ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 3, 111-115 (2017)

www.astesj.com

Special Issue on Recent Advances in Engineering Systems

ASTESJ

ISSN: 2415-6698

Paper Improving Rule Based Stemmers to Solve Some Special Cases of Arabic Language

Soufiane Farrah[*], Hanane El Manssouri, Ziyati Elhoussaine, Mohamed Ouzzif

RITM Laboratory, IT Department, EST - ENSEM, Casablanca, Morocco

A R T I C L E I N F O

Article history:
Received: 24 February, 2017
Accepted: 03 April, 2017
Online: 15 April, 2017

Keywords :
Rule Based
Arabic text
Stemming

A B S T R A C T

Analysis of Arabic language has become a necessity because of its big evolution; we propose in this paper a rule based extraction method of Arabic text to solve some weaknesses founded on previous research works. Our approach is divided on preprocessing phase, on which we proceed to the tokenization of the text, and formatting it by removing any punctuation, diacritics and non-letter characters. Treatment phase based on the elimination of several sets of affixes (diacritics, prefixes, and suffixes), and on the application of several patterns. A check phase that verifies if the root extracted is correct, by searching the result in root dictionaries.

1. Introduction and Related Works

Arabic is a Semitic language spoken by more than 400 million people as a native language and ranked at the seventh position of Internet users in 2010. However, the task of performing the retrieve of information of Arabic language is very problematic, because of many aspects, such as: polysemy, irregular and inflected derived forms, various spelling of certain words, various writing of certain combination character, short vowels (diacritics) and long vowels, and the spectacular availability of affixes in the Arabic words [1, 2]. Different methods and approaches have been introduced to retrieve Arabic information [2, 3, 4, 5, 6].

To study Arabic morphology effectively, we divide words in Arabic into three self-contained categories as follows:

- ✓ إسْم: It includes nouns, pronouns, adjectives, adverbs, etc
- ✓ فِعْل: Verbs
- ✓ حَرْف: Particles, articles, and conjunctions

Particles are completely unpredictable; they don't fall into the templatic system (i.e. they have no patterns) nor do they undergo any morphophonemic changes. They are what they are and must be memorized. The up side is that there are relatively few of them in the language – within one hundred.

Nouns and verbs do fall into the templatic system and have very systematic morphophonemic rules that govern them. This includes the study of how verbs are conjugated, how they move from pattern to pattern to enhance their meanings, how the participles and other nouns are derived, how nouns pluralize, etc.

Each declinable noun and each verb is made up of a certain set of base letters, called its root: جذر. (Nouns that are always indeclinable (such as pronouns) usually don't follow this system).

Verbs can have either 3 base letters, or 4. Nouns can have 3, 4 or 5. Now these base letters can be augmented with extra letters, and they can be dropped or changed due to morphophonemic rules as well.

Particles - The third part of speech in Arabic mentioned above is the particle. The meaning of a particle is often understood in the context of the sentence and words before and after the particle. The sign of the particle is that it does not accept the signs of nouns or verbs.

Analyzing Arabic text was treated by many researchers, all of them tried to extract an exact root or stem from a word, there is two ways to treat a text; morphological analyze, which consist to find roots, and there are statistical stemmers that group word variants using clustering techniques.

The first approach of morphological analyze is manually constructed dictionaries based on roots, Kharashi and Evans worked with small text collections, for which they manually built dictionaries of roots for each word to be indexed [7]. Tim

[*]Corresponding Author : Soufiane FARRAH, 46 Rue 17 Hay Anigret, No +212663782432, Email: f.soufiane@gmail.com

Buckwalter developed a set of lexicons of Arabic stems, prefixes, and suffixes, with truth tables indicating legal combination [1].

Nehar [5] and Taghva [8] introduces new stemming techniques that do not rely on any dictionary, the first one is based on the use of transducers. Nehar [5] proposed also a heavy stemmer that does not use any dictionary of roots. Khoja and R. Garside [2] developed a dictionary-based stemmer, and Larkey [5] developed a Java program based on their own Arabic stemmer that will develop and evolve to take in count some nouns and verbs categories described in the previous paragraph. Taghva [8] proposed IRSI Arabic Stemmer Algorithm, which does not use a Root Dictionary. ISRI stemmer per-forms better than the other approaches on the shorter title queries. For the long texts and narrative queries, stemming made a difference: the Khoja, ISRI, and Light stemmers were significantly better than not stemming. Ghwanmeh [9] presents an Arabic root-based algorithm based on patterns. This stemmer is restricted to native Arabic words that consist of four or more Arabic alphabets.

