An Algorithm for Arabic Lexicon Generator Using Morphological Analysis

Samer Nofal

Department of Computer Science, Hashemite University, Zarqa, Jordan

Abstract. Several Natural Language Processing Systems (NLPSs) use lexicons, which are files that store information about words, such as word category, gender, number and tense. Computing lexical information instead of storing them will improve the time complexity of NLPSs. This work designs, implements and examines an algorithm for the Arabic morphological analyzer and lexicon generator. The algorithm is based on segmenting the word into a prefix, a stem and a suffix. The algorithm then tries to decide the fillers of the lexicon entries from the information contained in these segments. The algorithm makes several tests on the compatibility between the word components: the prefix, stem and suffix. This algorithm consults three types of lists for assertion purposes: prefixes list, suffixes list and stem lists. The algorithm was tested on three social and political articles of nearly 1300 words. The evaluation shows that we can depend on computational morphological analysis with a percentage of at least 80 percent. The 20 percent failure percentage is due to language exceptions and the hidden diacritics of Arabic words.

1 Background: Arabic Language Structure

In this paper, we put -where applicable- an explanation in the English language beside Arabic language examples to let the non-Arabic speakers get the point. Arabic examples are put between double quotation marks. English explanations are put between brackets.

Arabic language is a right-to-left language. Its letters are divided into three types: consonants, such as: " س ، ض ، ص ، ل ، ف ، خ ، ج ، ... " [examples in English are: B, C, D, F, G,…etc], vowels : " و ، ي ، ا " [examples in English are: A, E, O , U, I], and small vowels (diacritics): " َ ، ُ ، ِ " [there is no applicable example from English]. Arabic words are divided into three types: particles such as: " في ، إذا ، كلا ، ثم " [examples in English are: in, through, if, on, at …], verbs such as: " لعب ، يبتسم ، ادرس ، ليس ، يذهبون " [in English like: play, smile, study, go…etc], and nouns such as: " الإنسان ، المعلم ، المؤدبات " [examples in English are: human, teacher, beautiful,…etc, all words other than the verbs and the particles].

Morphologically, Arabic words are divided into two types: frozen and derived words. Nearly, most verbs and nouns are derived words. A derived word is a word that has basic letters plus other augmented letters. The basic letters are called the root of the

Nofal S. (2006).
An Algorithm for Arabic Lexicon Generator Using Morphological Analysis.
In *Proceedings of the 3rd International Workshop on Natural Language Understanding and Cognitive Science*, pages 57-70
DOI: 10.5220/0002472800570070

word, and the augmented letters are called the affixes. An affix may come before any one of the basic letters, and then it is called a prefix, or may come through the basic letters, and then it is called an infix, or may come after the basic letters, and then it is called a suffix. Roots may be made of three letters (triliteral) or four letters (quadriliteral) or five letters (quintiliteral), but the most common is the triliteral root. A slight different definition of these four terms is introduced later in section 3.1 for computation purposes. How are words derived from roots? Inflection science is concerned with this problem. In general, the derivation process is controlled by rules, and each type of roots has its own derivational rules. However, in most languages there are exceptions of these rules. Frozen words are words that do not come from a known root and through which no infixes might come. An example will be given by the end of this section.

Typically, the Arabic language is a vowelized language. This means that every letter has a companion small vowel. Vowel letters in Arabic are three letters: " ا ، و ، ي ". These letters have counter-part small vowels: " ُ ، َ ، ِ " respectively. Beside these basic diacritics, there are secondary ones such as: " ّ " , which means that the letter must be stressed (duplicated) in pronunciation. Small vowels or diacritics are not usually written explicitly in Arabic texts. These diacritics are very important in the Arabic language. Two similar words (which have the same letters) with different small vowels have different meanings, such as the two words: " أهلُك ، أهلَك " . The first word from the left side is a verb which means that someone killed another one, and the second word is a noun that means your family. Without the use of diacritics, the word is ambiguous, and only the context may uncover the intended meaning. The derivation process consists of five components: diacritics, augmented letters (affixes), basic letters (the root of the word), template: the place of the infixes with the root and some of the prefix letters, and a set of guidelines for how we can augment a prefix or a suffix. For example, we can derive several words from the root " بحث " [in English : to search]:

