Building an Arabic Words Generator

Afnan Aqel
Department of Computer Science Faculty of Computing and Information Technology
King Abdulalziz University

Sahar Alwadei
Department of Computer Science Faculty of Computing and Information Technology
King Abdulalziz University

Mohammad Dahab, Ph.D
Department of Computer Science Faculty of Computing and Information Technology
King Abdulalziz University

ABSTRACT
Morphology studies the word structure considering its basic meaningful units. It has been always one of the most important components in nearly any application of Natural Language Processing (NLP). Through applying that concept on Arabic language, Arabic words were highly inflected and derived. In this paper an approach is going to be developed that will deliver almost all the words can be produced out of any submitted word. Beside that, this paper will answer the question 'whither or not inflected and derived words can be equally produced using the same methodology?'. Furthermore, many ideas of developing the algorithm presented here are discussed.

General Terms
Natural language processing.

Keywords
Arabic morphology, word structure, derivation, inflection, word generator.

1. INTRODUCTION
Given the enormous number of Arabic speakers and the expectations of meeting their technical needs, it has become imperative to produce what suits them and which can be in conformity with their own language. Thence, simulating that language is certainly a must, but the shortage of resources and projects in this aspect has become an obstacle. Therefore, Arabic Natural Language Processing (ANLP) – which is the scientific notion considers that field – has gained increasing importance, and several state-of-the-art systems have been developed for a wide range of applications, including machine translation, information retrieval and extraction, speech synthesis and recognition, localization and multilingual information retrieval systems, text to speech, and tutoring systems [10].

The development of NLP technologies for Arabic is a challenging task where it is a language of extensive morphology with both derivational and inflectional formats. Furthermore, that richness of Arabic morphology makes it difficult to be analysed or generated as a critical part of the whole process of simulation. On the one hand, analyzing Arabic morphemes has come a long way where number of analyzers is already in the field. Those analyzers offer rang of results using different approaches. Despite their known defects, they are providing a reliable source of computational analyzing even their flaws is considered to be useful in different circumstances. On the other hand, generating Arabic words has also received a lot of attention due to its usage in a large number of required daily applications like spell checking [5] as an instance.

In this paper a general-purpose approach of generating words in Arabic is presented. The method will apply an idea of producing words based on another given word in the form of added prefixes followed by a stem extracted from the given word then tailed by suffixes. The prefixes and suffixes are chosen based on the stem pattern that is going to be provided in the earlier stages of this process.

2. WHAT IS ARABIC MORPHOLOGY?
Morphology refers to the study of word structure or form [7]. Its basic concept is morpheme where the smallest expressive unit of a language is encountered. For example the word (ÇaAmiluwn, "عاملون", 'workers')1 consists of two morphemes (ÇaAmil, "عامل", 'worker') and (uwn, "ون", 's: suffix for plural, precisely masculine').

There are some key concepts should be defined first to explicate the formation of an Arabic word, Fig. 1.:

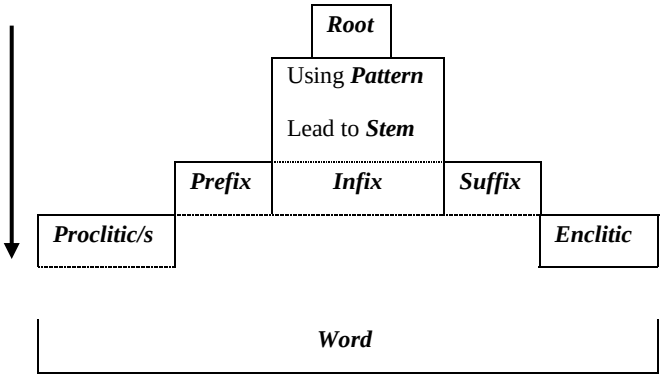

Fig. 1: Word structure in Arabic language

- *Root:* relatively invariable discontinuous bound morpheme, represented by two to five phonemes[2] [6], where they are the core part of a word that expresses the basic meaning [9].
- Pattern: bound and in many cases, discontinuous morpheme consisting of one or more vowels and slots for root phonemes [6]. Each pattern conducts a meaning that defines at least one grammatical feature.
- Affixes: each is a set of morphemes attached to the stem. It could be before the stem; prefix, within the stem; infix, or after the stem; suffix.
- Clitics: morphemes that attach to the stem after affixes. They are categorized by their position of words whither they are placed in the beginning or the end of the word to be consecutively proclitic or enclitic. On the one hand, proclitics are representing conjunctions where the enclitics are denoting pronouns [9][11].
- Stem: a word that is common in all of its inflected forms [12], which are the generated forms after adding syntactic features such as tense, number, person, case, etc. [1] through affixes and clitics. It can be a

