An Approach for Arabic Root Generating and Lexicon Development

Mohamed Osman Hegazi

Department of Computer Science, College of Computer Engineering and Science, Prince Sattam University, Saudi Arabia;
Department of Computer Science, College of Computer Science and IT, Alzaiem Alazhari University Sudan

Summary
This paper presents a novel approach for Arabic root generation and lexicon development. The approach provides three algorithms; in the first algorithm Arabic word root is generated using the concept of permutation and combination, the root generator algorithm generates roots by applying permutations to the Arabic alphabetic letters. Then, the second algorithm is used for developing difference words from that root using Arabic morphology template, the morphology developing algorithm develops the Arabic word by formulating the roots according to the Arabic template. Finally, lexicon is constructed by providing meaning and other information. The paper provides a contribution to the field of Natural Language Processing (NLP); and hence it provides a novel Arabic NLP approach that might be used in different ways in generating Arabic roots and developing Arabic lexicons. The approach presented in this paper can be used as a basis for processing, understanding and electronically using of Arabic language, because it provides a way for covering most of Arabic language components. The results show that this approach provides up to 21,924 different Arabic roots with their morphological information.

Key words:
Arabic Language; Stemming Algorithm; Arabic Lexicon; Arabic Root; Arabic Word; Morphology; Natural Language Processing (NLP).

1. Introduction

Arabic language is characterized by a set of features including:

1- Link revelation, because it is the language of the holy Quran, which was the holly book for all Muslims (more than 1.62 billion people) [19].
2- It is the native language for More than 422 million speaker.
3- Well-established language, which is rooted in the depths of history for more than fifteen centuries [20].
4- Abundance of vocabulary: we can derive dozens of words from one root, for example from the root (ك ت ب) (wrote) we can get (كتب ، يكتب ،اكتب، اكتبا ، اكتبي ، اكتبوا ، اكتبن ، وأكتب، ونكتب، ويكتب ،وتكتب ، وكاتب ،ومكتوب ومكتبة، ومكتب، وكُتُبٌ، وكتيبة، وكَتَبَةٌ، وكاتبة، وكاتبون، وكاتبات، وكويتب، وكويتبات، وكتيبات ….. etc).
5- The connotations or linguistic features are carried on the structural templates not like other languages that provide connotations by connecting to the body of words for example in English (ing) at the end of word to indicate continuous tense. In Arabic this is based on templates such as (فاعل): for the noun of the subject (كاتب- writer), (مفعول) the noun of the objective: (مكتوب - written)…etc.
6- Arabic language consists of 28 alphabetic letters and it is written from right to left. The letters in the Arabic words are written in cursive way not separated as in English language. Also, the shape of characters are different according to their position in the word.
7- The Arab lexicon is the richest dictionaries of languages in vocabulary and synonyms (wealth verbal); the large Arabic dictionaries can have more than one million vocabulary. The Arabic vocabulary does not have an inventory of the lexicon materials; because Arabic is derivatitive language, and the language article in traditional Arab lexicon is just the root. Many words with different meanings can come from one root, Figure **1**) shows an example of different words generated from one root كتب (wrote).

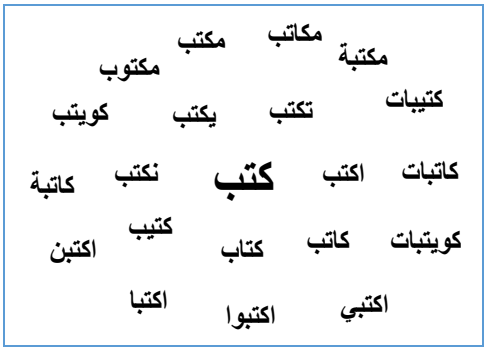


Figure 1: Example of difference words generated from one root "كتب" (wrote)

According to these characteristics; processing Arabic language and analyzing its morphology is not an easy task. Therfore many efforts have been done to provide the basis for Arabic language processing, but till now there is no

Manuscript received January 5, 2016
Manuscript revised January 20, 2016

satisfactory results or successful algorithms for generating Arabic word roots or morphological information.
This paper provides an algorithm that generates the root of Arabic words by applying permutations on the Arabic letters. The paper also provides another algorithm to build Arabic lexicon by generating different morphological information for these roots and then provides meanings and examples for each word from a holy Quran dataset [18] (the holy Quran is considered as a standard Arabic reference since around 15 century).