1. Derived word: " بَحَث" (searched), Template: "فَعَلَ" [these letters for templates will be explained later] , Prefix: nil, Suffix: nil, Category: verb, Tense: past, Number: singular, Gender: masculine.
2. Derived word: " بَاحِث "(researcher), Template: " فَاعِل" , Prefix: nil, Suffix: nil, Category: noun, Tense: undefined , Number: singular, Gender: masculine.
3. Derived word: " يَتَبَاحثون " (discuss), Template: "تَفاعَل" , Prefix: "ي ", Suffix: "ون", Category: verb, Tense: present, Number: plural, Gender: masculine.

The letters: "ف ، ع ، ل" in the template refer to the letters of the root, and the remaining letters are the infixes. For example, in the template: " تَفاعَل" in the second example, the three letters: "ف ، ع ، ل" refer to the three letters of the root: "بحث", and the letters: "ت ، ا" are the infixes. Again, the diacritics are part of the template. In the first example, the derived word does not change except in the small vowels. Changing the diacritic of one letter makes a difference in meaning besides a difference in morphological analysis. (e.g., the verb: " شَهِدَ " (witness) and the noun: "شُهْد " (honey)). Prefixes and suffixes are added to templates to add a new meaning. Prefixes and suffixes are common, that is one prefix such as "ال" can be added to any noun template, and the suffix "ني " can be added to any verb template. Prefixes and suffixes will be explained in more details later.

2 The Algorithm

2.1 Introduction

The proposed approach is hybrid from two approaches, the procedural-based approach and the slightly exhaustive approach. This is a procedural approach because it encompasses a syntactic analysis depending on the Arabic language grammars. It is a slightly exhaustive approach since the analyzer stores indispensable knowledge but not all the possible knowledge that the full exhaustive approach stores. To clarify the difference between this algorithm and the previous works two examples are presented here to touch one main difference and not the only difference. The first example is a work done by Al_Shalabi [3]. His work depends on consulting a list of all possible roots (which count in thousands) in the language to determine the root of the word. Our algorithm only stores a small subset of all possible roots in the language as we will explain in later sections. The largest subset (the triliteral roots which count in thousands) is excluded. Another example is the analyzer done by Timotly Buckwalter [4]. His analyzer uses a large list of stems instead of roots, which may be larger than the first list of Al-Shalabi work.

Particles and frozen words are not feasible for processing. Tthese kinds of words plus a list of other kinds (will be explained later) must be stored rather than processed. The ultimate aim of all works in the morphological analysis domain is to decrease the complexity time, so that the amount of information needed by a morphological analysis subsystem must be minimized to minimize the execution time of a natural language processing system. This is exactly what this work tries to achieve. This work tries to depend more on Arabic language inflectional grammars in analyzing words.

Our analyzer determines the following morphological analysis information for a word: root, category, gender, number and tense of the word. The category of a word may be a noun, verb or particle. The gender of a word is feminine or masculine. The number of a word is plural, dual or singular. The tense of a verb is command, present or past; for nouns: accusative, nominative or genitive.

This algorithm is based on the following assumption; each derived word in Arabic language is formulated according to the following formula:

Word = prefix + stem + suffix (1)
where stem = root (2) or stem = template (3)

A template consists of a root and one or more affixes (infixes or prefixes only) . The “+” in (1) means the traditional concatenation of letters. Formula (2) indicates that the stem may be equivalent to the root, which may be one of the following:

1. **Triliteral root:** For example, the word: "يلعبون " [in English: they play] could be segmented according to (1) as follows: prefix ="ي" [this prefix adds the present sense to verbs] , stem = "لعب " [play] , and suffix = "ون" [this suffix designates the subject which is a third person] . Note that the stem in this case is equivalent to the root.
2. **Quadriliteral root**: For example, the word: "فسيطروا" [in English: they controlled] is segmented to: prefix =" ف" [it is a connector designates ordering] , stem =" سيطر" [control] , and suffix ="وا" [designates a third person subject] . Note that the stem is a quadriliteral root.

