Arabic Text Preprocessing for the Natural Language Processing Applications

Arafat AWAJAN

Computer Science Department

Princess Sumaya University for Technology, Amman, Jordan

awajan@psut.edu.jo

Abstract

This article aims at describing a new approach for preprocessing vowelized and unvowelized Arabic texts in order to prepare them for Natural Language Processing (NLP) purposes. This approach is rule-based and made up of four phases: text tokenization, word light stemming, words' morphological analysis, and text annotation. The first phase preprocesses the input text in order to isolate the words and represent them in a formal way. The second phase applies a light stemmer in order to extract the stem of each word by eliminating the prefixes and suffixes. The third phase is a rule-based morphological analyzer that determines the root and the morphological pattern for each extracted stem. The last phase aims at producing an annotated text where each word is tagged with its morphological attributes. The preprocessor presented in this paper is capable of dealing with vowelized and unvowelized words, and provides the input words along with relevant linguistics information needed by different applications. It is designed to be used with different NLP applications such as machine translation, text summarization, text correction, information retrieval, and automatic vowelization of Arabic text.

Key Words:

Arabic Text Preprocessing, Stemming, Morphological Analysis, Text Annotation, Part of speech tagging.

معالجة النصوص العربية لغايات تطبيقات المعالجة الآلية اللغات الطبيعية

الدكتور عرفات عوجان

قسم علم الحاسوب

جامعة الأميرة سمية للتكنولوجيا - عمان – الأردن

awajan@psut.edu.jo

خلاصة

يهدف هذا البحث إلى وصف مقاربة جديدة لمعالجة النصوص العربية المشكلة و غير المشكلة من أجل تهيئتها للاستعمال في تطبيقات المعالجة الآلية للغات الطبيعية. بُنيت المقاربة الجديدة على قواعد محددة مسبقاً و تتكون من أربعة مراحل: فصل المكون الأساسي في النص (الكلمة)، فصل جذور الكلمات، التحليل الصرفي للكلمات و إضافة توصيف للكلمات على النص. تعالج المرحلة الأولى النص بدف عزل الكلمات و إعادة تشيلها بطريقة معيارية. و يخضع النص في المرحلة الثانية إلى معالج يقوم باستخراج جذوع كلمات النص وذلك بإزالة الإضافات السابقة و اللاحقة للكلمات. تشمل المرحلة الثالثة محللاً صرفياً مبنياً على قواعد محددة، يستخرج جذر و النمط الصرفي لكل كلمة. أما المرحلة الأخيرة فتضيف توصيفات على النص تشمل الخصائص الصرفية لكل كلمة. تستطيع الطريقة المقترحة في هذا البحث التعامل مع النصوص العربية المشكلة و غير المشكلة و تنتج لكل كلمة من النص مجموعة من المعلومات اللغوية الضرورية لكثير من التطبيقات. لقد تم تصميم المعالج بحيث يمكن استخدامه مع الكثير من تطبيقات المعالجة الآلية للغات الطبيعية مثل ترجمة النصوص، تلخيص النصوص، تصحيح النصوص، استخراج المعلومات و التشكيل الآلي للنصوص العربية.

1. INTRODUCTION

The Natural Language Processing (NLP) is one of the most important and evolving fields of investigation in Computer Science and Artificial Intelligence. NLP deals with the creation of programs that are capable of processing and understanding human languages. A natural language is a very complicated phenomenon, and its study involves many levels of analysis related to the phonology, morphology, syntactical rules, and semantics of the language [Allen 1995; Manning and Schutze 2000].

NLP covers a wide range of useful applications, i.e. machine translation, text summarization, text correction, document analysis, human-machine interaction and information retrieval. Although the objectives of each application determine the processing techniques and the transformations to be applied on the original texts and the order in which these transformations should be applied, the first and most critical step is the preprocessing of the input text.

The text preprocessing is a core natural language processing task. It aims at creating an intermediate form from the inputted text based on the extraction of words, the morphological analysis, and the text annotation. Many research papers and studies related to the preprocessing of natural language texts have been published, mainly for the European languages. These works cover the tokenization of text [Grefensette and Tapanainen 1994], the morphological analysis of words [Antworth 1994], and the part of speech tagging of words [Jurafsky and Martin 2000].