All algorithms mentioned before have some weaknesses. In this paper, we will prove that the best way of stemming is the one that have a strong preprocessing phase, and it is based on both "patterns check" and "root list".

This paper is an extension of work originally presented in conference [10].

In fact, we present the weakness in Heavy and Light Stemming Algorithms and we try to propose some new solutions for each point treated, then we will compare results of our new stemmer with other ones.

In section II, we present the different areas for improvement in Arabic text classification, in section III we present our approach, and in section IV we present some tested examples and, compare our solution with others.

2. Improvement Areas in Arabic Classification

Light stemming algorithms removes suffixes, and prefixes from words, producing a form of word called "stem" [11], there was many versions of the light stemming algorithms and the last one is light10 [12]. This algorithm after removing punctuation and non-letters, diacritics, Hamza from letter "أ", he replaces final letter "ة" with letter "ه" and then replace final letter "ى" with letter "ي". After that, the algorithm search in irregular word list to find out if the word exists on this table or not. Then the algorithm removes the letter "و" from the beginning of the words if the length of the word is more than three characters, because it considers that this letter is usually a conjunction.

This step generates several errors on stem extraction, I give below some examples:

Word	Stem extracted by Light10
وَبِيل	بِيل
وُجُوب	جوب
وَرِيث	رِيث

Table 1. Morphology of Arabic Word

As we can see, when removing the letter "و" from the beginning of those words, we change the meaning of the word, for the first word "وَبِيل" it means calamitous; disastrous, and when we remove the letter "و" the word means torch.

In Khoja's Approach, and TC system proposed by M.Hadni, A.Lachkar and S. Alaoui Ouatik in [13], and also in Mohammed N. Al-Kabi who proposed evolution of Khoja's algorithm [14], we find this same issue, so in our algorithm we will take care of this point and we propose to check if the word doesn't exist in the list of words that begins by 'و', and then remove diacritic 'و' (primarily weak vowels), this list is constituted manually and must be maintained regarding the evolution of Arabic language.

The second point we have improved is removing the letter "أ", in Light Approach, Khoja's Approach and M.Hadni's one, this letter is deleted because it is considered as a prefix. The issue is when this letter is a part of word as for the word "أباح" which means "permit", and when we delete it the word means "confide", to solve that, we built a list of words that starts with letter "أ".

The third point we involve in this paper is the stemming of five nouns (أب, أخ, حمو, فو, ذُو), those nouns are excluded from the other single nouns per the syntactic case. They have other marks to indicate them syntactic cases that the other doesn't have. The single noun always depends on rules to indicate its syntactic cases but the five nouns are contradicting those rules. The five nouns aren't depending on al Harakat (vowelization on system) rather than the letters. They have preconditions to be different from other single nouns:

- It has to be adjunct to another noun in other words there must be a noun after it that is genitive noun.
- The noun after must not be (ي) that indicates the speaker [15].

Therefore, for our algorithm, we handle the fives nouns separately.

It is true the orthography in Arabic is less ambiguous and more phonetic with the use of diacritics. For example, a word can be written using the same characters and be pronounced differently. The main purpose of diacritics including vowel marks, known as Harakat a phonetic is to provide, "حركات" aid to show the correct pronunciation. Arabic vowel marks include Fatha "فتحة", Kasra "كسرة", Damma "ضمة", Sukun "سكون", Shadda "شدة" and Tanwin "تنوين". The pronunciation of these vowel marks are represented in Table 2 below:

Double Constant	No Vowel	Tanwin			Vowel		
ـّ shadda	ـْ Sukun	ـٍ in	ـٌ un	ـً an	ـِ Kasrah	ـُ Ḍammah	ـَ Fatḥah

Table 2. Arabic Diacritics

However, in Modem Standard Arabic (MSA), vowel marks are not usually included in printed and electronic text, and the understanding and correct pronunciation of the word is determined within its context by the reader, so we decide not to remove (if it exists), the vowels as a step on preprocessing phase.

