AN IMPROVED ALGORITHM FOR THE EXTRACTION OF TRILITERAL ARABIC ROOTS

Dr. Raed Kanaan, Assistant Prof.
Management Information Systems Amman Arab University,
Faculty of Computer Sciences and Informatics, Amman, Jordan

Dr. Ghassan Kanaan, Full Prof.
Computer Science Amman Arab University,
Faculty of Computer Sciences and Informatics Amman, Jordan

Abstract

Stemming in the Arabic language is extracting the root form of the verb, removing inflectional affixes and derivational morphemes. Stemming is a share form of language processing in the systems of information retrieval. It is similar to the morphological processing used in natural language processing, but to some extent has different aims. Stemming is used to reduce word forms to common words. Stemming is the process of removing all affixes from a word to extract its root. This paper describes a stemming algorithm that has been developed for the Arabic language. The algorithm utilizes an important morphological aspect of the Arabic language. The algorithm examines the word and extracts its root. It examines the word letter by letter starting from the end of the word, i.e., from the last letter of the word to the first. The algorithm correctly stems most Arabic words that are derived from roots, and achieves high rate of accuracy. The algorithm has been tested on a corpus of 242 abstracts of Arabic documents from the Proceedings of the Saudi Arabian National Conference.

Keywords: Arabic Language; Extraction; Roots; Stemming; Light stemmer

1. Introduction:

1.1 Linguistic Affiliation:

Arabic is a Semitic language of the Arab-Canaanite subgroup (Ruhlen, 1987). It belongs to the Afro-Asiatic family of languages--the bulk of which are spoken in Africa which has several main sections: Semitic (such as Arabic); Berber; Chadic (such as Hausa); and Ancient Egyptian descendent of modern, Coptic, is to maintain the liturgical language. Arabic and Canaanite are distantly related to Aramaic. Other relatives are even more
distant Semitic languages of Ethiopia and Akkadian, an extinct language once spoken in Mesopotamia.

The major dialects of Arabic are Classical Arabic, Eastern Arabic, Western Arabic, and Maltese. A modern form of Classical Arabic referred to as Modern Standard Arabic is used in writing Arabic today. Eastern Arabic, sometimes called Mesopotamian Arabic, includes the Arabic dialects spoken in several countries such as Egypt, Sudan, Syria, Iraq, Arabian Peninsula, and the Arabic speaking communities in Asia (Bateson, 1967).

1.2 Orthography

Arabic language uses an alphabetic system that normally represented by symbols; consonants and long vowels. In addition, there is a close match between the written symbols and their linguistic function. Short vowels, however, are not written despite the fact that much morphological and grammatical meaning is signaled by vowels. Due to the fact that only roots and stems of an inflected word are written, therefore the reader has to infer its particular meaning from context. When vowels are represented, as in children's books or learners' manuals, super- and subscript diacritics are used. Arabic language is written from right to left (UCLA, 2013).

1.3 Linguistic Sketch

Modern Standard Arabic (MSA) has a grammatical system known as a "root and pattern system." Words are composed of roots and patterns. Roots consist of three consonants, however a few have four or five; the roots, unpronounceable of itself, are allied with a general meaning, therefore the sequence ktb has an association with the meaning "writing." Patterns are vowel sequences, which can be mediated on as templates, (sometimes as prefixes and suffixes, and sometimes with additional consonants). Patterns are then "added" to, or even within roots of the word following well-defined models. For instance, consider the root d-r-s. Despite the letters (i.e., consonants) drs will always remain the same, the following are examples to confirm that the scheme and vocalization will change depending upon usage (Semitic Languages, 2013):

darasa, "to study, learn" "he studied" (the third personal singular perfect form is the reference form for the verb).  
darrasa, "to teach" "he taught"  
durs, "lesson, "lessons"  
mudaaris, "teacher (male)"/mudarrisa, "teacher (female)"  
madrasa, "school"
These patterns then produce various nominal and verbal stems, which have a variety of functions; in nouns for example, they imply habitual occupations, diminutives, or colors, and in verbs, they form participles, causatives, and passives.

Nouns are inflected and morphologically marked for case (nominative, genitive, and accusative), gender (masculine and feminine), number (singular, plural, dual, and collective) and determination (definite and indefinite). Plural forms of many nouns are significant by ablauts, that is, “the vowel pattern within a root varies between singular and plural forms, similar to alternations in English like those in the verb sing, sang, and sung, or the noun mouse and mice” (Touregypt, 2013).

In verbs, which occur in two basic stems, the perfect and imperfective, person, number, mood, and aspect are marked by prefixes and suffixes. Templates for verbs consists of ten commonly, (also of four rarely) used shapes and meanings. Their meanings reveal verbs that relate intensity, repetition, causation, intention, and belief. In addition to the nominal and verbal systems there is another system of particles. Particles include such things as function words, which express syntactic relationships, for example, interrogatives, prepositions, conjunctions, and pronouns (Touregypt, 2013). Compared to the root-pattern system of other word categories these are quite simple in their formation (Al-Fedaghi, & Al-Anzi, 1989; Al-Fedaghi, and Yaseen, 1990; Ali, et.al, 1984; Beesley, 1989).