1.1 Contribution

This paper provides a contribution in the field of NLP by: providing a novel root generator algorithm for giving the roots of Arabic words using an approach that permutes and combines the Arabic alphabet. This way of generating the word root was used by Al Khalil Bin Ahmed in constructing his valuable and well-known lexicon before more than 1200 years ago.
Building a comprehensive electronic Arabic lexicon (Arabic language dataset) that can be used in Arabic language processing and for other purposes such as dictionaries, word processing, spelling checker.. etc.

1.2 Previous work

Most of Arabic language processing achievemnets are in the area of Arabic morphology. The research projects that work on Arabic stemming algorithms or root generating can be classified into two categories:

- Rule-Based stemming or root generating algorithm: This category is studied more than the other kinds of algorithms, because it provides more accurate results than other algorithms. Most of works on rule-based stemming algorithms depend on affix removal such as khoja stenmmer and Tim Buckwalter morphological analyzer. In general there are many examples of recently research work under this category such as [7], [4],[1] , [5] and [9].
- Classification or clustering algorithms: The stem in this category of algorithms is classified or clustered according to the stem type and language document. For example in [13] the association rules are used in clustering the stem of text. In [14] Stem is generated based on feature reduction technique which is used to categorize Arabic Text. In [21] algorithm based on Naive Bayes is developed to categorize Arabic document. In [15] the authors use classification with p-stemmer for Arabic text. Authors in [8] studied the effect of stemming on Arabic text classification.

In the area of lexicon development most of the previous work concentrated on morphological analysis either to provide a special corpus or documents set such as [2], [3], [10] and [12] or in grammar or morphological computational such as [11],[17] , [16].

2. Proposed Algorithms

This paper presents an approach for generating Arabic word roots by emulating the Arabic letters and then generate the corresponding word, morphology, and adding the meaning to resulted Arabic lexicon. Consequently, two main algorithms have been initiated, one for generating roots and the other for developing lexicons.

2.1. Generating the Roots

Most of the roots of Arabic words are made up of three characters in form of (فعل) template, no roots less than three characters [20] exist in Arabic language. The roots that consist of two characters are usually has repeated characters or bugs. There are some roots, which consist of 4 or 5 characters, these types of roots are for special words which can be studied individually such as (دحرج. بعثر. زلزل. طمأن. وسوس. دِرْهَم) as for 4 characters root, which is in form of (فعلل) template, and (سَفَرْجَل . زَبَرْجَد. غَضَنْفَر) as for 5 characters root, which is in form of (فعلّل) template. Therefore, most of Arabic linguistics and morphological analysis are based on three-character root. Accordingly, our proposed algorithms work in generating three characters root. The root generator algorithm is formulated according to the following approach:

1- State the framework (the mathematical permutation and combination theory).
2- Apply the algorithm.
3- Present the results.
4- Test the results.

2.1.1. The framework for our proposed algorithm

Our proposed algorithm for generating the Arabic roots based on the mathematical permutation and combination theory. According to the permutation theory (equation (1)) if we want to arrange unrepeated three items out N elements then the result will be equal to N×(N-1)×(N-2), and it is equal to N×N×N if it has repeated items.

$$P_r^n = \frac{n!}{(n-r)!} \qquad (1)$$

In Arabic language there are 28 alphabetical letters or characters (see table 1). The three characters in an Arabic root can be a combination or a permutation of any three letters without repeating of one character three times, repeating of two characters is allowed. Based on this

concept and on the permutation theory we can generate two types of correct Arabic roots:

1- Maximum Arabic word roots: the roots that can have adjacent repeated two characters. This kind of roots can be generated using the following formula:
   a. Find all possible roots, the maximum three arrangement groups of characters (i.e. with one repeated character). This is equal to 21952 (28×28×28)
   b. Remove the repeated one character root (28 roots: the number of Arabic alphabets), then the maximum number of roots that can be generated from 28 Arabic alphabetical letters will be 21924 roots (21952-28). These roots represent the maximum possible of all Arabic Language roots.
2- The strong Arabic roots: the roots that have no adjacent repeated two characters, these kinds of roots represent the roots of the common used Arabic words. To generate this kind of root we need to find a way for avoiding the occurrence of the adjacent repeated two characters, if we do that the total number of roots must be equal to 28×27×27= 20412.