3. Important Steps

The method we propose is based on preprocessing step, treatment and check steps, here is a description of each one:

3.1. Preprocessing

In this step we proceed to:

- Divide text into words
- Format the word by removing any punctuation, diacritics and non-letter characters
- Check if the word is a stop word (3)

(3) This step consists on eliminating (very frequent) words that contain no or little information to help discriminate the text they occur in. A large list of stop words are used (Table 3).

إن	أنما	إنما	بعد	ضد
أحد	إلي	في	من	حتى
تلك	كذلك	التي	وهو	به
لكن	عن	منذ	على	وكان

Table 3 Stop Word

3.2. Searching in strange words list

In this step our algorithm will check if the word is a part of strange words (it is a word that comes from another language than Arabic, and used in the modern Arabic language especially), those words exists in a list of Strange words constituted manually (Table 4).

ألمانيا	فرنسا	ديسمبر	أوروبا	إفريقيا
اكلنيكي	ميكانيك	ناهوند	أوباما	بنكيران

Table 4 Strange Words

If the word exists the algorithm returns the word, otherwise the treatment continues.

3.3. Check if the word exists in the list of words that begins by "و":

The stemmer removes letter "و" ("and") from the beginning of the words if the length of the word is more than three characters, and if the word doesn't exist in the list of "Words_begins_by_Waw.txt", because many common Arabic words begin with this character.

ومي	وقر	وصب	وخط	وخض
وهث	وكز	وعع	وغي	وغف

Table 5 Words starting with *"و"*

3.4. Check if the word exists in the list of words that begins by "ال"

The stemmer removes letter "ال" from the beginning of the word if the length of the word is more than three characters, and if the word doesn't exist in the list of "Words_begins_by_AL" (table 6), because many common Arabic words begin with this character.

أتم	أبط	أبض	أبر	أبب
أول	أوض	أوش	أمس	أذي

Table 6 Words starting with "ال"

3.5. Normalization

The third step in the stemmer is normalization of the words. Normalization process in the proposed stemmer is the similar to the normalization process in Light10 stemmer which runs as following:

- Remove Hamza from letter "أ" (Replace " آ إ أ" with "ا")
- Replace final letter "ة" with "ه".
- Replace final letter "ى" with "ي".

3.6. Removing prefixes and suffixes

This step consists of removing the prefixes and suffixes from the words

س	و	ب	ل	لل
وا	ها	كما	تما	هما
نا	كن	تن	كم	تم
هم	هن	ين	ون	تا

Table 7 Prefixes and Suffixes

3.7. Check if the word matches any of the patterns

The last step after deleting the prefixes and suffixes of the words is correcting any word that its meaning changed. In some cases, a letter in the pattern of the word is deleted affecting the process of root extraction, like in the word.

In (فعل) which is the pattern of (رأى), the letter "أ" is deleted, and in the present tense of pattern (فعل) is (يفعل) which is introduced to (يرأى) and not (يرى). If we take (يرى) as a present verb, the past will be (رى) so the letter 'ع ' is deleted for this reason becoming (يفل) instead of (يفعل).

There are three rules which apply in the stemmer for correcting some words their meaning was affected:

1. Adding "ي" to the end of the word if the suffix "يه" is deleted
2. Adding "ه" to the end of the word if the suffix "ته" is deleted
3. Replacing the letter "ئ" to the end of the word by "ء" if the suffix of the word is deleted.

4. Detailed Algorithm

To implement our algorithm, we have used Java code program, based on Khoja's one.

