife Suha Alkayed who endured this long process with me, always offering support and love.

To my lovely Sisters Fanan, and Taif and their husbands Dr. Zeiad Abo Qadoora and Anas Abo Qadoora, dear brothers Mohammad and Ali, to my lovely kids Fares, Abdullah, Fanan and Wasef , who endured this long process with me.

To my Dear grandparents Ahmad and Safeia'h, aunt Kefaya Ahmad Alfares and my uncle Wasfy Ahmad Alfares.

To my dear friend Mr. Said Abd Alrabei'a Alsaaidah, for his support in all times and especially in Critical and difficult conditions.

To my dear friend Dr. Motaz Khaled Saad (Faculty of IT, Islamic University of Gaza – Palestine), for his theoretical and technical support.

To my dear friend Dr. Ahmad Alqurneh (Faculty of IT, Middle East University– Jordan), for his theoretical support in NLP.

To my dear friend Dr. Ahmad Mohammad AbdelKhaleq Obaid (Future University in Qairo – Egypt), for his theoretical support and consultations.

To my dear friend Mr. Faris Alsmadi (Computer Center, Jordan University ,Amman –Jordan), technical support in Java.

To my dear brother and beloved friend Mr. Abobakr Bagais(Computer Science, King Abdulaziz University, Jeddah –KSA), theoretical, technical and formatting support especially in Critical and difficult conditions.

To my dear brother and beloved friend Mr. Abdelaziz Mirad-Abo Hamzah (Master Degree in Artificial Intelligence, especially in Arabic NLP-Algeria) for his support in theoretical, technical and formatting support especially in Critical and difficult conditions.

To my dear brother and beloved friend Mr. Mohamed Labidi (Master degree in Artificial Intelligence, especially in Arabic NLP, Higher Institute of Computer Science and Communication Technologies, Hammam Sousse,

Tunisia) for his support in theoretical, technical and formatting especially in Critical and difficult conditions.

# LIST OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF APPINDEXES

# محلل نحوي عربي إحصائي مبني على منهج التنقيب عن القواعد باستخدام خوارزمية النايف بييزين

إعداد

**أحمد واصف الفارس**

اشراف

**د.أحمد عادل أبوشريحة**

## الملخص

تواجه جمل اللغة العربية تحديا وذلك لأنها كثيرا ما تحمل أكثر من معنى واحد. والذي يحدد المعنى المطلوب هو التحليل النحوي(الإعراب). ويعرف التحليل النحوي على أنه عملية تحديد قسم الكلام النحوي/الإعرابي والحالة الإعرابية والحركة الإعرابية (على اخر حرف بالكلمة) لكل كلمة في الجملة. وهناك منهجين رئيسين يستخدمان في التعامل مع التحليل النحوي في اللغة العربية وهما المنهج القاعدي والمنهج الإحصائي. ومن ناحية أخرى فإن المنهج القاعدي يعاني من العديد من السلبيات ومنها محدودية مقدراته في التعامل مع الجمل حيث يتعامل مع القصيرة منها حصرا, وكذلك احتياجه لجهد كبير للحصول على المعرفة والموارد اللغوية واستهلاكه للوقت كذلك. أضف إلى ذلك فإن طبيعة حرية ترتيب الكلمات في الجملة العربية من جهة وحذف الضمير الشخصي من جهة أخرى يزيد الصعوبة ليس فقط في المنهج القاعدي ولكن أيضا في بناء قاعدة متحررة من السياق (CFG) وكَفُؤَة. وفي هذه الرسالة تم اقتراح منهج لحوسبة التحليل النحوي العربي في محاولة للتغلب على المشاكل والعقبات التي تنشأ من استخدام المنهج القاعدي. ويتضمن المنهج المقترح أربعة مراحل وهي: مرحلة المدخلات ومرحلة استخراج الخصائص وبناء البيانات المهيكلة ومرحلة التعليم ومرحلة الاكتشاف/الالتقاط. ففي المرحلة الأولى فإن كل كلمة يتم عنونتها بتحليلها النحوي الخاص بها يدويا. وفي المرحلة الثانية يتم استخراج 14 خاصية لكل كلمة من جمل الكوربس. وفي المرحلة الثالثة والتي تسمى مرحلة التعليم, يتم إدخال كوربس الجمل المعنونة للنظام

والذي بدوره يرسله لمصنف نموذج خوارزمية النايف بييز المُنشَأ. وفي المرحلة الرابعة والتي تسمى الاكتشاف/الالتقاط يتم إرسال كوربس الجمل غير المعنونة لعملية استخراج الخصائص بالمرحلة الثانية وباستخدام النموذج المُنشَأ بالمرحلة الثالثة وذلك لاختيار التحليل النحوي الأكثر صحة للكلمة. ومن بعض الخصائص التي استخدمت: التعريف, الزمن, الصيغة, الحالة الإعرابية, قسم الكلام. وعلى الرغم من وجود بعض المحددات (مثل: الطول المحدود للجمل المستخدمة, محدودية مجموعة الخصائص المستخدمة, ليس كل الكلمات يمكن تجذيرها بوضوح). كانت النتائج مرضية مع دقة كافية 75.38٪. وفي الختام، فإن الطريقة المقترحة هي محاولة لحل غموض الجمل العربية عن طريق جعل التحليل النحوي عملية أكثر سهولة.

**الكلمات المفتاحية:** معالجة اللغة العربية الطبيعية, التحليل النحوي العربي الإحصائي, التشكيل, المحلل النحوي, الصرف الإعرابي, التعلم الالي الإشرافي

# Statistical Arabic Grammar Analyzer Based on Rules Mining Approach Using Naïve Bayesian Algorithm

Prepared by

**Ahmad Wasef Alfares**

Supervisor

**Dr. Ahmad Adel Abu-shareha**

## ABSTRACT

Arabic sentences have always been a challenge because they, mostly, may carry more than one meaning. What determines the desired meaning is grammar analysis. Grammar analysis is the process of determining the grammatical tag, grammatical case and grammatical diacritic (at the last character in the word) of each word in an Arabic sentence. There are two approaches to deal with grammar analysis for arabic language which are: rule–based approach and statistical approach. However, rule-based approach suffers from various drawbacks, such as the limitation of its capabilities in dealing with short sentences only, required much hard-to-get language knowledge/resources and time consumption. Additionally, the free word order nature of Arabic sentences from one hand and the presence of an elliptic personal pronoun from other hand increase the difficulty not only for rule-based approach, but also for building an efficient context free grammar (CFG). In this thesis, an approach has been suggested to automate Arabic grammar analysis attempting to overcome the problems and setbacks that emerged in using the rule-based approach. The proposed approach consists of four stages: inputs stage, features extraction and building structured data stage, the learning stage and the discovery stage. In the First stage, each word in a sentence is annotated with its corresponding grammar analysis manually. In the second stage, a 14 features were

extracted for each word in sentences of the corpus. In the third stage, which called the learning stage, the annotated corpus of sentences is entered to the system which subjected to the classifier of the Naive Bayes algorithm model was constructed. In the fourth stage, which called the discovery stage, a non-annotated corpus of sentences subjected to features extraction process in the second stage and using the constructed model resulted in the third stage, to choose the most correct grammar category. Some of features used are: state, voice, aspect, mood, case, part-of-speech (POS). Although, there are some limitations (e.g.: the limited length of the utilized sentences, limited set of utilized features, not all words can be rooted clearly), the results were satisfactory with adequate accuracy of 75.38 % for 7204 sentences. In conclusion, the proposed method is an attempt to resolve the ambiguity of Arabic sentences by making grammar analysis an easier process.

**Keywords**: Arabic Natural Language Processing, Statistical Arabic Grammar Analysis, diacritization, Grammar analyzer, Inflectional Morphology, Supervised Machine Learning

# CHAPTER ONE

# INTRODUCTION

Arabic ranks fifth in the world's league table of languages, with an estimated 255 million native speakers **(Alansary & Nagi, 2014)**. As the language of the Qur'an, the holy book of Islam, it is also widely used throughout the Muslim world. It belongs to the Semitic group of languages which also includes Hebrew and Amharic, the main language of Ethiopia.

Natural language analysis serves as the basic block upon which natural language applications such as machine translation, natural language interfaces, and speech processing can be built **(Othman, Shaalan, & Rafea, 2003)**. A natural language parsing system must incorporate three components of natural language, namely, lexicon, morphology, and syntax. As Arabic is highly derivational, each component requires extensive study and exploitation of the associated linguistic characteristics. Arabic grammar is a very complex subject of study; even Arabic-speaking people nowadays are not fully familiar with the grammar of their own language.

Thus, Arabic grammatical checking is a difficult task. The difficulty comes from several reasons: the first is the length of the sentence and the complex Arabic syntax, the second is the omission of diacritics (vowels) in written Arabic', and the third is the free word order of Arabic sentence **(Shalaan, 2005)**.

The modern form of Arabic is called Modern Standard Arabic (MSA). MSA is a simplified form of classical Arabic, and follows the same grammar. The main differences between classical and MSA are that MSA has a larger (more modern) vocabulary, and does not use some of the more complicated. Arabic words are generally

classified into three main categories: noun, verb and particle. While an Arabic sentence has two forms: nominal sentence and verbal sentence **(Shaalan, 2010)**.

This study help Arabic to advance like other mature languages such as English. The feasibility of speedy developing using statistical-based approach due to requiring big effort when acquiring grammatical knowledge from experts, consuming time that needed when writing and maintaining the grammar analysis, rule-based approach has inefficient behavior when using too many cases (or too many exceptions), It's virtually impossible predicting all cases (grammar analysis) covering the zone, the hardness when treating with hand-crafted grammar rules and the rule-based approach may be slow and not lending the required quickly **(Ibrahim, Mahmoud, & El-Reedy, 2016)**.

Arabic grammar analysis is the process of determining the grammatical role and case ending diacritization of each word in an Arabic sentence **(Ibrahim, Mahmoud, & El-Reedy, 2016)**. Grammatical role of a word is determined based on its relation with its dependents words in the same sentence and their role. While, grammar analysis is highly similar with parsing process, grammar analyses are flatter than regular parsing since it assigns additional information like case ending diacritization of each word. The significant of grammar analysis is embodied in that once the Arabic grammar analysis of a sentence is completed, many problems can be simply solved such as automatic diacritics, Arabic sentences correction and accurate translation **(Alqrainy, Muaidi, & Alkoffash, 2012)**. An example of the grammatically analyze the sentence " الأولاد يلعبون في الحديقة مع بعضهم" is shown in Table 1.1 **(Ibrahim, Mahmoud, & El-Reedy, 2016).**

**Table 1.1 Grammar Analysis Example**

as adapted from **(Ibrahim, Mahmoud, & El-Reedy, 2016)**

| Word in Arabic | Transliterated word | Grammatical Role | Case and Sign |
|---|---|---|---|
| الأولاد | Alawlad | Subject | Nominative with Dammah |
| يلعبون | ylEbwn | Present verb | Nominative with existing noon |
| في | Fy | Uninflected Particle | ---------- |
| الحديقة | AlHadyqp | Genitive noun | Genitive with Kasrah |
| مع | mE | Uninflected Circumstance | ---------- |
| بعض | bED | Possessive | Genitive with Kasrah |
| هم | Hm | Uninflected Pronoun | ---------- |

The grammar analysis task is strongly related to the morphological and syntactic ambiguities in Arabic language. Thus, previous works on grammar analysis have focused on implementing a set of basic NLP tasks, these are: Tokenization, Part-of-Speech Tagger (POS tagger), and morphological analyzer. These tasks are followed usually by morphological analysis and grammar analysis based on Context Free Grammar (CFG). Besides the rule that depends on CFG, almost all the advance NLP tasks can be solved using a learning based technique. In which, a supervised learning mechanism (classification) is trained using input labeled corpus and the trained model is used in the testing stage to assign the correct output for a sentence with unknown labels. To the best of our knowledge, previous work on Arabic grammar analysis have not

investigated the potential of pure learning-based approach on delivering a correct analysis of the Arabic sentences **(Ibrahim, Mahmoud, & El-Reedy, 2016)**.

## 1.1. Natural Language Processing

**Natural Language Processing (NLP)** is a field of computer science and linguistics concerning in an interactions between the computer and the natural language. It starts as a field of artificial intelligence which is branched from informatics. The linguistics concentrates on theoretical sides in Natural Language Processing while Natural Language Processing modern algorithms founded on machine learning especially statistical approach which requires knowing a number of different fields such as linguistics, computer science and statistics. The goal for NLP to make the machine analyzing and understanding the languages that human naturally understands.

## 1.2. The importance of Arabic NLP

The NLP especially in computational linguistics help in seeking on a new theories and a modern theoretical questions corresponds in the language in general and also in the processing of digital writing. Arabic ranks fifth in the world's league table of languages, with an estimated 255 million native speakers **(Alansary & Nagi, 2014)**. And as we know the Arabic countries are a world market. And the producer realizes that Arabic language which is the target language for the purchaser haven't lowest important than the source language, so they need systems helping them for solving a multiple language issues **(Sadat, Kazemi, & Farzinda, 2014)**.

## 1.3. Arabic NLP tasks helping in solving translation challenges

Ambiguity is the big challenge in Arabic syntax. Which creates a problem for many Arabic NLP tasks such as automatic diacritics, Arabic sentences correction, accurate translation. Arabic morphology is extremely inflectional, which has many (pronouns, articles and prepositions) that called affixes **(Habash N. Y., 2010)**. Arabic morphology is extremely derivational, with 10,000 root and 120 patterns **(EZZELDIN & SHAHEEN, 2012)**. No capital letters (unlike in Latin's languages) for named entities which have many translated and transliterated forms. Shortage in Arabic language resources with high capacity **(Shaalan, 2010)**. Such as corpora(set of corpuses), lexicons and dictionaries(with machine readable form) **(Saad & Ashour, 2010)**. The word order freedom. Ambiguity resulted from orthographic ambiguity. The hard resulted from morpho-syntactic complexity.

## 1.4. Arabic Natural Language Processing (NLP) tasks

- **Tokenization** It (also sometimes called segmentation) refers to the division of a word into clusters of consecutive morphemes, one of which typically corresponds to the word stem, usually including inflectional morphemes

  **(Habash N. Y., 2010)**

- **Part-of-speech tagging** (POS-tagging) is the process of automatically assigning the proper grammatical tag for each word in the text according to its context in the sentence. POS is implemented by assigning each token a lexical category. POS-tagging is usually the first step in linguistic analysis. Also,

it is a very important intermediate step to build many natural language processing applications **(Alqrainy, Muaidi, & Alkoffash, 2012)**.

- **Base phrase chunker** also known as a shallow syntactic parser, is the process of grouping related words into phrases based on their context and their dictionary-based role. Phrases, not individual words, are the base of most advance NLP process, such as machine translation, spell checking and correcting, speech recognition, information retrieval, information extraction, corpus analysis, syntactic parsing and text-to-speech synthesis systems **(Habash N. Y., 2010)**.
- **Parsing** is the process of mapping the sentence (string of words) to its parse tree. To do that, an efficient Context-Free Grammar (CFG), which defines the language rule is used, CFG in natural languages represents a formal system which describes a language by specifying how any legal text can be derived from a distinguished symbol called the sentence symbol. Furthermore, a robust syntactical analysis system to check whether the parser input sentence may generate by a given CFG is also very important step, which requires an efficient Part-Of-Speech (POS) tagging system to assign the syntactic category (noun, verb, and particle) to each word in the input sentence. The main component of the CFG is the set of production rules. For example VP → V NP, represents one of the CFG production rules that may be used to describes the context of a verbal sentence. Furthermore, CFG is represented by a recursive nesting of phrases that efficiently describes the context of all languages, which is analyzed using CFG. Arabic language as many other natural languages has nominal (NP) and verbal sentences (VP). It well known that nominal sentences begin with noun while verbal begin with verb. Parsing Arabic sentences considered a

requirement to many NLP applications like information retrieval and machine translation and others **(Alqrainy, Muaidi, & Alkoffash, 2012)**.

## 1.5. Arabic NLP and grammar analysis task:

Arabic **grammar analysis** is the process of determining the grammatical role and case ending diacritization of each word in an Arabic sentence. The grammatical role of a specific word is determined based on its relation with its dependents words in the same sentence and their roles. While, grammar analysis is highly similar with the parsing process, grammar analysis are flatter than regular parsing because it assigns additional information like case ending diacritization for each word. The significant of grammar analysis is embodied in that once the Arabic grammar analysis of a sentence is completed, many problems can be simply solved such as automatic diacritics, Arabic sentences correction and accurate translation. An example of the grammatically analyze the sentence "الأولاد يلعبون في الحديقة مع بعضهم" is shown in Table 1.1 **(Ibrahim, Mahmoud, & El-Reedy, 2016)**.

## 1.6. The difference between Derivational, Inflectional and Cliticization Morphology:

Arabic Morphology can be divided into two parts:

- **Form (Habash N. Y., 2010)**
- Concatinative: *Prefix, Suffix, Circumfix*.
- Templatic: root+pattern
- **Function (Habash N. Y., 2010)**
- *Derivational*
    - Creating new words
    - Mostly templatic

- *Inflectional*
    - Modifying features of words
        - ❖ Tense, number, person, mood, aspect
    - Mostly concatinative. **(Habash N. Y., 2010)**

**Derivational morphology** is concerned with creating new words from the source word and by which the core meaning of the source word is modified **(Habash N. Y., 2010)**. For example, the Arabic كاتب *kAtib* 'writer' is resulting from the verb كتب (*to write/ katab), in* the same way the English word *writer* is resulting from the verb *write*. **(Habash N. Y., 2010)** Derivational morphology usually involves changing the part-of-speech (POS) of the source word **(Habash N. Y., 2010)**. The derived variants in Arabic typically come from a set of relatively well-defined *lexical relations*, e.g., *location*( اسم مكان), *time* (اسم زمان), *actor/doer/active participle* (اسم فاعل) and *actor/object/passive participle* (اسم مفعول) among many others **(Habash N. Y., 2010)**. The derivation of one form from another typically involves a pattern switch. In the example above, the verb كتب (*katab)* has the root ك ت ب *k-t-b has* the pattern *1a2a3,* which is changed to derive the active participle of the verb, to the pattern *1A2i3 in order* to produce the form كاتب *kAtib* 'writer'. **So, in derivational morphology the lexeme is approximately equal to the root plus pattern. (Habash N. Y., 2010)**

**Inflectional morphology**, the core meaning and POS of the word stay intact and the extensions are always predictable and limited to a set of possible features. Each feature has a finite set of associated values. The feature-value pairs *number:plur* and *case:gen*, indicate that that particular analysis of the word وكتبه *wakutubihi* is plural in number and genitive in case, respectively. **(Habash N. Y., 2010)** Inflectional features are all compulsory and must have a specific (non-nil) value for every word. Some features

have POS restrictions **(Habash N. Y., 2010).** In Arabic, there are eight inflectional features. *Aspect*, *mood*, *person* and *voice* only apply to verbs, while *case* and *state* only apply to nouns/adjectives **(Habash N. Y., 2010)**. *Gender* and *number* apply to both verbs and nouns/adjectives. **So, in inflectional morphology the word is equal to the lexeme plus features (Habash N. Y., 2010).**

**Cliticization** (Clitics are independent meaning-bearing units that are phonologically and orthographically merged with words, either as prefixes(proclitics) or suffixes(enclitics)). Cliticization is closely related to inflectional morphology **(Habash N. Y., 2010)**. Similar to inflection, cliticization does not change the core meaning of the word. However, unlike inflectional features, which are all compulsory, clitics (i.e., clitic features) are all optional **(Habash N. Y., 2010)**. Moreover, while inflectional morphology is expressed using both templatic and concatenative morphology (i.e., using patterns, vocalisms and affixes), cliticization is only expressed using concatenative morphology (i.e., using affix-like clitics) **(Habash N. Y., 2010)**.