Table 1: Arabic alphabetic

#	Arabic letter	English name	Pronounce
1	أ	Alef	A
2	ب	Baa'	Bah
3	ت	Taa'	Tah
4	ث	Thaa'	Thah
5	ج	Jeem	Jah
6	ح	Haa'	Hah
7	خ	Khaa'	Khah
8	د	Dal	Dah
9	ذ	Thal	Thah
10	ر	Raa'	Rah
11	ز	Zain	Zah
12	س	Seen	Sah
13	ش	Sheen	Shah
14	ص	Saad	Sua
15	ض	Daad	Dhah
16	ط	Taa'	Tah
17	ظ	Zaa'	Zhah
18	ع	Aain	Aaa'
19	غ	Ghain	Ghah
20	ف	Faa'	Fah
21	ق	Qaaf	Qah
22	ك	Kaf	Kah
23	ل	Lam	Lah
24	م	Meem	Mah
25	ن	Noon	Nah
26	هـ	HHaa'	HHah
27	و	Waw	Wah
28	ي	Yah	Yah

2.1.2 Root Generation algorithm

To generate all possible three characters root, either the maximum root or the strong root, we will use three nested FOR loops, as it shown in fFigure 2). For the maximum root we use the operation AND inside IF statement clause. For the Strong root we use the operation OR instead of AND. The results (the roots) are stored into structured database table because the number of roots is large and we are going to use this database table also to store electronic information about lexicon. We used visual basic.net and Microsoft access for implementing our algorithm. Figure 3) shows snapshot of actual code that used in generating our roots, while Figure 4) show snapshots for the database tables that used to store the maximum roots and strong roots respectively.

```
Array [أ, ب, ت, ث,……]
For i=0 to 27
For j=0 to 27
For k= 0 to 27
If (Array[i] <> Array[j]) AND/OR ( Array[i] <> Array[k])
Then
Root= Array[i] & Array[j] & Array[k]
New Database record = root
End if
i=i+1
j=j+1
k=k+1
```

Figure 2: Root generation Algorithm

2.1.3 Root Generation results

The implementation of the maximum root generation algorithm provides all possible Arabic roots, 21924 roots, including adjacent repeated two characters, (figure 4-a) which is equal to the number that is calculated by the emulation equation (see section 2.1.1). All roots are made of three characters; no two roots have three repeated characters.

The roots that are generated by this algorithm (maximum root) can be suitable to use as a basis and framework for all Arabic language, because it generates all possible roots in Arabic language. The only point against this generator is that, it may generate roots that are not an actual Arabic word root, especially when the root includes adjacent repeated two characters, but we think this is an advantage because it may cover the extinct Arabic language, or the special Arabic language. (There are varieties of Arabic language according to varieties of regions or tribes). Accordingly, we suggest that the maximum roots can be used as reference or a useful Arabic language dataset.

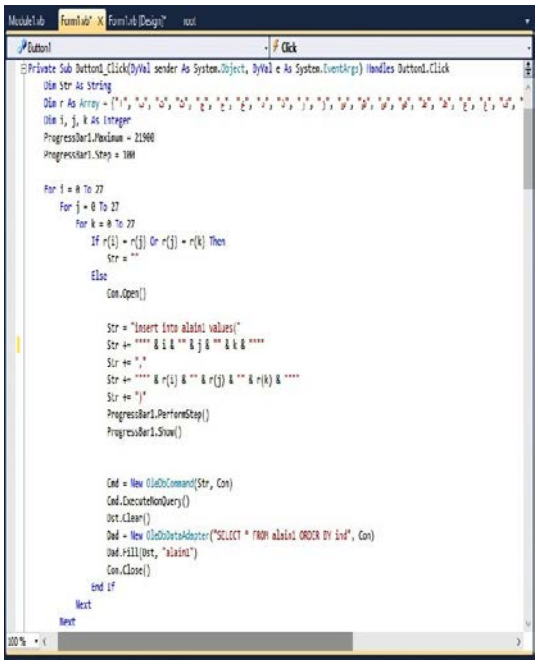


Figure 3: Snapshot of part of the root generation algorithm VB.Net code

The implementation of the strong root generator algorithm provides 20412 Arabic roots (figure 4-b) which is equal to the number that calculated by the emulation equation (see section 2.1.1). All roots are made of three characters without any three repeated characters or adjacent repeated two characters. The roots that generated by this algorithm has the characteristics of common and most used Arabic words. Accordingly, this root can be suitable for many of NLP process.

The suggested root generator algorithm is used for generating two kinds of roots, as it presented above. In addition, this algorithm can generate another kind of roots, for examples; if we remove one of the For Loops we can get two characters root, in the other hand if we add additional For-Loop we can get four characters root, and so on. Also, if we made any changes in the if-condition or if-operators we can get different kinds of roots, (figures (2) and (3)).

