A Rule and Template Based Stemming Algorithm for Arabic Language

Tengku Mohd T. Sembok, Belal Mustafa Abu Ata and Zainab Abu Bakar

***Abstract*—**Stemming is defined as the conflation of all variations of specific words to a single form called the root or stem. Stemming plays a vital role in natural language processing and understanding. As in other languages, there is a need for an effective stemming algorithm for Arabic words. Arabic is a language having a rich and complex morphological word structures and rules. An Arabic stemming algorithm based on morphological rules has been developed, and to enhance its effectiveness, a dictionary of root words is used to determine the right stems. The Arabic stemming algorithm developed by Al-Omari is studied and a new algorithm is proposed to enhance the performance. The improvements obtained relate to the order in which the dictionary is looked-up and the order in which the morphological rules are applied.

***Keywords*—**Stemming, indexing, information retrieval, natural language processing.

I. INTRODUCTION

One of the main modules of a document retrieval system is the text processing and indexing of the input documents to obtain the representation of the documents in the form of indexes. These indexes will be the surrogates to the documents and facilitate the process of retrieving relevant documents with respect to the given query. The process of selecting the representation or index terms constitutes a major operation and technique applied in information retrieval systems. Word stemming is one technique normally applied in the indexing process because it helps in reducing the size of the index terms and also proved to help in improving the degree of relevancy in retrieving documents. The stemming process constitutes word morphological analysis based on the language used in order to get the words' stems to represent the documents as well as to function as indexes to the documents for efficient and effective retrieval.

Stemming is defined as the conflation of all variations of specific words to a single form called the root or stem. Stemming algorithms for some languages have been published and applied in building of information retrieval systems, among which for English is the well known Porter's algorithm [1], for French we have Savoy's algorithm [2], and for the Malay language we have Fatimah's et al. [3]. Stemming techniques play a vital part in the development of a good document retrieval system. The stemming process will reduce the size of the documents representations by 20-50% compared to full words representations, according to van Rijsbergen [4]. Furthermore, the relevancy of the retrieved documents will be improved and their number will also be increased.

Stemming algorithms for the Arabic language are not widely available and published in journals. The current algorithms reported are either general in nature, or lack in the morphological aspect of getting to the correct Arabic stems. Pioneer works on Arabic stemming have been published by researchers such as Gheith & El-Sadany[5], El-Sadany & Hashish[6], Saliba & Al-Dannan [7], Hilal [8], AlKharashi & Even [9], and Al-Omari [10].

II. ARABIC LANGUAGE STEMMING ALGORITHM

In the previous sections, we mentioned English, Malay and French language stemmers. Approaches adopted by these stemmers are not fully appropriate for the development of Arabic stemmers due to differences in the morphological structures peculiar to each of the languages as well as their semantic differences. The main differences as put forward by El-Sadany & Hashish [6] are as follows:

i. Arabic is one of Semitic languages which differ in structure of affixes from Indo-European type of languages such as English and French;

ii. Arabic is mainly roots and templates dependent in the formation of words;

iii. Arabic roots consonants might be changed or deleted during the morphological process;

Stemmers such as Porter's are developed mainly to improve the retrieval performance of document retrieval systems. As a result, these stemmers do not make use of dictionary that checks for the correctness of the resulted stems. Whereas, for languages such as Malay, French and Arabic, it will be somehow impossible to develop a stemming algorithm that does not make use of such dictionaries for stems and phrases checking. More precisely, if such stemmers are developed, their accuracy and performance will be low [9].

Manuscript received April 9, 2011: Revised version received May 28, 2011..

Tengku M. T. Sembok is with the National University of Malaysia, Bangi, Selangor, Malaysia (phone: +60123373539; fax: +60389256732; e-mail: tmts@ftsm.ukm.my).

Zainab A. Bakar, is with Universiti Teknologi MARA, Shah Alam, Malaysia.. (e-mail: zainabab@uitm.edu.my).

Belal Abu Ata is with Bahrain University, Bahrain (e-mail: belal@yahoo. com).

Approaches for the development of Arabic stemmers are restricted due to its complicated structure. These approaches are mainly dependent on the understanding of the Arabic morphology. Hence, Arabic stemming is actually a process of morphological analysis applied for the word in order to extract the correct stem. The stemming approach adopted by most of the previous Arab researchers for the development of morphological analysers is mainly an iterative of the following processes:

i. Analysing the prefixes
ii. Analysing the postfixes
iii. Analysing the stem
iv. Recoding the root
v. Lexicon verification

This approach was used by several researchers such as El-Sadany & Hashish[6], and Hilal [8], Shahein & Youssef [11] and Al=Omari [10].