There have been few articles and research papers published on the subject of the Arabic text preprocessing for the NLP applications. These published works cover mainly the morphological analysis of Arabic words [Al-Sughaiyer and Al-Kharashi 2004]. Also, there are fewer papers which treat the tagging of words [Khoja et al. 2001] and the preparation of text for text understanding. These works generally ignore the presence of diacritics in Arabic texts or limit the analysis to words generated from 3-letter roots [Beesley 1996; Larkey et al. 2002]. Some of these works are based on the generalization of concepts used for European

languages to the case of the Arabic language [De Roeck and Al-Fares 2000; Larkey et al. 2002].

The work presented in this paper aims at presenting a new approach of preprocessing Arabic texts based on the features of the Arabic language and is able to analyze Arabic words as they appear in real texts. This new approach is capable of dealing with vowelized, unvowelized or partially vowelized Arabic words. The proposed techniques transform the input texts into a new format that is more appropriate and adequate for the different NLP applications. In addition to the original text, the new format contains additional information at the word level. The new format's main purpose is to make the NLP applications faster and more accurate.

Our approach consists of preprocessing the input text in four phases. The first phase, called the tokenization of the text, preprocesses the input text in order to detect and isolate the words. The second phase is a light stemmer that eliminates prefixes and suffixes in order to extract the kernel words or the stems. The third phase is the morphological analysis of words; it consists of a rule-based morphological analyzer that decomposes each word into its basic morphological components; the root and the morphological pattern. The fourth and last phase is the text's annotation that tags the words by adding morphological attributes in order to facilitate their analysis. Figure 1 describes schematically the transformations that the proposed preprocessor applies on the manipulated texts.

2. BACKGROUND

Words in the Arabic language may be classified into two categories: derivative words, and non-derivative words. The derivative Arabic words are generated from basic entities called roots or radicals according to a predefined list of standard patterns called morphological patterns or balances. These morphological patterns represent the major spelling rules of Arabic words.

The non-derivative words include two sub- categories: fixed words and foreign words. The fixed Arabic words are a set of words that do not obey the derivation rules. These words are generally functional words like pronouns, prepositions, conjunctions, question words and the like. The foreign words are nouns borrowed from foreign languages.

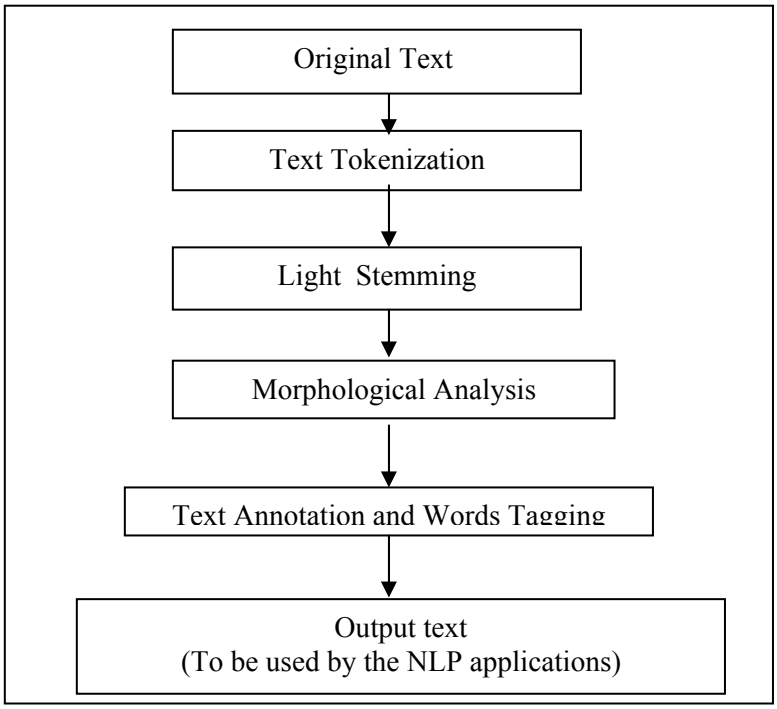

Fig. 1. The Main Phases of the Text Preprocessing

The majority of Arabic words belong to the category of derivative words. All kinds of words (verbs, nouns, adjectives and adverbs) can be generated from roots according to the standard patterns. The pattern associated to a word determines its various attributes such as gender (masculine/feminine), number (singular/plural), and tense (past, present, and imperative). Based on the above, a derivative Arabic word can be represented by its root along with its morphological pattern. For example, the word (لاعبون) (in English "players") is generated from the root (to play: لعب) according to the pattern (فاعلون). This pattern indicates that the word is a noun, its gender is masculine, and its number is plural. The final meaning of this Arabic word will be players (root (play): Attributes (noun; plural; masculine)).

