ARABIC ROOT BASED STEMMER

Mohammed Naji AL-Kabi
Computer Information Systems Department
Yarmouk University
Irbid, Jordan
mohammednaji@yahoo.com

Ronza S. Al- Mustafa
Computer Information Systems Department
Yarmouk University
Irbid, Jordan
ronza_malkawi@yahoo.com

ABSTRACT

This paper presents a new (root-based) stemming algorithm for Arabic language. As other natural languages not all the words used in Arabic language has roots, some of these are borrowed from other languages, e.g. as the word "تلفزيون" television, so in this case the stemmer will fail to get the right root because these foreign words have no root. This algorithm is based on affix removal beside a knowledge from structural linguistics. The implementation and evaluation of this algorithm shows a noticeable improvement in the accuracy relative to previous algorithms.

Keywords: *Arabic, Stemming, Root, negative suffix, negative prefix, Light Stemming, NLP.*

1. INTRODUCTION

The Arabic language is the fifth most widely spoken language in the world. It belongs to the Semitic family; so it differs from the Indo-European languages morphologically, semantically, and syntactically. The Arabic alphabet contains twenty-eight letters, always written from right to left in cursive form. Diacritical marks (harakat) (tashkiil تشكيل) appear either above or below the letters, and play an essential role in many cases in distinguishing semantically and phonetically between two identical words with the same characters, but with different diacritics. Diacritical marks are used in holy books, poems, and children's literature; newspapers, journals and other books for adults are usually printed without diacritics, which means that many strings are ambiguous. Most native Arabic words are derived from verbal roots. Arabized words, on the other hand, mainly nouns borrowed from other languages with a slight phonetic adjustment to suit the Arabic pronunciation, have no roots[8].

All Arabic words belong to three main categories: noun, verb or particle. Around 64% of Arabic words are derived from triliteral verbs (three consonants), but there are also biliteral verbs (two consonants), quadriliteral verbs (four consonants), and pentaliteral verbs (five consonants). Naturally these verbs represent the roots for which stemming algorithms typically search. This stemming process excludes words derived from nouns and particles[9].

A morpheme is the smallest meaningful lingual unit which has a semantic interpretation in the grammar of a language. There is a difference between stem and a root, a stem is a morpheme or a set of concatenated morphemes that can accept an affix, where a root is a single morpheme that provides the basic meaning of a word.

Stemming might be useful to Information retrieval systems, text classification systems, text clustering systems, dictionary automation, text compression, … etc.

Stemming is considered by a number of authors as word Standardization [12]. A number of writers thought that stemming is useful for improving retrieval performance because it reduces variants of the same root word to common concept, beside reducing the size of the indexing structure because the number of distinct index terms is reduced [3]. Other writers are not satisfied with the concept of using stemming in IR and Text mining [3]. Accordingly many search engines do not adopt stemming [3]. Several common types of stemming strategies are discussed by Frakes: affix removal, table lookup, successor variety, and *n*-grams [7]. Affix removal strategy tries to eliminate the prefixes and suffixes. The most important part in this strategy is suffix removal, since most variants of terms are generated by suffixes.

In Arabic language as with other natural languages the stemmer may face the problem of a *negative prefix*, where the prefix which eliminated is part of the word and not really a prefix. If a stemmer tries to strip the "الـ" which is a well known prefix from the following examples, the output will be definitely wrong, e.g. "الله" *Allah*, "ألمانيا" *Germany*, "ألوية" *Brigades*, "ألبانيا" *Albania*, …etc. It also includes other prefix such as "و" *And* which represents a frequently used conjunction, e.g., stripping off "و" *And* from "وفاء" *honesty* leads to a wrong stem.