3. **Quintiliteral root**: For example the word: "الإنسانية " [In English: The Humanity] is segmented to : prefix = " ال" [definite article "the"], stem = " إنسان" [human], and suffix = "ية " [designates the feminine]. Note that the stem is a quintiliteral root.
4. **Penetrated root:** A root that enters the Arabic language from other languages. For example, the word "التلفزيونات" [The televisions] is segmented to: prefix = "ال" [definite article "the"] , stem = "تلفزيون " [Television] , suffix = "ات" [designates feminine plural] . Note that the stem is a penetrated root which is taken from English "Television".

The traditional morphological analyzer may store these four lists, but attempts should be done to minimize the execution time. Our approach stores three lists only: quadriliteral roots, quintiliteral roots, and penetrated roots. The largest list of roots, namely the triliteral root list, is dropped. The other three lists are smaller since most of the derived words were derived from triliteral roots.

The stem according to formula (2) is called a root stem, and in case the stem is a template, as (3) indicates, it is called template stem. Templates are categorized into three types, each of which corresponds to one root type, except for the last root type whereas no templates for penetrated roots:

1. Triliteral root templates: For example, the word "سنستخرجهما" [in English: we will extract them] is segmented to: prefix = "سن"[designates the future tense], template stem = "ستخرج" , and suffix = "هما" [designates a third person object]. The template stem can be analyzed as: root = "خرج" , infix = "ست", and template = "ستفعل ".
2. Quadriliteral root templates: For example, the word "فالدراهم" [money] can be segmented to: prefix = "فال" [compound prefix designates a connector "and" and the definite article "the"] , template stem = "دراهم" , and suffix = no suffix . The template stem can be further analyzed as: root = "درهم" , infix = "ا ", and template = "فعالل ".
3. Quintiliteral root template: For example, the word "سفرجل " [kind of vegetables] can be analyzed as: prefix = no prefix, template stem = "سفرجل ", and suffix = no suffix. The template stem can be analyzed as: root = "سفرجل ", infixes = no infixes, and template = "فعللل".

Most derived words are derived from triliteral roots. This algorithm only considers the triliteral root templates. Quadriliteral and quintiliteral root templates are rare. This approach stores all possible template stems for quadriliteral and quintiliteral roots. To clarify this, the algorithm supposes that all the template stems for the root "دحرج" [he rolled something] are stored (which are: "متدحرج" [roller] " تدحرج" [something is being rolled]). Triliteral template stems will be explained in more details in a later section.

To summarize, to successfully analyze a word, it is important to segment it correctly into prefix, suffix and stem and identify the correct type of the stem. To accomplish this task, the analyzer must keep: a list of all particles, a list of all quadriliteral and quintiliteral and penetrated roots, a list of all quadriliteral and quintiliteral template stems, a list of all triliteral templates, a list of all prefixes, a list of all suffixes. At this point, we define the frozen word from the algorithm's point of view. A frozenword may refer to a particle, quadrilitiral root, quintilitiral root, penetrated root, quadrilitiral template, or quintilitiral template.

How can the lexical information be determined for a word? These information can be determined from the three parts of the word: prefix, stem and suffix. This algorithm uses the aforementioned lists that contain lexical information about the items listed. These lists will be explained in later sections.

2.2 The Major Steps of the Algorithm

This section introduces the algorithm in a high-level view. The algorithm is detailed in next sections.