2.2. Developing the Arabic Lexicon

The proposed Arabic lexicon is developed in two steps: generating the morphology and providing the meaning.

ind	root
1	بزد
2	بزذ
3	بزر
4	بزز
5	بزس
6	بزش
7	بزص
8	بزض
9	بزط
10	بزظ
11	بزع
12	بزغ
13	بزف
14	بزق
15	بزك
16	بزل
17	بزم
18	بزن
19	بزه
20	بزو
21	بزي
22	بسا
23	بسب
24	بست
25	بست

Record: 25 of 21925

(A)

alain

ind	root
1	ابا
2	ابت
3	ابث
4	ابج
5	ابح
6	ابخ
7	ابد
8	ابذ
9	ابر
10	ابز
11	ابس
12	ابش
13	ابص
14	ابض
15	ابط
16	ابظ
17	ابع
18	ابغ
19	ابف
20	ابق
21	ابك
22	ابل
23	ابم
24	abn
25	ابه

Record: 25 of 20412

(B)

Figure 4: Snapshot of part of the roots tables (a) part of the maximum root table (25 records out of 204120) (b) part of strong root table (25 records out 20412)

2.2.1 Generating the morphology

Arabic language is based on root [20]; hence, all Arabic linguistic and morphology analysis is carrying on the structural templates based on word root. Arabic morphology templates are providing the format in which words are structured (Table 2).

Table 2:An example of Arabic morphology templates

فعل	فاعل	فعول	فعيل
مفعل	مفاعل	مفعول	مفاعيل
تفعل	يفعل	فعلل	يتفوعل
افعل	افعال	افتعال	يفتعل
استفعل	استفعال	مستفعل	يستفعل

The morphology generation algorithm generates different words from one root by inserting or adding letters to that root at certain positions according to the morphology template (Table 2). As it mentioned in section (2.1) the three characters root is of "فعل" template can be used as a basis for generating new words by adding characters at different positions. For example, if we want to generate a word in "فاعل" template then we will insert the letter "ا" at the second position, from right of the root string.Figure 5) shows how the algorithm works.

```
Open the root database table
While not end of the root table
        String = root
        Word1 in "فاعل"  template = insert "أ" into
the second position  of string
        Word2 in "فعول"  template = insert "و" into
the third position  of string
        Word1 in "فعيل"  template = insert "أ" into
the second position  of string
        ……
        ……
        Update root set value = …., …,
        Move to next record
Update table
End loop
```

Figure 5:Morphology generator algorithm

The morphology generation algorithm is implemented using VB.Net and MS-Access database. Four templates are used to generate four words, as an example, given that the same algorithm can be used for generating large number of words (all possible words can be generated). Figure 6) shows snapshot of part of the code that used in generating these words and Figure 7) shows snapshot of part of the database table where the roots and the words that are generated as a morphology from that roots are stored. This table presents the first stage in our proposed lexicon.

2.2.2 Providing the meaning

The last stage in developing our proposed Arabic lexicon is to provide meaning and examples for our generated roots and the words that structured from those roots. To do this we can use any searching algorithm that searches for the root or the structured word in one of the well-known Arabic datasets and then add the searching results to our proposed lexicon (the database records).
Figure 8) shows a snapshot of the code that we developed to search for the meaning of the generated words from one of holy Quran datasets [18]. The searching results are then added to our proposed lexicon, these results include all available data in the dataset, such as the Arabic meaning, English meaning, an example sentences include the word (the Quran verse that contains the word or the root), the pronouns … etc. Figure 9) shows part from the resulted lexicon.

```
Con.Open()
Dad = New OleDbDataAdapter("SELECT * FROM alain1 ORDER BY ind", Con)
Dad.Fill(Dst, "alain1")

Con.Close()
Dim Str, s As String
Dim i, j As Integer
i = CurrentRow
j = Dst.Tables("alain1").Rows.count - 1
Try

    Con.Open()
    While i <> j

        s = Dst.Tables("alain1").Rows(CurrentRow)("root")
        Str = LSet(s, 1) & "ا" & Mid(s, 2)
        Dst.tables("alain1").rows(CurrentRow)("faail") = Str
        Str = "م" & LSet(s, 1) & "ا" & Mid(s, 2)
        Dst.tables("alain1").rows(CurrentRow)("mfaail") = Str

        Str = LSet(s, 2) & "و" & Mid(s, 3)
        Dst.tables("alain1").rows(CurrentRow)("faool") = Str
        Str = "م" & LSet(s, 2) & "و" & Mid(s, 3)
        Dst.tables("alain1").rows(CurrentRow)("mfaool") = Str

        Str = LSet(s, 2) & "ي" & Mid(s, 3)
        Dst.tables("alain1").rows(CurrentRow)("faiil") = Str
        Str = "م" & LSet(s, 2) & "ي" & Mid(s, 3)
        Dst.tables("alain1").rows(CurrentRow)("mfaiil") = Str


        Str = "ي" & s
        Dst.tables("alain1").rows(CurrentRow)("yfal") = Str
```