## 1.7. Rule-based approach drawbacks:

- Requiring big effort to acquiring grammatical knowledge from experts **(Ibrahim, Mahmoud, & El-Reedy, 2016)**.
- Consuming time that needed when writing and maintaining the grammar rules **(Shaalan, 2010)**.
- So bad when using too many cases(or too many exceptions). It's virtually impossible predicting all (grammar rules) covering the zone **(Shaalan, 2010)**.
- The hardness when treating with hand-crafted grammar rules **(Ibrahim, Mahmoud, & El-Reedy, 2016)**.

- It may be slow and not lending the required quickly **(Shaalan, 2010)**.
- In many times it can't handling with distorted data **(Shaalan, 2010)**.

## 1.8. Hypothesis

- In automating the process of grammar analysis there are features that affects mostly in determining the grammar analysis.
- A machine learning algorithm and a language models used can helps in features extraction and representation.
- machine learning-based approach for Arabic grammar analysis can be achieved by building a framework which used the determined and extracted features from an input set of annotated corpus

## 1.9. Problem Statement

In this thesis, a statistical approach used that applying supervised machine learning mechanism to extract the Arabic grammar analysis from an input set of annotated text**.**

This research will answer the following questionas:

- How to determine the most effective features that lend to automate the process of grammar analysis.
- How to extract and represent the features in the feature-vector.
- How to use the determined and extracted features in a machine learning mechanism to extract the correct grammar analysis from an input set of annotated text.

## 1.10. Objectives

This research try to utilize from an annotated corpus of text to predict the grammar analysis applying a supervised machine learning using a statistical approach.

This research will define the following objectives:

- To determine the most effective features that lend to automate the process of grammar analysis.
- To extract and represent the features in the feature-vector.
- To use the determined and extracted features in a machine learning mechanism to extract the correct grammar analysis from an input set of annotated text.

## 1.11. Research Significance

Once the Arabic grammar analysis of a sentence is completed, many problems can be simply solved such as:-

1) Automatic diacritization.
2) Grammar checking and correction.
3) Machine translation enhancing.

## 1.12. Research Contribution

1. Determine and represent the most effective features that influence the process of automatic grammar analysis for arabic language.

2. Building a supervised machine learning framework that use the determined and extracted effective features in a model, to discover or predict the most correct grammar analysis for a sentence of unknown analysis.

# CHAPTER TWO

# LITRATURE REVIEW AND RELATED WORKS

Chapter Two provides an overlook on arabic natural language processing as a whole focusing on the grammar analysis. It has three sections: Section 2.1 presents a background on natural language processing tasks. Section 2.2 presents the related works that cover the grammar analysis task. Section 2.3 is a summary.

## 2.1. Background

Grammar analysis is the process of determining the grammatical role and case ending diacritization of each word in an Arabic sentence. The grammatical role of a word is determined based on its relation with its dependents words in the same sentence and their roles. While, grammar analysis is highly similar to the parsing process, grammar analysis are flatter than regular parsing since it assigns additional information like case ending diacritization of each word. Subsequently, grammar analysis helps in diacritization process of the arabic words in the sentence. Furthermore, grammar analysis helps in grammar checker programs and the automatic or semi-automatic correcting of the sentences.

Many studies have been published over the past two decades that addressed the problem of automatic Arabic grammar analysis. However, these studies focused on simple tasks such as morphological analysis and part of speech tagging. Many frameworks were presented, such as those given by **(Attia M., 2006), (Attia M. ,2008)** and **(Daoud, 2010)**, that provided many functions to the NLP in arabic.

### 2.1.1. Diacritization

Diacritization is a means used to vocalize the Arabic letters using certain orthographic symbols. It is based, to a great extend, on the grammar analysis. Diacritics are categorized, according to its function, into two categories: lexemic diacritics, and inflectional diacritics.

The lexemic diacritics discriminate among two lexemes. "A lexeme is an abstraction over inflected word form which groups together all forms that differ only in terms of one of the morphological categories such as number, gender, aspect, or voice. The lemma is distinguished word form which serves as citation form". For example, referring to Table 2.1, the diacritization made the similar two words with totally different reading and meanings. From the other side, the inflectional diacritics discriminate inflected forms of the same lexeme.

**Table 2.1 The diacritization difference between the lexemes**

as adapted from **(Habash & Rambow, 2007)**

| | | |
|---|---|---|
| **كَاتِب** | ***kAtib*** | **'writer'** |
| **كَاتَب** | ***kAtab*** | **'to correspond'** |

Diacritics have been omitted by Arabic language users despite its importance. So, many scholars have spent much efforts to ease using the diacritics. Among those scholars **(Habash & Rambow, 2007)** who introduced an Arabic diacritization model utilizing tagger and a lexical language models. Both of the two models formed a lexical resource with many features which, in consequence, achieved great results. So, the diacritization has re-gained its role as a significant stage in natural language processing applications.

Also, **(Rashwan, Al-Badrashiny, Attia, & Abdou, 2009)** introduced an automatic Arabic diacritization system that works on a raw Arabic text. This system adopted a hybrid approach with two layers steps. The first layer approach(Statistical methods with fully non-factorizing works on full-form word), if possible, finding the most probable

diacrtizations sequence for the full-word that has the highest marginal probability through long A* lattice search and m-gram probability approximation. In other words, the first layer looks for the full word and puts all the possible probabilities to find the most right Diacritics . In case the first stage fails in its job, then the second layer approach, which is a (Linguistic factorizing analysis working on sub-form word). It breaks an Arabic word into its probable morphological portions (e.g.: prefix, root, pattern, suffix). Then, the portions of the word are subject to m-gram and A* lattice search to find the most probable diacrtizations sequence for the full-word.

The first layer is faster and more accurate than the second layer. Yet, the second is vital to use in case the first stage doesn't find the adequate results, particularly, for long size words or identical ones.

### 2.1.2. Grammar checker

**Shalaan, (2005)** developed a grammar checker for modern standard Arabic, called GramCheck, which provides services to average users such as checking the writing for specific and pre-definite grammatical errors. GramCheck detects problems and give the user a suggestion for enhancement. The system relies on a feature of relaxation approach for catching the arabic sentences with ill-formed structures and based on deeply analyzing syntax of the sentences. There are two parts of a tool

composed of firstly an arabic morphological analyzer and a standard bottom-up chart parser holding a handler for a grammatical checking.

## 2.2. Related Works

As aforementioned, the grammar analysis is the process of determining the grammatical role of each word in a sentence in natural languages. In Arabic, the grammar analysis includes, also, an additional task which is the determination of the case ending diacritization of each word too. So, the Arabic grammar analysis could not be implemented by a simple parsing technique. Another property of the Arabic grammar analysis is that it is flatter than other regular parsing tree structures. That is because the Arabic grammar does not contain finite verb phrase forms. Grammar analysis is strongly related to the morphological, syntactic, hard-to-analyze forms. Once the Arabic grammar analysis of a sentence is completed, many problems can be simply solved

**( Ibrahim M. N., 2015).** Thus, previous works, related to this field, have recommended that a set of basic NLP tasks (namely: tokenization, part-of-speech tagging, and morphological analyzing) should be used before implementing grammar analysis to make the process easier.

### 2.2.1. Rule-Based Approach

Rule-based approach is a traditional method that is concerned, mainly, with European languages.

**Al Daoud & Basata, (2009)** proposed system automates the grammar analysis of Arabic language sentences utilizing the rule-based frameworks. The proposed system consists of two consequent phases: the lexical (morphological) analysis and the syntactic analysis.

The first phase, the lexical analysis, has two tasks: the first task where the input stream (words) are broken into morphological items(morphemes). These morphemes go in two ways: the first way to form a single free form word (unbounded) , while the other to form a complexed form inflected word(bounded). The second task is assigning a suitable symbol to each lexeme(word).

The second phase is the syntax analysis . It has two tasks. The first task is determining all Arabic rules and, then, writing the equivalent Context free Grammar(CFG). The second task is choosing and building the recursive parser which has a top-down technique which is built from a group of mutually-recursive procedures that involve implementing the grammar rules for each case.

In summary, the syntax analysis receives all the tokens and finds the best grammar for the given sequence of the tokens by using CFG. The system considers only verbal sentences with different analysis forms. In addition, the proposed framework is restrict to right sentences, lexically and grammatically, with verbs in the active voice only.

**Al-Taani, Msallam, & Wedian, (2012)** this work presented an efficient top-down chart parser that parses simple Arabic sentences. The CFG has been used to represent the Arabic grammar depending, solely, on Arabic grammar rules to determine the sentence structure. The grammar rules encode the syntactic and the semantic constrains that help resolve the ambiguity of parsing Arabic sentences. The proposed parsing technique provides a promising impact on many language applications such as question answering and machine translation. That is because the source sentences are analyzed according to the grammar rules that go with the sentence's meaning. Consequently, the syntactic and semantic ambiguity is reduced.

Using the proposed top-down chart parser has advantages over the other existing approaches as follows :

i- It analyses both Arabic nominal and verbal sentences regardless their length.

ii- It uses efficient parsing techniques, the top-down chart parser, which showed the effectiveness of the system for analyzing both verbal and nominal sentences.

In summary, this study presents a parsing system using the top-down chart parser technique. The system consists of three steps: word classification, Arabic grammar identification using CFG, and parsing. The system was tested on 70 sentences with different sizes, from 2-6 words, achieving accuracy of 94.3%.

**Attia M. ,(2006)** and **Attia M. ,(2008)** used a parsing-based technique to resolve the disambiguity in Arabic texts. He constructed an Arabic parser using Xerox linguistics environment to write grammar rules and symbolics that follow the LFG formalisms. Attia verified his approach on short sentences arbitrarily that were selected from a corpus of news articles. The accuracy obtained was 92%.

**Bataineh & Bataineh, (2009)** developed a new parser aiming to analyze and extract the attributes of Arabic words. The methodology was, mainly, based on studying and analyzing the Arabic grammar rules conforming to gender and number. Then, it formulizes the rules using the CFG- the context free grammar. After that, the system uses transition networks for representing the rules, and then constructing a lexicon of words that construct the sentence structure, implementing the recursive transition network parser and evaluating the system using real Arabic sentences.

A top-down algorithm technique with a recursive transition network was used in the parser development. The efficiency of the developed parser was put to evaluationy using a sample of 90 sentences for testing. The results showed that 85.6% of the sentences were parsed successfully, 2.2% of sentences were parsed unsuccessfully and 14.4% of the sentences were not parsed for various reasons such as lexical problem(4.4%), incorrect sentences(2.2%), or not recognizable by linguists according to Arabic grammar rules(5.6%).

In conclusion, the parser was an efficient and a satisfactory system due to its remarkable performance.

**Alqrainy & Alkoffash**, (**2012**) built a parser to test whether the syntax of an Arabic sentence is grammatically right or not by building new effective Context-Free Grammar. A Top-Down technique was used in this model for parsing schemes constructing a parse from the initial symbol {S}. The method of this system that it chooses a production rule and tries to match the input sentence's words with the chosen

rule. A set of experiments were made on a dataset that holds 150 Arabic sentences. The system reached an average accuracy of 95%.

**Othman, Shaalan, & Rafea, (2003)** This paper proposed an Arabic bottom-up chart parser based on rule-based approach. This method was devised in order to analyze the Modern Standard Arabic (MSA) sentences and judge the syntax which leads to reduce their ambiguity . The process consists of performing a morphological analysis based on the Augmented Transition Network (ATN) technique which is used to signify the context-sensitive information regarding the relation of the stem and the inflectional extractions. The augmented transition network (ATN) contains a total number of 170 rules that are partitioned into 22 groups, each group has a grammatical category ( such as: the subject, the object, defined, conjunction forms, etc...) were used. Also, additional linguistic features, such as lexical and semantic features, have been used to disambiguate the sentence .

### 2.2.2. Statistical-Based Approach

**Roth, Rambow & Habash, (2009)** proposed a system called "MADA+TOKAN" which is one of the greatest well-known Arabic NLP systems. This framework implements morphological disambiguation, POS tagging, diacritization, lexicalization, lemmatization, stemming, etc...

The MADA+TOKAN system is made of two major parts : firstly , the MADA, which does a morphological analysis and disambiguation. The second part is TOKAN, which is a general tokenizer for the MADA-disambiguated text. The both parts collaborate together to find a solution to the different Arabic NLP problems. The

system, as a whole, follows a statistical-based approach. It inspects a list of all probable analysis for each word, and then chooses the analysis that match the existing context best. This is done, in addition, by support vector machine models that has 19 different weighted morphological features. The MADA+TOKAN's chosen analysis includes diacritics, lexemic, glossary, and morphological information. These all disambiguation tasks are made in one step.

**Diab M., (2009)**, devised a system called AMIRA which is a framework that was designed for Arabic tokenization, POS tagging, Base Phrase Chunking, and Named Entities Recognition. AMIRA is made of a clitic tokenizer (TOK), part of speech tagger (POS) and base phrase chunker (BPC)-shallow syntactic parser. The technology of AMIRA is built on supervised learning technique using Support Vector Machine(SVM) algorithm with implicit dependence on the knowledge of deep morphology. Therefore, in contrast to other systems such as MADA, AMIRA depends on surface data to learn generalizations.

**Manning, Klein, & Toutanova, (2003)** used the Stanford natural language processing group to develop Arabic NLP tools. This group includes a word segmenter, a part-of-speech tagger and a probabilistic parser. The dataset used in this system is the Penn Arabic Treebank. The Arabic Stanford word segmenter, Stanford tagger, and Stanford tagger are all written in java code, and based on machine learning technique using a Conditional Random Field model, Maximum-Entropy probabilistic context free grammar (PCFG) depending on the hand-parsed sentences, consecutively.

**Diab, (2007)** presented a supervised learning technique using a support vector machine algorithm(SVM) for Arabic Base Phrase Chunking(BPC). The system achieved an F-score of 96.33% over 10 base phrase chunk types. Diab versified the feature sets according to two factors: the usage of explicit morphological features, and the usage of different part of speech (POS) tag sets. 75 POS tags were used that represented information about definiteness, gender and number features.

In details, ERTS comprises 75 tags. For the current system, only 57 tags are initiated. The author developed a POS tagger based on this new set. Also the author adopted the YAMCHA sequence model based on the TinySVM classifier. The tagger trained for ERTS tag set uses lexical features of +/-4 character ngrams from the beginning and end of a word in focus. The context for YAMCHA is defined as +/-2 words around the focus word. The words before the focus word are considered with their ERTS tags. The kernel is a polynomial degree 2 kernel. The author adopt the one-vs.-all approach for classification, where the tagged examples for one class are considered positive training examples and instances for other classes are considered negative examples.

**Habash & Roth, (2009)** built the Columbia Arabic Treebank (CATiB), which is a database of syntactic analysis of Arabic sentences. CATiB is distinguished for its speed despite some constraints regarding linguistic richness. Two basic ideas encourage using the CATiB approach: 1) no annotation of redundant information and 2) using illustrations and terminology inspired by traditional Arabic syntax. The grammar analysis is done by applying a guileless parsing approach.

**Dukes & Buckwalter, (2010)** built the Quranic Arabic Dependency Treebank (QADT), which is an annotated grammar resource containing of 77,430 words from the Quran. This project offers a language training model, Hidden Markov model part-of-speech taggers, based on traditional Arabic grammar affiliated by a Linguistic research in the Quran that uses the annotated corpus, an automatic categorization of Quranic chapters, and prosodic analysis of the text.

**Khoufi, Louati , Aloulou , &Belguith,** (**2014**) presented an approach for parsing Arabic sentences based on supervised machine learning using Support Vector Machine (SVMs). This system selects syntactic labels of the sentence. This proposed method has two stages: 1) the learning stage and 2) the prediction stage. The first stage is based on a training corpus, extraction features, and a set of rules that are obtained from the corpus of learning. The second stage implements the results of learning obtained from first stage to accomplish parsing. Promising results for this method were achieved with an f-score of 99% .

**Shahrour, Khalifa, &Habash, (2015)** presented a methodology of using models with access to additional information of exact syntactic analysis and rules to offer an enhanced estimation of case and state. The expected case and state values are, then, used to re-tag the Arabic morphological tagger MADAMIRA output by choosing the best match its graded morphological analysis. They edge their retagging to nominals. Since what they are learning to expect is how to correct MADAMIRA's baseline choice (as opposite to a generative model of case-state), they also re-apply the

model on its output to repair mainly spreaded agreement errors in a way similar to **(Habash et. al., 2007)** agreement classifier.

**Habash & Rambow, (2005)** In this paper the writers extend the Morphological Analysis and Disambiguation of Arabic (MADA) system . They reused the tool and training set features, that had been used by others, to improve the results and make it easier for comparison. The improvements of results in all categories in numbers: As for WER(word error rate) the Zitouni et al. mistake was reduced by 17.2%, while the DER(diacritic error rate) error diminution was only 10.9%.

### 2.2.3. Hybrid-Based Approach

Hybrid-based approach is a combination of both the rule-based and the statistical annotated corpora approaches . This method is used for the following reasons: firstly, the shortage of language resources, such as parallel or bilingual big corpora, and secondly that the Arabic language, although it is rich and has an availability of corpora, suffers data scarcity. The above reasons encouraged several researchers to follow the rule-based approach combined with statistical annotated corpora (and that is called hybrid-based approach) for developing tools and systems in arabic natural language processing.

**Ibrahim, Mahmoud, & El-Reedy, (2016)** proposed a hybrid system composed of the learning-based and rule-based approaches for arabic grammar analysis. This system showed an adequate accuracy and easy to implement. However, the system requires deep knowledge of Arabic despite the use of learning portions availability. Many

experiments were made on a dataset that holds 600 Arabic sentences. The system reached an average accuracy of 90.44%.