[1] (Arabic Transliteration [8], "Arabic Form", 'English Meaning')
[2] Phonemes are any of the perceptually distinct units of sound in a specified language that distinguishes one word from another [6].

derivative that will be identified with a root and pattern, or can be a non-derivative [11].

Applying the previous terms constructing the word on an example would considerably declare that structure, Fig. 2.

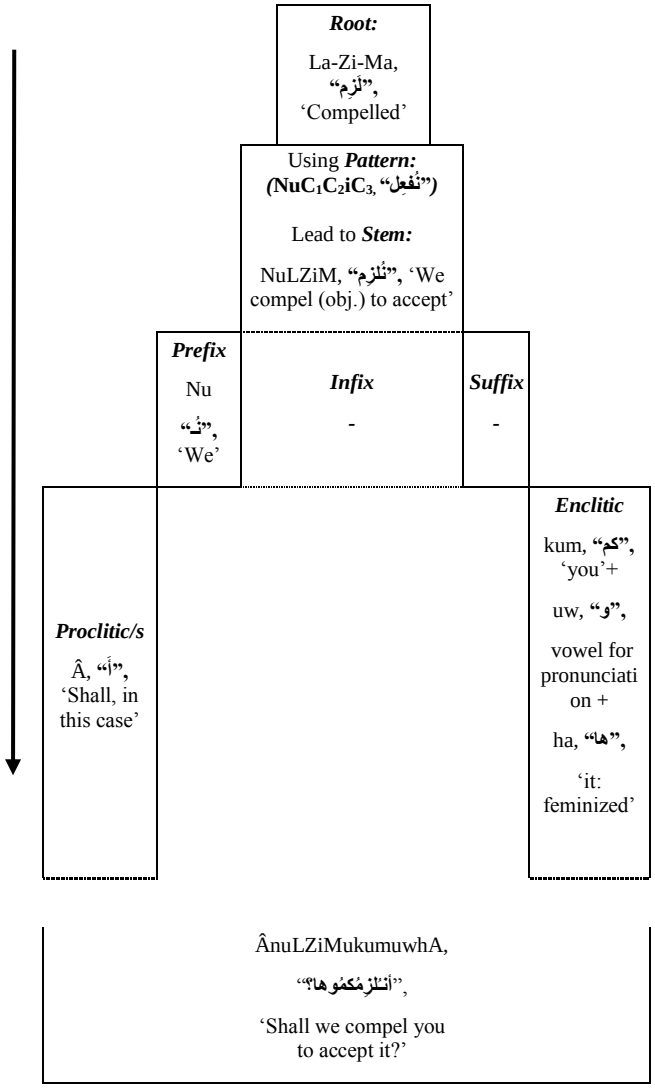

Fig. 2: Word structure in Arabic language applied to "أَنلزِمُكمُوها"

In fact, Arabic morphology is extremely systematic and it normalizes those approaches associated with the clarified concepts are cataloged into two models: derivation and inflection.

2.1 Derivation

Arabic morphology is using morphemes based on a system of consonant roots that interlock with patterns to form a new word [6]. Also the roots could contain vowels and those will be considered as main parts of the root too. For example, the root of the word (ϛaAmil, "عامل", 'worker') as a sample of a root consists of consonants is (ϛamil, "عَمِل", 'work'), and that word is a stem of this root formed by following the pattern (C1aAC2iC3, "فاعل", Active Participle [13]), where the notations C1, C2 and C3 are the main components of the root (i.e. ϛ-m-l in this case) [2][9]. Another instance can be shown for other roots having a vowel as a main part, is the word (qaAŷil, "قائل", 'the man who said'), where its root is (qaAl, "قَال", 'says'), and its core components are q-A-l (i.e. the "A" is a vowel here). Principally, neither an Arabic root nor a pattern can be used in isolation; they need to connect with each other in order to form actual words [6]. Detailed example is shown in Table 1.