Nevertheless, the morphological analysis of Arabic words presents many challenges that must be considered. The first challenge is the fact that some letters of the root may be dropped or modified during the derivation of words from roots. The second challenge is that many affixes can be attached to the beginning of the word (prefixes) and the end of the word (suffixes). These affixes may be formed from one or more letters. The third challenge is that the Arabic language words are written without short vowels. Different diacritical marks are used to

replace the short vowels. The number of these marks is eight diacritics: three diacritical marks to indicate the short vowels (ِ ، ُ ، َ), three double diacritic marks which combine the single ones (ٍ ، ٌ ، ً), one diacritical mark to indicate the absence of vowelization (ْ), and a single diacritical mark to indicate the duplicate occurrence of a consonant (ّ). These marks play a vital role in determining the possible meaning of the word. Actually, two different patterns may have the same sequence of consonants, but they differ from each other in terms of the diacritical marks.

According to the extent of using the diacritical marks, Arabic texts may be classified into three different categories: unvowelized, partially vowelized, and fully vowelized texts. The first category represents the texts without diacritics such as many of the typed, printed texts and newspapers. The second category represents the texts partially vowelized such as text books and scientific texts. The last category represents the fully vowelized Arabic texts, in which every consonant is followed by a diacritical mark such as the Holy Quran text, children books and literature texts.

3. TEXT TOKENIZATION

Tokenization is the process of isolating word-like units from a text [Grefensette and Tapanainen 1994]. In addition to words, text documents often contain white spaces, punctuation marks, and a number of mark-ups that indicate font changes, text subdivision, and special characters. The aim of the tokenization phase is to detect and isolate the individual words by eliminating these additional components. For the purposes of this work, it is assumed that an Arabic word is a sequence of Arabic letters and diacritical marks without separators (space or punctuation marks).

The detection of individual words is based on a simple text matching algorithm. A new word is a sequence of Arabic letters and diacritical marks starting by a letter and ended by a letter or diacritical marks. The detection of a separator or special charter marks the end of the word.

4. WORD REPRESENTATION

In order to deal with the different forms that an Arabic word can take (unvowelized, partially vowelized, and fully vowelized words), an additional diacritical mark called “EXTRA-SEKOUN” is introduced in this work. This special diacritical mark “EXTRA-SEKOUN”, represented by a period in the different examples given in this paper, is used to replace the missed diacritical marks in a word.

A rule based procedure Check_Diacritics takes the string of characters forming a word and checks the presence of diacritic marks after each consonant. It produces the new presentation of the word "NewWord" by replacing the missed diacritical marks by the special character EXTRA-SEKOUN. It calls a function IsDiacritic defined for finding out the diacritical marks. The function IsDiacritic returns TRUE if a character is one of the diacritical marks of the Arabic language. The Prolog-style description of the rule Check_Diacritics is presented as follows:

```
//Stopping case when the new presentation is finalized
Check_Diacritics([ ], NewWord , False).
//The last consonant is not followed by a diacritic; an EXTRA-SEKOUN //mark is then added
Check_Diacritics([],NewWord,True):- Check_Diacritics([],['.'|NewWord], False).
// Detection of a consonant at the first call of the rule or after a diacritic
Check_Diacritics([Head|Tail], NewWord, False):-
        Check_Diacritics(Tail, [Head|T1] , True).
// Detection of a diacritic
Check_Diacritics([Head|Tail], NewWord, True):-
        IsDiacritic(Head),
        Check_Diacritics(Tail, [Head | NewWord], False).
//Adding EXTRA-SEKOUN if  2 consecutive consonants are detected
Check_Diacritics([Head|Tail], NewWord, True):-
        Check_Diacritics([Head|Tail],['.',NewWord],False).
```

The word is then represented by a list of characters NewWord with the following format: [C_1 V_1 C_2 V_2 …… C_n V_n] , where C_i is a consonant and V_i is one of the diacritical marks including the EXTRA_SEKOUN mark replacing the missed diacritical marks in the input word.