Figure 6 :Snapshot of part from morphology generator VB.Net code

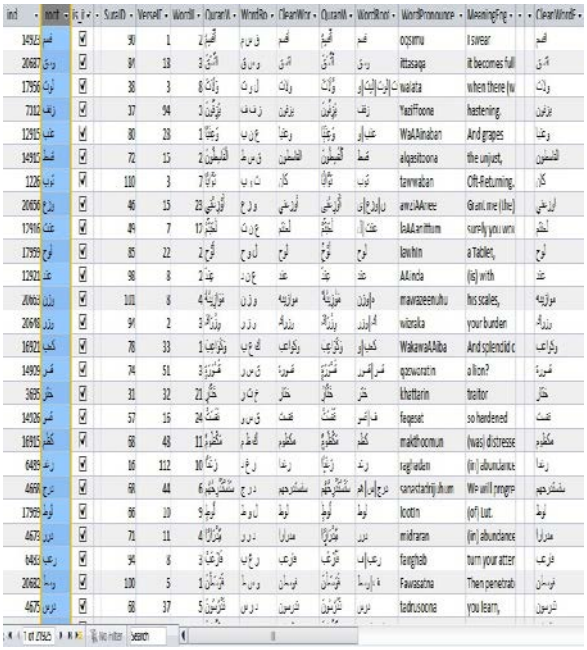
ind	root	faail	faool	faiil	mfaail	mfaool	mfaiil	yfal	tfal
3875	خلل	خالل	خولل	خيلل	مخالل	مخوّل	مخيّل	يخلل	تخلل
14195	فصل	فاصل	فصول	فصيل	مفاصل	مفصول	مفصيل	يفصل	تفصل
17764	لقح	ألقح	لقوح	لقيح	ملاقح	ملقوح	ملقيح	يلقح	تلقح
17343	لمج	ألمج	لموج	لميج	ملامج	ملموج	ملميج	يلمج	تلمج
9288	صقر	صاقر	صوقر	صيقر	مصاقر	مصوقر	مصيقر	يصقر	تصقر
4265	خفق	خافق	خوفق	خيفق	مخافق	مخوفق	مخيفق	يخفق	تخفق
17388	لحف	لاحف	لحوف	لحيف	ملاحف	ملحوف	ملحيف	يلحف	تلحف
14215	فضض	فاضض	فضوض	فضيض	مفاضض	مفضوض	مفضيض	يفضض	تفضض
12861	عشر	عاشر	عشور	عشير	معاشر	معشور	معشير	يعشر	تعشر
1757	نصب	ناصب	نصوب	نصيب	مناصب	منصوب	منصيب	ينصب	تنصب
12250	مدد	مادّ	مودّ	ميدّ	ممادّ	ممدود	ممديد	يمدد	تمدد
8936	شكك	شاكك	شوكك	شيكك	مشاكك	مشوكك	مشيكك	يشكك	تشكك
8937	شكل	شاكل	شوكل	شيكل	مشاكل	مشوكل	مشيكل	يشكل	تشكل
8941	شكو	شاكو	شوكو	شيكو	مشاكو	مشوكو	مشيكو	يشكو	تشكو
3894	خرص	خارص	خورص	خيرص	مخارص	مخورص	مخيرص	يخرص	تخرص
19630	هنو	هانو	هونو	هينو	مهانو	مهونو	مهينو	يهنو	تهنو
17391	لحن	ألحن	لحون	لحين	ملاحن	ملحون	ملحين	يلحن	تلحن
17390	لءم	ألءم	لءوم	لءيم	ملائم	ملءوم	ملءيم	يلءم	تلءم
14550	قرض	قارض	قورض	قيرض	مقارض	مقورض	مقيرض	يقرض	تقرض
17387	لحق	ألحق	لحوق	لحيق	ملاحق	ملحوق	ملحيق	يلحق	تلحق
12608	عسب	عاسب	عوسب	عيسب	معاسب	معسوب	معسيب	يعسب	تعسب
3890	خزر	خازر	خوزر	خيزر	مخازر	مخزور	مخزير	يخزر	تخزر
14206	فصح	فاصح	فصوح	فصيح	مفاصح	مفصوح	مفصيح	يفصح	تفصح
3885	خرج	خارج	خورج	خيرج	مخارج	مخروج	مخريج	يخرج	تخرج
14538	قوت	قاوت	قووت	قويت	مقاوت	مقووت	مقويت	يقوت	تقوت