Al-Omari Morphological Analyser

This morphological analyser is implemented at the Department of Computer Science, Universiti Kebangsaan Malaysia, as part of a multilingual document retrieval system (Al-Omari, 1994). This analyser uses a set of rules that must be loaded before the actual analysis begins. The analyser consists of three main modules: the *article analyser module*, the *word analyser module*, and the *system knowledge base module*.

The article analyser module is where the input word is checked whether it is an article or not, similar to Hilal's algorithm. This module contains an article's knowledge base that contains rules for the purpose of determining valid articles and non-valid ones. There are around 175 articles defined in this analyser. The second module is the word analyser module. The word analyser will try to generate the roots.

The word analyser consists of three sub-modules: the prefix analyzer, the postfix analyser, and the root extractor. The prefix analyser will try to find all the possible valid prefixes that are attached to the input word (which is not an article). Thus, this sub-module will produce a list of possible prefixes that are attached at the beginning of the word.

The postfix analyser will try to find all the possible valid postfixes that are attached to the input word (which is not an article). Thus, this sub-module will produce a list of possible postfixes that are attached at the end of the word.

The root extractor module will try to find all the possible valid root(s) that can be extracted from the word after removing the prefixes and the postfixes. This module consists of five sub-modules as shown. One of these sub-modules is the stem generator which will generate all valid stems from the triple (prefix, postfix, the stem). This process will be performed for all combinations of the prefix list obtained from the prefix analyser and the postfix list obtained from the postfix analyser. The output of this formation will be a set of possible stems.

Another sub-module of the root extractor is the root generator. This sub-module will find the possible roots from the generated stems. All the information needed to process the various modules and sub-modules of this morphological analyser are stored in the knowledge base in sets of rules. There is a set of rules that concerns with the prefixes and their application to words, another set is used for the postfixes and their application to words, and other sets of rules for the various sub-modules. Lists of valid Arabic roots and templates are also found in the knowledge base [10].

This morphological analyser does perform quite well but with few limitations. The system refuses to stem words which do not have any match in the knowledge base. The number of Arabic articles, templates and roots catered by the system is relatively few. Based on these limitations, an improved stemming algorithm has been developed. The development of this new algorithm is described in the following sections.

III. ARABIC WORD FORMATION

The grammatical system of the Arabic language is based on a root-and-pattern structure and considered as a root-based language with not more than 10,000 roots [12]. A Root in Arabic is the base verb form which can be trilateral, which is the overwhelming majority of Arabic words, and to a lesser extent, quadrilateral, pentaliteral, or hexaliteral, each of which generates additional verb forms and noun forms by the addition of derivational affixes [13].

A stem is a combination of a root and derivational morphemes to which an affix (or more) can be added [14]. However, when applying this definition to Arabic, the verb roots and their verb and noun derivatives are considered as stems. *Affixes* in Arabic are: prefixes, suffixes and infixes (see Appendix-B for examples). Prefixes are attached at beginning of the words, where suffixes are attached at the end, and infixes are found in the middle of the words. For example, the Arabic word **الطالبات** (*altalibat*) which means "*female students*", consists of the elements as shown in Table 1**:**

Table 1: Example of Arabic Affixes

Word	prefix	suffix	infix	root
الطالبات	*ال*	*ات*	*ا*	*طلب*

There are 8 prefixes in Arabic language which form a small set of prefixes compared to languages such as Malay and Slovene [3][15]. However, Arabic allows up to three concurrent prefixes to be added to the same word [16][17], such as the word وبالوالدين which contains three prefixes (و , ب, ال). Table 2 contains all the 8 Arabic prefixes and their meanings.

Table 2: Arabic Prefixes and Their Meanings

Prefix	Meaning	Example
ب	with, in, by	بالسيارة
ك	same as,	كالدخان
س	will,	سأذهب
و	and	ورجالهم
ال	the	النساء
أ	questioning	أأكلت
ف	then	فذهبوا
ل	to, because	لتنام

Arabic prefixes do not follow a systematic standard for their attachment to Arabic words. In order to find all the rules that cover prefixes, rules of their combination and letters of words they are allowed to precede, a thorough and extensive study and analysis of the Arabic words and roots are needed.