5. WORD LIGHT STEMMING

In the Arabic language as is the case in many other languages, some lexical elements can be attached to a word in order to add new information to the word or to form sometimes a sentence or a part of sentence. Examples of these additive parts, called affixes, are the conjunctions (ex.: ف), the prepositions (ex.: ل), the pronouns (ex.: هم) and the article (ex.: ال).

The number of affixes is limited, and they may be added at the beginning of the word "prefixes", or at the end of the word "suffixes". A word may have up to two prefixes and up to three suffixes [Al-Sughaiyer and Al-Kharashi 2004].

The process of extracting the kernel word or stem from the original text by eliminating the suffixes and prefixes is called stemming. As opposed to the English language, the removal of prefixes and suffixes from an Arabic word does not usually reverse the meaning of the word [Abu Salem et al. 1999; Xu et al. 2002; Moukdad 2006].

The stemmer that we are using is developed based on the light stemming approaches described in [Darwish 2002; Larkey et al. 2002] with additional restrictions in the list of strippable prefixes and suffixes. The light stemming refers to a process of stripping off a small set of prefixes and / or suffixes, without trying to neither deal with infixes nor recognize patterns and roots. On one hand, this approach reduces the risk of root consonant loss that can be produced if a heavy stemming is used. On the other hand, the drawback of the use of light stemmer will be corrected by the morphological analyzer that will be applied on the stems in the next phase.

The stemmer decomposes the input string into the additive parts (prefixes, suffixes) and the stem. The decomposition is realized by applying a set of identification rules that test the first characters and last characters of the word against the possible and strippable additive parts.

The decomposition of the word into three parts is achieved provided that the first part is in the list of possible prefixes and the third part is in the list of possible suffixes. The first and third parts may be null. The light stemmer developed in this paper is based on the following assumptions:

1. A word is composed of three parts: prefix, stem and suffix.
2. Any of the additive parts (prefixes and suffixes) may be empty.
3. The stem of the word has at least three letters. Any word with less than four letters will be left without decomposition.
4. A prefix can have 0 to 3 letters and exist in the list of prefixes.
5. A suffix can have 0 to 3 letters and exist in the list of suffixes.
6. Only the consonants of the word (letters) are taken into account for the purpose of this decomposition.

The process of word-decomposition and suffix-removal is repeated until one of the following conditions is verified:

1. The number of letters in the word is less than or equals 3.
2. There are no prefixes, nor suffixes detected.

The final output of the light stemmer takes the following structure:

[Prefix1][Prefix2] stem [Suffix1][Suffix2][Suffix3]

An explicit list of strippable affixes is provided in a table. They are classified according to their type (prefix/suffix) and their length (1, 2 and 3 letters). This restricted list contains mainly conjunctions, prepositions, pronouns and the article. Table I shows examples from this list. The affixes used to determine the person, number and gender are not included in this table and therefore not strippable in this phase. The priority of detection and removal is given for the three-letter affixes over the two-letter affixes, for the two-letter affixes over the one-letter affixes, and for prefixes over suffixes.

However, some words contain letters that may be detected as prefixes or suffixes. To solve this problem, we will consider the results and feedback of the morphological analyzer to

correct this type of error. If the morphological analyzer fails in detecting the morphological structure of the stem, the last action taken by the light stemmer will be disregarded.

Table I. Examples from the list of strippable affixes

Types of Affixes	Group	Examples of Affixes	Examples(Words)
Prefixes	P_G1 (one letter)	و، ف ، ب، ك ، ل	فسمع
	P_G2 (two letters)	أل	الكتاب
	P_G3 (three letters)	وال ، فال، كال	كالعصفور
Suffixes	S_G1 (one letter)	ي ، ك، ـه	كتابي
	S_G2 (two letters)	هم، هن، ها، كن	يعلمهم
	S_G3 (three letters)	هما	رافقهما

6. THE MORPHOLOGICAL ANALYSIS

The proposed morphological analyzer is a rule-based technique, designed to identify the morphological structure of vowelized and unvowelized Arabic words. The morphological analyzer processes the extracted stems in order to determine their roots and patterns (morphological knowledge).

As the affixes used to determine the person, number and gender are not removed in the previous phase, the list of morphological patterns used to generate verbs and nouns is extended to include new computational patterns generated from the classical patterns by adding these affixes. Table II shows examples of the extended morphological patterns represented by their list of consonants.