1. The first step of the algorithm converts the compound letter "آ" into " أا" for processing purposes. For example, the word " مآكل" [restaurants] has a template "مفاعل" . To make the matching possible, the word is converted into "مأاكل" to match the template "مفاعل ".
2. A word may be made up of one or two letters and this occurs only in command verbs such as: " كل" (eat) and "ع" (understand). The algorithm complements these words by using two lists for each type, which means that "كل" has the root "أكل" and "ع" has the root "وعي" . Afterwards, the algorithm concludes that the word is a command verb and singular, and then it exits. Note that in case the word is a verb, then the number refers to the number of the subject of the verb, not to the verb.
3. The algorithm then tries to find a frozen word in the word by consulting the frozen words file. Look at Table 1 that shows a sample of the contents of the frozen words file.

Table 1. Sample of frozen words file.

tense	number	gender	category	root	word
غ [undefined]	غ [undefined]	غ [undefined]	حرف [particle]	إذا [if]	إذا [if]
غ [undefined]	مفرد [singular]	مذكر [masculine]	فعل [verb]	ليس [Not]	ليس [Not]
غ [undefined]	جمع [plural]	مؤنث [feminine]	اسم [noun]	عنصر [element]	عناصر [elements]

4. Two special cases arise:

- When the word ends with "ة" : In this case, we remove the "ة" and conclude that the suffix is a feminine noun. What does this mean? It means that the suffix only comes with feminine nouns.
- When the word ends with "وا" : In this cas, we remove "وا" and conclude that the word is a plural masculine verb.

After determining that the resulted word is made up of three letters, the algorithm goes through the following steps:

- In case the removed suffix is" ة", the algorithm concludes that the word is a feminine noun and the root is equal to the word, then exit.
- In case the removed suffix is "وا", the algorithm concludes that the word is a plural masculine verb and the root is equal to the word, then exit.

5. The algorithm then tries to find a template in the word. If we successfully determine a template in the word, it is easy to extract the root and determine the prefix and the suffix and all other needed information. We will explain how to

find a template in a later section. If a template is found, the algorithm tries to determine the lexical information, and then the program terminates.

6. In step 4, we remove "ة" and "وا" as a special case. If steps 4 and 5 are passed , then we return both of them to the word to complete the analysis process. " ة" and "وا" may be essential in the word such as in the word "شفة" [lip] , in which the " ة" was converted from the letter "و" , and so the root for this word is "شفو" .
7. If the algorithm does not find any template in the word, the algorithm assumes that the word has the following structure as the formula: Word = prefix + trilateral root + suffix, The algorithm tries to extract the root (see the root finding section) from the word according to the formula. The algorithm then tries to find the lexical information.
8. As a last step, if the root is not found, then the algorithm keeps the word as it is with no analysis. In case the algorithm found a root for a given word the algorithm then makes a further correction step by using a list of problematic roots. A problematic root is a triliteral root that has one of its letters altered by the phonetic changes phenomenon. For example the root "قال" [to say] has the letter "ا " that has been altered from" و". To give the right root, the algorithm must consult this list for correction purposes. This list is not very large because it contains only special rare roots, and it is not necessary to use it with every root for correction, but it is used in the roots that have one of the following letters: "و ، ا ، ي ، ء". Next sections discuss the files used by the algorithm and contain details of the algorithm.

2.3 Templates File

A sample of the file is presented in Table 2. All the templates in this file are triliteral templates. The symbol "؟" in the template refers to the holes in which roots are interdigitated with the template to form template stems. This list omits some rare triliteral templates such as: "ا؟؟ي؟ا؟" . Diacritics are dropped from the templates because this algorithm is concerned only with analyzing non-vowelized Arabic texts. This list is ordered from the longest template to the shortest one because one template may be a subset from another. For example the template "؟؟ا؟" is a subset of the template "أ؟؟ا؟ " . The templates: " ست؟؟؟ ، ن؟؟؟ ، ؟ت؟؟" are originally derived from the templates: "است؟؟؟، ا؟ت؟؟، ان؟؟؟" respectively. When a prefix is augmented with one of these original templates, the first letter " ا "is dropped. For example, the verbs: " يستنجد ، يندفع ، ينتظر" [from left to right, these verbs mean : is waiting, to go ahead, to ask for help] are derived from the verbs: " استنجد ، اندفع ، انتظر " respectively.