These are the schema of our algorithm:

```
{
  // check if the word consists of two letters:
  if ( word.length ( ) == 2 )
    if the word consists of two letters, we treat two cases:
- A root consisting of two letters (though I can't think of any!)
- A letter was deleted as it is duplicated or a weak middle or last letter.
  // if the word consists of three letters
  if( word.length ( ) == 3 && !rootFound )
- If the last letter is a weak letter or a hamza, then check for last week words list.
- If the second letter is a weak letter then check for second week words list.
```

```
        // if the word consists of four letters
        if( word.length ( ) == 4 )
            // check if it's a root
    -   Check on the list of four letters root.

        // if the root hasn't yet been found
        if( !rootFound )
        {
            // check if the word is a pattern
    -   Try and find a pattern that matches the word
        }
        // if the root still hasn't been found
        if ( !rootFound )
        {
            // check for a definite article, and remove it
            word = checkDefiniteArticle ( word );
    -   look through the vector of definite articles search through
each definite article, and try and find a match
    -   Check to see if the word is a root of three or four letters
    -   If the word has only two letters, test to see if one was
removed
    -   If the root hasn't been found, check for patterns/ check for
suffixes/ prefixes
        }
        // if the root still hasn't been found
        if ( !rootFound && !stopwordFound )
        {
            // check for the prefix waw
            word = checkPrefixWaw ( word );
    -   Check to see if the word is a stopword
    -   Check to see if the word is a root of three or four letters,
that begin by 'waw'
    -   If the word has only two letters, test to see if one was
removed
    -   if the root hasn't been found, check for patterns
    -   Check for suffixes
    -   check for prefixes
        }
        // if the root STILL hasnt' been found
        if ( !rootFound && !stopwordFound )
        {
            // check for suffixes
            word = checkForSuffixes ( word );
        }
        // if the root STILL hasn't been found
        if ( !rootFound && !stopwordFound )
        {
            // check for prefixes
            word = checkForPrefixes ( word );
    -   Check to see if the word is a stopword
    -   Check to see if the word is a root of three or four letters.
    -   If the word has only two letters, test to see if one was
removed
    -   if the root hasn't been found, check for patterns
    -   Check for suffixes
    -   check for prefixes
        }
        return word;
    }
```

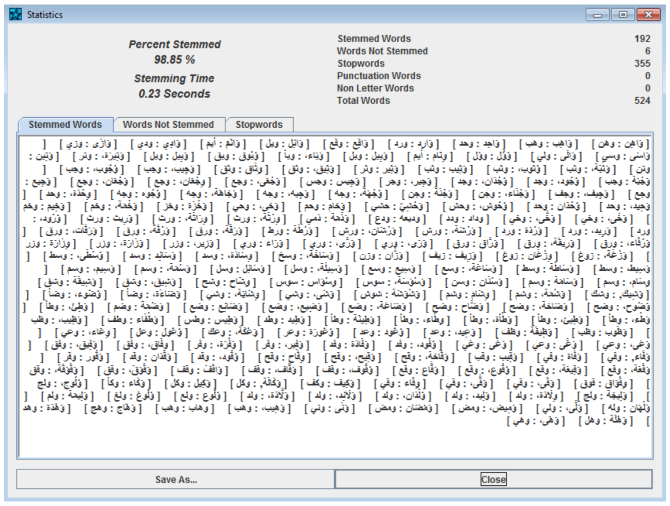
5. Results

The stemming result of the word will be correct, if the output form of the word is the same as the target form of the word. Otherwise, the result of the word will be incorrect. We have used Khoja's java code and we applied our approach on it. This program take in input a text file and returns in output a list of words theirs stems and the type of the each word. We have used as a first test a list of 524 words that begins by "waw" letter, we give below results of our stemmer comparing it to Khoja's one.

	Khoja's Stemmer	Our Stemmer : EST.Stemmer
Number of words used	524	534
Stemmed words	96,75 %	98,85 %
Not Stemmed words	3,25 %	1,15 %