When a sentence is inputted to the proposed framework , the system assigns each token an appropriate tag, case, and a sign. Then, the system determines for every token its POS tag, Base Phrase chunk and its morphological features (such as token definiteness). The rule-based system is responsible for determining the tag, case, and the sign of each word in the sentence. From the other side, the grammar analyzer input and features could be characterized as follows:

"Input': A complete sentence of Arabic words.

"Context": The whole sentence.

"Features": To extract the grammatical role of the words in the sentence.

A stemmer, POS tagger, BP chunker, and a morphological analyzer are used to extract extra morphological features of the words in the sentence. The Arabic grammar analyzer module uses stemmer to separate proclitics and enclitics of the word. Then, the POS tagger assigns an adequate POS tag to each token. Then, the base phrase chunker groups words belonging to the same phrases. Additional morphological information are extracted for each word using the morphological analyzer. Finally, it applies the Arabic grammar rules to assign a tag, case and sign for each word.

As an evaluation of this framework, the developer of the system generated 600 sentences. The 600 sentences consisted of 3452 tokens. The overall accuracy of the tokens, that have correct tag, case and sign, was 90.44% which is a good precision for this complex task.

## 2.3. Summary

Overall, the existing Arabic grammar analysis approaches concentrated, only, on short sentences with hand-crafted grammars, and that made the system very slow and not easy to evaluate. Furthermore, these approaches were run on simple verbal sentences or nominal sentences. Some researchers realized that they need a reliable Arabic grammar analyzer that can be used easily. They adopted three core approaches:

The first approach applies rule-based technique which relies on deep knowledge of Arabic morphology and grammatical rules.

The second uses statistical-based technique that uses annotated data and tries to fit a grammatical tag to every word.

The third approach is a hybrid approach which is a combination of the both techniques shown above.

Following, a table that displays a summary of the properties of the information frameworks ordered by the utilized approach and the publishing date used Table 2.2.

**Table 2.2 Summary for the properties of the frameworks ordered by the utilized approach and the publishing date**

| **Author** | **Year (Z-A)** | **Tasks** | **Technique** | **Approach Category (ORDER BY)** |
|---|---|---|---|---|
| *Shalan et.al.* | **2003** | Parsing | Bottom-up chart parser | **Rule-based approach** |
| *Attia .M* | **2008** | Parsing for disambiguation | Parsing-based technique using Xerox linguistic environment to write grammar rule and formalization that follow the LFG formalism | **Rule-based approach** |
| *Bataineh et.al.* | **2009** | Parsing | Top-Down parser | **Rule-based approach** |
| *AlTaani et.al.* | **2012** | Parsing | Top-Down chart parser using CFG and word classification | **Rule-based approach** |
| *Alqrainy et.al.* | **2012** | Parsing | Top-Down by building effective CFG | **Rule-based approach** |
| *Manning et.al.* | **2003** | POS tagging, Segmentation, Parsing | Supervised Machine Learning using Conditional Random Field model for Segmentation and Maximum-Entropy for POS tagging | **Statistical-based approach** |
| *Diab M.* | **2007** | Base Phrase Chunking | Supervised Machine Learning using SVM algorithm | **Statistical-based approach** |
| *Diab M.* | **2009** | POS tagging, Base Phrase Chunking, Named Entity Recognition | Supervised Machine Learning using SVM algorithm | **Statistical-based approach** |
| *Habash et.al.* | **2009** | Treebank for Parsing Purposes | No annotation of redundant information and using representations and terminology inspired by traditional Arabic syntax used for building models | **Statistical-based approach** |
| *Roth et.al.* | **2009** | Morphological disambiguation, POS tagging, Stemming, | Supervised Machine Learning | **Statistical-based approach** |

| | | | | |
|---|---|---|---|---|
| | | Lemmatization, Lexicalization, Diacritization, etc. | | |
| *Dukes et.al.* | **2010** | Quranic Treebank for Parsing Purposes | Annotated corpus includes training HMM POS taggers for Arabic based on traditional Arabic Grammarإعراب | **Statistical-based approach** |
| *Khoufi* | **2014** | Parsing | Supervised Machine Learning using SVM algorithm | **Statistical-based approach** |
| *Sahrour* | **2015** | Parsing | Supervised Machine Learning | **Statistical-based approach** |
| *Ibrahim et.al* | **2016** | Parsing | Morphological analyzer and Arabic grammar database are rule-based, while learning-based component using CRF classifier. | **Hybrid approach** |

# CHAPTER THREE

# PROPOSED WORK

This chapter presents the proposed model for Arabic grammar analysis. The proposed model has four stages: the first stage handles a set of sentences maintained in a corpus and each sentence is consist of words that are annotated by their corresponding grammar analysis. The second stage extracts a set of features for the arabic words, in order to built a new corpus with structured data. Then, in the third stage, a learning model is constructed by applying naïve Bayesian algorithm/classifier on the a part of structured data(with grammar analysis). After that, in fourth stage, the grammar analysis is discovered/predicted by naïve Bayesian classifier using another part of structured data (without/hidden grammar analysis). These processes will be fully explained in the following sections in this chapter.

## 3.1. Introduction

Automatic Arabic grammar analysis is an important task in natural language process (NLP) as it helps broadening the research in many related NLP tasks. However, only few researchers have worked on this issue. This was the motivation behind this work. As described in chapter two, there are two main techniques used to deal with grammar analysis for Arabic language: The rule-based technique, and the statistical-based technique. The previous works, mentioned in chapter two, premised on short sentences and used hand-crafted grammars. Therefore, long time was required to give an output. Further, the used techniques were difficult to scale with unstructured data. Also, these approaches used traditional parsing techniques (e.g.: top-down and bottom-up) by which parsers demonstrated on simple verbal or nominal sentences with short

lengths. Subsequently, this chapter proposes a statistical approach for analyzing Arabic grammar. This approach depends on analyzing the linguistic and grammatical rules and extracting the most significant features to be used in a machine learning process. The significance of this framework is in using a statistical approach in allocating the adequate grammar analysis depending on a set of determined features extracted toward analyzing the words**.** The Proposed Framework is devoted to present the general architecture of the proposed framework. As shown in Figure 3.1 ,

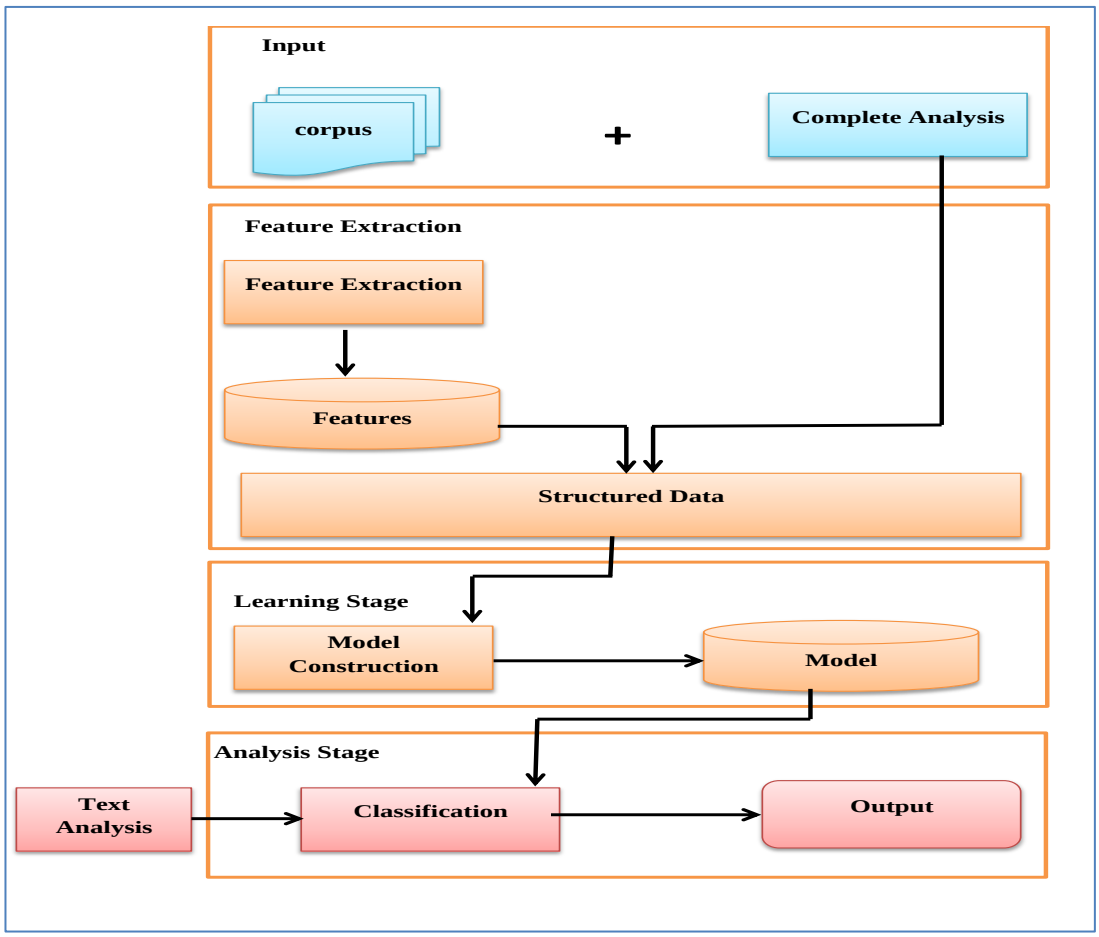



**Figure 3.1 Framework of the proposed methodology**

The presented framework is composed of four stages, inputs stage, features extraction and building structured data stage, learning stage and discovery stage. In the learning stage the framework receives as input the word, the feature-vector of each word and its

corresponding grammar analysis. These inputs are used to learn a model by applying the naïve bayes algorithm. In the discovery stage, the learning model which resulted in the learning stage will be employed in order to discover the most correct grammar analysis (which is hidden) for the words in the input sentence.

## 3.2. Determining the most effective features in Grammar Analysis

In this section, we will answer the first research question:

- **How to determine the most effective features that lend to automate the process of grammar analysis?**

  The features are overviewed in the next subsections.

### 3.2.1. Nouns

**Noun** is a part of speech that refers to persons, places, or things. Nouns can be used as "Mobtada" and "Khabar" in the nominal sentences, as a subject or a predicate in the verbal sentences, or with Particles. Among the eight inflectional features used in the Arabic language, only four of them are applied to nouns which are : case state, gender, and number.

The markers of nouns are used to distinguish nouns from verbs (الْأَفْعالُ) and particles (الْحُروفُ). The noun is characterized by the following characteristics:

**First**, the noun can be in genitive case, which can be discriminated by using the **case feature**. In this case, the noun has a "Kasraat" diacritic sign (الْجَرُّ) which is a small hyphen below the end letter of the word. If the noun is not proceeded by (ال ) double

diacritic signs, that is “Tanween”, are used and pronounced as “n”. The diacritic signs can be discriminated by using the **state feature**. Also, nouns can be preceded by a preposition (حَرْفُ جَرٍّ), and that can be discriminated by using the **proclitic1 feature**.

**Second**, **Mubtada** (المبتدأ) is another form of nouns. It is the starting word in the nominal sentences, considered as the subject. It has a nominative case and a naked pronunciation factor. Subsequently, as **Mubtada** is a noun, it can be discriminated by using **POS feature**. As a nominative, **Mubtada** can be discriminated using the **case feature**. As a naked pronunciation factor, **Mubtada** can be discriminated by using **proclitic3** and **proclitic0** features.

**Third**, **Khabar (**الخبر) is a noun that provides the information about the **Mubtada** in nominal sentences, and it is described as its predicate, with Raf'a as nominative case**.** As a noun, **Khabar** can be discriminated by using **POS feature**. As a nominative, **Mubtada** can be discriminated by using the **case feature**.

**Fourth**, **The subject (**الفاعل**)** is a noun that refers to the doer of the verb in the verbal sentences. **The subject** is always in the nominative case with a Damma (ُ), an Alif (ا) or a Waw (و). This form can be discriminated by using **POS feature** and **Case feature**.

**Fifth**, **The direct object (**المفعول به**)** is an accusative noun refers to the party that undergoes the action of the verb. There are three types of **The direct object :** a noun in accusative case (اسْمٌ مَنْصُوبٌ), a separate pronoun (ضَمِيرٌ مُنْفَصِلٌ), and an affixed pronoun (مُتَّصِلٌ ضَمِيرٌ). These types can be discriminated by using **POS feature**, **Case feature** and **enclitic0 feature**.

**Sixth**, **The genitive noun (**الاسم المجرور**)** is a noun in the genitive case as it comes after a preposition. Also, **The genitive noun** comes as the second part- the added to- of an

annexation phrase following the first term- the added "المضاف". Subsequently, both of the terms, the added "المضاف" and the added to " المضاف ليه" can be discriminated by using **POS feature** and **case feature**.

### 3.2.2. Particles

**Particle (**الحرف) is a word that does not have a meaning by itself. The purpose of particles is to signify words with different attributes. Particle is almost equivalent to English prepositions, conjunctions, articles, and other particles. The particle words' grammar analysis is, always, "Mabni". The word 'mabni' means that the particle's word's last letter is never changed regardless its position or the rule used in the sentence. In other words, words that are "Mabni" always have the same diacritics (tashkeel تشكيل) on the last letter. On the other hand, "morab", which is opposite to "Mabni", is a word that changes the form of its ending (last letter) according to its position in the sentence or the function it performs in the sentence (Grammatical Case). Unlike the alphabet ( حُرُوفُ الْأَبْجَدِيَّةِ), particles do not have a specific marker. Following types of Arabic particles are briefed.

**En'na and its 'sisters'** are particles that have a special effect on the nominal sentences. When any of these particles is added to the beginning of the nominal sentence, the subject (الْمُبْتَدَأُ) of that sentence is then called the noun of en'na "اسْمُ إِنَّ" and is put in the accusative case (مَنْصُوبٌ), while the predicate of that subject "خَبَرُإِنَّ" remains in the nominative case. These characteristics can be discriminated by using **POS feature** and **Case feature**.

**Kan and its 'sisters'**كان وأخواتها are incomplete verbs that have special effects on the nominative sentence. When any of these verbs proceeds the nominal sentence, the subject (الْمُبْتَدَأُ) of that sentence is then called the noun of "kan ".(اسْمُ كَانَ) ,and is in

the nominative case (مَرْفُوعٌ), while the predicate of that subject (الْخَبَرُ) becomes the predicate of "Kan " (خَبَرُ كَانَ) or one of its sisters, and takes the accusative case (مَنْصُوبٌ). These characteristics can be discriminated by using **POS feature** and **Case feature**.

### 3.2.3. Verbs

**Verb (**الفعل) is a word that explains the action in the verbal sentences. It can be in two parsing forms: "Mabni" or "Morab". The inflectional features which specify the verbs are Aspect, mood, person, voice, gender and number. Verb is distinguished by the characteristic of allowing prefixes. For instance, using the prefix of the Arabic letter, سَ, indicates a future tense verb which can be discriminated by using the **proclitic1 feature**. In addition, Arabic verbs allow suffix too. For example, the suffix, تْ, non-decline (consonant) indicates feminine, which can be discriminated using the **enclitic0 feature**.

**Perfect/Past Verb (**الفعل الماضي**)** indicates the past tense. This verb is always mabni (its last character's diacritic sign is not inflected/changed regardless its position in the sentence). The default grammar analysis for the past tense verb is "mabni" on "fat'h". However, it may be "mabni" on "sukun" or on "dham" in certain cases. These characteristics, the past tense verb and its diacritics, can be discriminated by using **POS feature**, **Mood feature**, **Aspect feature**, **Voice feature**, **Person feature**, **Gender feature** and **Number feature**.

**Order/Imperative Verb (**فعل الأمر**)** indicates giving instructions, commands, or prohibitions. This type of verb is always mabni. Mostly, the grammar analysis for this verb is "mabni" on "fath" . Yet, in some cases, it may be "mabni" on "sukun" or "mabni" on "dham". These characteristics can be discriminated by using **POS feature**,

**Mood feature**, **Aspect feature**, **Voice feature**, **Person feature**, **Gender feature** and **Number feature**.

**Imperfect/Present Verb (**الفعل المضارع**)** indicates that the verb tense is in the present, or in the future. This verb always stay morab. This kind of verbs has three moods: indicative, subjunctive, and jussive. Unless proceeded by a subjunctive or a jussive tool, this verb is always in the indicative mood. These characteristics can be discriminated by using **POS feature**, **Mood feature**, **Aspect feature**, **Voice feature**, **Person feature**, **Gender feature** and **Number feature**.

### 3.2.4. Adjectives

**The adjective (**النَّعْتُ) describes a noun (persons or things). Unlike English, the adjective in the Arabic language comes, always, after the one/thing that it describes "الْمَنْعُوت ". The attributive adjective agrees with the noun it describes in case, gender, definite, and number agreement. **The adjective** can be discriminated by using **POS feature** , **case feature**, **gender feature**, the **state feature**, and the **number feature**.

### 3.2.5. Adverb

**An adverb (**الظرف**)** describes the verb's time, or place. It is always in accusative case (مَنْصُوبٌ). It can be discriminated by using the **POS feature** and **Case feature**. In Arabic, the adverbs of time is called time adverbials (ظَرْفُ الزَّمَانِ) and the place adverbials (ظَرْفُ الْمَكَانِ).

### 3.2.6. Others

**Al' atf (**العطف**) ,**or junctioning, it is a set of conjunctions that junction nouns or verbs. In this type of grammar, there are two parts : the first word is called the "junctioned to" or "معطوف عليه" , and the second one is called "the junctioned" or "معطوف" . Grammatically, the second word has the same case and mood (only the jussive verb

mood) as the first one. So, the second word follows the first word in the grammar parsing (I'raab). These characteristics can be discriminated by using **POS feature**, **Case feature**, and **Proclitic2 feature**.

**The Per-mutative (**البدل**)** is a structure of two parts: the first of which is called the permutated, or "المبدل منه", which is the part being replaced, and the second part is called the permutating ,or "بدل" is the part replacing the first one. The per-mutative follows the word changes in all of its cases (inflections, declension). These characteristics can be discriminated by using **POS feature** and **Case feature.**

**Table 3.1 Summary of the features specifies the grammar analysis categories**

| Grammar Analysis Category | Features Specifies |
|---|---|
| **The Mubtada, المبتدأ** | Pos, case, state, proclitic3, proclitic0 |
| **The Khabar, الخبر** | Pos, case |
| **The adjective,النّعْتُ** | Pos, case, state, proclitic3, proclitic0, number, gender |
| **The genitive noun, الاسم المجرور** | Pos, case, proclitic1 |
| **The moḍaf, مضاف** | Pos, case, state |
| **An adverb, الظرف** | Pos, case |
| **An atf, العطف** | part of speech , case and proclitic2 |
| **The permutative البدل** | part of speech , case |
| **En'na and its 'sisters' إِنَّ واخواتها** | part of speech , case |
| **Kan and its 'sisters' كان وأخواتها** | part of speech , case |
| **The Subject الفاعل** | part of speech , case |
| **The direct objectالمفعول به** | speech, case and enclitic0 features. |
| **Perfect/Past verbالفعل الماضي** | part of speech, mood, aspect, voice, person, gender and number |
| **Order/Imperative verbفعل الأمر** | part of speech, mood, aspect, person, gender and number features. |
| **Imperfect/Present verb الفعل المضارع** | part of speech, mood, aspect, voice, person, gender and number features. |

## 3.3 Feature Extraction

This stage is designed to develop the feature(s) that suit the input which is a corpus of Arabic sentences written in Modern Standard Arabic (MSA), which is the formal language used in education and official multimedia.