The negative prefix problem in Arabic language stemmer is not restricted to the "الـ" and "و" prefixes, but it also includes other prefixes such as "كالـ","والـ", "للـ", "فالـ", …etc. The Arabic light stemming in this case for the term "والي" *Governor* will be wrong, if the prefix "والـ" strip off from the term. Similarly the stems of the words "كالح" *glum*, "الله" *Allah*, "فالح" *successful*, if we strip from them the prefixes "كالـ", "للـ", "فالـ" respectively. Similarly Arabic stemmers face another problem of a *negative suffix*, where the suffix in natural languages face which eliminated is part of the word and not really a suffix. If a stemmer

try to strip off the "ان" which is a well known suffix from the following examples, the output will be definitely wrong, e.g. "لعمان" *To Amman*, "اليابان" *Japan*, …etc. Table 5 in the Appendix illustrate a number of examples.

Table lookup is the simplest strategy among the four, it simply looks for the root of the term in the lookup table. The performance of this strategy is highly affected by the number of words (terms) and their root in the table, as the lookup tables gets larger the performance get higher too. Large lookup tables might need a considerable storage space. Successor variety is not straightforward as the others, and depends on algorithms which is based on structural linguistics and attempts to determine morpheme boundaries. N-grams stemming searches for digrams, trigrams or more term successive letters. This strategy is a term clustering procedure not a stemming procedure.

The above two problems (negative prefix & negative suffix) of Arabic stemmers leads to a wrong grammatical root, so the accuracy of IR & Text mining systems which rely on these stemmers will be deteriorated.

The two main problems of stemming have been described by Chris D. Paice [12]. In the first place, pairs of etymologically related words sometimes differ sharply in meaning [12] for example, consider "سل" *ask*, "سلب" *stole*, and "سلام" *Peace*. In the second place, the transformations involved in adding and removing suffixes involve numerous irregularities and special cases [12]. Stemming errors are of two kinds: *understemming errors*, in which words which refer to the same concept are not reduced to the same stem, and *overstemming errors*, in which words are converted to the same stem even though they refer to distinct concepts. In designing a stemming algorithm there is a trade-off between these two kinds of error.

A *light stemmer* plays safe in order to avoid overstemming errors, but consequently leaves many understemming errors. A *heavy stemmer* boldly removes all sorts of endings, some of which are decidedly unsafe, and therefore commits many overstemming errors [12].

Shereen Khoja addressed the problems that might face the Arabic stemmer [9]:

"If the root contains a weak letter (e.g. "أ" alif, "و" waw or "ي" yaa), the form of this letter may change during derivation. To deal with this, the stemmer must check to see if the weak letter is in the correct form. " If not, the stemmer produces the correct form of this weak letter, which then gives the correct form of the root. If any triliteral rooted verb's one of the three root letters contains either "أ" alif (hamza, a), "و" waaw (w) or "ي" yaa (y) then that is defined as a weak verb, e.g. "وَهَبَ" *gave*, "وَجَدَ" *found*, "وَضَعَ" *put*, "وَقَفَ" *stood*, "وَعَدَ" *promised*, "بَاعَ" *bought*, "جاء" *came*, "قَرَأَ" *read*. Also weak verbs includes a triliteral rooted verb's where the second letter is doubled with a ّshadda, e.g. "شَمَّرَ" *prepared*. Shadda (Germination mark (tashdeed)) is written above the consonant that is doubled, and it look like the w shape. Strong verb is a triliteral rooted verb's which does not have any of the above three weak letters.

- "Some words do not have roots. For example the Arabic equivalents of "نحن" we, "بعد" after, "تحت" under and so on. If the stemmer comes across any of these words, it does nothing. "
- "Sometimes a root letter is deleted during derivation. This is especially true of roots that have duplicate letters (e.g. the last two letters are the same), e.g., "دُجِجَ" get dressed, "دَلَّلَ" dandle, "خَلَّلَ" souse, "عَلَّلَ" explained, "قَلَّلَ " reduced, "بَلَل" wet, …etc. The stemmer can detect this, and return the letter that was removed. - If a root contains a hamza, this hamza could change form during derivation, e.g., "قَالَ" talk, "قَامَ" stand up, ...etc. The stemmer detects this, and returns the original form of this hamza. "