Table II. Examples from the list of extended morphological pattern

Standard Pattern	Person	Number	Masculine	Feminine	Example
Verb standard pattern: فعل	First	Singular	فعلت	فعلت	ذهبت
		Dual	فعلنا	فعلنا	ذهبنا
		Plural	فعلنا	فعلنا	ذهبنا
	Second	Singular	فعلت	فعلت	ذهبت
		Dual	فعلتما	فعلتما	ذهبتما
		Plural	فعلتم	فعلتن	ذهبتم
	Third	Singular	فعل	فعلت	ذهبت
		Dual	فعلا	فعلتا	ذهبتا
		Plural	فعلوا	فعلن	ذهبوا
Noun standard pattern: فاعل		Dual	فاعلان	فاعلتان	لاعبان
		Plural	فاعلون	فاعلات	لاعبون

6. 1. Stem Decomposition

The stem [C_1 V_1 C_2 V_2 … C_n V_n] is split into two lists: the first one LC contains the sequence of consonants [C_1, C_2, … , C_n] and the second one LV contains the sequence of diacritical characters [V_1, V_2, … , V_n]. Table III illustrates this representation for the three situations of Arabic texts where the EXTRA-SEKOUN character marked by a dot is used to replace the missed diacritical marks in the original word.

Table III. Decomposition of Stems

Word	Case	List of Consonants LC	List of Diacritics LV
يَـذْهَـبُـوْنَ	Fully vowelized	[ي ذ هـ ب و ن]	[َ ْ ُ َ ْ َ]
يَـذْهَـبـونَ	Partially vowelized	[ي ذ هـ ب و ن]	[َ . . َ . َ]
يـذهـبـون	Unvowelized	[ي ذ هـ ب و ن]	[. . . . . .]

The recursive procedure Decompose performs the decomposition of the word in the two lists: LC and LV. The procedure Decompose may be presented by the following Prolog-style description:

```
// Base (Stopping) case when the decomposition is terminated
Decompose ([ ] , LC,LV).
// Detection of a diacritic
Decompose ([Head|Tail], _ , LV ):-
        IsDiacritic (Head),
        Decompose(Tail , _ , [Head|LV]).
// Detection of a consonant
Decompose ([Head|Tail], LC , _ ):- Decompose(T , [Head|LC] , _ ).
```

The list LC of consonants represents the letters of the word's root and the letters added to the root to form the stem according to a standard pattern.

6. 2. Root Representation

In order to extract the root of a stem, the list LC can be represented by the following general description:

$$[X_1\ [X_2[X_3]]]\quad R_1\ \ [Y_1]\ \ R_2\ \ [Y_2]\ \ R_3\ \ [\ [Y_3]\ R_4\ [[Y_4]\ R_5]]\ \ [Z_1\ [Z_2[Z_3]]\ ]$$

where $X_1X_2X_3$ represent a prefix of maximum 3 letters, $Z_1Z_2Z_3$ represent a suffix of maximum three letters and $Y_1Y_2Y_3Y_4$ represent the possible infixes. The maximum number of letters considered for the prefixes and suffixes takes into account that some of these components may not be detected by the light stemmer used before. The slots R_1, R_2, R_3, R_4, and R_5 represent the letters of the root used to generate the word. This representation allows us to manipulate words generated by all kinds of roots (3-letter roots, 4-letter roots and 5-letter roots).

The three examples in Table III share the same list of consonants LC. This list LC contains a prefix with one consonant X_1 = " ي ", a suffix with two consonants Z_1 = " و ", Z_2= " ن " , and a 3- letter root $R_1R_2R_3$ where R_1 = "ذ ", R_2 = "ـهـ ", and R_3 = "ب ". Table IV shows additional examples illustrating the decomposition of the list of consonants LC into prefixes, suffixes, infixes and root' letters.