Record: 1366 of 21925 | No Filter | Search

Figure 7: Snapshot of part of database table shows the roots and the words that generated as morphology for those roots

```
Module1.vb   Form1.vb   Form1.vb [Design]   root
Button3                                    Click

Do While Not EOF()
    s1 = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("CleanRoot")
    s2 = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("CleanRoot1")

    If s = s1 Or s = s2 Then
        Dst.tables("alain1").rows(i)("new") = s



        Dst.tables("alain1").rows(i)("SuraID") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("SuraID")
        Dst.tables("alain1").rows(i)("VerseID") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("VerseID")
        Dst.tables("alain1").rows(i)("WordID") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("WordID")
        Dst.tables("alain1").rows(i)("QuranWordArshad") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("QuranWord
        Dst.tables("alain1").rows(i)("NurRuulArshad") = Dst1.Tables("tblQuranNurCleanTashkeelRuut").Rows(k)("NurRuulAr
        Dst.tables("alain1").rows(i)("CleanWord") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("CleanWord")
        Dst.tables("alain1").rows(i)("QuranWordUsmani") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("QuranWord
        Dst.tables("alain1").rows(i)("WordRootNoorSoft") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("WordRoot
        Dst.tables("alain1").rows(i)("WordPronounce") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("WordPronounc
        Dst.tables("alain1").rows(i)("MeaningEnglish") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("MeaningEng
        Dst.tables("alain1").rows(i)("Intonation") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("Intonation")
        Dst.tables("alain1").rows(i)("MeaningArabic") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("MeaningArabi
        Dst.tables("alain1").rows(i)("CleanWordFull") = Dst1.Tables("tblQuranWordCleanTashkeelRoot").Rows(k)("CleanWordFu
        Dst.tables("alain1").rows(i)("is_it_in_Quran") = True


        Dim cond As New OleDbCommandBuilder(Dad)
        Dad.Update(Dst, "alain1")
        ProgressBar1.PerformStep()
        ProgressBar1.Show()

    End If

    k += 1

100 %
```

Figure 8 :Snapshot of the meaning developer algorithm.



ind	wordR	is_it_	SuraID	VerseID	WordID	QuranWord	WordRoot	CleanWord	QuranWord	WordRoot	WordPronounce	MeaningEng	CleanWordF
14562	قسم	☑	90	1	3	أُقْسِمُ	ق س م	قسم	أُقْسِمُ	قسم	oqsimu	I swear	قسم
20687	وعى	☑	84	18	3	ٱتَّسَقَ	و س ق	اتسق	ٱتَّسَقَ	وسق	ittasaqa	it becomes full	اتسق
17956	ولت	☑	38	3	8	وَلَاتَ	ل ا ت	ولات	وَلَاتَ	ت+لا+و	walata	when there (w	ولات
7102	رهف	☑	37	94	5	يَرْفُونَ	ر ف ف	يرفون	يَرْفُونَ	رفف	YarFoone	hastening.	يرفون
12815	عنب	☑	80	28	3	وَعِنَبًا	ع ن ب	وعنبا	وَعِنَبًا	عنب	WaAlnaban	And grapes	وعنبا
14912	قسط	☑	72	15	2	ٱلْقَٰسِطُونَ	ق س ط	القاسطون	ٱلْقَٰسِطُونَ	قسط	alqasitoona	the unjust,	القاسطون
1226	كاب	☑	110	3	7	تَوَّابًا	ت و ب	كانا	تَوَّابًا	توب	tawwaban	Oft-Returning,	كانا
20656	برح	☑	46	15	23	أَوْزِعْنِيٓ	و ز ع	اوزعني	أَوْزِعْنِيٓ	ا+وزع+ا+ي	awziAAnee	Grant me (the)	اوزعني
17916	عنت	☑	49	7	17	لَعَنَّتَكُمْ	ع ن ت	لعنتم	لَعَنَّتَكُمْ	عنت+ا	laAAanittum	surely you wou	لعنتم
17959	لوح	☑	85	22	2	لَّوْحٍ	ل و ح	لوح	لَّوْحٍ	لوح	lawhin	a Tablet,	لوح
12821	عند	☑	98	8	2	عِندَ	ع ن د	عند	عِندَ	عند	AAinda	(is) with	عند
20665	وزن	☑	101	8	4	مَوَٰزِينُهُۥ	و ز ن	موازينه	مَوَٰزِينُهُۥ	موازين+ه	mawazeenuhu	his scales,	موازينه
20648	وزر	☑	94	2	3	وِزْرَكَ	و ز ر	وزرك	وِزْرَكَ	وزر+ك	wizraka	your burden	وزرك
16921	كعب	☑	78	33	1	وَكَوَاعِبَ	ك ع ب	وكواعب	وَكَوَاعِبَ	كعب+ا	WakawaAAiba	And splendid c	وكواعب
14905	قصر	☑	74	51	3	قَٰصِرَٰتُ	ق ص ر	قصرت	قَٰصِرَٰتُ	قصر+ات+ا	qasiratu	alice?	قصرت
3695	خطر	☑	31	32	21	خَتَّارٍ	خ ت ر	ختار	خَتَّارٍ	ختار	khattarin	traitor	ختار
14926	قسو	☑	57	16	24	قَسَتْ	ق س و	قست	قَسَتْ	قسو+ت	fegeset	so hardened	قست
16915	كظم	☑	68	48	11	مَكْظُومٌ	ك ظ م	مكظوم	مَكْظُومٌ	مكظوم	makhoomun	(was) distress	مكظوم
6489	رغد	☑	16	112	10	رَغَدًا	ر غ د	رغدا	رَغَدًا	رغد	raghadan	(in) abundance,	رغدا
4669	درج	☑	66	44	6	سَنَسْتَدْرِجُهُم	د ر ج	سنستدرجهم	سَنَسْتَدْرِجُهُم	س+ن+درج+هم	sanastarrijuhum	We will progre	سنستدرجهم
17995	لوط	☑	96	10	9	لُوطٍ	ل و ط	لوط	لُوطٍ	لوط	lootin	(of) Lut.	لوط
4679	مدر	☑	71	11	4	مِّدْرَارًا	م د ر	مدرارا	مِّدْرَارًا	مدرار	midraran	(in) abundance	مدرارا
6466	رغب	☑	94	8	3	فَارْغَبْ	ر غ ب	فارغب	فَارْغَبْ	رغب+ا+ف	fargheb	turn your atter	فارغب
20682	وسط	☑	100	5	1	فَوَسَطْنَ	و س ط	فوسطن	فَوَسَطْنَ	ف+ا+وسط+ن	Fawasatna	Then penetrati	فوسطن
4675	درس	☑	68	37	5	تَدْرُسُونَ	د ر س	تدرسون	تَدْرُسُونَ	تدرسون	tadrusoona	you learn,	تدرسون