L. S. Larkey, and M. E. Connell [11] conducted a good study based on a modified version of Shereen Khoja stemmer. The modified version includes a few changes to enhance the accuracy of the stemmer. These changes are summarized as follows:

- If a root were not found, the normalized form would be returned, rather than returning the original unmodified word.
- List of place names are considered "unbreakable" words exempt from stemming.
- In addition to the Arabic stop word list included in the Khoja stemmer, a script was to remove stop phrases.
- A light stemmer used to strip off definite articles (فالـ, كالـ, بالـ, والـ, الـ, ولـ, لـ and و) from the beginnings of normalized words and strips 10 suffixes from the ends of words (ات, ان, ها, ي, ة, ه, ية, يه, ين, and ون).

Table 5 in the appendix shows that light stemming leads to wrong results if it carried out unconditionally, so we record our reservation on the last step. Larkey, and Connell stemmer seems to be better than its parent (Khoja stemmer).

Morphology is a branch of linguistics that concerned with studying of the internal structure of word forms. Semitic languages have a complex morphology and so the Arabic language is a complex language for stemming. Arabic stemmers have to deal with affixes (prefixes, infixes, and suffixes), in addition to diacritics marks (harakat), in order to get the right root with its appropriate diacritics marks on it. Furthermore Arabic stemmer has to deal with

Arabized words (foreign words) which have no root, and in this case has to be excluded from stemming.

This study uses morphological patterns to obtain the trilateral and quadriliteral roots. The algorithm used simply tries to extract the root, in case there is a match between pattern infix and word infix.

Shereen Khoja is a pioneer in this field, but unfortunately we failed to get her original work entitled "Stemming Arabic Text " with her colleague Roger Garside. Leah S. Larkey and Margaret E. Connel and others headed a team at University of Massachusetts, Amherst to conduct a number of studies which depends on Khoja work. Their work [10][11] represent an improvement to Khoja work. Although their work include an improvements to Khoja but it does not solve the problems of negative prefix and negative suffix which discussed before. Al-Kharashi, I.A. et. Al. [2] presents pattern based stemming for Arabic language, also Taghva K. et. Al. [13] used the same approach which is different from Khoja, with an equivalent performance. Pattern based stemming does not use root dictionary. This approach based on matching the word with a number of Arabic patterns to extract the root. Chen A. et. Al. [4] conducted a study to find Arabic roots using Machine Translation (MT) based stemmer. Although this study depends on Ajeeb machine translation system, stopword removing, clustering, light stemming, and morphological analysis, but it does not presents a solution to the problems of negative prefix and negative suffix. Kareem Darwish [5] shows how to extract a root from the word, by first removing the prefix and suffix of the word to get a stem, then match a stem to a number of templates to get the root. In this study the researcher did not mention how many templates used in comparisons, beside the absence of an algorithm. Darwish, K. et. Al. [6] used an approach which is similar to his previous one[5], but with more details about the prefixes, and suffixes being removed. Table 6 shows the patterns used within our algorithm.

2. THE ALGORITHM

The first step of the Arabic Rooter under study is to normalize the text. Afterward a matching is performed between the stem and the verbal and noun patterns, in order to obtain the root. To conduct this study, a system (stemmer) is built to find the Arabic roots using Visual Basic 6.0. This stemmer kept the words unchanged if it failed to find a root, and this a normal case when the stem is an Arabized word or when it represent the names of places, such as continents, regions, countries, states, districts, cities, villages, rivers, mountains, deserts, … etc.