The text is then passed to the MADAMIRA morphological Analyzer component which develops a list of all possible analyses (independent of context) for all the words that cover all determined morphological features of the word (POS, and 13 inflectional and cliticization features).

To produce a prediction for feature(s) suitable for words entered (independent of context), a set of models is applied for ranking the possible alternatives . Several SVM classifiers are trained to predict morphological features in the MADA uses a morphological analyzer to produce, for each input word, out-of-context, a list of analyses specifying every possible morphological interpretation of that word, covering all morphological features of the word (diacritization, POS, lemma, and 13 inflectional and clitic features) by using a Buckwalter Arabic morphological analyzer (BAMA). MADA then applies a set of models (support vector machines and N-gram=4 language models), which have 14 SVM Classifiers (one for each feature) determining a prediction for that feature value for each word and language models trained using Penn Arabic Treebank by (MSA PATB3 v3.1) to produce a prediction, per **word in-context,** for different morphological features, such as POS, lemma, gender, number or person for closed-class features, while language models with 4-gram(take 4 words to the left and right vicinity) predict open-class features such as lemma and diacritic forms as shown in

Figure 3.3. The analysis that gains most of the features will be selected by MADA. SVMs are used an Analysis Ranking component then scores each word's analysis 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(have star sign '*') is chosen as the predicted interpretation for that word in-context as shown in Figure 3.2. Also, MADA in this part work as a Morphological Disambiguators (POS taggers) given a word **in-context**, render best possible analysis as Figure 3.3 shown.

| INPUT | wsynhY | Alr}rys | jwlth | bzyArp | AlY | trkyA | . |
|---|---|---|---|---|---|---|---|
| GLOSS | and will finish | the president | tour his | with visit | to | Turkey | . |
| ENGLISH | The president will finish his tour with a visit to Turkey. | | | | | | |

```
;;; SENTENCE wsynhY Alr}ys jwlth bzyArp AlY trkyA .
;;WORD wsynhY
;;MADA: wsynhY art-NA aspect-IV case-NA clitic-NO conj-YES def-NA mood-I num-SG part-NO per-3 pos-V
       voice-ACT
*0.930061 wasayunohiy=[>anohaY_1 POS:V +IV s+ MOOD:I +S:3MS w+
       BW:wa/CONJ+sa/FUT+yu/IV3MS+nohiy/IV+(null)/IVSUFF_MOOD:I]=complete/finish/communicate
^0.780654 wasayanohaY=[nahaY-i_1 POS:V +IV s+ MOOD:I +S:3MS w+
       BW:wa/CONJ+sa/FUT+ya/IV3MS+nohaY/IV+(null)/IVSUFF_MOOD:I]=forbid/restrain
_0.739338 wasayunohaY=[>anohaY_1 POS:V +IV s+ +PASS MOOD:I +S:3MS w+
       BW:wa/CONJ+sa/FUT+yu/IV3MS+nohaY/IV_PASS+(null)/IVSUFF_MOOD:I]=be_completed/be_communicated
[ ... 7 additional options omitted ...]
```

**Figure 3.2 Ranking for weights of morphological analysis (Habash, 2016)**

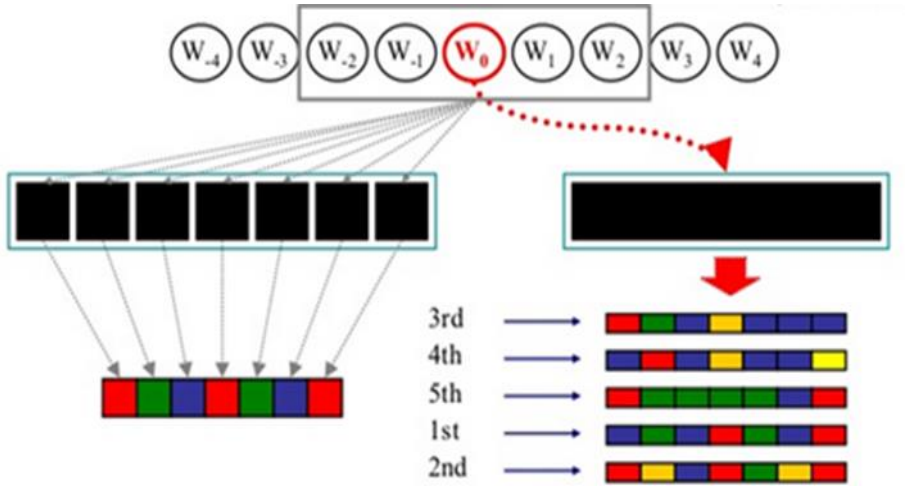



**Figure 3.3 Disambiguation by Ranking scores based on (SVM & 4-gram) language model (Habash, 2016)**

That analysis can then be used to deduce a proper tokenization for the word. MADAMIRA currently provides 11 different ways (schemes) for **Tokenization** of the input which done as follows:

•**Input:** disambiguated morphological analysis + tokenization scheme

•**Output:** highly-customizable tokenized text

**For example:** Tokenization for the sentence وسينهي الرئيس جولته بزيارة الى تركيا

| ARABIC | | | | | | | |
|---|---|---|---|---|---|---|---|
| ORIGINAL | wsynhý | Alrŷys | jwlth | bzyArħ | Alý | trkyA | . |
| GLOSS | and will finish | the president | tour his | with visit | to | Turkey | . |
| ENGLISH | The president will finish his tour with a visit to Turkey. | | | | | | |
| SCHEME | BASELINE | | | | | | |
| D1 | w+ synhy | Alrŷys | jwlth | bzyArħ | Ălý | trkyA | . |
| D2 | w+ s+ ynhy | Alrŷys | jwlth | b+ zyArħ | Ălý | trkyA | . |
| TBold | w+ synhy | Alrŷys | jwlħ + h | b+ zyArħ | Ălý | trkyA | . |
| TB | w+ s+ ynhy | Alrŷys | jwlħ + h | b+ zyArħ | Ălý | trkyA | . |
| D3 | w+ s+ ynhy | Al+ rŷys | jwlħ + h | b+ zyArħ | Ălý | trkyA | . |
| EN | w+ s+ Ânhý/VBP +S:3MS | Al+ rŷys/NN | jwlħ/NN + h | b+ zyArħ/NN | Ălý /IN | trkyA/NNP | . |
| READOFF | wasayun.hiy | Alr~aŷiy.su | jaw.latahu | biziyaraħī | Ăilaý | tur.kiyA | . |

**Figure 3.4 Tokenization for the words in a sentence (Habash N. ,2014)**

Tokenization, morphological features and parts-of-speech are all directly provided by the implemented analysis. From Figure 3.4, displays the morphological disambiguation significance since its dependency is in-context.

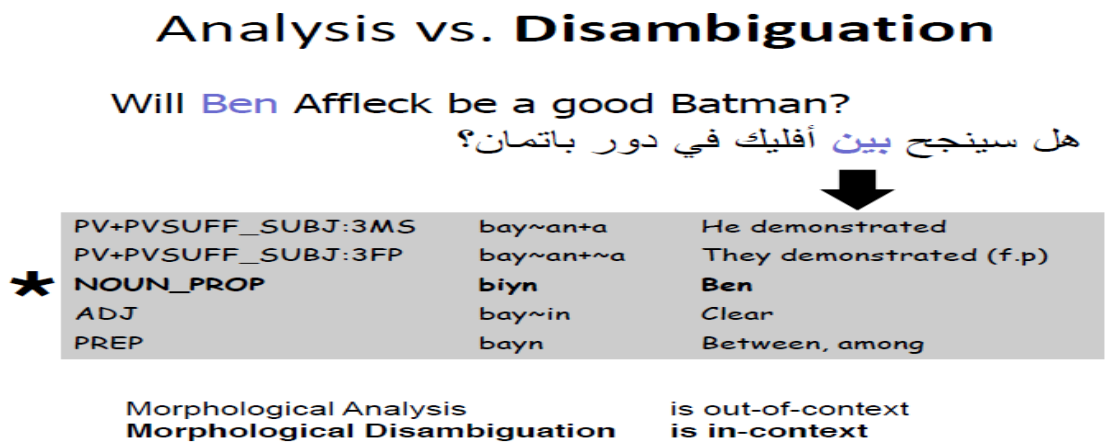
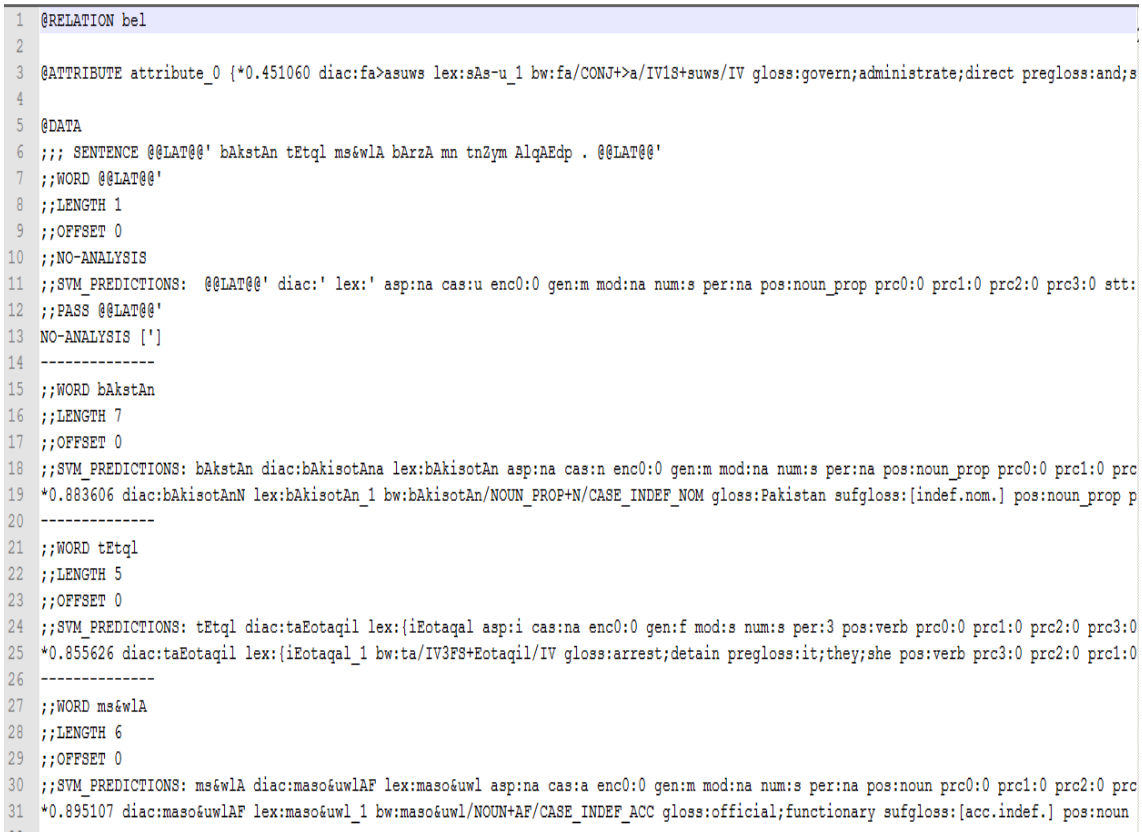



**Figure 3.5 Morphological analysis vs. disambiguation(POS-Tagging)**

**( Habash N. ,2014)**

Taking a real snapshot for the corpus of sentences after the determined features of words were extracted and before it is annotated with the grammar analysis categories shown in Figure 3.6.



```
@RELATION bel

@ATTRIBUTE attribute_0 {*0.451060 diac:fa>asuws lex:sAs-u_1 bw:fa/CONJ+>a/IV1S+suws/IV gloss:govern;administrate;direct pregloss:and;s

@DATA
;;; SENTENCE @@LAT@@' bAkstAn tEtql ms&wlA bArzA mn tnZym AlqAEdp . @@LAT@@'
;;WORD @@LAT@@'
;;LENGTH 1
;;OFFSET 0
;;NO-ANALYSIS
;;SVM_PREDICTIONS:  @@LAT@@' diac:' lex:' asp:na cas:u enc0:0 gen:m mod:na num:s per:na pos:noun_prop prc0:0 prc1:0 prc2:0 prc3:0 stt:
;;PASS @@LAT@@'
NO-ANALYSIS ['']
--------------
;;WORD bAkstAn
;;LENGTH 7
;;OFFSET 0
;;SVM_PREDICTIONS: bAkstAn diac:bAkisotAna lex:bAkisotAn asp:na cas:n enc0:0 gen:m mod:na num:s per:na pos:noun_prop prc0:0 prc1:0 prc
*0.883606 diac:bAkisotAnN lex:bAkisotAn_1 bw:bAkisotAn/NOUN_PROP+N/CASE_INDEF_NOM gloss:Pakistan sufgloss:[indef.nom.] pos:noun_prop p
--------------
;;WORD tEtql
;;LENGTH 5
;;OFFSET 0
;;SVM_PREDICTIONS: tEtql diac:taEotaqil lex:{iEotaqal asp:i cas:na enc0:0 gen:f mod:s num:s per:3 pos:verb prc0:0 prc1:0 prc2:0 prc3:0
*0.855626 diac:taEotaqil lex:{iEotaqal_1 bw:ta/IV3FS+Eotaqil/IV gloss:arrest;detain pregloss:it;they;she pos:verb prc3:0 prc2:0 prc1:0
--------------
;;WORD ms&wlA
;;LENGTH 6
;;OFFSET 0
;;SVM_PREDICTIONS: ms&wlA diac:maso&uwlAF lex:maso&uwl asp:na cas:a enc0:0 gen:m mod:na num:s per:na pos:noun prc0:0 prc1:0 prc2:0 prc
*0.895107 diac:maso&uwlAF lex:maso&uwl_1 bw:maso&uwl/NOUN+AF/CASE_INDEF_ACC gloss:official;functionary sufgloss:[acc.indef.] pos:noun
```

**Figure 3.6 Snapshot for the corpus sentences after extracting the words features**

The grammar analysis categories are the output of a deep analysis of the Arabic text corpus. The mixture of features and grammar analysis categories, mined from the training set, assigns each token of the word in a sentence to its most possible grammar analysis category by a supervised-learning algorithm.

Referring to Table 3.2, which displays the sequence order for the fourteen (14) features in the analyzed corpus as the following order in the corpus:

{f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14}.

**Table 3.2 The list of features order in the utilized corpus**

| Feature's Order | Feature Name |
|---|---|
| **f1** | Aspect(الزمن) |
| **f2** | Case(الحالة) |
| **f 3** | Enclitic0(0 لاحقة) |
| **f 4** | Gender(الجنس) |
| **f5** | Mood(الصيغة) |
| **f6** | Number(العدد) |
| **f7** | Person(الشخص) |
| **f8** | POS(قسم الكلام) |
| **f9** | Proclitic0(0 سابقة) |
| **f10** | Proclitic1(1 سابقة) |
| **f11** | Proclitic2(2 سابقة) |
| **f12** | Proclitic3(3 سابقة) |
| **f13** | State(التعريف) |
| **f14** | Voice(البناء) |

An example of the list of extracted features order (from f1 to f14) for the words of a sentence from the utilized corpus before annotated with the grammar analysis is given in Figure 3.7.

```
bAkstAn     na   n   0   m   na   s   na   noun_prop   0   0   0   0
tEtql   i   na   0   f   s   s   103   verb   0   0   0   0   na   a
ms&wlA  na   a   0   m   na   s   na   noun   0   0   0   0   i   na
bArzA   na   a   0   m   na   s   na   adj   0   0   0   0   i   na
mn      na   na   0   na   na   na   na   prep   na   0   0   0   na   na
tnZym   na   g   0   m   na   s   na   noun   0   0   0   0   c   na
AlqAEdp     na   g   0   f   na   s   na   noun_prop   Al_det   0   0   0
.       na   na   na   na   na   na   na   punc   na   na   na   na   na   na
```

**Figure 3.7 The fourteen (14) extracted features for a sentence in the corpus**

Now we manually annotate the words in the sentences of the corpus by its corresponding grammar analysis category number in which part of them are shown in Table 3.3 (all grammar analysis categories and its numbers found in Appendix A).

**Table 3.3 Part of grammar analysis categories**

| Category –number | Grammar analysis Category (الإعراب) |
|---|---|
| **0** | مبتدأ مرفوع وعلامة الرفع الضمة |
| **1** | فعل مضارع مرفوع وعلامة الرفع الضمة |
| **2** | مفعول به منصوب وعلامة النصب الفتحة |

For each word in the structured data which holds features-vectors attached with the specified grammar analysis for each word that looked like 1, 2, 3 in Figure 3.8.

1) bAkstAn, na, n, 0, m, na, s, na, noun_prop, 0, 0, 0, 0, i, na, ***0***
2) tEtql, i, na, 0, f, s, s, 3, verb, 0, 0, 0, 0, na, a, ***1***
3) ms&wlA, na, a, 0, m, na, s, na, noun, 0, 0, 0, 0, i, na, ***2***

" category-number" ***0*** is ('مبتدأ مرفوع وعلامة الرفع الضمة') :
na, n, 0, m, na, s, na, noun_prop, 0, 0, 0, 0, i, na
" category-number" ***1*** is ('فعل مضارع مرفوع وعلامة الرفع الضمة') :
i, na, 0, f, s, s, 3, verb, 0, 0, 0, 0, na, a
" category-number" ***2*** is ('مفعول به منصوب وعلامة النصب الفتحة') :
na, a, 0, m, na, s, na, noun, 0, 0, 0, 0, i, na

**Figure 3.8 Extracted features and grammar analysis category number association**

From the Figure 3.7, we notice the association between the word extracted features and its corresponding analysis **which appears for the word 'bakestan' as in the feature-vector like : (na, n, 0, m, na, s, na, noun_prop, 0, 0, 0, 0, i, na, 0** ) which represent morphological features' values for the " category-number " 0, that represent the category of specific noun case, known as:('مبتدأ مرفوع وعلامة الرفع الضمة'). Noting that na means not applicable and noun_prop means proper noun.