Table 2. Sample of Templates File.

Tense	Number	Gender	Category	Example	Template
غ [undefined]	مفرد [singular]	مذكر [masculine]	اسم [noun]	استغفار [asking forgivness]	است؟؟ا؟
غ [undefined]	مفرد [singular]	مذكر [masculine]	اسم [noun]	انتصار [win]	ا؟ت؟ا؟
غ [undefined]	مفرد [singular]	مذكر [masculine]	اسم [noun]	انكسار [refraction]	ان؟؟ا؟
غ [undefined]	مفرد [singular]	مذكر [masculine]	اسم [noun]	متقاعس [lazy]	مت؟ا؟؟

This list contains one special synthesized template: "تت؟؟؟ ", which was added to solve the ambiguity that may occur in some words such as the word: "تتجمدون" [which means you will be frozen] if the template"تتفعل" is not added to the list then the template"تفعل " will be matched to the subword: "تتجمم" . Further analysis shows that this match is incorrect.

Every template has a specification for: category, gender, number and tense. These specifications mean that a template stem that has one of these templates has the specifications associated with that template. For example, the template stem "استخدام" [which means: using] it is according to the template "است؟؟ال", so it obtains the specification associated with this template, namely: "اسم مذكر مفرد" [from left to right, this means: singular, masculine, and noun] . These obtained specifications play a major role in determining the specifications for a word that has either a prefix or a suffix as we will see in a later section.

2.4 Suffixes File

A sample of this file is presented in Table 3. Entries: Category, Gender, Number and Tense mean that a suffix only comes in these cases. For example, the suffix "تم" only comes with a past tense verb, and the subject of the verb is masculine and plural . The Gender and Category entries refer to subjects in case the suffix is a verb suffix.

The suffixes are listed in the file in a special order. This ordering eliminates the need to enumerate all the possible combinations of suffixes. For example, consider the word: "أعطيناكموها " [we gave it to them] which can be segmented into the suffix =" ناكموها"and the template stem = "أعطي ". There is no entry for the suffix "ناكموها", so how can this word be analyzed? The algorithm starts searching for a suffix in the word from left to right. The first entry, which has a match in the given word, is the suffix "ها". The algorithm then removes it from the word and keeps the lexical information attached with the suffix (the remaining entries in the same row) for further processing. The procedure then continues looking in the table below the entry "ها" and does not restart from the beginning of the table. Similarly, the suffixes "كمو " and "نا" will be removed.

Table 3. Sample of suffixes file.

Tense	Number	Gender	Category	Example	Suffix
غ [undefined]	غ [undefined]	مؤنث [feminine]	اسم [noun]	الإنسانية [Humanity]	ية
غ [undefined]	غ [undefined]	مؤنث [feminine]	اسم [noun]	المدرسة [school]	ة
غ [undefined]	جمع [plural]	مذكر [masculine]	فعل [verb]	جلسوا [they sat]	وا
غ [undefined]	غ [undefined]	غ [undefined]	غ [undefined]	الأردني [Jordanian]	ي

2.5 Prefixes File

A sample of this file is presented in Table 4. The prefixes in this file are ordered from longer to shorter. Prefixes in the Arabic language are divided into two types: verb prefixes: which go with verbs, and noun prefixes: which go with nouns. Verb prefixes are: present tense letters "ي، ت،أ، ن", ordering letter "ل", question letter "أ", succession letter "ف", willing letter "س", waw al-atf "و" (the equivalent in English is the conjunction "and"). Noun prefixes are: genitive letter "ل", question letter "أ", succession letter "ف", genitive letter "ب", genitive letter "ك" , the definite article "ال", waw al-atf "و" (the equivalent in English is the conjunction "and"). Prefixes can somehow be combined:

1. Verb prefixes are formulated in two ways:
   a. Verb + [ي، ت،أ، ن] + [س ، ل] + [ف ، و].
   b. Verb + [ي، ت،أ، ن] + [أ].