Figure 9: Snapshot of our suggested lexicon

3. Conclusion

This paper presents a new root generation algorithm for developing Arabic word root, the algorithm is based on permutation being applied to the Arabic alphabet letters (28 letters), the result are all possible Arabic roots. This approach was used by Al Khalil Bin Ahmed in developing his well-known Arabic lexicon "Alain" more than 1200 years ago.

The paper also presents an automated approach for developing all possible Arabic roots, including developing phase, the implementation phase, and then finally presenting the results. Therefore, this way of developing Arabic roots can be recognized as a base approach for dealing with Arabic language processing, understanding, learning, using …etc.

The finding shows that the maximum roots generation algorithm generates 21,924 roots. When these roots are compared with the Quranic dataset roots [18] we found that 1476 of these roots are in common with the Quranic dataset roots. The total number of roots in Quranic dataset that we used is 1613 root which composes 92% of Quran words roots. In addition, the strong roots generation algorithm generates 20,412 roots, 1325 from these roots are in Quranic dataset roots, with percentage of 82%. These results indicate two main conclusions:

- The first is that the results don't cover all Quranic dataset roots and that because most of roots that was not covered in our data contain HAMZA "ء". The algorithm can be enhanced if we considered HAMZA "ء" as the 29th Arabic alphabet. In our suggested algorithms, we can easily add the HAMZA. We can do that by adding it to the array and changing the end of the FOR-LOOPs from 28 to 29 (figure (2) and (3)). Because HAMZA is special character and its came in different shapes, therefore most of Arabic language forms do not consider it as the 29th alphabet letter. Accordingly, we did not include it in this study. This issue can be covered in future work.
- The second is that the generated roots are more than the roots that are included in the holy Quran (according to the dataset we used), and that is the case in Arabic language, because not all Arabic roots are in Quran, according to [6] only 15% of them are in the Quran.

Moreover, the paper presented an algorithm for generating the Arabic morphology by formulating the roots according to Arabic templates. This way of generating Arabic morphology can be used for generating any Arabic word because all Arabic words are generated from roots. The implementation of the algorithm provides correct Arabic words according to the templates that are used.

Finally, the paper shows how the Arabic lexicon can be generate and developed automatically. The paper provides two Arabic lexicons in MS-Access database tables, one

with 21,924 records, and the other with 20,412 records. Each record presents a root with its corresponding words, meaning, English translation ...etc (see figure (9)). In this paper, our concern was mainly to provide an algorithm that can be used to generate Arabic language dictionary including roots and other comprehensive lexicon.