In order to produce the most correct grammar analysis for a certain sentence, the set of extracted features and the correct grammar analysis are used an input for the proposed framework to obtain the correct analysis using the utilized classification approach. Noting that features are extracted automatically by MADAMIRA tool, while the class/label (grammar analysis category) for a certain words in the corpus sentences were done manually.

Thus, eight inflectional features were extracted and used with the analysis in Arabic language as shown in the Table 3.4. State and case are used only with nouns and adjectives; while aspect, mood, voice and person are used only with verbs. Number and gender are used with verbs, nouns and adjectives.

**Table 3.4 Morphological inflectional features used in grammar analysis**

| Feature Name | Some Feature Values in Arabic | Some Feature Values in Arabic (Translation) |
|---|---|---|
| Person (الشخص) | 1st, 2nd, 3rd | متكلم, مخاطب, غائب |
| Aspect (الزمن) | Perfect, Imperfect, Common | ماضي, مضارع, أمر |
| Voice (البناء) | Active, Passive | للمعلوم, للمجهول |
| Mood (الصيغة) | Indicative, Subjunctive, Jussive | مرفوع, منصوب, مجزوم |
| Gender (الجنس) | Feminine, Masculine | مؤنث, مذكر |
| Number (العدد) | Singular, Dual, Plural | مفرد, مثنى, جمع |
| State (التعريف) | Indefinite, Definite, Construct | نكرة, معرفة, مضاف |
| Case (الحالة) | Nominative, Accusative, Genitive | مرفوع, منصوب, مجرور |

There are five cliticization (prefixes and suffixes) features also extracted and used to help determining the grammar analysis in Arabic language. These are shown in the following Table 3.5.

**Table 3.5 Morphological cliticization features used in grammar analysis**

| Feature Name | Some Feature Values in Arabic | Some Feature Values in Arabic(Translation) |
|---|---|---|
| Proclitic3(سابقة 3) | The 'question' proclitic<br>Interrogative Particle >a | أداة استفهام |
| Proclitic2 (سابقة 2) | The 'conjunction' proclitic<br>Conjunction fa, Connective particle fa, Response, Conjunction wa, Particle wa | حروف العطف |
| Proclitic1 (سابقة 1) | The 'preposition' proclitic<br>bi_prep, li_prep, sa_fut | حروف جر, سين الاستقبال |
| Proclitic0 (سابقة 0) | The 'article' proclitic<br>Determiner, Negative particle mA | ال التعريف, أداة نفي |
| Enclitic0 (لاحقة 0) | Enclitics(pronominal)<br>3ms_dobj, 3ms_poss | ضمير مفعول به مباشر<br>ضمير مذكر للغائب |

The proposed grammar analysis approaches depends on using set of features that affects and can be used to extract the analysis of words. These features are chosen by analyzing a set of arabic grammar categories and determine the related features that can be used for automatic generation of the word analysis.

## 3.3. The Learning Stage

The learning stage performed using 10-folds cross-validation includes the use of the output of the training corpus which produce a learning model. In order to determine a correct grammar analysis, a set of Arabic morpho-syntactic features are used as input. The determined features extracted using MADAMIRA tool affect in the grammar analysis task. The learning stage is illustrated in Figure 3.9.

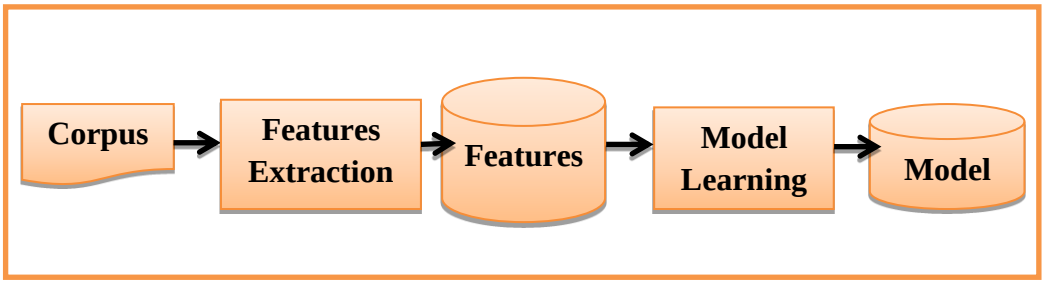



**Figure 3.9 Flowchart of the Learning Stage**

In order to perform supervised machine learning, two types of datasets must be available:

1) The first type has the input connected to the correct/expected output. Determining the correct/expected output for each data row is very important for applying supervised machine learning. The tokens and the values of the features attached to the grammar analysis, for each token in the corpus, were used for training. This dataset is considered as a "gold standard" and called the training set.
2) The second type has the input not attached with the correct/expected output, in which the input stands alone. The model (obtained from the first type) is applied on this type of data. However, at this point it hasn't any correct/expected output yet and called the testing set.

The experiments evaluation of our analyzer is achieved in cross-validation manner using the Weka tool. Thus, k-fold process was used by setting the parameter k, to ten, so the corpus and the annotated data are randomly partitioned into ten portions of equal size. In each iteration of the cross validation, nine portions were used for training the model and one portion was used for testing the model. The cross-validation process is then repeated ten times (the folds). The ten results from the folds were averaged to produce the model evaluation.

The naïve bayes classifier is based on a frequency table, which is widely used because it is often outperforms more complicated classification methods. The naïve bayes classifier, also, based on the Bayes' theorem with independent suppositions between attributes/predictors. Building a naïve Bayesian model is so easy, especially for very large dataset. That is because there is no iterative parameter estimation. Bayesian technique described the feature likelihoods that obtained from data, and then, the classification is performed by calculating the class posteriors given features.

Bayesian classifier is implemented as follows:

- Bayes theorem supplies a method for determining the posterior probability P(c|x), from P(c), P(x),P(x,c).
- Naïve bayes classifier supposes that the impact of the value of feature(x) on a given label or class(c) is separated or independent of other features.
- This supposition is called conditional independence.
- The following formula finds the posterior probability P(c|x):

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)} \quad (3.1)$$

Where the following parameters mean:

- P(c|x): is the posterior probability of class(c) given feature(x).
- P(x|c) is the likelihood which is the probability of the feature (x) given class(c).
- P(c) is the prior probability of class(c) are calculated based on their frequency in the training corpus.
- P(x) is the prior probability of feature (x) are calculated based on their frequency in the training corpus.
- P(c|x)= P(x1|c) × P(x2|c) ×… ×P(xn|c) × P(c).

An example for implementing Bayesian classification in a grammar analysis task, taking for explanation the **case** feature with value of nominative ($\boldsymbol{case} == '\mathbf{nominative}'$) with grammar analysis for **Mobtada** as follows:

- Bayes theorem supplies a method for determining the posterior probability P(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة | $case == 'nominative'$) , from $P(Mubtada$ مبتدأ مرفوع وعلامة رفعه الضمة$)$, $P(case == 'nominative')$ and $P(case == 'nominative')|$Mubtada مبتدأ مرفوع وعلامة رفعه الضمة).
- Naïve bayes classifier supposes that the impact of the value of feature ($case == 'nominative'$) on a given label or class($Mubtada$ مبتدأ مرفوع وعلامة رفعه الضمة) is separated or independent of other features (i.e.: 13 features), this supposition is called conditional independence.
- The following formula finds the posterior probability P(c|x):

  P(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة | $case == 'nominative'$) =

$$\frac{P(case == 'nominative')|\text{Mubtada مبتدأ مرفوع وعلامة رفعه الضمة} \ * \ P(\text{Mubtada مبتدأ مرفوع وعلامة رفعه الضمة})}{P(case == 'nominative')}$$

Where the following parameters mean:

- P(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة | $case == 'nominative'$): is the posterior probability of class(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة) given feature ($case == 'nominative'$).
- $P(case == 'nominative')\big|\ Mubtada$ مبتدأ مرفوع وعلامة رفعه الضمة$)$ is the likelihood which is the probability of the feature ($case == 'nominative'$) given class($Mubtada$ مبتدأ مرفوع وعلامة رفعه الضمة).
- P($Mubtada$ مبتدأ مرفوع وعلامة رفعه الضمة) is the prior probability of class(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة).

- P($case == 'nominative'$) is the prior probability of feature ($case == 'nominative'$).

For more understanding we will take a real example :

- The following formula finds the posterior probability P(c|x):

P(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة | $case == 'nominative'$) =

$$\frac{P(case == 'nominative')|\text{Mubtada مبتدأ مرفوع وعلامة رفعه الضمة} \ * \ P(\text{Mubtada مبتدأ مرفوع وعلامة رفعه الضمة})}{P(case == 'nominative')}$$

- $\boldsymbol{P(case == 'nominative')|\text{Mubtada مبتدأ مرفوع وعلامة رفعه الضمة}}$ =

4844/8729 = 0.5549

- Noting that, the number 4844 means the number of words with label of category Mubtada that have a case = 'nominative'.
- Noting that, the number 8729 means the total number of words with label of category Mubtada in the corpus.

- $\boldsymbol{P(\text{Mubtada مبتدأ مرفوع وعلامة رفعه الضمة})}$ =

8729/65430 = 0.1334

- Noting that, the number 8729 means the total number of words with label of category Mubtada in the corpus.
- Noting that, the number 65430 means the total number of words in the corpus.

- $\boldsymbol{P(case == 'nominative')}$ =

8224/65430 = 0.1256

- Noting that, the number 8224 means the total number of words with label case = 'nominative' in the corpus
- Noting that, the number 65430 means the total number of words in the corpus.

So, the result for the above formula :

P(Mubtada مبتدأ مرفوع وعلامة رفعه الضمة | $case == 'nominative'$) =

(0.5549 * 0.1334) / 0.1256 = 0.0740/0.1256 = 0.5893

## 3.4 The Discovery stage (Testing Stage)

In this stage, a new text is used with hiding the annotation of grammar analysis in order to test the training model's accuracy. This stage, actually, tests whether the classifier is able to detect the predicted/correct classification for the tokens of the corpus as the classifier trained or not. In discovery/testing stage, the performance of the model, which has been trained, is estimated. The performance of the model depends upon the size of data, the value of prediction (label/class), and the input. This stage has been performed as shown in Figure 3.10.

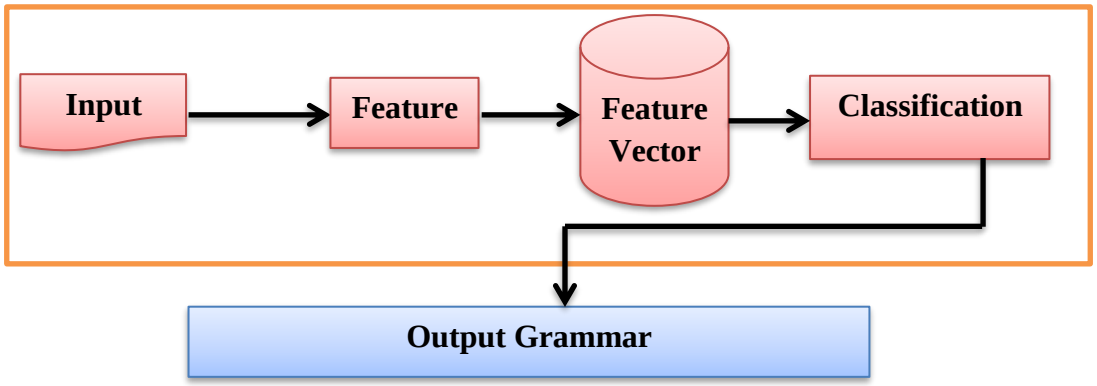



**Figure 3.10 Flowchart of the Testing Stage**

## 3.5 Summary

In this chapter, an Arabic grammar analyzer has been presented. It is based on the supervised learning approach. The purpose of this method is to allocate the suitable features by using the SVM classifiers. The process is performed as follows: the first step is to enter a corpus of words(in sentences) along with the corresponding grammar analysis categories to designate the features depending on the characteristics of the words entered. After that, the input is passed over for Morpho-Syntactic feature extraction. Then, the words and the extracted features are merged with the grammar rules of analysis to produce the Structured Data. This data is prepared and passed to apply the Supervised Machine Learning approach. This is done in two steps. The first step is called the Learning stage, while the second one is the Discovery stage. In the Learning stage, the structured data is entered to the Naive Bayes algorithm to establish a linguistic model. In the discovery stage, unlabeled/unclassed words of sentences are entered to the classifier to predict the right grammar analysis for each word based on the linguistic model established in the previous stage. In general, this method looks

promising since it depends on the supervised learning approach and the Naive Bayes model which enable analyzing relatively long sentences due to it’s the approach used.

# CHAPTER FOUR

# THE EXPERIMENTAL RESULTS

In this chapter, the experimental results for the proposed Arabic grammar analysis are presented. Section 4.1 briefly presents the dataset that is used to conduct the experiments. The details of the experimental design and techniques are given in Section 4.2. The actual results are given in Section 4.3. Finally, summary of this chapter is given in Section 4.4.

## 4.1. Dataset

The Arabic text corpus, which is used to conduct the experiments for the proposed Arabic grammar analysis, was developed by **(Ibrahim, Mahmoud, & El-Reedy,2016)**. The corpus contains enormous sentences that were collected from newspapers of general and various topics. The corpus has a total number of 65430 tokens corresponding to 48646 words contained in 7204 of individual sentences. Example sentences in the utilized corpus before they are annotated with the features set are given in Figure 4.1.

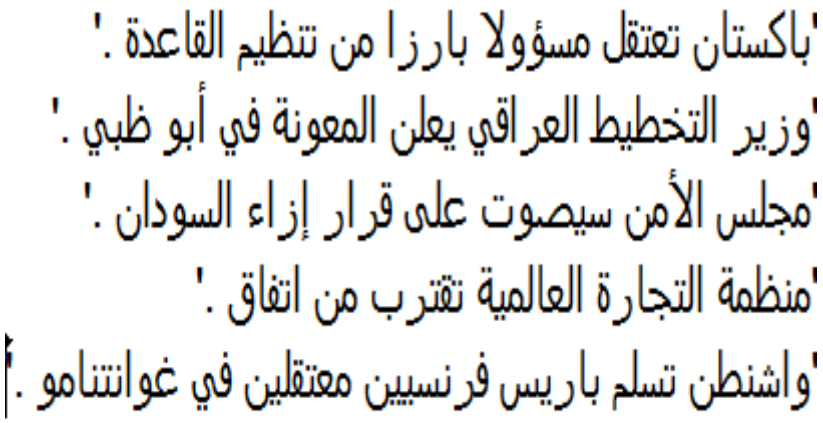



**Figure 4.1 Some individual sentences in the corpus before the annotation**

Also, example of some grammar analysis categories and a category number in the

corpus is given in Figure 4.2.

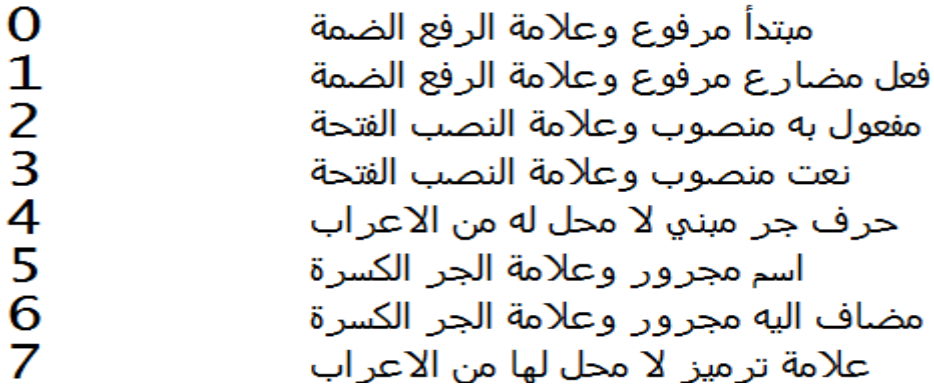



**Figure 4.2 Sample of grammar analysis categories and a category number in the corpus**

Then, each sentence in the corpus is used as input to MADAMIRA tool for Morphological Analysis and Disambiguation. MADAMIRA then extracts the assigned values for the 14 features to each word.. After that, each word in the sentence was manually annotated with its correct Arabic grammar analysis. Example of sentences in the corpus after the annotation with the extracted features and grammar analysis is given in Figure 4.3. Note that each of the grammar analysis output takes a number from 0 to 71( as given in Appendix A).

```
bAkstAn, na, n, 0, m, na, s, na, noun_prop, 0, 0, 0, 0, i, na, 0
tEtql, i, na, 0, f, s, s, 3, verb, 0, 0, 0, 0, na, a, 1
ms&wlA, na, a, 0, m, na, s, na, noun, 0, 0, 0, 0, i, na, 2
bArzA, na, a, 0, m, na, s, na, adj, 0, 0, 0, 0, i, na, 3
mn, na, na, 0, na, na, na, na, prep, na, 0, 0, 0, na, na, 4
tnZym, na, g, 0, m, na, s, na, noun, 0, 0, 0, 0, c, na, 5
AlqAEdp, na, g, 0, f, na, s, na, noun_prop, Al_det, 0, 0, 0, d, na, 6
., na, na, na, na, na, na, na, punc, na, na, na, na, na, na, 7
wzyr, na, n, 0, m, na, s, na, noun, 0, 0, 0, 0, c, na, 0
```

**Figure 4.3 Words in the corpus annotated with extracted features and grammar analysis**

## 4.2. Tools and Environment

WEKA and MADAMIRA are used in experimental design of the proposed arabic

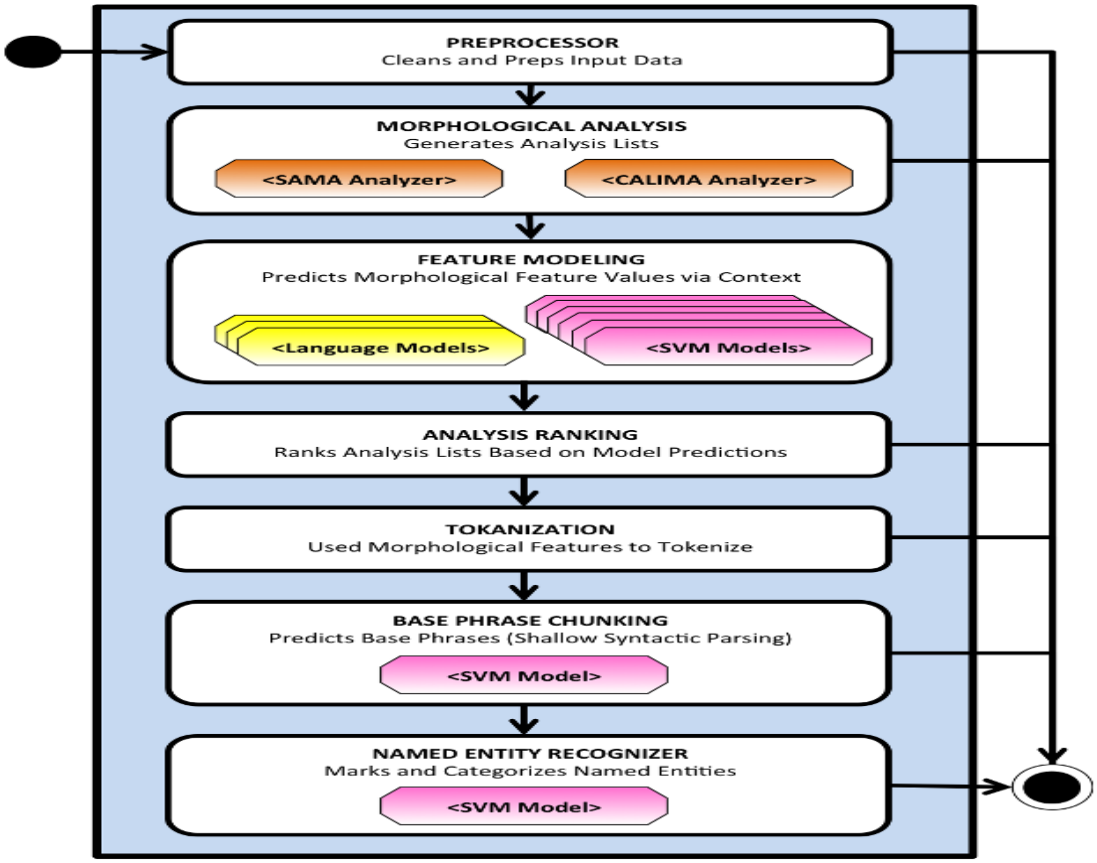



**Figure 4.4 MADAMIRA architecture overview (Pasha, et al., 2014)**

grammar analysis. WEKA (http://www.cs.waikato.ac.nz/ml/weka/) is abbreviation to Waikato Environment for Knowledge Analysis. This software tool is written in java programming language, which used in machine learning tasks. The University of Waikato in New Zealand is the developer for this freeware software, which is licensed under General Public License(GNU).