Table 1. Example of Derived Words from Their Root

Root	Pattern	Derived Word	Gloss	Derivation Category
s-n-ϛ, "ص ن ع", 'made'	C1aAC2iC3	saniϛ, "صَانِع"	'maker'	Active Participle
	maC1C2owC3	masnowϛ, "مَصْنُوع"	'something has been made'	Passive Participle
	maC1C2aC3	masnaϛ, "مَصْنَع"	'manufactory'	Adverb
	C1iC2aAC3aħ	sinaAçaħ, "صِناعَة"	'manufacture/ industry'	Craft Noun

2.2 Inflection

The term "inflection" generally refers to phonological changes a word undergoes as it is being used in context, where the core meaning of that word remains unchanged [6] [9].

Arabic words are generally marked for grammatical categories to represent the inflectional process upon those words. There are eight major grammatical categories in Arabic: tense/aspect, person, voice, mood, gender, number, case, definiteness [6]. Employing those categories upon a stem would result in an integer of words annotated with their grammatical meaning, as presented in Table 2.

Table 2. Example of Inflected Words from a Stem

Stem	Inflected Word	Inflectional Categories			
		Gender	Number	Case	Definiteness
saAriq, "سَارِق", 'Thief' From the Root: s-r-q, "س ر ق", 'stole' Using the Pattern: C1aAC2iC3	saAriqaħ, "سَارِقة"	Feminine	Singular	Nominative Genitive Accusative	Indefinite
	saAriqataAn, "سَارِقَتان"		Dual	Nominative	
	saAriqatayn, "سَارِقَتين"			Genitive Accusative	
	saAriqaAt, "سَارِقات"		Plural	Nominative Genitive Accusative	
	saAriqaAn, "سَارِقان"	Masculine	Dual	Nominative	

	saAriquwn, "سَارِقون"		Plural	Nominative	
	saAriqayn, "سَارِقين"		Dual Plural	Genitive Accusative	

3. GO TO DERIVATIONAL OR INFLECTIONAL GENERATION?

Generating words given a specified word could be accomplished through one of the morphological approaches enlightened above. On the one hand, using the derivational technique will produce variant collection of words in which the meaning of the original input could be even altered. The root of the word entered is needed where all the words generated will be formed as a combination of that root and the suitable patterns. On the other hand, inflectional procedures will reserve the pure meaning of the word and will turn out the proper words based on grammatical factors. The pattern of this word as well as its Part Of Speech (POS) should be declared where it accordingly will be inflected. In this paper the second approach is implemented since keeping the same meaning for the produced words compared to the original entered one is the defined goal of the process developed in this paper.

4. METHODOLOGY

In the following two subsections the method of generating is elucidated. Section 4.1, accompanied by Fig. 3, presents the algorithm while section 4.2 clarifies its symbols and mechanism and section 4.3 shows a trace of the algorithm.

4.1 Algorithm

Generate Words:
Input: String Word.

Output: Result.
Declare variables: Word, AR [], Result, i.

Word ← Input from the user

AR [] ← Analyze(Word)

Result ← Null

i ← Size(AR)

if AR [i].Type ← Tool_Word or Proper_Noun
or Except_Word

return Result

else

Word ← Stem(Word)

Result +Generate(Word, Common_Affixes)

do

if AR [i].Pattern = Nominal

Result +Generate(Word, Noun_ Affixes)

else

if AR [i].Pattern = Verbal

Result +Generate(Word, Verb_ Affixes)

else

return Result
i --
while(i ≠ 0)

return Result

4.2 Demonstration

Receiving a word from the user is the start point of the Generate Words algorithm after declaring different variables to monitor its procedures. These variables are:

- Word: String variable to store the user input, then later store the returned value of Stem method.
- AR []: The Analyzing Results list of objects.
- Result []: A list that collects all the possible generated words.
- i: The counter that maintains the loop based on the size of the AR list.