References

[1] Ababneh, Mohamad, et al. "Building an Effective Rule-Based Light Stemmer for Arabic Language to Improve Search Effectiveness." International Arab Journal of Information Technology (IAJIT) 9.4 (2012).

[2] AbdelRaouf, Ashraf, et al. "Arabic Corpus Enhancement using a New Lexicon/Stemming Algorithm." ICPRAM. 2013.

[3] AbdelRaouf, Ashraf, et al. "Building a multi-modal Arabic corpus (MMAC)." International Journal on Document Analysis and Recognition (IJDAR) 13.4 (2010): 285-302.

[4] Abu-Errub, Aymen, et al. "Arabic Roots Extraction Using Morphological Analysis." International Journal of Computer Science 11 (2014): 2.

[5] Algarni, Mohammed, et al. "Simple Arabic Stemmer." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.

[6] Ali Helmi Moussa, 1980 " A Comparison Study Between the Words Roots in Quran and Words Roots in Taj Alaroos" Alam Alfikir, Kuwait. Quran research special edition in 12- 1982.

[7] Al-Kabi, Mohammed N., et al. "A novel root based Arabic stemmer." Journal of King Saud University-Computer and Information Sciences 27.2 (2015): 94-103.

[8] Al-Kabi, Mohammed, Emad Al-Shawakfa, and Izzat Alsmadi. "The Effect of Stemming on Arabic Text Classification: An Empirical Study." Information Retrieval Methods for Multidisciplinary Applications (2013): 207.

[9] Atwan, Jaffar, Masnizah Mohd, and Ghassan Kanaan. "Enhanced arabic information retrieval: Light stemming and stop words." Soft Computing Applications and Intelligent Systems. Springer Berlin Heidelberg, 2013. 219-228.

[10] Bekkali, Mohammed, and Abdelmonaime Lachkar. "ARABIC TWEETS CATEGORIZATION BASED ON ROUGH SET THEORY." International Journal of Computer Science & Information Technology 6.6 (2014).

[11] Boudelaa, Sami, and William D. Marslen-Wilson. "Morphological units in the Arabic mental lexicon." Cognition 81.1 (2001): 65-92.

[12] Darwish, Kareem, Walid Magdy, and Ahmed Mourad. "Language processing for arabic microblog retrieval." Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012.

[13] Haralambous, Yannis, Yassir Elidrissi, and Philippe Lenca. "Arabic Language Text Classification Using Dependency Syntax-Based Feature Selection." arXiv preprint arXiv:1410.4863 (2014).

[14] Harrag, Fouzi, Eyas El-Qawasmah, and Abdul Malik S. Al-Salman. "Stemming as a feature reduction technique for Arabic Text Categorization." Programming and Systems (ISPS), 2011 10th International Symposium on. IEEE, 2011.

[15] Kanan, Tarek, and Edward A. Fox." Automated Arabic Text Classification with P-Stemmer". Machine Learning, and a Tailored News Article Taxonomy. Department of Computer Science, Virginia Polytechnic Institute & State University, 2015.

[16] Khaliq, Bilal, and John Carroll. "Induction of Root and Pattern Lexicon for Unsupervised Morphological Analysis of Arabic." (2013): 1012-1016.

[17] Maamouri, Mohamed, et al. "The penn arabic treebank: Building a large-scale annotated arabic corpus." NEMLAR conference on Arabic language resources and tools. 2004.

[18] Mohamed Osman Hegazi, Anwer Hilal, and Mohammad Alhawarat 2015. "Fine-grained Quran Dataset" International Journal of Advanced Computer Science and Applications(IJACSA), 6(12): 262-267, 2015.

[19] Mohammad Alhawarat, Mohamed Hegazi and Anwer Hilal, 2015. "Processing the Text of the Holy Quran: a Text Mining Study" International Journal of Advanced Computer Science and Applications(IJACSA), 6(2): 262-267, 2015.

[20] Mustafa Ahmed Adel-Alim, 2010, "Arab characteristics between the ancient and modern" United Arab Emirates University,2010

[21] Nehar, Attia, Djelloul Ziadi, and Hadda Cherroun. "Rational Kernels for Arabic Stemming and Text Classification." arXiv preprint arXiv:1502.07504 (2015).

Mohamed Osman Hegazi has got his Ph.D. in Distributed Database from Sudan University of Science & Technology, Khartoum Sudan in 2004. He has worked as an associate professor in Alzaiem Alazhari University, Khartoum Sudan for 4 years, he then joined the College of Computer Engineering and Sciences of Prince Sattam Bin Abdulaziz University since 2013 as an associate professor of computer science.