IV. ARABIC SUFFIXES

There are fifteen(15) suffixes in Arabic language which form a small set of suffixes compared to languages such as Malay and the Slovene. However, Arabic allows up to three concurrent suffixes to be attached at the end of the same word, for example, the word ضربناهم contains three prefixes (هم , ا, ن). Arabic suffixes are mostly made of attachable pronouns. Table 3 contains all the 15 Arabic suffixes and their English meanings [16][17].

Table 3: Arabic Suffixes and Derivative Meanings

Suffix	Derivative Meaning	Example
ين	singular female	تلعبين
ان	male dual	يلعبان
و	plural male,	ينمو
ه	missing singular	ضربته
ك	addresser singular	ضربك
ا	male dual	أكلا
ي	singular female	أكلتي
ن	plural	أكلن
ت	singular female	أكلت
ات	female plural	لاعبات
ون	absent male plural	يلعبون
وا	absent male plural	أكلوا
تم	addresser male	أكلتم
هم	absent male plural	ضربهم
كم	addresser male	ضربكم

The characteristic of Arabic suffixes is similar to that of prefixes, which does not have a systematic rule for their attachment to Arabic words.

V. RULE-BASED STEMMING ALGORITHM

There are a few approaches taken in the development Arabic stemming algorithms. Among them are neural network approach [18][19], Support Vector Machine (SVM) application[20], and the rule based approach [21]-[24]. The first two approaches use machine learning technique to run the algorithms, and the later approach use the morphological rules created by the Arabic language experts to do the stemming. In this paper, we are adopting the rule based approach.

The stemming algorithm was implemented using Standard C language with Arabic language support. The stemming algorithm consists of the following main modules:

- Prefix and suffix removal module;
- Root generator and checking module;
- Pattern generator and checking module;
- Intensification module which handle double letters (تشديد);
- Recoding module.

A. Prefix and Suffix Removal Module

This module will try to find all the valid affixes in a given Arabic word and remove such affixes. Arabic language contains just a few number of affixes, however, affixes attachment rules to words are not easy to list out. After some thorough study of the Arabic morphological structure and word formation, we came out with around 800 rules that cover both Arabic prefixes and suffixes attachment rules to words. The prefix and suffix rules are define accordaing to the following syntax:

1. Prefix rules: **prefix** + ***let(s)***
where *let(s)* is a set of valid letters to follow the prefix,
example: أ + تأ
which means أ *is considered a prefix if the next two letters are* تأ *such as in the word* أتأتي

2. Suffix rules: ***let(s)*** + **Suffix**
where let(s) is a set of valid letters preceding the suffix,
example: بب + ت
which means ت *is considered a suffix if the previous 2 letters are* بب *such as in the word* أحببت

Table 4 shows example of prefixes and suffixes in the given words.

Table 4: Examples of Word Letters that match the Arabic Affixes

Word	**Letter(s)**	**Type of Affix**
فارس	ف	*Prefix*
لاعبون	ل	*Prefix*
بارد	ب	*Prefix*
بنات	ت	*Suffix*
متم	تم	*Suffix*
القرون	ون	*Suffix*

B. Root Generator and Checking Module

This module will try to find all the valid possible roots for a given word. The module will check for the root validity by using the hashing technique to search for it in the root dictionary. This module will invoke the *Intensification submodule* that checks for words of double letters in order to change it to the normal form.

C. Pattern Generator and Checking Module

This module will take the word to be stemmed and one possible root (generated by the root module) as an input and then derive a template that matches both of them. This process will be repeated for the entire possible root generated from the root generator module. The module will also check the resulted template for its correctness by matching it to a set of valid Arabic templates [see Appendix-A]. An example of this validation process is as follows:

- for the word فاسقين, some of the possible roots generated are فسق, فاس, سقي, قين, فاق , where the roots قين, فاق are ignored as they are not valid Arabic roots.
- the templates for the remaining 3 roots are constructed with reference to the word فاسقين, the resulting templates and their validity in Arabic are shown in Table 5.