MADAMIRA tool by **(Pasha et al., 2014)** was used to extract the features that are utilized in the proposed arabic grammar analysis approach. MADAMIRA composed of two products MADA and AMIRA. MADAMIRA is a set of machine learning sub-tools that are used to analyze an Arabic text (either MSA or EGY). MADA, which relies on

deep morphological analysis and disambiguation. As described in **(Pasha et al., 2014),** Several SVM classifiers are trained to predict morphological features. These features are then used to rank the morphological analyses retrieved from a dictionary, and the analysis with the highest score is taken as the final analysis for the given word. This deep analysis results in accurate and detailed tagging albeit slower than simple SVM methods. The advantage of using MADAMIRA over using a morphological analyzer is that MADAMIRA performs contextual disambiguation of the analyses produced by the

morphological analyzer, hence reducing the possible options for analyses per word. For training data, MADAMIRA used the Penn Arabic Treebank corpus (parts 1, 2 and 3) for MSA **(Maamouri et al., 2009)**. Figure 4.4 summarizes the MADAMIRA architecture. SVM used to compute a ranked list of 14 features for each word/token.

**For another describing to the MADAMIRA work we can say that:**

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 Analysis Ranking component then scores each word's analysis 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. The chosen analyses and tokenizations can then be used by the Base Phase Chunking component to divide the input text into chunks (using another SVM model). Similarly, the Named Entity Recognizer component uses a SVM to mark and categorize named

entities within the text. The top-scoring analysis is chosen as the predicted interpretation for that word in context.

**In steps we summarize MADAMIRA work as the following :**

**First**, MADAMIRA clean the text by removing all non-textual data and converts it to the Buckwalter representation. Buckwalter is a representation of the arabic text using English characters. An example of the Buckwalter representation is given in Table 4.1.

**Table 4.1 Example of Buckwalter representation**

| Input Arabic Text | Buckwalter Representation |
|---|---|
| يُولَدُ جَمِيعُ ٱلنَّاسِ أَحْرَارًا مُتَسَاوِينَ فِي ٱلْكَرَامَةِوَٱلْحُقُوقِ. وَقَدْ وُهِبُوا عَقْلًا وَضَمِيرًا وَعَلَيْهِمْ أَنْ يُعَامِلَ بَعْضُهُمْ بَعْضًا بِرُوحِ ٱلْإِخَاءِ. | YuwladujamiyEu {ln~aAsi>aHoraArFAmutasaAwiynafiy {lokaraAmapiwa{loHuquwqi. WaqadowuhibuwAEaqolFAwaDamiyrF AwaEalayohimo>anoyuEaAmilabaEoD uhumobaEoDFAbiruwHi |

**Second**, the text is then sent to the morphological analysis component (SAMA & CALIMA Analyzers), which develops a list of all possible analyses (independent of context) for each word. The analysis produced by the morphological analyzer contains a list of morphological features that covers all the morphological characteristics of the word (diacritization, POS, lemma, and 13 inflectional and cliticization features).

**Third**, model-based decision is implemented, per-word, regardless of its context, which examines the co-presents of the different morphological characteristics for each word, such as POS, lemma, gender, number or person and produce a weight for each analysis based on two techniques, these are: Support Vector Machines (SVMs) and N-gram language models. SVMs are used for closed-class features, while language models predict open-class features such as lemma and diacritic forms.

**Fourth**, the produced analysis, based on the given weight in the previous step, is ranked

using a tuned weighted sum of matches with the predicted characteristics.

**Fifth**, the top-scoring analysis is selected as the predicted interpretation for that word in its context. The top-scoring analysis can then be used to deduce the appropriate tokenization for the word. Each word is passed to the Tokenization component to generate a customized tokenization for the word.

**Sixth**, the chosen analysis and tokenization can then be used by the Base Phase Chunking component to divide the input text into chunks (using another SVM model).

**Finally**, the Named Entity Recognizer component uses a SVM to mark and categorize named entities within the text.

As a result, tokenization, base phrase chunks and named entities, the diacritic forms, lemmas, glosses, morphological features, parts-of-speech, and stems can be extracted from by the chosen analysis.

The following Figure 4.5 and Figure 4.6 shows an examples on the difference between MADAMIRA morphological Analysis and disambiguation process.

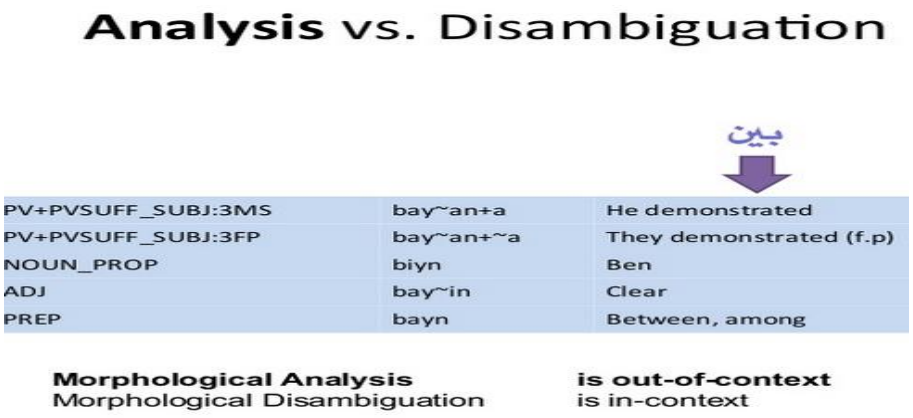



**Figure 4.5 Example of MADAMIRA morphological analysis for the word بين**

**(Habash, 2016)**

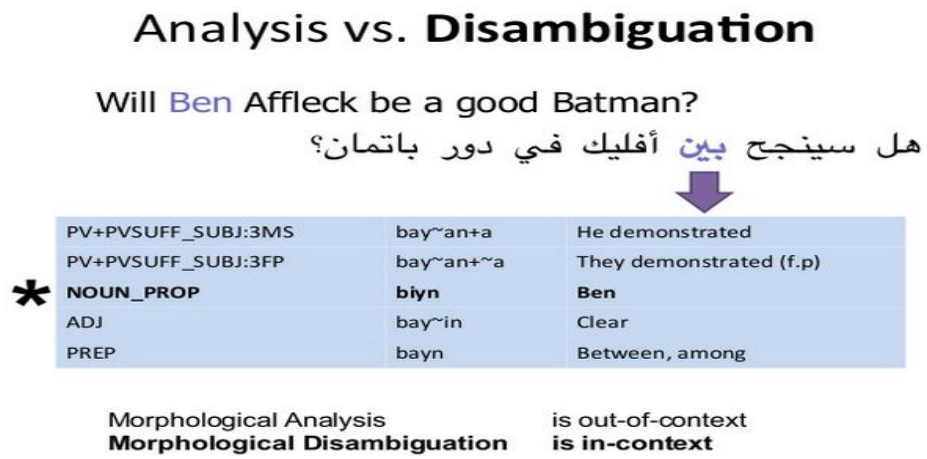



**Figure 4.6 Example on MADAMIRA morphological disambiguation for the word بين (Habash, 2016)**

As given in Figure 4.5, MADAMIRA produces all possible morphological analysis for each input word. Figure 4.5 shows the list of analyses specifying every possible morphological interpretation of that word بين, covering all morphological features of the word (diacritization, POS, lemma, and 13 inflectional and clitic features). Figure 4.6 shows that MADAMIRA selects only one morphological analysis(neighboring star) after morphological disambiguation process is done.

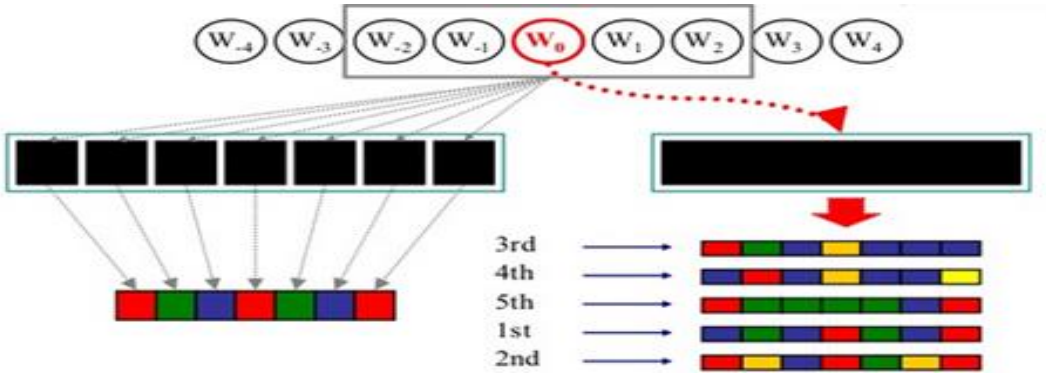



**Figure 4.7 Example on how MADAMIRA morphological disambiguation done by using language (n-gram) model works for the words in the sentence.**

**(Habash, 2016)**

As we see from Figure 4.7**,** MADAMIRA morphological disambiguation done by using language model with n-grams equal to four word. When we say that the language model with n-gram = 4, that means MADAMIRA return the disambiguated analysis of the word(W0 as shown in Figure 4.7) taking in regards the word in-context to their neighbors of words in the sentence before and after the word.

## 4.3. Experimental Results

In this section, the experiments are conducted and the results are collected. The experimental results are conducted as given in Figure 4.3.

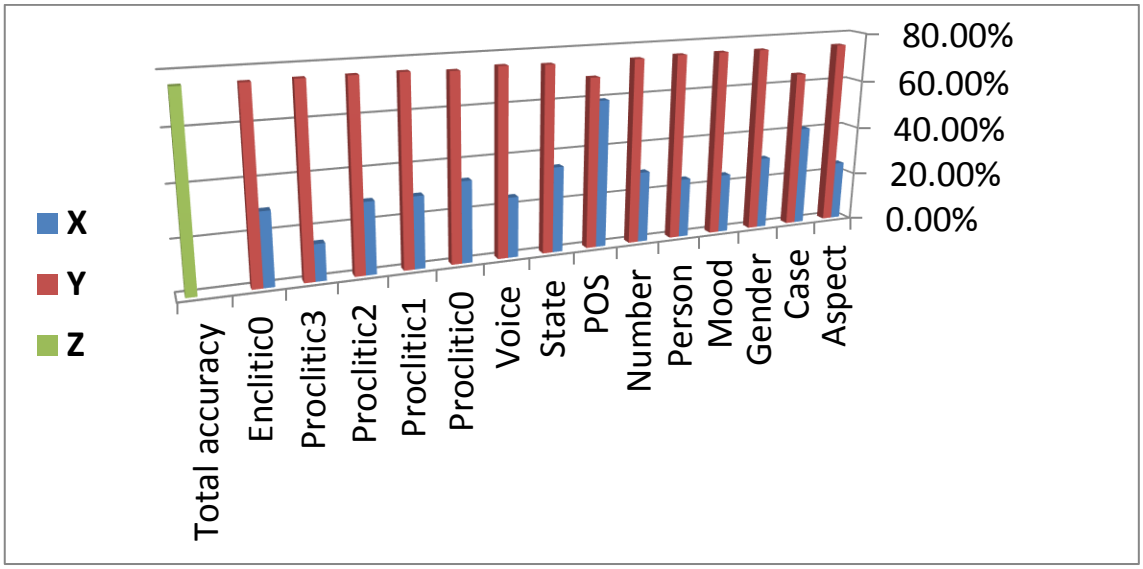



**Figure 4.8 Experimental results conducted**

Overall, the process is initiated as following: First, the input text is fed into MADAMIRA tool to extract the desirable features. The extracted tag, case, aspect, case, enclitic0, gender, mood, number, person, person, proclitic0, proclitic1, proclitic2, proclitic3, state and voice of each word is used. Then, the output is normalized and fed into WEKA tools to implement the classification task.

### 4.3.1. The Evaluation Measures

Accuracy, Precision and Recall are used in as measurements for evaluating the generated output. **Accuracy** is how close a measured value is to the **actual (true) value**, or the proportion of correct classifications (**true positives and negatives**) from **overall** number of cases. Accuracy is calculated as given in Equation 4.1. **Precision** (also called positive predictive value) is the fraction of retrieved instances that are relevant, or the proportion of correct positive classifications (**true positives**) from cases that are **predicted as positives**. Precision is calculated as given in Equation 4.2. **Recall** is (also known as sensitivity) is the fraction of relevant instances that are retrieved, or the proportion of correct positive classifications (**true positives**) from cases that are **actually positive**. Recall is calculated as given in Equation 4.3.

$$\boldsymbol{accuracy = \frac{TP}{TP+TN+FP+FN}} \quad \textbf{(4.1)}$$

$$\boldsymbol{precesion = \frac{TP}{TP+FP}} \quad \textbf{(4.2)}$$

$$\boldsymbol{recall = \frac{TP}{TP+FN}} \quad \textbf{(4.3)}$$

where, TP is the number of true positive, TN is the number of true negative, FP is the number of false positive and FN is the number of false negative.

### 4.3.2. The Results of the Proposed Approach

The overall accuracy of the proposed approach, as listed in Table 4.2, is 75.1994%, which is considered acceptable for this complex task. Each feature is used, by its own, in the task of Arabic grammar analysis. For 14 features, 14 different experiments are conducted; one feature in each experiment. The accuracy of sole feature in producing

the correct grammar analysis is collected. Moreover, the accuracy of all features excluding one of them each time are also collected, as given in Table 4.3 and illustrated in Figure 4.4. In Figure 4.4, the symbol *X* refers to using specific features, *Y* refers to use all features excluding one and *Z* refers to the accuracy of using all features.

**Table 4.2 Results of the proposed approach**

| Factor | Numbers | Percentage |
|---|---|---|
| **Correctly Classified Instances** | 49203 | 75.1994 % |
| **Incorrectly Classified Instances** | 16227 | 24.8006 % |
| **Total Number of Instances** | 65430 | |

**Table 4.3 Feature-based results**

| The Feature | Accuracy of (X) | Accuracy of All Features Excluding (X) → Y | Summarizing the Influence |
|---|---|---|---|
| **POS** | 59.37% | 68.56% | Good |
| **Case** | 40.80% | 64.10% | Good |
| **State** | 34.49% | 74.64% | Good |
| **Proclitic0** | 32.63% | 74.49% | Good |
| **Gender** | 29.61% | 75.18% | Good |
| **Number** | 28.90% | 74.70% | Good |
| **Enclitic0** | 28.86% | 75.07% | Fair |
| **Proclitic1** | 28.65% | 75.24% | Fair |
| **Proclitic2** | 28.47% | 75.21% | Fair |
| **Voice** | 24.26% | 75.23% | Fair |
| **Mood** | 24.22% | 75.23% | Fair |
| **Aspect** | 24.19% | 75.21% | Bad |
| **Person** | 24.19% | 75.21% | Bad |
| **Proclitic3** | 14.48% | 75.20% | Bad |

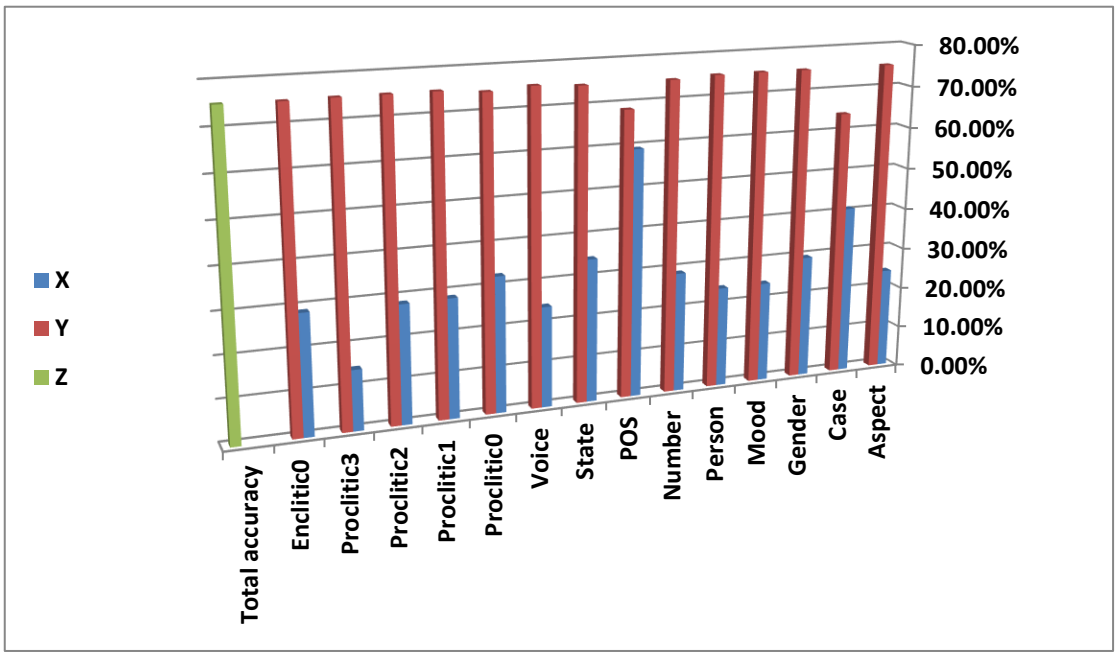



**Figure 4.9 Feature-based results with overall accuracy**

The symbol **( X )** refers to using **specific feature(only one)**, **( Y )** refers to use **all features excluding one** and **( Z )** refers to the accuracy of using **all features**. **Aspect** feature accuracy compares to the overall accuracy, which is 75.1994%, is 24.1938%. When **Aspect** feature is removed the accuracy is increased slightly to 75.2086%. The result reveals that **Aspect** has **almost no influence** the grammar analysis.