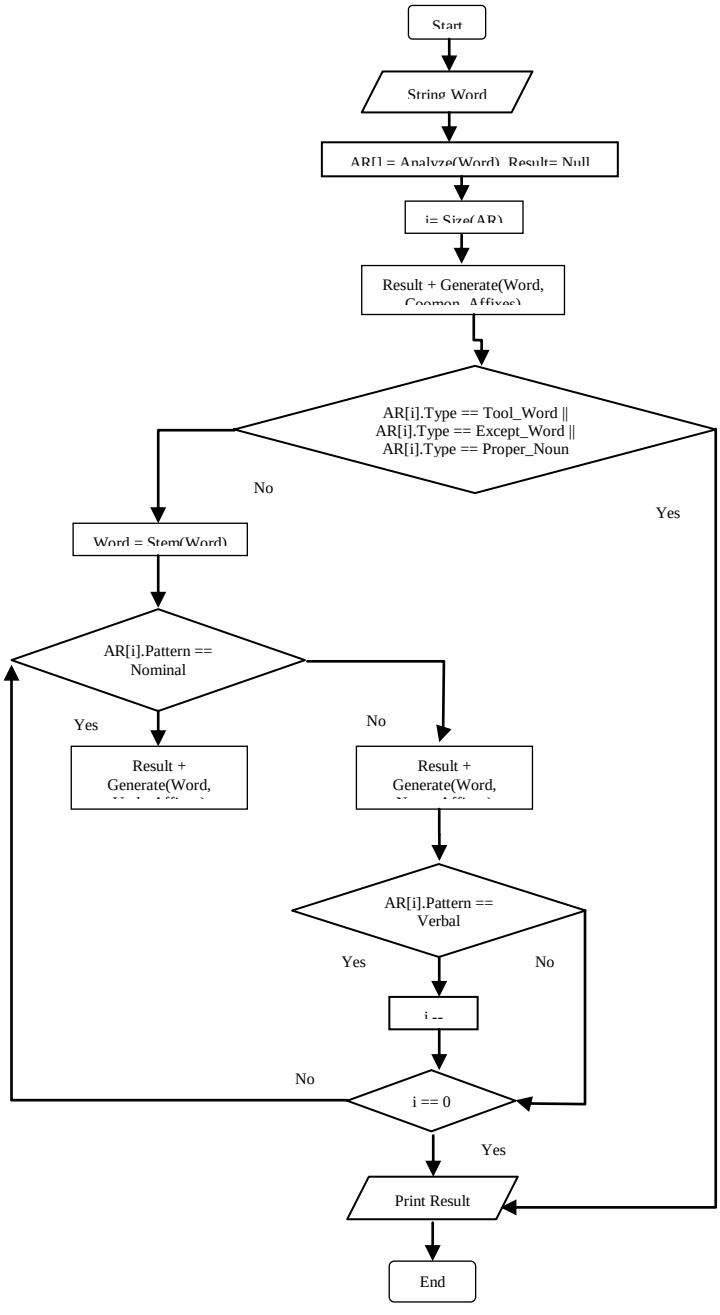

Fig. 3: Word Generator Flowchart

Besides, there are two important methods by which the algorithm task is accomplished. On one hand, there is the Analyze method, which takes one parameter and returns a list of objects. On the other hand, the Generate method takes two parameters and returns a list of strings represents the probable generated words.

Mainly this algorithm gives a great attention to the returned analyzing results from the Analyze method. Based on those results different dissensions are taken. Essentially, Word value is sent to the Analyze method where it is going to be examined and its features are defined. Each possible result of this method is stored as an object to be a part of the list returned from that method – the AR list.

The values of AR objects variables have a significant effect on the generating process. The first parameter of Generate method is a new value returned from sending Word to the method Stem, which provides the stem of the word entered by the user. Given the indicated value of AR[i].Pattern of a Word, the list of affixes (see Appendix) to be sent to the Generate method as a second parameter is chosen accordingly. Those list of affixes are categorized into three main classes: Common_Affixes that is suitable to be used along with any word submitted by the user then later stemmed; Noun_Affixes and Verb_Affixes are picked to be added also based on the Pattern value: Nominal or Verbal respectively.