Table IV. Decomposition of the list of consonants

Input Word	List of Consonants	Root R1R2R3	Prefix X1X2X3	Infix Y1 Y2	Suffixes Z1Z2Z3
سيـدرسون	[س ي د ر س و ن]	[د ر س]	[س ي]	[ ]	[و ن]
دارسـون	[د ا ر س و ن]	[د ر س]	[ ]	[ا]	[و ن]
مدارس	[م د ا ر س]	[د ر س]	[م]	[ا]	[ ]

6. 3. Morphological Pattern Representation

Each one of the morphological patterns is represented by a list L of characters with the same structure as we proposed for the words. The slots of the root letters are marked by '*', and may be replaced by any consonant. For example, the morphological pattern "يَـفْـعَـلْـوْنَ" is represented by the list [ي َ * ْ * َ * ْ و ن َ]; and decomposed into two lists: the list of consonants LC (ي * * * و ن) and the list of diacritical marks LV (َ ْ ْ َ ْ َ). This partition of consonants and diacritics reduces significantly the number of patterns to be tested. The characters '*' represent slots where consonants can be inserted to form a real word.

The morphological patterns can be regrouped in classes according to their list of consonants. The patterns of the same class share the same list of consonants and they are different among one another in terms of the lists of diacritical marks. Table V shows an example of three different patterns of the same class; these patterns have the same list LC and have different lists of diacritical marks LV. The set of patterns will be represented by the set of consonant lists LC, where we associate with each entry all the possible and correct combinations of diacritical marks LV. The couplet LC and LV will determine the morphology of the word.

Table V. Grouping Patterns according to their list of consonants

Pattern	List of Consonants LC	List of Diacritical Marks LV
يَـفْـعَـلُـوْنْ	[ي *** و ن]	[◌ْ ◌َ ◌َ]
يُـفْـعَـلُـوْنْ	[ي *** و ن]	[◌ْ ◌َ ◌ُ]
يُـفْـعَّـلُـوْنْ	[ي *** و ن]	[◌ْ ◌َّ ◌ُ]

6. 4. Root and Pattern Identification

The step of identification of the root and pattern is realized by two recursive procedures: FindPattern and FindRoot. The recursive procedure FindPattern (LC(word), LC(pattern)) receives the list of consonants of the stem and returns the corresponding standard pattern. The recursive procedure FindRoot (LC(word), LC(Pattern), LC(Root)) receives the word and its pattern and extracts the root by comparing the two entries and applying the rules relaying the pattern to the root. The following Prolog style code represents these two rule based recursive procedures:

```
//Base (Stopping) case when the decomposition is terminated
FindPattern ([],_).
//Skip the letters of the stem located in the root letters slots of the pattern
FindPattern([Head|Tail1], ['*'|Tail2]):- FindPattern (Tail1, Tail2).
//Finding the letters of the standard pattern
FindPattern([Head|Tail1],[Head|Tail2]):- FindPattern(Tail1, Tail2).
//Base (Stopping) case when the Root is completely detected
FindRoot([ ], [ ] , Root).
```

// Skip of the letters added to the root according to the standard pattern

FindRoot ([Head|Tail1], [Head|Tail2], Root):- FindRoot (Tail1, Tail2, Root).

// Find the root letters, corresponding to slots '*' in the morphological pattern

FindRoot ([Head1|Tail1], ['*' |Tail2], Root]:- FindRoot (Tail1, Tail2, [Head1|Root).

7. TEXT ANNOTATION

The text annotation is based on a categorical approach that uses the constituent parts of the words generated from by previous phases**.** The words are automatically categorized and classified into predefined categories. Three main categories are defined: (1) the derivative words, (2) non-derivative words and (3) undefined words. The derivative words are the words that our system is able to extract their roots, patterns, prefixes, and suffixes. The pattern will determine for this kind of words, additional features such as type (name/verb), gender, number, etc.

The non-derivative words are Arabic words that are not generated from standard pattern such as pronouns, prepositions, conjunctions, question words, foreign names and the like. These words will be stored in predefined tables. The last category “undefined” word class is used to categorize the words that our system fails in classifying them into one of the first two categories. Table VI shows the main categories and subcategories used for the purposes of this study.

A categorical grammar notation is used to describe the results of the last phase. This notation determines the category and subcategory of each word. For the derivative words, their constituent parts (the root, morphological pattern and the prefixes and suffixes produced by the light stemmer) are also attached to this notation.

Additional attributes are attached to the derivative words based on the features and attributes of their standard pattern discovered by the morphological analyzer. These attributes are:

1. For the nouns: (Gender(M/F), Number(Singular/Dual/Plural), (Definite/Un-definite)).
2. For the verbs: (Gender (M/F), Number (Singular/Dual/Plural), Person (First/Second/Third), Time (Present/Past)

Table VII shows some examples of the results of the annotation phase.