- Germination mark (tashdeed) (ّ) "shaddah" is placed above a consonant letter as a sign for the duplication of the consonant
- ***T(i)*** be any term
- Let ***LenT(i)*** be the length of each term
- Let ***n*** be a number of terms within a document
- Let ***chr(i)*** be the character position within a term
- Let ***LenP(j)*** be the length of the pattern
- Let ***Infixes_String*** be a string generated manually, consisting of the pattern, and the affix of that pattern, e.g., the stem "مـ سابح" *swimming pools*, match with the pattern of "مَفاعِ لْ", so the ***Infixes_String*** in this case is the string "مـا", where "م" lie in the first position, and "ا" lie in the third position.
- Let ***T_String*** be the corresponding string of the word which corresponds the string of the pattern ***Infixes_String*,** i. e., to clarify the idea suppose we want to find the root of the stem "مسابح" *swimming pools*, the system has to check this word with all 5 characters patterns, one of these pattern is "تَّفعيلْ", so the ***Infixes_String*** in this case is "تـي" and the ***T_String*** is "مـب", the mismatch is obvious in this case, when matching the stem with the pattern "مَفاعِلْ" the ***Infixes_String*** & ***T_String*** will be "مـا".

Table 1 shows how to get ***Infixes_String*** for each of the patterns used.

Table 1: An example of patterns and their infixes, and the position of each infix

Pattern	Infixes_String	Infix : Infix position			
فعال	ا	3 : ا			
مفعول	مو	1 : م	4 : و		
يستفعلن	يستن	1 : ي	2 : س	3 : ت	7 : ن

1. **Stop word removal** depending on a list of (1281) stop words consists of prepositions, pronouns, article and conjunction.

2. **Normalization**
   - **2.1** Remove *tatweel* (kasheeda) symbol ("_")
   - **2.2** Remove *punctuations* using a list of punctuation characters

2.3 Remove *diacritics* depending on a list of diacritics characters

3. If ***LenT*(*i*)** ≥ 5 **then**
Remove initial definite article (ال، لل)
Else if ***LenT*(*i*)** ≥ 6 **then**
Remove initial definite article (كال، فال، بال)
End if

4. If ***LenT*(*i*)** > 4 and the final character of the *T*(*i*) like "ء" then
Replace final "ء" with "ي"
End if

5. Replace *initial* (إ,), (أ) with bare alif (ا)

6. Replace *initial* (آ) with bare alif (ا)

7. Replace *final* (ة) with (ه)

8. Replace *final* (ى) with (ي)

9. For ***i*** ß 1 to ***n*** do
9.1 If *LenT*(*i*) = 3 **then**
9.1.1 If ***T*(*i*)** ends with germination mark (tashdeed) (ّ) **then** *Root*(*T*(*i*)) = *chr*(1)& *chr*(2)& *chr*(2)
Else return the *normalized term* (Stem)
9.1.2 If ***LenT*(*i*)** = 3 **then**
Root(***T*(*i*)**) = ***T*(*i*)**
Return *Root*(***T*(*i*)**)
9.2 If ***LenT*(*i*)** ≥ 4 **then**
9.2.1 For ***j*** ß 1 to number of patterns of length = ***LenT*(*i*) do**
9.2.1.1 If ***T_String*** match ***Infixes_String*** **then**
9.2.1.1.1 Remove the infix characters from ***T*(*i*)**
9.2.1.1.2 Replace "ى" with "ي"
9.2.1.1.3 Replace with "أ "
9.2.1.1.4 Replace "ء" or "ى" with "ي"
9.2.1.1.5 Return *Root* (***T*(*i*)**)
Else
Return the *normalized term* (Stem)
Next ***j***
Next ***i***

3. EVALUATION

In order to test the accuracy of our algorithm, we selected a number of words randomly. Table 2 shows the manual trace of the execution of the above algorithm to extract the root of the selected terms.

Table 3 shows the strength and weakness of the above algorithm, using a small data sets containing 1,827 words. The system failed to analyze 55 words, since their patterns are unknown. This failure mostly due to foreign (Arabized) words. The system accepts to analyze the rest (1,772 words), but we found that accuracy of extracting the right roots is 91%.