This algorithm focus on generating words of nouns and verbs but it avoids particular types of word. Those Types are Tool Words, Proper Nouns and some Exceptional Words, and they are defined earlier to be checked at the very beginning of the algorithm course.

4.3 Trace

The algorithm has been applied on the word (masnaç, "مَصْنَع", 'manufactory') where it operates as follows:

Input: مصنع.

Word ← Input from the user

AR [] ← Analyze(مصنع)

Result ← Null

i ← Size(AR)

if AR [i].Type ← Tool_Word or Proper_Noun
or Except_Word
return Result

else
Result +Generate(مصنع, Common_Affix)

do

if AR [i].Pattern = Nominal
Word ← Stem(Word)
Result +Generate(مصنع, Noun_Affix)

return Result

Output:

مصنعك	أفمصنع	لمصنع	فللمصنع
مصنعكما	لمصنع	ولمصنع	أوللمصنع
مصنعكم	ولمصنع	فلمصنع	أفللمصنع
مصنعكن	فلمصنع	ألمصنع	كالمصنع
مصنعه	المصنع	كمصنع	وكالمصنع
مصنعيه	والمصنع	وكمصنع	فكالمصنع
مصنعها	فالمصنع	فكمصنع	أكالمصنع
مصنعيها	أالمصنع	أكمصنع	لكالمصنع
مصنعهما	أو المصنع	لكمصنع	أوكالمصنع
مصنعيهما	أفالمصنع	فومصنع	أفكالمصنع
مصنعهم	للمصنع	بالمصنع	مصنعان
مصنعهن	وللمصنع	وبالمصنع	مصنعين
مصنعنا	فللمصنع	فبالمصنع	مصنعون
مصنعي	بمصنع	وبالمصنع	مصنعات
ومصنع	وبمصنع	أوبالمصنع	مصنعا
فمصنع	فبمصنع	أفبالمصنع	مصنعوا
أمصنع	أبمصنع	للمصنع	
أومصنع	لبمصنع	وللمصنع	

5. CAN THIS METHODOLGY PRODUCE DERIVED WODS?

The generated result of the demonstrated methodology above has achieved a high accuracy where the words produced are valid and precise. Those words in general are inflected where none of them is derived since they kept the pure meaning of the submitted word and their stems follow exactly the same pattern of the submitted word.

Thus, producing the derived forms of a word would have a very different model where the Roots and Patterns are the main factors. The first step of such an algorithm is to get the root of the input word then direct its main components to the suitable patterns. As these letters are sent, the generating method will return the new-formed words as an output. Obviously, inflectional and derivational morphologies are following dissimilar approaches. Therefore, the exact algorithm developed here will not serve this purpose.

6. EVALUATION

The process used to evaluate the method is to compare the forms resulted from the algorithm declared above (section 4.1) to the forms founded by a corpus. The corpus used in this practice is "arabicCorpus" which has a total of 173,600,000 words out of five main categories or genres: Newspapers, Modern Literature, Nonfiction, Egyptian Colloquial, and Premodern [14].

Submitting the same word "مَصْنَع" to the corpus examining all the corpora and declaring the word type as a noun will reveal the following forms:

مصنعهما	مصنعيه	فالمصنع	المصنعين
بمصنعي	ومصنعنا	ومصنعان	مصنعي
كمصنعين	بمصنعين	ومصنعه	لمصنع
مصنعك	لمصنعيها	فمصنع	مصنعين
بمصنعه	للمصنعين	كالمصنع	مصنعها
وللمصنع	لمصنعي	مصنعيهما	بمصنع

والمصنع	ومصنعها	ومصنعين	المصنعان
بالمصنع	لمصنعين	لمصنعها	ويصنع
مصنعه	مصنع	مصنعان	ومصنعيه
ومصنعا	المصنع	كمصنع	بالمصنعين
والمصنعين	مصنعا	مصنعنا	ومصنعيها
ومصنعي	ومصنع	مصنعيها	لمصنعنا
	للمصنع	مصنعهم	لمصنعه

Exploring this result alongside with the previous one displayed within the 4.3 section would lead to:

- The number of words generated by the algorithm is larger than the one brought by the corpus. Fig 4 is presenting the word "مَصْنَع" case.