Table VI. Word categories and subcategories

	Level 1	Level 2	Level 3
Word	Derivative (1)	Noun (1.1)	Constituent parts: root pattern Attributes: Gender Number Definite or Un-definite.
		Verb (1.2.)	Constituent parts: root pattern Attributes: Gender Number Person Time
	Non Derivative (2)	Fixed Words(2.1)	Pronoun (2.1.1)
			Preposition (2.1.2)
			Conjunction (2.1.3)
			Question (2.1.4)
		Foreign Words (2.2)	
		Proper Names (2.3)	
	Undefined (3)		

Table VII. Examples of annotated words

Word	Category	Constituent Parts	Attributes
اللاعبون The players	(1.1) Derivative words, Noun	Light Stemming Phase Prefix: ال (Article the) Suffix: Morphological Analyzer Phase Root: لعب Pattern: فاعلون	Gender (M) Number (P) Definite
تأكلها She eats it	(1.2) Derivative word Verb	Light Stemming Phase Prefix: -- Suffix: ها (Pronoun it) Morphological Analyzer Phase Root: أكل Pattern: تفعل	Gender (F) Number (S) Person (Third) Time (Present)
أين Where	(2.1.4) Non Derivative word Fixed word Question	--	--
أوروبا Europe	(2.2) Non Derivative Foreign word	--	--

8. EXPERIMENTS AND EVALUATION

8.1 Implementation

For the purpose of evaluating the performance and accuracy of our approach, a prototype of the proposed preprocessor has been implemented and tested on real Arabic texts. This implementation includes the different tables needed in the different phases of the preprocessing. A table entitled ROOT is defined to represent the roots in the language to generate the derived words; 672 three-letter roots and 41 four-letter roots are stored in this table for the purposes of this experiment. A Table entitled PATTERN aims at representing the standard and extended patterns; whereby each entry contains the list of consonants of the pattern along with all the possible combinations of diacritical marks. A table entitled AFFIXES contains the strippable affixes in the light stemming phase. A table entitled NON-DERIVATIVE is defined and contains the fixed words and tools of the language. In addition, the foreign words that the Arabic language has borrowed from other languages as well as the proper nouns have been listed.

In the Arabic language, as well as in many other languages, a word may belong to more than one category. The implemented prototype is designed to be able to produce all the possible and detected categories of the input words.

8.1. Data Sets

The performances of the proposed preprocessing technique have been empirically tested and evaluated on real data sets. Three different data sets have been used representing the three cases of texts: vowelized, partially vowelized and unvowelized texts. The texts have been collected from different resources. Manual treatments were necessary in some cases to eliminate or insert the diacritical marks to create these three different categories of sets. Table VIII shows more details about the data sets. It has been noticed that in despite the fact that the number of fixed words is limited, their frequency in real text is very high.

Table VIII. Data Sets

Data Set Category	Number of Words	Number of Derivative Words	Number of Non-Derivative Words
Vowelized text	1180	542 (46%)	638
Partially vowelized text	1467	820 (56%)	647
Unvowelized texts	1452	650 (45%)	802

8.2 Performance Measurement

In order to evaluate the quality and accuracy of the results, four performance measures have been used: the recall, the precision, the error rate, and the category precision. The last measure indicates the ratio of correct categories detected at the word level. These three measures are defined as:

Precision = C / N

Recall = C / TN

Error -Rate = E / TN

Category-Precision = TC / TD

Where: TN is the total number of words in the data set

E is the number of words incorrectly categorized

C is the number of words correctly categorized

N is the number of words categorized by the system (N = C+E).

TC is the total number of categories correctly detected

TD is the total number of categories detected.

8.3. Results

The results of the preprocessor are controlled manually to check their correctness. A word is considered correctly categorized if all the categories associated to this word are correct. A word is considered incorrectly categorized if any of the categories associated to the word is incorrect.

As the results depend directly on the above-mentioned tables, the words that the preprocessor might fail to identify and categorize due to the fact that their roots or pattern are not included

in the predefined tables are not taken into consideration in the performance analysis of the prototype. The results of the evaluation test for the three data sets are shown in Table IX.