**Case** feature accuracy is 40.8024%. When **Case** feature is removed the accuracy is decreased, compares to the accuracy of using all features, to 64.1036%. The results reveal that **Case** has **good influence** the grammar analysis.

**Gender** feature result is 29.6087%. When **Gender** feature is removed the accuracy is decreased slightly, compares to the accuracy of using all features, to75.1796%. The results reveal that **Gender** has **almost no influence** the grammar analysis.

**Mood** feature result is 24.2152%. When **Mood** feature is removed the accuracy is increased compares to the accuracy of using all features, to 75.2346%. The results reveal that **Mood** has **bad influence** the grammar analysis.

**Person** feature result is 24.1907%. When **Person** feature is removed the accuracy is increased slightly, compares to the accuracy of using all features, to 75.2056%. The results reveal that **Person** has **almost no influence** the grammar analysis.

**Number** feature accuracy is 28.9011%. When **Number** feature is removed the accuracy is decreased, compares to the accuracy of using all features, to 74.7202%. The results reveal that **Number** has **good influence** the grammar analysis.

**POS** feature accuracy is 59.3681%. When **POS** feature is removed the accuracy is decreased, compares to the accuracy of using all features, to68.5634%. The results reveal that **POS** has **good influence** the grammar analysis. Moreover, **POS** by itself, with a satisfactory accuracy of, can be used by itself for Arabic grammar analysis.

**State** feature accuracy is 34.4872%. When **State** feature is removed the accuracy is decreased, compares to the accuracy of using all features, to 74.6416%. The results reveal that **State** has **good influence** the grammar analysis.

**Voice** feature accuracy is 24.2626%. When **Voice** feature is removed the accuracy is increased, compares to the overall accuracy, to 75.2285%. The result reveals that **Voice** does not influence the grammar analysis. On the contact, this feature has **bad influence** on the grammar analysis task.

**Proclitic0**feature accuracy is 32.6318%. When **Proclitic0** feature is removed the accuracy is decreased, compares to the accuracy of using all features, to 74.4888%. The results reveal that **Proclitic0** has **good influence** the grammar analysis.

**Proclitic1** feature accuracy is 28.6535%. When **Proclitic1** feature is removed the accuracy is increased, compares to the overall accuracy, to 75.2361%. The result reveals that**Proclitic1** does not influence the grammar analysis. On the contact, this feature has **bad influence** on the grammar analysis task.

**Proclitic2** feature accuracy compares to the overall accuracy, which is 75.1994%, is 28.4686%. When **Proclitic2** feature is removed the accuracy is increased slightly to 75.2086%. The result reveals that**Proclitic2** has **almost no influence** the grammar analysis.

**Proclitic3** feature accuracy compares to the overall accuracy, which is 75.1994%, is 14.4781%. When **Proclitic3** feature is removed the accuracy is increased slightly to 75.2025%. The result reveals that**Proclitic3** has **almost no influence** the grammar analysis. It is noted that **Proclitic3** feature has the worst accuracy when it ias used solely for arabic grammar analysis. Thus, it cannot be used alone for the underlying task.

**Enclitic** feature accuracy is 28.8629%. When **Enclitic** feature is removed the accuracy is decreased, compares to the accuracy of using all features, to 75.065 %. The results reveal that **Enclitic** has **good influence** the grammar analysis.

Overall, there are some features that shown to have a good influence on the underlying task, some with bad influence and other with no influence. Table 4.4 categorizes the features based on their influences.

**Table 4.4 Feature-categorization based on their influence and accuracy values**

| *Good* | Accuracy of Each(X) | *Fair* | Accuracy of Each(X) | *Bad* | Accuracy of Each(X) |
|---|---|---|---|---|---|
| **POS** | **59.37%** | **Enclitic0** | **28.86%** | **Aspect** | **24.19%** |
| **Case** | **40.80%** | **Proclitic1** | **28.65%** | **Person** | **24.19%** |
| **State** | **34.49%** | **Proclitic2** | **28.47%** | **Proclitic3** | **14.48%** |
| **Proclitic0** | **32.63%** | **Voice** | **24.26%** | | |
| **Gender** | **29.61%** | **Mood** | **24.22%** | | |
| **Number** | **28.90%** | | | | |

More experimental results are conducted based on the categories initiated and accuracy values in Table 4.4. The Figure 4.5, Figure 4.6 and Figure 4.7 shows the accuracy for features with good influence only, features with no influence only and those with bad influence only.

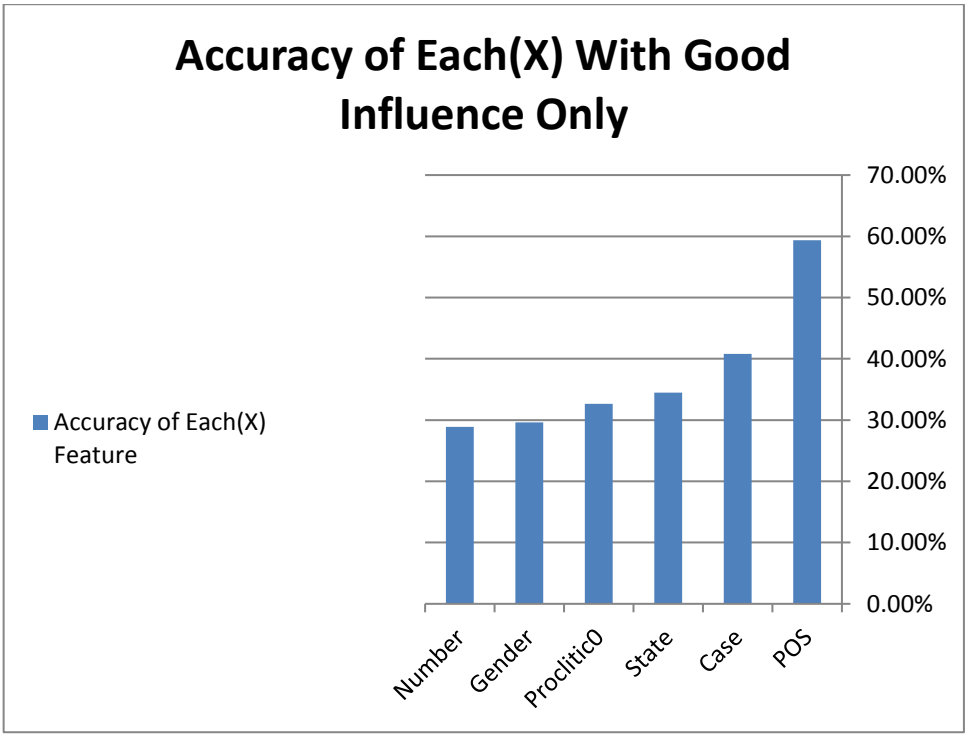



**Figure 4.10 Accuracy for features with good influence only**

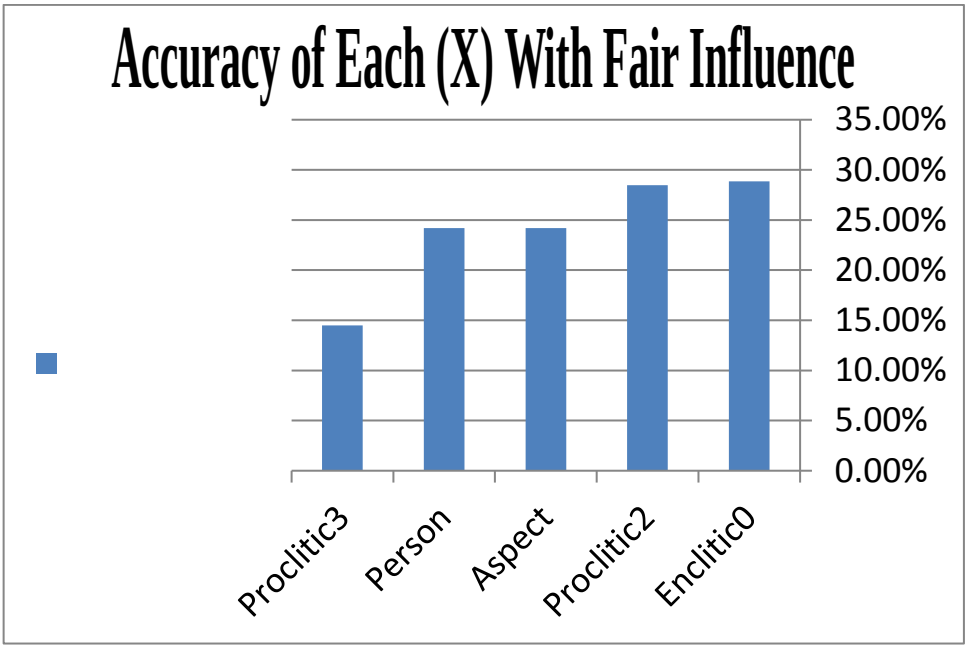



**Figure 4.11 Accuracy for features with fair influence only**

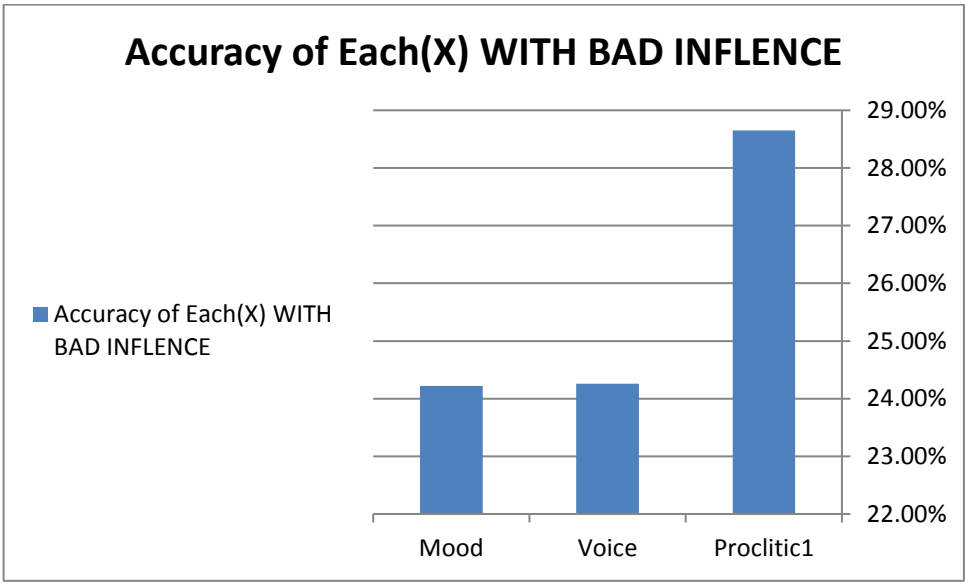



**Figure 4.12 Accuracy for features with bad influence only**

A sample for the confusion matrix and accuracy results of the proclitic3 feature were found in Appendix B.

### 4.3.3. The Results Comparison with Previous Works

The comparison of results of the proposed approach is compared with these exist in the literature as shown in Table 4.5.

**Table 4.5 Results comparison**

| ACCURACY FOR FEATURES | Accuracy of Ibrahim et al., (2016) | Accuracy of Proposed Approach |
|---|---|---|
| **POS** | 93.33% | 59.3681% |
| **CASE** | 94.09% | 40.8024 % |
| **OVERALL ACCURACY (With All Features)** | 90.44% | 75.1994 % |
| **OVERALL ACCURACY (With Good Features Only)** | 90.44% | 75.38% |

As noted in Table 4.5, the accuracy of the proposed approach when using **POS** and **Case** feature alone is 59.3681% and 40.8024% respectively, which appears to have low accuracy compared with the previous study **Ibrahim, Mahmoud, & El-Reedy, (2016),** that achieved 93.33% and 94.09% respectively for both features. The interpretation for this results refers to a two reasons:

**Firstly**, the size of data for our study is very large which is 7204 sentences compared with 600 sentences for the previous study **Ibrahim, Mahmoud, & El-Reedy, (2016).**

**Secondly,** the approach in our study is statistical approach while the approach for the previous study **Ibrahim, Mahmoud, & El-Reedy, (2016)** is a hybrid approach**.**

The accuracy of the proposed approach when using all features is 75.1994% and when using all Good features only is equal to 75.38%, which appears to have low accuracy

compared with the previous study for **Ibrahim, Mahmoud, & El-Reedy, (2016),** that achieved for all (only three) features **(Pos, Sign and Case)** a percentage equal to 90.44%, since that study using a hybrid approach.

The explanation for this results refers for the two reasons we showed above and also for the number of features we used are 14 features which are large compared with 3 features that used by the previous study **Ibrahim, Mahmoud, & El-Reedy, (2016)**.

### 4.3.4. The Results Comparison With Previous Works Results

The comparison of results with the previous works shown in the following table:

**Table 4.6 Results comparison with previous works**

| Study no. | The System | Data Tested | Results | | | |
|---|---|---|---|---|---|---|
| | | | Precession | Recall | F-score | Accuracy |
| 1. | (Khoufi, Louati , Aloulou , & Belguith, 2014) | 20% of corpus | 78.12 % | 73.24% | 75.37% | ---------- |
| 2. | (Al-Taani, Msallam, & Wedian, 2012) | 70 sentences | ---------- | ---------- | --------- | 94 % |
| 3. | (Bataineh & Bataineh, 2009) | 90 sentences | 85.6% correct | ---------- | --------- | 2,2% wrong 14,4% rejected |
| 4. | (Ibrahim, Mahmoud, & El-Reedy, 2016) | 600 sentences | ---------- | ---------- | --------- | 90.44% |

- Comparing our results with the study number one in the table 4.2 noticing that the total accuracy in our study is approximately equals 75% which is around the result of the study number one.
- Comparing our results with the study number two in the table 4.2 noticing that the total accuracy in our study is approximately equals 75% which is lower than result of the study number two, since that study have small dataset tested which contains 70 sentences only, while our study have 7204 sentences . Also the study number two based on rule-based approach which can't depends on prediction for result like our statistical approach.
- Comparing our results with the study number three in the table 4.2 noticing that the total accuracy in our study is approximately equals 75% which is lower than result of the study number three, since that study have small dataset which contains 90 sentences only, while our study have 7204 sentences.
- Comparing our results with the study number four in the table 4.2 noticing that the total accuracy in our study is approximately equals 75% which is lower than result of the study number four, since that study have small dataset which contains 600 sentences only, while our study have 7204 sentences. Also the study number four used a hybrid approach which utilize the benefits for both rule-based and statistical approaches.

# CHAPTER FIVE

# CONCLUSION AND FUTURE WORK

## 5.1. Conclusion

This thesis proposed a supervised machine learning approach for the grammar analysis of Arabic text in the attempts to improve the performance of identifying the Arabic grammar analysis using Naïve Bayesian(NB) algorithm classifier.

It was concluded that machine learning-based approach for Arabic grammar analysis can be achieved by building a framework which used the determined and extracted features for each word in an annotated corpus. Moreover, It was concluded that out of 14 features that were used in the experiments, the following features (Pos, Case, State, Proclitic0, Gender and Number) are effective and useful in automating the arabic grammar analysis. Depending on these effective features, the accuracy of the proposed machine learning based on naïve Bayesian algorithm classifier is 75.38%.

In conclusion, the proposed work is an attempt to resolve the ambiguity of Arabic sentences by automating the process of arabic grammar analysis and determine the most effective features that influences the arabic grammar analysis task.

## 5.2. Future work

- The empirical study over the proposed work was implemented over a corpus of arabic sentences, which includes a total number of 65430 tokens corresponding to 48646 words contained in 7204 sentences. Each word in the corpus is annotated with its complete Arabic grammar analysis.

- Thus, for future works, we will test the system with other sentences and we will collect more data to increase the size of corpus in-order to increase the accuracy of implemented approach, as increasing the portion of the training set which will help significantly to enhance the classification outcomes and improve the overall approach.

- Determine and extract more features, from various language categories, such as morphological and lexical, will also be investigated in the future, in the same way, as those utilized in the proposed work. Subsequently, increasing the size of the corpus and increasing the feature set will enable increasing the involved grammar analysis categories.

- In the future, we will work on combining two classification techniques in order increase the accuracy percentage. Furthermore, we will work on optimizing the results of grammar analysis task by using the hybrid-based approach which uses both rule-based and statistical approaches.

- Finally, Arabic language contains a lot of grammar rules. Therefore, it is recommended to enhance the data machine translation systems with a grammar dictionary in order to be used in a WEB-API's.

# References

Al Daoud, E., & Basata, A. (2009, April 20). A Framework to Automate the Parsing of Arabic Language Sentences. *No. 2*. Jordan: The International Arab Journal of Information Technology.

Alqrainy, S., Muaidi, H., & Alkoffash, M. S. (2012, September). Context-Free Grammar Analysis for Arabic Sentences. *Volume 53 - No. 3*. Salt, Jordan: International Journal of Computer Applications (0975 - 8887).

Al-Taani, A., Msallam, M., & Wedian, S. (2012). A Top-Down Chart Parser for Analyzing Arabic Sentences.

Attia, M. (2000). A Large-Scale Computational Processor of The Arabic Morphology, and Applications.

Attia, M. (2005). Theory and Implementation of a Large-Scale Arabic Phonetic Transcriptor and Applications.

Attia, M. (2006). An ambiguity-controlled morphological analyzer for modern standard Arabic modeling finite state networks. *pp. 155-159*. London: the Challenge of Arabic for NLP/MT Conference.

Attia, M. (2008). Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation. (M. Attia, Trans.) UK.

Bataineh, B. M., & Bataineh, E. A. (2009). An Efficient Recursive Transition Network Parser for Arabic Language.

Beesley, K. (2001). Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. In K. Beesley (Ed.). Toulouse: Status and Prospect-- 39th Annual Meeting of the Association for Computational Linguistics.

Ben Ali, B., & Jarray, F. (2013, June 3). GENETIC APPROACH FOR ARABIC PART OF SPEECH TAGGING. *No.3*(International Journal on Natural Language Computing (IJNLC)). Tunisia.

Boubas, A., Lulu, L., Belkhouche, B., & Harous, S. (2014). A Genetic-Based Extensible Stemmer for Arabic. (Vol.VI). Al-Ain, UAE: LINGUISTICA COMMUNICATION.