2. Related Work:

According to Al-Nashashibi, et al. (2010) he suggested a new technique for root extraction as a pre-processing step in the Arabic text mining. They claimed that the available approaches in the literature does not tackling the elimination of long vowels, geminated, and hamzated. Therefore, they proposed an algorithm to handle such issues. This algorithm improved the accuracy by 14%. However, Beesley (2001) and El-Sadany and Hashish (1989) handled the elimination of long vowel words.

Khoja (1999) proposed a light stemmer to handle weak and geminated words; however, Al-Nashashibi, et al. (2010) further asserted that none of the available approaches handling weak, geminated, hamzated and eliminating long vowels. As a result, Al-Nashashibi, et al. (2010) benefited from Al-Ameed (2006) linguistic approach and improved the accuracy by 14%.

To support the reason behind conducting this research, Aljlayl, and Frieder (2002) light stemmer for instant remove the most frequent prefixes and suffixes from the words instead of the prefix and/or suffix list must be removed from the selected words, hence our proposed algorithm.
The work of Al-Ameed et al. (2005) or what so called TREC-2001, enhanced the performance of Larkey’s (2002) light8 stemmer in two ways. In one hand, by changing the sequence of the components of algorithm execution. On the other hand, by additionally adding new affixes to the already existing ones. He claims that the prefixes list contains 17 two characters, however the only found is 15.

Apparantly, the above work can be considered as a development of Darwish and Oard (2002) stemmer. The below prefixes have been removed by his stemmer:

In addition to the following suffixes:

Below section describes the proposed algorithm.

3. Algorithm Description:

We start by examining the length of the word. If it is a one or two letter word, it is probably a particle; it is treated as a stop word and the algorithm returns it as is. If the word consists of three letters we accept it as a root and do not process it further. Every letter is checked separately to determine whether it is considered additional (احرف زيادة) or not. These letters are:

The rest of the letters are considered original.

The first step is to find the stem by deleting the definite article “ال” and what precedes it in addition to removing the letters “ة” “ه” “م” from the end. Then we examine the length of the word, i.e., if it is three letters long, then it is a root, otherwise we continue.

If the first letter is "ل" we delete it and if the first two letters are "ال" we delete them too. If the second letter is "س" preceded by one of the following letters:

we delete the three letters, if the length of the word is more than five letters.

If "س" is the first letter followed by "ي""ت" or "ت" and the word length is more than four, these letters are deleted. If the last letter is "ي" or "ة" it is changed to "ه".

And we start implementation to apply the main part of the algorithm:

We examine the last letter, if we have two similar letters, with a vowel between them; we delete the vowel, and add the two letters to the root.
As for the preposition "ب" we always delete it except in two cases: if it is followed by the letter "ي" or "ل". We also remove the letter that precedes it, then continue till the first letter, i.e., until the length of the word becomes equal to zero. The processes that are applied on letters are as follows:

Let us take the word مدرس as an example. When we delete its additional letters it becomes مدرس so we take the last letter مدرس and find it an original letter, we examine the next letter "ر" and find it also an original letter, then we examine "م" to find it also original, but when we examine "م" we find it to be an additional or extra letter so the root is مدرس.

Note: if "يا" or "لي" are found, they are changed into "لي"

**Letter "ه"**

If preceded by "س" and followed by "ل", the root is سل.

If preceded by "ه" they are deleted ساقوم= قام.

If it was the initial letter, it is added to the list of root characters; otherwise, it is not original.

**Letter "لم"**

If preceded by "س" and followed by "ل", the root is سل as in:

If the length of the root is more than five letters, and the number of original letters is more than two and preceded by "س" they are deleted سير= ساسير=سار.

If it is a word initial character or in second position, it is deleted, but if it is in the third position, it is listed.

If it is a postposition and proceeded by "هم" or "كم" we delete all three letters.

If the word length equals the stem word length and "لم" is preceded by one of the following letters درسنا و درسوا = درس و ملكها = ملك، "ن" "ه" "و" "ت" درستنا و درستنا و

If it does not match one of these cases, it is listed.

**Letter "ت"**

If "س" is the second letter, preceded by one of these letters "ه" "ك" "ن" "ت" "ن" "ي" "م" "ل" and followed by "ت" we deleted all three of them since the word length exceeds five letters. مستعمل، نستعمل، نستعمل، يستعمل.

If "س" is in initial position, followed by "ات", and the word is more than four letters, we delete both characters. ستمسهم= نهم.
If we have the two letters "ت" or "ب", we delete the last one and keep the first if the word length is four, but if it is more, we delete the two.

If it is the last letter, and it is preceded by "" or "" or "" or "", we delete both letters.