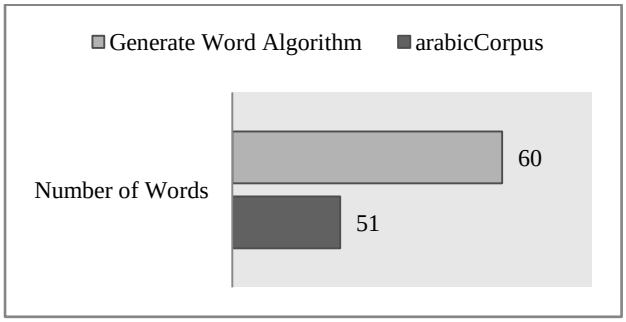

Fig. 4: Comparison between the number of words generated from the word "مَصْنَع" by the algorithm and the corpus

- Some of the generated forms are not used frequently Fig 5 even they are grammatically valid. Thus, they are not included in the corpus results.

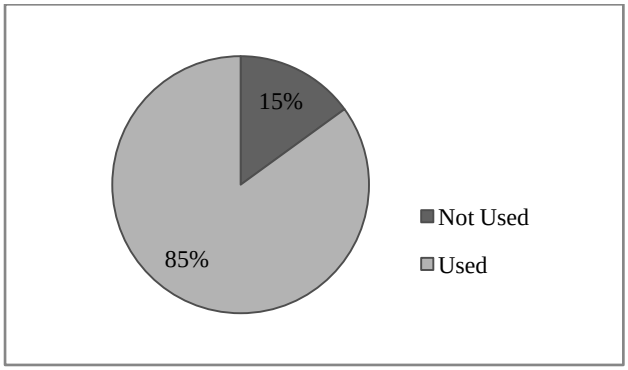

Fig. 5: Words generated by the algorithm usage

- Generated forms are applying one level of affixes modulation in which either prefixes or suffixes are added, where the resulted forms are not limited to that.

7. FUTURE WORK

In the future, this method can be enhanced further more to be even more accurate and amusing. Different sets of patterns could have defined affixes to get the exact expected words. Accordingly, the databases of patterns and affixes would be reformed or even follow more precise classifications. More precisely, affixes should be cataloged in such a way that the morph-tactic problem can be avoided. Different types of prefixes do not suit some suffixes to be attached to the same word.

Besides, other categories of words could be studied too to be involved in the generating procedure developed here, where the special types were eliminated in the beginning of the method could have their own process as well.

Moreover, the input of the system can be more than a word and the obtained results would be stored distinctly and subsequently revealed in a sequence.

Likewise, the system output can be improved by operating the method in an iterative way, where the generated words are submitted again to be considered as a new input to the Generate method. This attitude will provide an advanced level of the words generated.

Another practice can have a noticeable affect on the output utility for the user of the system is declaring the modifications have been made at each word generated.

Finally, integrating this system within larger systems used for different applications such Information Retrieval and Extraction, Machine Translation, Data Mining, ..., etc. Doing so will clarify additional features to be developed and combined with the system.

8. CONCLUSION

In this paper a method that provides nearly all the words can be formed out of any entered word was developed. First, the word will be submitted and its features will be specified after analyzing that word. Accordingly, new inflected words are produced using the method developed and demonstrated through this paper based on the suitable affixes to the word obtained from the user. Additionally, this paper answers the question that is 'whether or not inflected and derived words can be equally produced using the same methodology?', it was proven that another model based on the Root and Patterns is required rather the one developed based on the Stem and Affixes to achieve that. Lastly, many ideas of developing the algorithm presented above has bee discussed.

9. ACKNOWLEDGMENT

We would like to express our gratitude to our supervisor Dr. Mohammad Dahab for the useful comments, remarks and engagement through the process of preparing this paper. We also would like to thank every contribution was towards development of this paper.

10. REFERENCES

[1] K. Shaalan. Rule-based Approach in Arabic Natural Language Processing. International Journal on Information and Communication Technologies, Vol. 2, No. 3, June 2010.