Table 8 Results Comparison

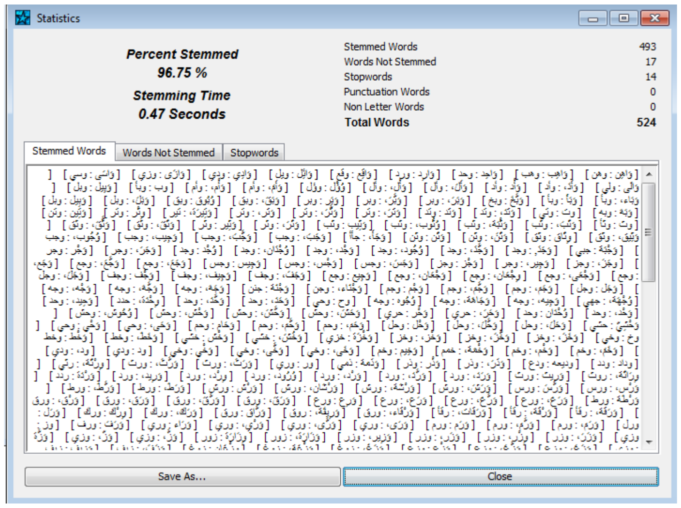
Figures below shows the running result of both stemmers:



```
Percent Stemmed          Stemmed Words          192
98.85 %                  Words Not Stemmed        6
Stemming Time            Stopwords              355
0.23 Seconds             Punctuation Words        0
                         Non Letter Words         0
                         Total Words            524
```

```
[ وَاهِن : وهن ] [ وَاهِب : وهب ] [ وَاجِد : وجد ] [ وَارِد : ورد ] [ وَاقِع : وقع ] [ وَائِل : وبل ] [ وَائم : أيم ] [ وَادِي : ودي ] [ وَازَى : وزي ] [
وَاسَى : وسي ] [ وَالَى : ولي ] [ وَوُل : وول ] [ وِئام : أيم ] [ وَبِيل : وبل ] [ وَبَاء، : وبأ ] [ وُبُوق : وبق ] [ وَبِيل : وبل ] [ وَتِيرَة، : وتر ] [ وَتِين :
وتن ] [ وَثِيَة، : وثب ] [ وُثُوب، : وثب ] [ وَثِيب : وثب ] [ وَثِير : وثر ] [ وَثِيق، : وثق ] [ وِثَاق : وثق ] [ وَجِيب، : وجب ] [ وُجُوب، : وجب ] [
وَجْبَة : وجب ] [ وُجُود، : وجد ] [ وُجْدَان، : وجد ] [ وَجِير، : وجر ] [ وَجِيس : وجس ] [ وَجْعَى، : وجع ] [ وَجْعَان، : وجع ] [ وَجْعَان، : وجع ] [ وَجِيع :
وجع ] [ وَجِيف، : وجف ] [ وِجْنَاء، : وجن ] [ وَجْنَة : وجن ] [ وُجْهَة، : وجه ] [ وَجِيه، : وجه ] [ وَجَاهَة، : وجه ] [ وُجُوه : وجه ] [ وِحْدَة، : وحد ] [
وَحِيد، : وحد ] [ وُحْدَان : وحد ] [ وُحُوش، : وحش ] [ وَحْشِيّ : حشي ] [ وَخَام : وحم ] [ وَحْي، : وحي ] [ وَخْزَة : وخز ] [ وَخْمَة، : وخم ] [ وَخِيم : وخم
] [ وَخْي، : وخي ] [ وَخْي، : وخي ] [ وداد : ودد ] [ وديعه : ودع ] [ وَذَمَة : ذمي ] [ وِرْثَة، : ورث ] [ وِرَاثَة، : ورث ] [ وَرِيث : ورث ] [ وُرُود، :
ورد ] [ وَرِيد، : ورد ] [ وَرْدَة : ورد ] [ وَرْشَة، : ورش ] [ وَرْشَان، : ورش ] [ وَرْطَة : ورط ] [ وَرَقَة، : ورق ] [ وَرَقَة، : ورق ] [ وَرَقَات، : ورق ] [
وَرْقَاء، : ورق ] [ وَرِيقَة، : ورق ] [ وَرَّاق : ورق ] [ وَرَى، : وري ] [ وَرَى، : وري ] [ وَرَاء : وري ] [ وَزِير، : وزر ] [ وَزَارَة، : وزر ] [ وِزَارَة : وزر
] [ وَزَغَة، : زوغ ] [ وِزْغَان : زوغ ] [ وَزِيف : زيف ] [ وَزَّان : وزن ] [ وَسَاخَة، : وسخ ] [ وِسَادَة، : وسد ] [ وَسَائِد : وسد ] [ وُسْطَى، : وسط ] [