Table 5: Possible Templates for the Word فاسقين

Root	**Generated Template**	**Template Validity**
فسق	فاعلين	*valid*
فاس	فعلقين	*invalid*
سقي	فافعلن	*invalid*

D. Handling Double Letter Module

There are many Arabic words and roots with *double* letters, which means that two similar adjacent letters are combined into one letter. This module will check for such words and its root and reconstruct the word by adding that letter. This will help in obtaining the correct root. Examples of words with *Intensification* are shown in Table 6.

Table 6: Examples of Arabic words with *Intensification*

Root	**Generated Template**	**Template Validity**
فسق	فاعلين	*valid*
فاس	فعلقين	*invalid*
سقي	فافعلن	*invalid*

E. Recoding Module

The *recoding module* main concern is to change some of the letters to their correct form. These letter changes will most probably occurs during the process of template formation in Arabic when a word is formed from a root. Some letter may dropped, changed or replaced by other letters. Table 7 lists some of the most recoded Arabic letters.

F. Stemming Algorithm Flowchart

The flow chart of the stemming algorithm is shown in Figure 1. The stemming process begins by processing a word and trying to find its correct stem. In case the word does have a correct stem, then the word without its affixes will be returned.

Table 7: Examples of Letter Recoding for Arabic Words

Word	**Recoding Rule (from→ to)**	**Word after Recoding**
هزئ	ئ → ؤ	هزؤ
	ئ → أ	هزأ
نبئ	ئ → أ	نبأ
خطئ	ئ → أ	خطأ
خسئ	ئ → أ	خسأ
صبئ	ئ → أ	صبأ
سيئ	ئ → أ	سيأ
نبء	ء → أ	نبأ
دنى	ى → ا	دنا
تؤمن	ؤ → أ	تأمن
يؤمر	ؤ → أ	يأمر
يؤخذ	ؤ → أ	يأخذ
ؤمر	ؤ → أ	أمر
راد	ا → و	رود
حيا	ا → ي	حيي

The stemming algorithm will take as input an Arabic word (not a stop word), and the output will be the extracted root (or stem). In cases where the algorithm cannot find a root for the a specific word, the word itself will be taken as a root. Such cases are few based on the performance of the algorithm

Table 8 shows the number of errors obtained by our stemming algorithm compared to the results obtained by Al-Omari's algorithm. Hence, we can conclude that the our algorithm does performance better than Al-Omari's algorithm.

Table 9 shows all the 21 words that has been stemmed wrongly and the types of errors for each word.

Figure 1: The steps to find the stem for the word: فسيأكلون:

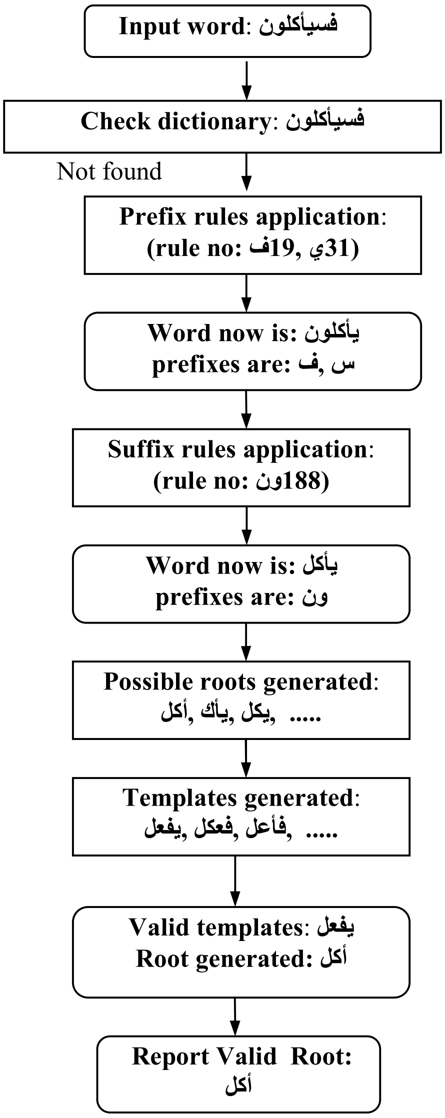
Input word: فسيأكلون
Check dictionary: فسيأكلون
Not found
Prefix rules application: (rule no: ف19, ي31)
Word now is: يأكلون
prefixes are: ف, س
Suffix rules application: (rule no: ون188)
Word now is: يأكل
prefixes are: ون
Possible roots generated: أكل, يأك, يكل, ...
Templates generated: فأعل, فعكل, يفعل, ...
Valid templates: يفعل
Root generated: أكل
Report Valid Root: أكل