The letters between brackets are mutually exclusive. For example, if the willing letter "س" is augmented to a given verb, then there is no ordering letter "ل" in that verb. The brackets sign means that this is optional, and the plus sign "+" refers to ordinary string concatenations. The question letter "أ" does not come before any of the prefixes except for the present letters "ي، ت،أ، ن", as shown in the second formula. Consider as an example the formulation of the prefix : "فسن" . Obviously, this prefix is formulated according to the first formula, whereas the succession letter "ف" is concatenated with the willing letter "س" , then the resulted string is concatenated with the present letter "ن".

2. Noun prefixes are formulated according to the following formula:
   Noun + [ال] + [ل ، ب ، ك] + [ف ، و]+ [أ]

Table 4. Sample of prefixes file.

Tense	Number	Gender	Category	Example	prefix
مجرور [genitive]	غ [undefined]	غ [undefined]	اسم [noun]	فبالعلم [then with education]	فبال
مجرور [genitive]	غ [undefined]	غ [undefined]	اسم [noun]	وكالأسد [and as a lion]	وكال
مجرور [genitive]	غ [undefined]	غ [undefined]	اسم [noun]	وبالتقدم [and with progress]	وبال

2.6 Finding Frozen Words

The following are the steps to find a frozen word in a word:

1. Search in the given word for a frozen word using the frozen word file.
2. If a frozen word is found, then:
   Prefix = the letters before the frozenword,
   Suffix = the letters after the frozenword.
3. The generated prefix and suffix from step 2 must be asserted (see the next two sections).

The content of the frozen word file has been explained previously. If a frozen word is found in a word then the function must assert that the match is correct. The assertion step is important because a frozen word may match incorrectly a subset of the given word. For example, the frozen word "إن" is a subset of the word "الإنسانية " [Humanity], but when the assertion step is performed, this frozen word is thrown away and another one is tried because the suffix "سانية " is not correct.

2.7 Asserting Suffixes

The steps to assert a suffix are:

1. Let temporary suffix (the suffix that is to be asserted).
2. Let possible suffix (the suffix that is read from the suffixes file.)
3. If the length of the temporary suffix is equal to the length of the possible suffix, then:
   a. If the possible suffix is equal to the temporary suffix, then the algorithm tests the compatibility between the temporary suffix and the stem. For example, a noun

stem cannot come with a verb suffix. One of the following conditions must be met to achieve the compatibility:

i. temporary suffix category = stem category, and one of the following conditions must be met:
   1. Temporary suffix tense = stem tense.
   2. Temporary suffix tense = undefined.
   3. Stem tense = undefined.

ii. Temporary suffix tense = undefined.

iii. Stem tense = undefined.

b. If the possible suffix is a subset of the temporary suffix, then temporary suffix=temporary suffix - possible suffix. So far, a subset of the suffix has been asserted. The algorithm must build the lexical information of the whole suffix from the lexicon information of part suffixes. For example, the word "حملوني" [they carried me], the suffix "وني" is composed of two suffixes: "و ، ني ". How can we determine the lexical information of the whole suffix from the part suffixes information? This will be discussed in later sections.

If the possible suffix is a subset of the temporary suffix, then the algorithm removes the possible suffix from the temporary suffix and continues to assert the whole temporary suffix. This is according to the fact that the suffix file is ordered in a special form as explained in the suffixes file section. For example, the suffix "ناكموها" is asserted in three steps:

1. Firstly, the suffix "ها" is asserted then removed from the entire suffix.
2. Secondly, the suffix "كمو" is asserted then removed from the entire suffix.
3. Thirdly, the suffix "نا" is asserted then removed from the entire suffix.

Notice the importance of the order of the suffixes file. The suffix "ها" must come before the suffix "كمو". Otherwise a big error would arise.

2.8 Asserting Prefixes

This step is simple because the prefixes file contains all possible combinations of prefixes as shown before. The compatibility between the prefix and the stem is similar to the compatibility test procedure as illustrated in the asserting suffixes section.