Table 2. Trace of the manual extraction of the correct root**.**

Original Word *T*(*i*)	Normalized *T*(*i*) (Stem)	T_String	Root (*T*(*i*))	Status
التعليمات	تعليمات	تيات	عِلْمْ	Right
الميزان	ميزان	ان	ميز	Wrong
الإستثمارية	استثماريه	استايه	ثَمَرْ	Right
للمعلمين	معلمين	مين	عَلِمَ	Right
الإسترحام	استرحام	استا	رَحِم	Right
سيتركانها	سيتركانها	سيتاها	تَرَكَ	Right
للمرشدين	مرشدين	مين	رَشَدَ	Right
مدّ	مدّ	-	مَدَدَ	Right
ميزان	ميزان	ان	مَيَزَ	Wrong
تسائلوا	تسائلوا	تاوا	سَأَلَ	Right
المدارس	مدارس	ما	دَرَسَ	Right
كريم	كريم	ي	كَرُمَ	Right
بالمكتبة	مكتبه	مه	كَتَبَ	Right
الطائر	طاأر	ا	طَأَرْ	Wrong
يستجيبون	يستجيبون	يستون	جيبْ	Wrong
محيطها	محيطها	ها	مُحِيطْ	Wrong

Table 3 Accuracy of root extraction for three Arabic text files

Number of words	Words not Analyzed	Number of incorrect Roots	Number of Roots extracted correctly
147	3 (2%)	16 (10.8%)	130 (87.2%)
244	7 (2.8%)	24 (9.8%)	215 (87.4%)
579	19 (3.3%)	33 (5.7%)	527 (91%)
857	26 (3%)	39 (4.6%)	791 (92.4)
1827	**55 (3%)**	**112 (6.1%)**	**1663 (91%)**

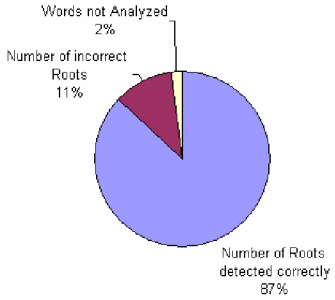

Figure 1 Statistics for root extraction

Table 4 shows the precision, recall and the harmonic mean (*F*-measure). Here we used the precision, recall and *F*-measure as shown in the following formulas:

$$Precision = \frac{Correct}{Correct + Incorrect} \quad \text{..... (1)}$$

$$Recall = \frac{Correct}{Correct + UnAnalyzed} \quad \text{..... (2)}$$

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \text{..... (3)}$$

Table 4 shows that the system obtains about 92% overall precision for the analyzed words, note that words that doe not match any of the verbal and noun patterns have been ignored as illustrated in table 6 from the computations of the accuracy measures, because these words are foreign words.

Table 4. Accuracy of root extraction for three Arabic text files

Number of words	Recall	Precision (Accuracy of Analyzed word)	*F*-measure
147	0.9771	0.8889	0.9309
244	0.9682	0.8987	0.9322
579	0.9652	0.9411	0.9530
857	0.9682	0.9531	0.9606
1827	**0.9697**	**0.9204**	**0.9442**

4. CONCLUSIONS

In order to increase the accuracy of the system, and to reduce the probability of facing the problems of negative suffix and negative prefix, the system shall not remove the prefixes ("فـ" ,"ب" ,"لـ" ,"و" ,"فـ") and suffix ("ـه").

Furthermore the system uses a conditional removing, e.g., in case the term length is six or more the system will remove the following prefixes ("وال", "بال" , "كال", "فال") otherwise when the term length is the length is less than six the term will be unchanged.

As mentioned in Thabet [14] root-based algorithm increases word ambiguity, where many word variants have different meaning, and this will affect the accuracy of IR, Text mining, …etc systems which rely on root based stemmers. Table 5 presents a number of ambiguous cases, one of these is the term "والدين", this can be interpreted by the reader as parents, religion, and debt, since this word is bare of diacritics, and it is in its own, not within a statement. As we said the diacritics used to distinguish the words semantically and phonetically.