Table 8: Results of the Experiments for 10 Chapters of the Quran

Number of words wrongly stemmed			
		Ours	Al-Omari's
Chapter 1	684+	9	25
	126~	6	15
Chapter 2	450+	8	23
	83~	6	17
Chapter 3	444+	6	24
	100~	4	20
Chapter 4	462+	9	27
	103~	5	19
Chapter 5	202+	6	15
	56~	3	12
Chapter 6	278+	5	19
	48~	2	17
Chapter 7	341+	9	29
	73~	4	19
Chapter 8	299+	5	12
	73~	4	10
Chapter 9	341+	4	14
	88~	3	9
Chapter 10	181+	4	7
	46~	2	4
	3682+	**65**	**195**
Totals	**796~**	**39**	**142**
	330*	**21**	**85**

Keys:
+ Total number of all words in the chapter
~ Total number of unique words in the chapter
*Total number of unique words in all the chapters

Table 9: Stemming Errors on Ten Chapters of the Quran

Word	Actual Root	Resulting Root	Error Type
ربه	ربب	ربه	unchanged
موته	موت	موة	spelling
الظا	ظنن	ظان	spelling
الريا	ريح	راح	spelling
بالبا	بطل	اطل	spelling
وبار	برك	ارك	spelling
فويل	ويل	يل	overstemming
الفلك	فلك	لك	overstemming
ليبلو	بلو	بل	overstemming
بآل	بآل	آل	overstemming
بأس	بأس	أس	overstemming
بوالد	ولد	ديه	others
	وقى	قا	others
تنزي	نزل	زيل	others
كرها	كره	رها	others
والف	فسق	سوق	others
المبي	بين	مبي	others
فتبيذ	بين	تبي	others
مبين	بين	مبي	others
بالهم	بآل	هم	others
فأتنا	أتي	تنا	others

Table 10: Distribution of Unique Errors on Quranic Data Set

Error Type	Number (%)
Overstemming	5 (23.8%)
Understemming	0 (0 %)

Unchanged	1 (4.7%)
Spelling	5 (23.8%)
Others	10(47.6%)

There are a total of 21 unique errors as shown in Table 10. These errors are classified into 5 groups, namely, *understemming, over-stemming, spelling, unchanged*, and *others*. The names of the groups describe the type of errors. Understemming and overstemming indicate that the resultant stems are uder stemmed or over stemmed. The group *spelling* indicates there is one letter in the resulted stem that is different from the correct root. As for unchanged group indicates that the resultant stem is the same as the original word which is not the correct root. Others indicate other types of stemming errors.

VI. CONCLUSION

Our experiments have shown that our new stemming algorithm performs better than that of Al-Omari. Could it be improved further? Our analysis suggests that most of the remaining errors are due to the precise order in which the rules are applied, and we are currently considering ways in which this ordering can be best applied.

References

[1] Porter, M. F. (1980). An algorithm for suffix stripping’. *Program, 14*, 130-137.

[2] Savoy, J. (1993). ‘Stemming of French words based on grammatical categories. *Journal of the American Society for Information Science,44*,1-9.

[3] Ahmad, F., Mohammed Yusoff, Sembok, T.M.T. 1996. Experiments with A Malay Stemming Algorithm, *Journal of American Society of Information Science*.

[4] van Rijsbergen, C. J. (1979). *Information Retrieval*. London: Butterworths.

[5] Gheith, M.& El-Sadany, T. 1987. Arabic Morphological Analyzer on A Personal Computer. *Arabic Morphology Workshop*. Stanford University.

[6] El-Sadany, T.A. & Hashish, M.A. 1988. Semi-Automatic Vowelization of Arabic Verbs. *Proceedings of 10th National Computer Conference:* 45-56.

[7] Saliba, B., & Al-Dannan, A. 1990. Automatic Morphological Analysis of Arabic: A study of Content Word Analysis. *Proceedings of the First Kuwait Computer Conference*: 231-243.

[8] Hilal, Y. 1990. Automatic Processing of Arabic Language and Applications. *Proceedings of the Arabic Language Processing Using Computer Conference*: 213-219.

[9] Al-Kharashi, I.A. & Evens, M.W. 1994. Comparing words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System. *Journal of the American Society for Information Science*. **45**(8): 548-560.