2.9 Finding Suffix Category

This section explains how to determine the category and tense of the whole suffix. The following steps are self-explanatory:

1. If the whole suffix category is defined, and the part suffix category is also defined , and whole suffix category does not equal part suffix category, then the whole suffix category is undefined, and the whole suffix tense is also not defined.
2. If the whole suffix category is undefined, then the whole suffix category is equal to the part suffix category, and the whole suffix tense is equal to the part suffix tense

2.10 Finding Suffix Gender

This section explains how to determine the gender of the whole suffix.

1. If the Whole Suffix Gender or the Part Suffix Gender is feminine, then the Whole Suffix Gender is feminine.
2. If the Whole Suffix Gender is undefined and the Part Suffix Gender is masculine, or the Whole Suffix Gender is masculine and the Part Suffix Gender is undefined, then the Whole Suffix Gender is masculine.

2.11 Finding Suffix Number

This section explains how to determine the number of the whole suffix. The singular form is the default form and the plural or dual forms are specific forms. This means that there are marks that must exist to prove the plural form or the dual form, but there are no such marks for singular forms. This approach starts from the specific form to the general one because the specific form outperforms the general form. The following are the steps:

1. If the Whole Suffix Number is equal to plural, then go to step 6.
2. If the Part Suffix Number is plural, then the Whole Suffix Number is plural, and go to step 6.
3. If the Whole Suffix Number is dual, then go to step 6.
4. If the Part Suffix Number is equal to dual, then the Whole Suffix Number is dual, and go to step 6.
5. If Part Suffix Number is singular, then the Whole Suffix Number is singular, and go to step 6.
6. Terminate.

2.12 Finding Word Category

Verbs are more specific than nouns because verb marks are more than noun marks. Any evidence for a verb must be considered first. The steps are below:

1. If the category of the prefix or suffix or stem is verb, then the word category is verb, and the word tense is equal to the prefix tense or suffix tense or stem tense, then go to step 4. Notice that the tense is associated with the category; this is true because the tense describes the category of the word not the word.
2. If the category pf the prefix, suffix or stem is noun, then the word category is noun and the word tense gets the tense of the prefix , suffix or stem. Go to step 4.
3. Otherwise, the word category and the word tense are undefined.
4. Terminate.

2.13 Finding Word Gender

This process is based on the rule: the feminine evidence is the determining factor because the masculine is the default, and the feminine is the special case. The Arabic language has feminine marks such as "ة" but has no masculine marks.

1. If the gender of the prefix, suffix or stem is feminine, then the word gender is assigned the gender of the prefix, suffix or stem. Go to step 4.
2. If the gender of the prefix, suffix or stem is masculine, then the word Gender is assigned the gender of the prefix, suffix or stem. Go to step 4.
3. Otherwise, the word gender is undefined.
4. Terminate.

2.14 Finding Word Number

The following is the ordering of evidence from the most evident to the least evident for number determination: plural, dual then singular.

1. If the number of the prefix, suffix or stem is plural, then the word number is plural. Go to step 5.
2. If the number of the prefix, suffix or stem is dual, then the word number is dual. Go to step 5.
3. If the number of the prefix, suffix or stem is singular, then the word number is singular, Go to step 5.
4. Otherwise, the word number is undefined.
5. Terminate.

2.15 Finding Templates

Firstly, this process tries to find a template assuming that the word has no suffix. If it fails to find a template, then it tries to find a template with the assumption that the word may have a prefix and a suffix. At every possible template in the word, the algorithm must assert the possible segmentation of the word by asserting the suffix and the prefix which are generated from the segmentation as explained in previous sections. For example, in the word "فاستخرج " [then he extracted], the template "استفعل" occurs in the word, so the prefix = "ف "and the root is "خرج".