Arabic stemmers can be used to enhance the efficiency of a number of systems such as, Spell checkers, Information retrieval systems, Text mining systems, Text Analysis systems, Compression systems… etc.

This algorithm is incapable of extracting Arabic roots of some imperative verbs ("أفعال الأمر") that is made up of one Arabic letter with the fact that its root being of three letters (trilateral verbs), e.g., " "عِ" , with the root of "وعِي". In addition, the problem of defective roots (weak roots) is still not solved by this algorithm. Defective roots are roots that contain vowels ("أ"،"و"،"ي") which are classified as irregular roots, since some vowels in these roots are altered to other vowels or removed in the derivational process [1], e.g., "رما" and "رمي" these two words have the same meaning *throw*, and both of them represent the same root. As a future research, we hope to solve these problems within our next enhancement to this work.

REFERENCES

[1] Aljlayl. M, Frieder. O. "On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach", *CIKM 02*, November 4-9, 2002, McLean, Virginia, USA. Pages 340 -- 347. ACM 1-58113-492-4/02/0011.

[2] Al-Kharashi, I.A., & Al-Sughaiyer, I.A. (2002e). "Pattern-based Arabic stemmer". *In Proceedings of the 2nd Saudi Technical Conference and Exhibition (STCEX2002*), Volume II (pp. 238-244), Riyadh, Saudi Arabia.

[3] Baeza-Yates, R., & Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, 1999.

[4] Chen A. and Gey Fredic. 2002. "Building an arabic stemmer for information retrieval". *In Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, National Institute of Standards and Technology, November.

[5] Darwish K. 2002. "Building a shallow Arabic Morphological Analyzer in one day", *In proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, Association for Computational Linguistics , July.

[6] Darwish, K. and D. Oard. "CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval". *In TREC*. 2002. Gaithersburg, MD.

[7] Frakes W. B., *Introduction to Information Storage and Retrieval Systems*, chapter 1, pages 1--12. Prentice-Hall, 1992.

[8] Kanaan, G.; Al-Shalabi, R.; AL-Kabi, M.N.; Jaam, J.M.; Hasnah, A.; . 2004. "New Approach for Extracting Quadriliteral/Quadrilateral Arabic Roots ", *In proceedings of 1st International Conference on Information & Communication Technologies: from Theory to Applications*, ICTTA'04, (Damascus, Syria, April 2004). IEEE-France.

[9] Khoja S., Research Interests, Pacific University, 2043 College Way, Forest Grove, Oregon 97116, http://zeus.cs.pacificu.edu/shereen/research.htm, July 8, 2006.

[10] Larkey L., Ballesteros L., and Connell M., "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis," *SIGIR 2002*: 275-282, 2002.

[11] Larkey L. S., and Connell M. E., "Arabic information retrieval at UMass in TREC-10". *In TREC 2001*.

[12] Paice C.D., "An evaluation method for stemming algorithms". In W.B. Croft and C.J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 69-90. Springer-Verlag, July 1994.

[13] Taghva, K., Elkoury, R., and Coombs, J. "Arabic Stemming without a root dictionary".2005. www.isri.unlv.edu/publications/isripub/Taghva2005b.pdf

[14] Thabet, N. (2004). "Stemming the Qur'an". *In Proceedings of Arabic Script-Based Languages Workshop*, COLING-04, Switzerland, August 2004.