[10] Al-Omari, H. 1994. *ALMAS: An Arabic Language Morphological Analyzer System*. National University of Malaysia. Bangi, Selangor.

[11] Shahein, H. I. & Youssef, S.A. 1990. A Model for Morphology As A Production System. *Proceedings of the Second Cambridge Conference on Bilingual Computing in Arabic and English*.

[12] Ali, A.Y. 1983. *The Holy Quran: Text, Translation and Commentary*. Maryland: Amana Corp.

[13] Saliba, B., & Al-Dannan, A. 1990. Automatic Morphological Analysis of Arabic: A study of Content Word Analysis. *Proceedings of the First Kuwait Computer Conference*: 231-243.

[14] Gleason, H.A. 1970. *An Introduction to Descriptive Linguistics*. 3rd Edition. New York: Holt. Rinehart and Winston.

[15] Popovic. M. 1991. *Implementations of A Slovene Language-Based Free-Text Retrieval System*. University of Sheffield. UK.

[16] Al-Fedaghi, S.S., & Al-Sadoun, H.B. 1990. Morphological Compression of Arabic Text. *Information Processing & Management*. **26**(2): 303-316.

[17] Dahdah, A. 1985. *Arabic Language Grammar Dictionary.* “ معجمقواعد اللغة العربية”.Beirut: Lebanon. Lebanon Library.

[18] Manal El-Obaid. 2006. Arabic Phoneme Recognition Using Neural Networks. *Proceedings of the 5th WSEAS International Conference on Signal Processing.*

[19] Alserhan, H.M. & Ayesh, A.S. 2006. An Application of Neural Network for Extracting Arabic Word Roots. *Proceedings of the 10th WSEAS International Conference on COMPUTERS.*

[20] Mesleh, A. 2007. Support Vector Machines Based Arabic Language Text Classification System: Feature Selection Comparative Study, *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*.

[21] Shquier, Mohammed M. Abu; Sembok, T.M.T. 2008. Word Agreement and Ordering in English-Arabic Machine Translation. *Proceedings of ITSim 2008: International Symposium on Volume 1, 26-28.*

[22] Awajan A. 2003. A Rule-Based Morphological Analyzer of Arabic Word. WSEAS *Transactions on Computers. No. 2. Volume 2.*

[23] Sameh Ghwanmeh. 2011. Effect of Excessive Letter Location in Arabic Lexical Items: A Stemmer Algorithm Approach. WSEAS Transaction.

Appendix-A :Arabic Word Templates

ع
أع
تع
عل
فع
فل
نع
يع
أعل
أفع
أفع
أفل
إتع
إفع
اتع
اعل
افع
افل
تعل
تفع
تفل
عال
علة
فاع
فال
فعة
فعل
فلة
معل
مفع
مفل
نتع
نعل
نفع
نفل
يتع
يعل
يفع
يفل
أتعل
أتفل
أعطل
أفطع
أفطل
أفعل
ؤفعل
إتعل

اتعل
اعتل
اعطل
افتع
افتل
افطع
افطل
افعل
انعل
انفع
انفل
تتفع
تعال
تعتل
تفاع
تفال
تفتع
تفتل
تفعل
عالة
علاء
علائ
علاة
عوال
فائل
فاعة
فاعل
فالة
فعاء
فعائ
فعاة
فعال
فعلة
فعلى
فعول
فعيل
فلاء
فلائ
فلاة
فواع
فوال
فوعل
فيعل
متعل
متفل
معال
معتل

معطل
مفاع
مفال
مفتع
مفتل
مفطع
مفطل
مفعل
منعل
منفع
منفل
نتعل
نتفل
نعال
نعتل
نعطل
نفاع
نفال
نفتع
نفتل
نفطع
نفطل
نفعل
ننعل
ننفع
ننفل
يتعل
يتفع
يتفل
يعال
يعتل
يفاع
يفال
يفتع
يفتل
يفعل
ينعل
ينفع
ينفل
أستعل
أستفل
أعلاؤ
أفاعل
أفتعل
أفطعل
أفعاؤ
أفعال