2.16 Finding Roots

This process assumes that the word is composed from: prefix + triliteral root + suffix. The process is initiated if the template finder process fails to find an appropriate template. This failure implies that the word is derived from augmenting the triliteral root by a prefix and a suffix. The following are the steps involved in this process:

1. If a prefix is found in the word and the length of the word after the prefix is removed is greater or equal to three letters, then the prefix is removed. This test is important to assure that the algorithm does not remove a root letter.

2. After removing the prefix in step 1, the algorithm considers the root is the first three letters of the word.
3. If the length of the word after removing the prefix in step 1 is greater than 3, then the suffix is considered the remaining letters after the root letters.
4. In this step, the extracted suffix in step 3 must be asserted as shown in section 3.8. If the suffix assertion fails, then the root is undefined and the word remains as it is.

For example, the word "يذهبون"[they go] is segmented according to this function into: prefix =" ي ", root = "ذهب ", and suffix = "ون" .

3 Evaluation

The algorithm is implemented using Microsoft Visual Basic. The program is examined using three social and political articles from Al-Hayat newspaper. These articles contain nearly 1300 words. The frozen word file and problematic roots have not been fully filled, but they are filled with a few examples to show the algorithm behavior. Any word that contains a frozen word is excluded from the evaluation. Only words that contain derived words are evaluated. The program fails to analyze some words for two reasons:

1. A template may incorrectly match a part of a word. Table 5 shows some examples. This failure is natural since this approach depends primarily on templates in analyzing words, and as it has been shown, that the diacritics are part of these templates, so when a part (diacritics) is dropped, then the whole template is distorted and failure takes place. This phenomenon is called Ambiguity in the Arabic language and it takes place only in non-vowelized Arabic texts. For example, the word "نقود" may be a verb meaning “we are driving” or a noun meaning “money”.
2. Some words have one of their letters converted from an original letter or dropped or unified with a similar letter. This phenomenon is called the phonetic change phenomenon. For example, the word "اتصل" [contact] is derived from the root "وصل" according to the template "افتعل" . But the letter "و " is converted to "ت" then unified with the original "ت" . Another example is the word "إعانة" [help] which is derived from the root "عون" according to the template "إفعالة". But the letter " و " is dropped. The objective of this phenomenon is to make the pronunciation easier; and it really does.

Table 5. Examples of words that the program fails to analyze.

Word	**Incorrect Similar Template**
توجد [there exist]	فوعل
أصول [roots]	أفعل
دارنا [our home]	فاعل
غبائه [his stupidity]	فعائل
فواجب [it is compulsory]	فواعل
يضيق [it is getting narrow]	فعيل
مقال [article]	فعال
منتجاتها [its products]	منفعل

4 Conclusion

A morphological analysis algorithm and lexicon generator algorithm for non-vowelized Arabic texts is designed and tested. The algorithm is based on segmenting the word into the triplet: the prefix, suffix and stem. The algorithm then uses the lexical information contained in these triplet to determine the lexical information for the whole word. This work concludes from the evaluation that we can depend on algorithmic morphological analysis in determining the lexical information for Arabic words with a percentage of 80 percent. The 20 percent failure percentage is due to language exceptions and hidden diacritics.

References

1. Darwish, k. (2002). Building a shallow Arabic Morphological Analyzer in One Day. In Proceedings of the Association for Computational Linguistics (ACL-02), 40th Anniversary Meeting (pp.47-54), University of Pennsylvania, Philadelphia.
2. http://www.xrce.xerox.com, Access Date: march/15/2004.
3. A1-Shalabi, R, and Evens, M. (1998). *A Computational Morphology System for Arabic*. Workshop on Computational Approaches to Semitic Languages, COLING-ACL.
4. Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0, Linguistic Data Consortium (LDC) catalog number LDC2002L49 and ISBN 1-58563-257-0.
5. Al-Jlayl, M., & Frieder, O. (2002). On Arabic search: Improving the Retrieval Effectiveness via Light Stemming Approach. In Proceedings of the 11th ACM International Conference on Information and Knowledge Management, Illinois Institute of Technology (pp. 340-347). New York: ACM Press.