Appendix A:

Table 5: The problem of negative prefixes and negative suffixes

Full word	Removing the suffix ات	Full word	Removing the suffix ان	Full word	Removing the suffix ون	Full word	Removing the suffix ين
البركات	البرك	الأمان	الأم	بالعون	بالع	الأمين	الأم
التعليمات	التعليم	الإنسان	الإنس	البالون	البال	التامين	التام
الثورات	الثور	الأوان	الأو	بطون	بط	تحسين	تحس
الجماعات	الجماع	الأوطان	الأوط	بلون	بل	حنين	حن
الحملات	الحمل	بركان	برك	التعاون	التعا	الدين	الد
الدورات	الدور	الجنان	الجن	الحسون	الحس	الذين	الذ
دوريات	دوري	الحنان	الحن	حنون	حن	سجين	سج
الذات	الذ	خلجان	خلج	الستون	الست	سكين	سك
السلطات	السلط	الريان	الري	سكون	سك	سنتين	سنت
السنوات	السنو	الريحان	الريح	صابون	صاب	سنين	سن
السياسات	السياس	الضمان	الضم	العيون	العي	عين	ع
الشركات	الشرك	عجمان	عجم	قرون	قر	قوانين	قوان
طبقات	طبق	عنوان	عنو	كانون	كان	كدين	كد
القوات	القو	لبنان	لبن	مرهون	مره	لين	ل
لجأت	لج	لعمان	لعم	المليون	الملي	متين	مت
لذوات	لذو	للبنان	للبن	الهرمون	الهرم	مدللين	مدلل
للهواة	للهو	مرجان	مرج	يدرون	يدر	مسكين	مسك
لنزلات	لنزل	الميزان	الميز	يُصلون، يَصلون	يصل	المعلقين	المعلق
مداخلات	مداخل	نيسان	نيس	مضمون	مضم	معين	مع
النقاشات	النقاش	الهوان	الهو	مسكون	مسك	يمين	يم
وذرات	وذر	اليابان	الياب	مفتون	مفت		