Table IX. Results of the preprocessor on the three Data Sets

Data Set category	Precision	Recall	Error Rate	Category Precision
Vowelized text	0.97	0.88	0.027	#1.0
Partially vowelized text	0.93	0.85	0.062	0.94
Unvowelized texts	0.91	0.83	0.073	0.87

In general, the preprocessor has achieved the best results in the fully vowelized texts, whereby there was a very low rate of inappropriate results. The incorrectly categorized words are generally the words whose some of their letters are mixed up with the affixes, and then stripped during the light stemming phase. The Error rate is higher in the case of the partially vowelized and unvowelized texts, mainly because of the missed diacritical marks especially the diacritical mark (ّ) used to indicate the duplicate occurrence of a consonant.

9. CONCLUSION

We have developed, successfully, a four phase approach to the task of preprocessing Arabic texts: (1) text tokenization, (2) light stemming (3) morphological analysis and (4) text annotation. These four phases are applied in sequence and collaborate together in order to transform an Arabic text into a new format designated to be used for the NLP applications.

Our approach works on Arabic texts as they appear in real world documents. This approach can be used as a part of a larger application of NLP. The output of the preprocessing steps determine whether the word is generated from a root or not as well as a set of attributes that can be used by the NLP application to determine the part of speech of the word in addition to its possible meanings.

The proposed technique gives accurate results for the fully vowelized Arabic texts. In the absence of diacritical marks, it produces a list of possible morphological patterns, in general between 1 to 5 patterns for each word that share the same consonants. However, they are different from the others in terms of the list of diacritics.

In spite of the good, accurate and general results obtained by our approach, it has been noticed that it requires many tables and rules. It is assumed that an expansion of the proposed preprocessing techniques in the direction of using the syntactical information will help in getting more accurate results.

REFERENCES

- Abu Salem, H., Al-Omari, M., and Even, M. (1999) Stemming methodologies over individual query words for an Arabic information retrieval system. Journal of the American Society for Information Science, 50, 6, 524-529.
- Al-Sughaiyer, I. A., and Al-Kharashi I. A. (2004) Arabic Morphological Analysis Techniques: A Comprehensive Survey. Journal of the American Society for Information Science and Technology 55, 3, 189-213.
- Allen, J. (1995) Natural Language Understanding, The Benjamin/Cummings Publishing Company, USA. 654 pages.
- Antworth, E. (1994) Morphological Parsing with a Unification-based Word Grammar, North Texas Natural Language Processing Workshop, May 23, 1994. http://www.sil.org/pckimmo/ntnlp94.html.
- Beesley, K. R. (1996) Arabic Finite-State Morphological Analysis and Generation. In Proceedings of the 16th International Conference On Computational Linguistics, (Vol. 1, Aug. 9-15,1996) 89-94
- Darwish, K. (2002) Building a Shallow Arabic Morphological Analyzer in One Day. Proceeding of the Association for Computational Linguistics (ACL – 02, 40th meeting, University of Pennsylvania, Philadelphia). 47-54**.**
- De Roeck, A. and Al-Fares, W. (2000) A Morphological Sensitive Clustering Algorithm fo Identifying Arabic Roots. Proceeding of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000, Hong Kong, China).
- Grefensette, G. and Tapanainen, P. (1994) What is a word, What is a Sentence, Problems of Tokenization. The 3rd Conference on Computational Lexicography and Text Research. Complex'94, Budapest, July 7-10, 1994.

- Jurafsky, D. and Martin J.H. (2000) Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 1st ed., Upper Saddle River: Prentice Hall.
- Khoja, S., Garside, R., and Knowles, G. (2001) A Tag set for the Morphosyntactic Tagging of Arabic. Proceedings of the Corpus Linguistics. Lancaster University. http://archimedes.fas.harvard.edu/mdh/arabic/CL2001.pdf
- Larkey, L. S., Ballesteros, L. and ConnelL, M. E. (2002) Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. Proceeding of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02. August 11-15, 2002, Tampere, Finland.) 275-282.
- Manning, C. D. and Schutze, H. (2000) Foundation of Statistical Natural Language Processing, MIT Press, Second Edition, USA.
- Moukdad, H. (2006) Stemming and root-based approaches to the retrieval of Arabic documents on the Web. Webology 3, 1. http://www.webology.ir/2006/v3n1/a22.html
- Xu, J., Farser, A., and Weischedel, R. (2002) Empirical Studies in Strategies for Arabic Retrieval. Proceeding of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02. August 11-15, 2002, Tampere, Finland.). 269 – 274.