BOUDLAL, A., LAKHOUAJA, A., MAZROUI, A., MEZIANE, A., OULD ABDALLAHI OULD BEBAH, M., & SHOUL, M. (2010). Alkhalil Morpho Sys1: A Morphosyntactic analysis system for Arabic texts.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. n proceedings of the Ninth European Conference on Artificial Intelligence.

Chalabi, A. (2004). Sakhr Arabic Lexicon. *NEMLAR International Conference on Arabic Language Resources and Tools.*

Daoud, A. M. (2010). Morphological analysis and diacritical arabic text compression. *1*, pp. 41-47.

Darwish , K. (2002). Building a shallow Arabic Morphological Analyzer in one day.

Diab, M. (2009). Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking. *2nd International Conference on Arabic Language Resources and Tools.* Citeseer.

Diab, M., Ghoneim, M., & Habash, N. (2007). Arabic Diacritization in the Context of Statistical Machine Translation.

Domingos, P., & Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple Bayesian classifier. in proceedings of the 13th International Conference on Machine Learning.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis. New York: John Wiley & Sons.

Dukes, K., & Buckwalter, T. (2010, Mach 28-30). A dependency Treebank of the Quran using traditional Arabic grammar. Cairo, Egypt: the 7th International Conference on Informatics and Systems (INFOS).

Eltibi, M. F. (2013). Author Attribution from Arabic Texts. Gaza, Palestine.

EZZELDIN, A. M., & SHAHEEN, M. (2012). A SURVEY OF ARABIC QUESTION ANSWERING: CHALLENGES, TASKS,APPROACHES, TOOLS, AND FUTURE TRENDS. *The 13th International Arab Conference on Information Technology ACIT'2012 Dec. 10-13 ISSN: 1812-0857.*

Fayed, D. M., Fahmy, A. A., Rashwan, M. A., & Fayed, W. K. (2014, March). Towards Structuring an Arabic-English Machine-Readable Dictionary Using Parsing. *Number 1*(Volume 5). Cairo, Egypt: International Journal of Computational Linguistics Research.

Frank, E., & Witten, I. H. (2005). *Practical Machine Learning Tools and Techniques.* Morgan Kaufmann series in data management systems.

Green, S., & Manning, C. D. (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis.

Habash, N. Y. (2010). *Introduction to Arabic Natural Language Synthesis Lectures on Human Language Technologies.* Morgan and Claypool Publishers.

Habash, N. Y., & Roth, R. M. (2009). CATiB: The Columbia Arabic Treebank. (the ACL-IJCNLP 2009 Conference Short Papers). Association for Computational Linguistics.

Habash, N., & Rambow, O. (2007, April). Arabic Diacritization through Full Morphological Tagging. New York.

Ibrahim, M., Mahmoud, M., & El-Reedy, D. (2016, May). Bel-Arabi: Advanced Arabic Grammar Analyzer. *No. 5*(Vol. 6). International Journal of Social Science and Humanity.

Ibrahim, M. N. (2015). Statistical Arabic Grammar Analyzer. *9041*, pp. 187-200. Computational Linguistics and Intelligent Text Processing.

Khoufi, N., Louati , S., Aloulou , C., & Belguith, L. H. (2014). Supervised learning model for parsing Arabic language. Sfax, Tunisia.

Kulick, S., Gabbard, R., & Marcus, M. (2006). Parsing the Arabic Treebank: Analysis and Improvements.

Maamouri, M., Kulick, S., & Bies, A. (2006). THE CHALLENGE OF ARABIC FOR NLP/MT Diacritization: A Challenge to Arabic Treebank Annotation and Parsing.

Manning, C. D., & Toutanova, K. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000).*

Manning, C., Klein, D., & Toutanova, K. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *HLT-NAACL*, pp. 252-259.

Michie, D., Taylor, C., & Spiegelhalter, D. J. (1994). Machine learning, neural and statistical classification. New York: New York: Ellis Horwood.

Othman, E., Shaalan, K., & Rafea, A. (2003). A Chart Parser for Analyzing Modern Standard Arabic.

Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Habash, N., Eskander, R., et al. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. *http://www.lrec-conf.org/proceedings/lrec2014/pdf/593_Paper.pdf*.

Rashwan, M. A., Al-Badrashiny, M., Attia, M., & Abdou, S. M. (2009). A Hybrid System for Automatic Arabic Diacritization. Cairo, Egypt.

Roth, R., Rambow, O., & Habash, N. (2009). Mada+ tokan: A toolkit for arabic tokenization diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. *In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR),* (pp. 102-109). Cairo.

Rumelhart, D., & Chauvin, Y. (1995). Backpropagation: Theory, architectures, and applications. Lawrence Erlbaum Assoc.

Saad, M. K., & Ashour, W. (2010). Arabic Morphological Tools for Text Mining. *6th International Conference on Electrical and Computer Systems (EECS).* Lefke, North Cyprus.

Sadat, F., Kazemi, F., & Farzinda, A. (2014, August 27). Automatic Identification of Arabic Language Varieties and Dialects Social Media. Dublin, Ireland.

Shaalan, K. (2010, June). Rule-based Approach in Arabic Natural language Processing. *3*. International Journal on Information and Communication Technologies.

Shahrour, A., Khalifa, S., & Habash, N. (2015, Sep). Improving Arabic Diacritization through Syntactic Analysis. Conference on Empirical Methods in Natural Language Processing.

Shalaan, K. F. (2005, March 11). Arabic GramCheck:a grammar checker for Arabic.

Sobh, I. M. (2009). AN OPTIMIZED DUAL CLASSIFICATION SYSTEM FOR ARABIC EXTRACTIVE GENERIC TEXT SUMMARIZATION. Giza, Egypt.

# Appendix A:
## The Complete Grammar Analysis Categories

| Category-number | Grammar analysis Category(الإعراب) |
|---|---|
| **0** | مبتدأ مرفوع وعلامة الرفع الضمة |
| **1** | فعل مضارع مرفوع وعلامة الرفع الضمة |
| **2** | مفعول به منصوب وعلامة النصب الفتحة |
| **3** | نعت منصوب وعلامة النصب الفتحة |
| **4** | حرف جر مبني لا محل له من الاعراب |
| **5** | اسم مجرور وعلامة الجر الكسرة |
| **6** | مضاف اليه مجرور وعلامة الجر الكسرة |
| **7** | علامة ترميز لا محل لها من الاعراب |
| **8** | نعت مجرور وعلامة الجر الكسرة |
| **9** | ظرف مبني في محل نصب |
| **10** | مفعول به ثاني منصوب وعلامة النصب الياء |
| **11** | مفعول به منصوب وعلامة النصب الكسرة |
| **12** | نعت مرفوع وعلامة الرفع الضمة |
| **13** | مبتدأ مرفوع وعلامة الرفع الواو |
| **14** | فعل مضارع مرفوع وعلامة الرفع ثبوت النون |
| **15** | نعت مرفوع وعلامة الرفع الواو |
| **16** | ضمير مبني في محل جر بالإضافة |
| **17** | خبر مرفوع وعلامة الرفع الضمة |
| **18** | حرف نصب للفعل المضارع مبني لا محل له من الاعراب |
| **19** | فعل مضارع منصوب وعلامة النصب الفتحة |
| **20** | ظرف مبني في محل نصب على الظرفية الزمانية |

| | |
|---|---|
| **21** | نعت مجرور وعلامة الجر الياء |
| **22** | مفعول به منصوب وعلامة النصب الياء |
| **23** | فاعل مرفوع وعلامة الرفع الضمة |
| **24** | فعل ماضي مبني علي السكون والتاء ضمير مبني في محل رفع فاعل |
| **25** | حرف عطف مبني لا محل له من الاعراب |
| **26** | معطوف مجرور وعلامة الجر الكسرة |
| **27** | بدل مجرور وعلامة الجر الكسرة |
| **28** | معطوف مرفوع وعلامة الرفع الضمة |
| **29** | مبتدأ مرفوع وعلامة الرفع الألف |
| **30** | نعت مرفوع وعلامة الرفع الألف |
| **31** | حرف مبني لا محل له من الاعراب |
| **32** | بدل منصوب وعلامة النصب الفتحة |
| **33** | بدل مرفوع وعلامة الرفع الضمة |
| **34** | اسم مجرور وعلامة الجر الياء |
| **35** | ضمير مبني في محل نصب مفعول به |
| **36** | نعت منصوب وعلامة النصب الياء |
| **37** | فعل مضارع مبني للمجهول مرفوع وعلامة الرفع الضمة |
| **38** | مضاف اليه مجرور وعلامة الجر الياء |
| **39** | ضمير فصل مبني في محل مبتدأ مرفوع |
| **40** | فعل أمر مبني علي ثبوت النون والواو ضمير مبني فى محل رفع فاعل |
| **41** | فعل ماضي مبني علي الفتح |
| **42** | اسم إشارة مبني في محل مبتدأ مرفوع |
| **43** | ضمير مبني في محل جر بحرف الجر |

| | |
|---|---|
| **44** | ضمير مبني في محل رفع فاعل |
| **45** | فاعل مرفوع وعلامة الرفع الواو |
| **46** | معطوف منصوب وعلامة النصب الفتحة |
| **47** | ظرف زمان منصوب وعلامة النصب الفتحة |
| **48** | معطوف منصوب وعلامة النصب الياء |
| **49** | مفعول به ثاني منصوب وعلامة النصب الفتحة |
| **50** | اخ من أخوات كان مبني علي الفتح |
| **51** | خبر مرفوع وعلامة الرفع الواو |
| **52** | حرف تحقيق مبني لا محل له من الاعراب |
| **53** | حرف تقليل مبني لا محل له من الاعراب |
| **54** | فعل ماضي مبني علي الضم والواو ضمير مبني فى محل رفع فاعل |
| **55** | حرف جزم للفعل المضارع مبني لا محل له من الاعراب |
| **56** | فعل مضارع مجزوم وعلامة الجزم السكون |
| **57** | ظرف مكان منصوب وعلامة النصب الفتحة |
| **58** | حرف ناسخ من أخوات إن مبني لا محل له من الاعراب |
| **59** | اسم إن منصوب وعلامة النصب الفتحة |
| **60** | خبر إن مرفوع وعلامة الرفع الضمة |
| **61** | اخ من أخوات كان منصوب وعلامة النصب الفتحة |
| **62** | اسم كان مرفوع وعلامة الرفع الضمة |
| **63** | فعل ماضي مبني علي الضم |
| **64** | خبر كان منصوب وعلامة النصب الفتحة |
| **65** | ضمير مبني في محل نصب اسم إن |
| **66** | خبر مرفوع وعلامة الرفع الألف |

| | |
|---|---|
| **67** | اخ من أخوات كان مرفوع وعلامة الرفع ثبوت النون |
| **68** | خبر كان منصوب وعلامة النصب الياء |
| **69** | اسم إشارة مبني في محل مضاف اليه مجرور |
| **70** | فاعل مرفوع وعلامة الرفع الألف |
| **71** | معطوف مجرور وعلامة الجر الياء |

# Appendix B:

## Confusion matrix with accuracy result for Proclitic3 feature

===Run information===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation: Arabic-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.attribute.Remove-R1-11,13-14

Instances: 65430

Attributes: 2

prc3

result

Test mode: 10-fold cross-validation

===Classifier model (full training set=== (

Naive Bayes Classifier

Class

| Attribute | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
| 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 |
| 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | | | | | | |
| (0) | (0) | (0) | (0.05) | (0.14) | (0.14) | (0.14) | (0.14) | (0.02) | (0.06) | (0.1) | (0.13) | | | | |
| (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0.03) |
| (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) |
| (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | (0) | | | | |

==========

==========

==========
==========
==========
==========
==========
==========
==========

prc3

3.0 293.0 3164.0 1.0 9472.0 9110.0 9371.0 1040.0 4188.0 6490.0 8729.0 0
59.0 79.0 44.0 3.0 18.0 19.0 166.0 135.0 31.0 227.0 126.0 2259.0 199.0
14.0 11.0 5.0 48.0 18.0 6.0 19.0 6.0 12.0 123.0 41.0 89.0 229.0 15.0
5.0 6.0 7.0 10.0 2.0 16.0 13.0 2.0 3.0 3.0 11.0 66.0 9.0 12.0 127.0
3.0 4.0 3.0 7.0 2.0 3.0 2.0 2.0 7.0 9.0 7.0 4.0 4.0 2.0 6.0
2.0 2.0 2.0 2.0

na 2.0 1.0 1.0 1.0 2.0 2.0 3.0 9160.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 108.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 3.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 3.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

< a_ques 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0

] total] 8732.0 6492.0 4190.0 1042.0 9374.0 9113.0 9476.0 9162.0 3166.0 295.0
5.0 201.0 2261.0 128.0 229.0 33.0 244.0 168.0 21.0 20.0 5.0 46.0 81.0 61.0
17.0 231.0 91.0 43.0 125.0 14.0 8.0 21.0 8.0 20.0 50.0 9.0 13.0 16.0
129.0 14.0 11.0 68.0 13.0 7.0 5.0 4.0 15.0 18.0 4.0 12.0 9.0 8.0
7.0 8.0 4.0 6.0 6.0 9.0 11.0 9.0 4.0 4.0 5.0 4.0 9.0 5.0 6.0 5.0
4.0 4.0 4.0 4.0

Time taken to build model: 0.03 seconds

===Stratified cross-validation===

===Summary===

| | | |
|---|---|---|
| Correctly Classified Instances | 18630 | 28.4732% |
| Incorrectly Classified Instances | 46800 | 71.5268% |
| Kappa statistic | 0.1643 | |
| Mean absolute error | 0.0208 | |
| Root mean squared error | 0.102 | |
| Relative absolute error | 84.8606% | |
| Root relative squared error | 92.0756% | |
| Total Number of Instances | 65430 | |

===Detailed Accuracy By Class===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.155 | 0.582 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 1 | 0.115 | 0.579 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 2 | 0.074 | 0.575 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 3 | 0.018 | 0.572 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 4 | 0.167 | 0.583 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 5 | 0.162 | 0.582 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 6 | 0.169 | 0.583 | 0.167 | 0.289 | 1.000 | 0.169 | 0.834 | 1.000 | |
| 7 | 0.985 | 0.999 | 0.993 | 0.994 | 1.000 | 0.987 | 0.002 | 1.000 | |
| 8 | 0.056 | 0.574 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 9 | 0.005 | 0.569 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 10 | 0.000 | 0.086 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 11 | 0.003 | 0.568 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 12 | 0.040 | 0.573 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 13 | 0.002 | 0.562 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 14 | 0.004 | 0.567 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 15 | 0.001 | 0.571 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 16 | 0.007 | 0.627 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 17 | 0.003 | 0.564 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 18 | 0.000 | 0.533 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 19 | 0.000 | 0.518 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 20 | 0.000 | 0.086 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 21 | 0.001 | 0.550 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 22 | 0.001 | 0.562 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 23 | 0.001 | 0.559 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 24 | 0.000 | 0.497 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 25 | 0.004 | 0.568 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 26 | 0.002 | 0.563 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 27 | 0.001 | 0.571 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 28 | 0.002 | 0.565 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 29 | 0.000 | 0.536 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 30 | 0.000 | 0.215 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 31 | 0.000 | 0.533 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 32 | 0.000 | 0.214 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 33 | 0.000 | 0.518 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 34 | 0.001 | 0.552 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 35 | 0.000 | 0.405 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 36 | 0.000 | 0.571 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 37 | 0.000 | 0.502 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 38 | 0.002 | 0.563 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 39 | 0.000 | 0.536 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 40 | 0.000 | 0.485 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 41 | 0.001 | 0.554 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 42 | 0.000 | 0.571 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 43 | 0.000 | 0.479 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 44 | 0.000 | 0.086 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 45 | 0.000 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 46 | 0.000 | 0.514 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 47 | 0.000 | 0.499 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 48 | 0.000 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 49 | 0.000 | 0.528 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 50 | 0.000 | 0.257 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 51 | 0.000 | 0.214 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 52 | 0.000 | 0.172 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 53 | 0.000 | 0.214 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 54 | 0.000 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 55 | 0.000 | 0.129 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 56 | 0.000 | 0.129 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 57 | 0.000 | 0.257 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 58 | 0.000 | 0.485 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 59 | 0.000 | 0.257 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 60 | 0.000 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 61 | 0.000 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 62 | 0.000 | 0.086 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 63 | 0.000 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

64 0.000 0.257 0.000 0.000 0.000 0.000 0.000 0.000

65 0.000 0.086 0.000 0.000 0.000 0.000 0.000 0.000

66 0.000 0.129 0.000 0.000 0.000 0.000 0.000 0.000

67 0.000 0.086 0.000 0.000 0.000 0.000 0.000 0.000

68 0.000 0.043 0.000 0.000 0.000 0.000 0.000 0.000

69 0.000 0.043 0.000 0.000 0.000 0.000 0.000 0.000

70 0.000 0.043 0.000 0.000 0.000 0.000 0.000 0.000

71 0.000 0.043 0.000 0.000 0.000 0.000 0.000 0.000

Weighted Avg. 0.285 0.121 0.163 0.285 0.181 0.163 0.638 0.250

===Confusion Matrix===

a b c d e f g h i j k l m n o p q r s t u v w x y z aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp bq br bs bt <-- classified as

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 8728 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 a = 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6489 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 b = 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4187 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 c = 2

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1039 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 d = 3

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 9370 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 e = 4

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 9109 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 f = 5

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 9471 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 g = 6

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9159 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 h = 7

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3163 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 i = 8

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 292 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 j = 9

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0k = 10

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 198 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 l = 11

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2258 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 m = 12

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 125 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 n = 13

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 226 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 o = 14

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 30 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0p = 15

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 107 134 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 0q = 16

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 165 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 r = 17

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 18 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0s = 18

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0t = 19

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0u = 20

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 43 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0v = 21

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 78 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0w = 22

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 58 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0x = 23

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 14 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0y = 24

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 228 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 z = 25

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 88 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0aa = 26

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 40 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ab = 27

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 122 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 ac = 28

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ad = 29

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ae = 30

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 18 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0af = 31

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ag = 32

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ah = 33

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 47 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ai = 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 4 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0aj = 35

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ak = 36

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 13 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0al = 37

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 126 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0 am = 38

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0an = 39

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ao = 40

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 65 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ap = 41

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0aq = 42

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ar = 43

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0as = 44

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0at = 45

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0au = 46

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0av = 47

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0aw = 48

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ax = 49

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ay = 50

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0az = 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0ba = 52

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bb = 53

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bc = 54

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bd = 55

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0be = 56

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bf = 57

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bg = 58

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bh = 59

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bi = 60

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bj = 61

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bk = 62

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bl = 63

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bm = 64

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bn = 65

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bo = 66

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bp = 67

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bq = 68

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0br = 69

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bs = 70

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
| 0 0bt = 71