Table 6: Verbal and noun patterns used within the algorithm

Full word	Pattern's used
Length 3 patterns	فعّ
Length 4 patterns	فعال فعول فعيل افعل فعلي فعلن فعله فعلا مفعل تفعل نفعل يفعل فعلت فوعل فيعل فتعل
Length 5 patterns	تفعيل افعال فعليه فعليا فعيلا فعيله يفتعل نفتعل تفتعل فوعله فيعله فتعله فواعل مفاعل فعائل فعالي فعالى فعلنا فعولا مفعله فعلته اتفعل ايفعل انفعل افعلن افتعل افاعل افعلي مفعول منفعل متفعل فعلون فعلين فعلان مفعال مفتعل مفعيل بفعال لفعال نتفعل يتفعل تتفعل فعلتم فعلهم فعلتن فعلكم فعلهن فعلها فعلهم فعلكن فعلوا فعلات فعلتك فعلوك فاعول تفاعل نفاعل يفاعل فعالى فعالي فاعلي فاعيل بفاعل لفاعل بفعول لفعول بتفعل لتفعل بفعيل لفعيل بمفعل لمفعل مفعلك بفعلك لفعلك يفعلك نفعلك تفعلك مفعلا كفعول كفعله لفعله بفعله افعله فعالة فعالل فعللي فعلله ففعله مفعلل تفعلل لفعلل بفعلل فويعل فعفله فيعلا فعلال فتفعل فنفعل فيفعل فعلتي فعلني يفعله نفعله تفعله فاعلا فاعله افعلت
Length 6 patterns	مفعليه افعوله يفاعله تفاعله نفاعله فاعلات فاعلان تفاعلن نفاعلن يفاعلن مفاعلن تفاعلت افعلاء فعاليه مفعلات تفعلات كفعلات لفعلات مفاعله استفعل مستفعل يستفعل نستفعل تستفعل افعالي افتعال افعوعل فعلتنا ففعلنا يفعلنا تفعلنا مفعلنا انفعال افعلنا افاعيل مفاعيل تفاعيل متفاعل نتفاعل يتفاعل تتفاعل فواعيل بمفاعل لمفاعل كمفاعل بتفعيل لتفعيل فعلتهم فعلتهن فعليهم فعليكن فعلتكن فعلتكم فعليكم بمفعلك لمفعلك فعولات فعيلات فعيلان فعلتوا فعالتي افعالا فعلليه تفعلوا يفعلوا فعلكما فعلهما مفعليا فعيليا لفوعله ففعلها ففعلهن ففعلكن ففعلكم لفعلها لفعلهن لفعلكن لفعلكم لفعلهم فعلاها فعلاهن فعلاكن فعلاكم فعلاهم فعللها فعللهن فعللكن فعللكم فعللهم فعلتان فاعوله مفعوله تتفعلل نتفعلل يتفعلل متفعلل افعلوا بفاعلك لفاعلك مفاعلك يفاعلك نفاعلك تفاعلك ففاعلك افاعلك كفعلهن كفعلهم بفعلهن مفعلهن لفعلهن افعلهن افعلهم افعلكم افعلكن مفعلكن مفعلكم مفعلهم مفعلهن كفعلكن كفعلكم كفعلهم كفعلهن لفعلتك مفعلتك كفعلتك بفعلتك فعالات فعالان فعالته مفتعله يفعلان تفعلان مفعلان بافعال تفعلين يفعلون تفعلون فاعلون فاعلين
Length 7 patterns	مفعولات مفعولان افعواله تفعيلات تفعيلان فعلانيه بالتفعل تفعيلها يفاعلون تفاعلون افعالات استفعلت استفعلن مستفعله استفعال افاعيل فانفعلت فعالتنا تفاعلوا مفاعليه مفاعلته مفاعلون مفاعلان مفاعلين مفاعلهن مفاعلكم مفاعلكم فافعلها فافعلكن فافعلكم فافعلهن لافعلها لافعلكن لافعلكم لافعلهن لفعللها لفعللهن لفعللهم لفعللكم لفعللكن فعلاهما فعلاكما فعللهما فعللكما فاعلتان فاعليان افتعالا تتفعلون تتفعلين تتفعلان مستفعلن تستفعلن يستفعلن فعلائهم فعلائهن فعلائكم فعلائكن فافعلهم افعالهم افعالهن افعالكم افعالكن فعلائها سافعلهم سافعلكم سافعلكن سافعلهن تفاعلنا يفاعلنا بفاعليه لفاعليه بفاعلوه لفاعلوه بفاعلوك لفاعلوك بفاعلون لفاعلون بفاعلين لفاعلين بفاعلهن لفاعلهن بفاعلكن لفاعلكن بفاعلكم لفاعلكم بفاعلهم لفاعلهم متفاعله بتفاعله لتفاعله كتفاعله بفعلتهن كفعلتهن لفعلتهن مفعلتهن بفعلتكن كفعلتكن لفعلتكن مفعلتكن بفعلتكم كفعلتكم لفعلتكم مفعلتكم بفعلتهم كفعلتهم لفعلتهم مفعلتهم بفعولكم كفعولكم لفعولكم مفعولكم بفعولهم كفعولهم لفعولهم مفعولهم بفعولهن كفعولهن لفعولهن مفعولهن بفعولكن كفعولكن لفعولكن مفعولكن تفاعلهن يفاعلهن كفاعلهن تفاعلهم يفاعلهم كفاعلهم نفاعلهم مفاعلهم تفاعلكم يفاعلكم كفاعلكم نفاعلكم تفاعلكن يفاعلكن كفاعلكن نفاعلكن فعاليات فعاليان
Length 8 patterns	افتعاليه افعاليان افعاليات يستفعلون تستفعلون مستفعلون يستفعلان تستفعلان مستفعلات مستفعلان مستفعلون مستفعلين مستفعلهن مستفعلكم مستفعلهم مستفعلكن افعلائهم افعلائكن افعلائكم افعلائهن بفاعلكما لفاعلكما بفاعلهما لفاعلهما بفعلتكما لفعلتكما بفعلتهما لفعلتهما بفعليكما لفعليكما بفعليهما لفعليهما افتعالات افتعالها فاعليهما فاعليكما فاعلتهما فاعلتكما
Length 9 patterns	فعلتموهما استفعالات استفعاليه سيفعلانها ستفعلانها