وَسِيط، : وسط ] [ وَسَاطَة : وسط ] [ وَسَاعَة، : وسع ] [ وَسِيع : وسع ] [ وَسِيلَة، : وسل ] [ وَسَائِل : وسل ] [ وَسْمَة، : وسم ] [ وَسِيم، : وسم ] [
وِسَام، : وسم ] [ وَسَامَة : وسم ] [ وَسْنَان : وسن ] [ وَسْوَسَة، : سوس ] [ وَسْوَاس : سوس ] [ وِشَاح : وشح ] [ وَشِيق، : وشق ] [ وَشِيقَة : وشق
] [ وَشِيك، : وشك ] [ وَشْمَة، : وشم ] [ وِشَام : وشم ] [ وَشْوَشَة : شوش ] [ وَشْي، : وشي ] [ وِشَايَة، : وشي ] [ وَضَاءَة، : وضأ ] [ وَضُوء، : وضأ ] [
وَضُوح، : وضح ] [ وَضَاحَة، : وضح ] [ وَضَّاح : وضح ] [ وَضَاعَة، : وضع ] [ وَضِيع، : وضع ] [ وَضَائِع : وضع ] [ وَضْمَة : وضم ] [ وَطِئ، : وطأ ] [
وَطْء، : وطأ ] [ وَطِيئ، : وطأ ] [ وَطْأَة، : وطأ ] [ وِطَاء، : وطأ ] [ وَطِيئَة : وطأ ] [ وَطِيد : وطد ] [ وَطِيس : وطس ] [ وَطْفَاء : وطف ] [ وَظِيب، : وظب
] [ وُظُوب : وظب ] [ وَظِيفَة : وظف ] [ وَعِيد، : وعد ] [ وُعُود : وعد ] [ وَعُورَة : وعر ] [ وَعْكَة، : وعك ] [ وَعُول : وعل ] [ وِعَاء، : وعي ] [
وَعْي، : وعي ] [ وَغْى : وعي ] [ وَغَى : وغي ] [ وُفُود، : وفد ] [ وِفَادَة : وفد ] [ وَفِير، : وفر ] [ وَفْرَة، : وفر ] [ وِفَاق، : وفق ] [ وَفِيق، : وفق ] [
وَفَاء، : وفي ] [ وَفَاة : وفي ] [ وَقِيب : وقب ] [ وَقَاحَة، : وقح ] [ وَقِيح، : وقح ] [ وَقَّاح : وقح ] [ وُقُود، : وقد ] [ وُقْدَان : وقد ] [ وُقُور : وقر ] [
وَقْعَة، : وقع ] [ وَقِيعَة، : وقع ] [ وُقُوع، : وقع ] [ وَقَّاع : وقع ] [ وُقُوف، : وقف ] [ وِقَاف، : وقف ] [ وَاقِف : وقف ] [ وُقُوق، : وقق ] [ وَقْوَقَة، : وقق
] [ وَقْوَاق : قوق ] [ وُقِي، : وقي ] [ وُقِي، : وقي ] [ وِقَاء : وقي ] [ وَكِيف : وكف ] [ وَكَالَة، : وكل ] [ وَكِيل : وكل ] [ وَكَاء : وكأ ] [ وُلُوج، : ولج
] [ وَلِيجَة : ولج ] [ وِلَادَة، : ولد ] [ وَلِيد، : ولد ] [ وِلْدَان، : ولد ] [ وَلَائِد : ولد ] [ وِلَادَة، : ولد ] [ وَلُوع : ولع ] [ وَلُوغ : ولغ ] [ وَلِيمَة : ولم ] [
وَلْهَان : وله ] [ وَلِي، : ولي ] [ وَمِيض، : ومض ] [ وَمَضَان : ومض ] [ وَنَى : وني ] [ وَهِيب، : وهب ] [ وَهَّاب : وهب ] [ وَهَّاج : وهج ] [ وَهْدَة : وهد
] [ وَهْلَة : وهل ] [ وَهِي، : وهي ]
```