أفعلة
أفعول
أفعيل
أفلاؤ
إتعال
إتفاع
إتفال
إستعل
إستفع
إستفل
إعلال
إفتعل
إفعاع
إفعال
إفعلة
إفعول
إفعيل
إفلال
إنفعل
اتعال
استعل
استفع
استفل
افاعل
افتعل
افطعل
افعال
افعلة
افعول
افلال
انفعل
تفاعل
تفتعل
تفعلة
تفعيل
تنفعل
فائلة
فاعال
فاعلة
فاعول
فعائل
فعالة
فعالل
فعالى
فعالي
فعاول
فعايل

فعلاء
فعلاؤ
فعلائ
فعلاة
فعلال
فعلان
فعللة
فعلية
فعليل
فعولة
فعيعل
فعيلة
فواعل
فياعل
فيعال
متفعل
مستعل
مستفع
مستفل
مفاعل
مفتعل
مفعال
مفعلة
مفعول
مفيعل
منفعل
نتعال
نتفاع
نتفال
نتفعل
نستعل
نستفع
نستفل
نعلال
نفاعل
نفتعل
نفطعل
نفعاع
نفعول
نفعيل
نفلال
نفوعل
نفيعل
ننفعل
يتعال
يتفاع
يتفال

يتفعل
يستعل
يستفع
يستفل
يعلال
يفاعل
يفتعل
يفعاع
يفعلن
يفعول
يفعيل
يفلال
يفوعل
يفيعل
ينفعل
أستفعل
أفاعيل
أفعلاء
أفعلاؤ
أفيعال
إستفعل
إفتعال
إفعالة
إفعلال
إفعنلل
إفعوعل
إنفعال
استفعل
افاعيل
افتعال
افعالة
افعلاء
افعلال
افعنلل
افعوعل
افيعال
انفعال
تتفعل
تستفعل
تفاعلة
تفعيلة
فاعلية
فعاعيل
فعالات
فعالية
فعاليل
فعلياء

فعيعيل
فعيلان
فواعيل
متفاعل
مستفعل
مفاعلة
مفاعيل
مفتعلة
مفعوعل
مفعولة
منفعلة
نتفاعل
نستفعل
نفاعيل
نفعنلل
نفعوعل
يتفاعل
يستفعل
يفاعيل
يفعلون
يفعنلل
يفعوعل
إستفعال
إستفعلة
إفتعالة
إفعاليل
إفعلالة
إفعنلال
إفعوعال
إفعيلال
إنفعالة
استفعال
افتعالة
افعلالة
افعنلال
افعوعال
افعيلال
انفعالة
تفاعيلن
تفعيلية
متفاعلة
مستفعلة
مفعوعلة
إستفعالة
إفعنلالة
إفعيلالة
استفعالة

استفعلة
افعنلالة
افعوعالة
افعيلالة

Appendix-B: Examples of Prefix and Suffix Rules

Prefix Rules for أ

No.	Rule	No.	Rule	No.	Rule
1	إ +أ	2	آ +أ	3	أ +أ
4	بال +أ	5	تت +أ	6	تج +أ
7	تأ +أ	8	تد +أ	9	تذ +أ
10	تح +أ	11	تص +أ	12	تع +أ
13	تس +أ	14	ته +أ	15	تخش +أ
16	تن +أ	17	تقت +أ	18	تقو +أ
19	تري+أ	20	دع +أ	21	صا +أ
22	تمد +أ	23	صح +أ	24	صد +أ
25	صب +أ	26	صغ +أ	27	صن +أ
28	صط +أ	29	صي +أ	30	ذك +أ
31	صو +أ	32	رأ +أ	33	فأ +أ
34	ذه +أ	35	فب +أ	36	فح +أ
37	فإ +أ	38	فغ +أ	39	فت +أ
40	فع +أ	41	فتأ +أ	42	فتت +أ
43	فات +أ	44	فتع +أ	45	فتم +أ
46	فتط +أ	47	فرأ +أ	48	فنج +أ
49	فته +أ	50	نط +أ	51	نل +أ
52	فنض +أ	53	وعج +أ	52	يأ +أ
55	نؤ +أ	56	يح +أ	57	يمس +أ
58	يب +أ				

Suffix Rules for و

No.	Rule	No.	Rule	No.	Rule
1	ئ + و	2	ء + و	3	ب + و
4	ت + و	5	ج + و	6	ر + و
7	ص + و	8	ع + و	9	ف + و
10	ق + و	11	ل + و	12	م + و
13	ن + و	14	بج + و	15	ا ـ د+ و