[2] M. Gridach, N. Chenfour, Developing a New System for Arabic Morphological Analysis and Generation, Mathematics and Computer Science, Department Faculty of Science Dhar, El Mehraz Fez.

[3] M.G. khayat, A. Al-othman and S. Al-safran, An Arabic Morphological Analyzer/Synthesizer, Department of Electrical & Computer Engineering, KAAU, Jeddah and KFUPM, Dhahran, Saudi Arabia, JKAU: Eng. Sci., vol. 13 no. 1, pp. 71-93,1421 A.H. / 2001 A.D.

[4] Tengku Mohd T. Sembok, Belal Mustafa Abu Ata and Zainab Abu Bakar, A Rule and Template Based Stemming Algorithm for Arabic Language, International Journal Of Mathematical Models And Methods In Applied Sciences, Issue 5, Volume 5, 2011.

[5] Khaled Shaalan and others, Arabic Word Generation and Modeling for Spell Checking, Institute of Formal and Applied Linguistics and others, Charles University in Prague, Czech Republic, School of Computing, Dublin City University, Ireland.

[6] Karin C. Ryding, A Reference Grammar of Modern Standard Arabic, Cambridge University Press, August 2005.

[7] Ritchey, T. General morphological analysis. 16th EURO Conference on Operational Analysis. 1998.

[8] Nizar Habash, Abdelhadi Soudi and Tim Buckwalter , On Arabic Transliteration , In Arabic Computational Morphology: Knowledge-based and Empirical Methods . Soudi, Abdelhadi; van den Bosch , Antal; Neumann, Günter (Eds.), 2007.

[9] Maha Althobaiti, Udo Kruschwitz, Massimo Poesio , AraNLP: A Java-based Library for the Processing of Arabic Text, School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK.

[10] Farghaly, K. Shaalan. Arabic Natural Language Processing: Challenges and Solutions, ACM Transactions on Asian Language Information Processing (TALIP), the Association for Computing Machinery (ACM). TALIP Vol 8, Issue 4, December 2009.

[11] Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane M. Ould Abdallahi Ould Bebah and M. Shoul., 2011: Alkhalil Morpho Sys 1: A Morphosyntactic analysis system for Arabic text, University Mohamed I, Oujda, Morocco.

[12] Kroeger, P., Analyzing Grammar: An Introduction, Cambridge University Press, 2005.

[13] Moulana Ebrahim Muhammad, From the Treasures of Arabic, Academy for Islamic Research, Safar 1427 A.H. March 2006.

[14] arabicCorpus, http://arabicorpus.byu.edu/search.php

11. APPENDIX

11.1 Common Affixes

11.1.1 Prefixes:

و	ف	أ	أو	أف	ل	ول	فل	

11.1.2 Suffixes:

ك	كما	كم	كن	ه	ها	هما	هم	ەن	

11.2 Noun Affixes

11.2.1 Prefixes

ال	وال	فال	أ ال	أو ال	أفال	لل
ولل	ب	فب	أب	لب	ل	ول
أل	ك	وك	فك	أك	لك	و
فو	بال	وبال	فبال	أبال	لبال	أو بال
أفبال	لل	ولل	فلل	أو لل	أفلل	كال
وكال	فكال	أكال	لكال	أو كال	أفكال	

11.2.2 Suffixes

يه	يها	يهما	ي	وا	ا	ان
ون	ين	ات				

11.3 Verb Affixes

11.3.1 Prefixes

ت	ي	س	وس	فس	أس	أو س
أفس	ل	ول	فل	ل	ول	فل
وهن	ني	نيه	نيها	نيهما	نيهم	نيهن

11.3.2 Suffixes

ناه	ناها	ناهما	ناهم	ناهن	كه
كها	كهما	كهم	كهن	كماه	كماها
كماهما	كماهن	كموه	كموها	كموهما	كموهم
كموهن	كنه	كنهما	كنهم	كنهن	وني
ونيه	ونيها	ونيهما	ونيهم	ونيهن	ونا
وناها	وناهما	وناهم	وناهن	وه	وها
وهما	وهم				