Figure 1 EST.Stemmer



Percent Stemmed	96.75 %
Stemming Time	0.47 Seconds
Stemmed Words	493
Words Not Stemmed	17
Stopwords	14
Punctuation Words	0
Non Letter Words	0
Total Words	524

[illegible]

Figure 2 Khoja's stemmer

A second test that we have done with an article that contained 1418 words, results are presented in the table below:

	Khoja's Stemmer	Our Stemmer: EST.Stemmer
Number of words used	1418	1418
Stemmed words	93,16 %	93,23 %
Not Stemmed words	6,84 %	6,77 %

Table 9 Results Comparison

Since the text didn't contain words that begins by "waw", some strange words, and five nouns, the difference between the two results is not large.

EST.Stemmer is able solve some cases, for example if we take وَسَامَة ، وَسْمَة.

With Khoja's stemmer we have as stemmed text:

Word	Stem	Type
وَسْمَة	سمي	ROOT
وَسَامَة	سوم	ROOT

Table 10 Results Example

With our algorithm EST.Stemmer:

Word	Stem	Type
وَسْمَة	وسم	ROOT
وَسَامَة	وسم	ROOT

Table 11 Results Example

We have also review and modify stop word list to solve some issues detected is our tests, for example we have added 'ففي' and 'منهم' in this list.

6. Conclusion

In this paper we proposed new methods to improve the detection of the stem for Arabic language. Indeed, specific cases related to the five nouns and words starting with a vowels are processed successfully by the algorithm.

In the future work we will test the accuracy of our algorithm and compare it with Light and Heavy stemmers.

References

[1] Buckwalter,T.Qamus: Arabic lexicography http://members.aol.com/ArabicLexicons/

[2] Khoja, S.and Garside, R. Stemming Arabic Text. Computing Department, Lancaster University, Lancaster. Internet Home Page, 1-7 (1999). [Available at:http://www.comp.lancs. ac.uk/computing/users/khoja/stemer.ps].

[3] H. AlSerhan, R. A. Shalabi, and G. Kannan, "New approach for extracting arabic roots," in Proceedings of The 2003 Arab conference on Information Technology (ACIT 2003), Alexandria, Egypt, Dec. 2003, pp. 42–59.

[4] Nehar, Attia, Djelloul Ziadi, Hadda Cherroun, and Younes Guellouma. "An Efficient Stemming for Arabic Text Classification." In *Innovations in Information Technology (IIT), 2012 International Conference on*, 328–32. IEEE, 2012.

[5] Larkey, Leah S., et Margaret E. Connell. « Arabic Information Retrieval at UMass in TREC-10. » In *TREC*, 2001.

[6] Saudagar, Abdul Khader Jilani, Habeeb Vulla Mohammed, Kamran Iqbal, et Yasir Javed Gyani. « Efficient Arabic text extraction and recognition using thinning and dataset comparison technique ». In *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*, 1–5. IEEE, 2015.

[7] Al-Kharashi, I. A. and Evans, 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 (JASIS) 45(8), pp 548-560.

[8] Taghva, Kazem, Rania Elkhoury, et Jeffrey S. Coombs. « Arabic Stemming Without A Root Dictionary. » In *ITCC (1)*, 152–157, 2005.

[9] Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., Rabab'ah, 2009. "Enhanced Algorithm for Extracting the Root of Arabic Words". In: the Sixth International Conference on Computer Graphics, Imaging and Visualization, (CGIV '09). pp. 388–391.

[10] Soufiane Farrah, Hanane El Manssouri, El Houssaine Ziyati. « Proposition of improvement areas in most heavy an light stemmer algorithms novel stemmer: EST.Stemmer». In *Information Science and Technology (CIST), 4th IEEE CiSt'16 Program*. IEEE, 2016.

[11] M. A. H. Omer and S. Ma, Stemming Algorithm to Classify Arabic Documents , Journal of Communication and Computer, Vol. 7, No. 9, Sep. 2010.

[12] Elrajubi, Osama Mohamed. « An improved Arabic light stemmer ». In *Research and Innovation in Information Systems (ICRIIS), 2013 International Conference on*, 33–38. IEEE, 2013.

[13] Hadni, Meryeme, Abdelmonaime Lachkar, et S. Alaoui Ouatik. « A new and efficient stemming technique for Arabic Text Categorization ». In *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, 791–796. IEEE, 2012.

[14] Mohammed N. Al-Kabi, Saif A. Kazakzeh, Bilal M. Abu Ata, Saif A. Al-Rababahc, Izzat M. Alsmadi« A novel root based Arabic stemmer ». In *Journal of King Saud University - Computer and Information Science,* 94–103, Elsevier, April 2015.

[15] The Five Nouns In Arabic : https://fr.scribd.com/doc/93320581/The-Five-Nouns-In-Arabic.

[16] Chen, A., Gey, F., 2002. "Building an Arabic stemmer for information retrieval". In: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002). National Institute of Standards and Technology, pp. 631–639.

[17] A.F.A. Nwesri, S.M.M. Tahaghoghi, F. Scholer "Stemming Arabic Conjunctions and Prepositions". Lect. Notes Comput. Sc., 3772 (2005), pp. 206–217 http://dx.doi.org/10.1007/11575832_23.