If "ت" is in initial position, and the number of original letters is greater than zero, we delete "ت" and retrieve the last letter that was listed except "" or "" or "".

or otherwise we delete "" or "" or "".

If "ت" is in third position, and the initial character is "" or "" or "", we delete both the first and the third character and keep the second 

Letters "ﱴ" and "ﺉ"

If one of them occurs in the initial position, it is changed into "ﱴ".

If we find "ا" or "ا" or "ا", we turn them into "ا" or "ا", when the length of the word is four letters.

If the word consists of five letters, we turn "ا" or "ا" or "ا", into "ا" or "ا".

In addition, we add it to the list. Otherwise, it is original.

Letter "ي"

If "س" occurs as the second letter, preceded by one of the following letters "ن" or "م", "ي", "ت", "ل" followed by one by "ي", we deleted the three of them if the word is more than five letters.

If "س" is in initial position followed by "ي" and the word is more than four letters, we delete both letters if the word is more than five letters.

If it occurs as the second letter, preceded by "" or "" we delete "ي" and keep "ي"

if is in initial position and there are only two original letters or fewer, and the list is nonempty, we delete "ي" and retrieve the last letter added except for "".

Letter "د"

If it is preceded by "ل" then it is an original letter.

If "و" is in initial position, and there are two original letters or fewer, and the list is nonempty, we delete "و" and retrieve the last letter added to the list except for "".

if occurs in third position, and "و" is the initial letter, we delete the third position and keep the two letters remaining, otherwise they are listed.
If it is in initial position and there are two original letters or fewer, and the list is empty, we keep the {ورد=ورد}.

**Letter "م"**

If "م" is in initial position, and there are two original letters or fewer, and the list is non empty, we delete "م" and retrieve the last letter added to the list.

{موعد=وقت، موعد=وعد}.

If it is initial position, and there are fewer than three original letters the list is empty, then it is an original letter.

If it is in final position in the base word, preceded by "ه" or "ك" but not followed by a letter in the original word, we delete the last two letters: "هم"

If the number of letters in the base word is greater than or equal to five letters. {كتبتكم، كتبكم = كتاب}. If it occurs initially followed by "ال"، the word followed the base {مالية=مال} otherwise it is original.

**Letter {ن}**

If there are more than five letters in the base word, and "ن" is the final letter preceded by a vowel, we delete both "ن" and the vowel. {درس=درس}

If "ن" is initial position, and there are fewer than three original letters, and the list is nonempty, we delete "ن" and retrieve the last letter added to the list so long as it is not preceded by "ي"،"ن"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"，"ي"،"و"،"ت"،"و"،"ي"،"و"，"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"،"ي"،"و"،"ت"،"و"，"ي"،"و"，"ي"،"و"،"ي"،"و"،"ي"،"و"，"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"，"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"，"ي"،"و"،"ي"،"و"،"ي"，"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"،"و"，"ي"،"و"،"ي"،"و"،"ي"،"و"،"ي"，"و"他表示

**Post Algorithm**

If there are fewer than three original letters
If there are two original letters, we retrieve the last letter in the list.
If there is only one original letter, we retrieve the last letter in the list
If there are no original letters, we retrieve the last three letters in the list.

**Sorting**

We arrange the letters in order of their appearance in the original word.
If there are four letters, we re-examine the resulting word as a new word if it is the same, we examine the first letter, so if it is one of the following letters "ل"،"ن"،"ب" we delete it, and otherwise the word has a quadruple root.

If there are two letters in the word, we examine the first letter, if it is "ت" or "ل" we retrieve it, otherwise we report it as the root as it is.

### Before Print
- If "س" it found, it is turned into "س".
- If the first letter is "ل" or "س", it is changed into "س"، on condition that it is not followed by a vowel.
- If the second letter is "ل" it is turned into "س", except when "ب" is initial or the third is "ل".

### 4. Results:

<table>
<thead>
<tr>
<th>Number</th>
<th>Number of words on text</th>
<th>Percentage result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>113</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>98.5%</td>
</tr>
<tr>
<td>3</td>
<td>550</td>
<td>96.6%</td>
</tr>
<tr>
<td>4</td>
<td>650</td>
<td>95.2%</td>
</tr>
</tbody>
</table>

The percentage of the correct words in the four texts =97.6%

### Problems And Weaknesses
- The existence of "ت" in the end of the word, either original or not.
- The existence of "ب" as extra letters the beginning of the words.

### 5. Conclusion
In this paper an Arabic Stemming Algorithm have been designed and implemented, which has been developed for the Arabic language. The algorithm utilizes an important morphological aspect of the Arabic language. The algorithm was implemented in Vbasic. Sample output of the program is shown in Figure 1. We have tested the algorithm with sample data from the Proceedings of the Saudi Arabian National Computer Conferences with a total of 3500 words, the algorithm runs very well and achieves an accuracy rate that reached 97.6%. The words that the algorithm failed to analyze are foreign names and proper nouns.
References: