Building an Effective Rule-Based Light Stemmer for Arabic Language to Improve Search Effectiveness

Mohamad Ababneh[1], Riyad Al-Shalabi[2], Ghassan Kanaan[2], and Alaa Al-Nobani[1]
[1]Computer Information Systems Department, Al-Balqa Applied University, Jordan
[2]Computer Information Systems Department, Arab Academy for Banking and Financial Science, Jordan

Abstract: *Building an effective stemmer for Arabic language has been always a hot research topic in the IR field since Arabic language has a very different and difficult structure than other languages, that's because it is a very rich language with complex morphology. Many linguistic and light stemmers have been developed for Arabic language but still there are many weakness and problems, in this paper we introduce a new light stemming technique and compare it with other used stemmers and show how it improves the search effectiveness.*

Keywords: *Stemmer, natural language, light stemmers.*

Received February 22, 2010; accepted May 20, 2010

1. Introduction

The main goal behind building any stemmer is to improve the search effectiveness so an IR system can match user's queries with relevant documents. Users form their query terms in many different formats but they are looking for the same thing [1]. Now an IR system should be able to translate all these forms that have the same meaning to a standard form and thus grouping all these different formats in a singular or standard format and this should be done on both sides, on users queries and on index terms. The big growth of the Arabic internet content in the last years has raised up the need for an effective stemming techniques for Arabic language [2]. Many stemming methods have been developed for Arabic language. Although they suffer from many problems, they have been in use in many IR systems. These stemmers are classified into two categories. The first one is root extraction stemmer like the stemmer introduced by Khoja [5]. The second is light stemmers like the stemmer introduced by Leah *et al.* [6]. We will describe these two approaches and show the problems in each one in the next section.

2. Stemming Approaches

As we mentioned before the two most successful approaches to Arabic stemming have been a root extraction stemmer developed by Khoja and the light stemmer developed by Larkey, now we will describe these two approaches and discuss the problems in each one.

2.1. Root Extraction Stemmer

Arabic words are formed from abstract forms named roots, the root is the basic form of word from which many derivations can be obtained by attaching certain affixes so we produce many nouns and verbs and adjectives from the same root [3]. A root based stemmer main goal is to extract the basic form for any given word by performing morphological analysis for the word [2], Table 1 shows an example root "لعب" and a set (not all) derivations can be obtained from this root:

Table 1. Some derivations of the root لعب.

يلعب	ملعب	لاعب	ملعوب	لعبة
Play	Playground	Player	Played	Game

Khoja [5] stemmer basically attempts to find roots for Arabic words which are far more abstract than stems. It first removes prefixes and suffixes, then attempts to find the root for the stripped form [4]. The problem in this stemming technique is that many different word forms are derived from an identical root, and so the root extraction stemmer creates invalid conflation classes that result in an ambiguous query which leads to a poor performance. For example the word "الاقتصادية" using Khoja stemmer will be stemmed to its root which is "قصد" where this root is very far abstract from the stem and many words with very different meanings can be formed from this root like the words: مقصود، قاصد so they will always be stemmed to this root and this will lead to a very poor search effectiveness because many words are different in meaning but they originate from one identical root.

So we end up with the fact that root extraction stemmers increase word ambiguities and that inflected and derived words can have a vigorous impact on the retrieval effectiveness of any information retrieval system and a good stemmer should recognize the different forms of a word [2].

2.2. Light Stemmer

Light stemming is to find the representative indexing form of a word by the application of truncation of affixes [1]. The main goal of light stemming is to retain the word meaning intact and so improves the retrieval performance of an Arabic information retrieval system. Many light stemming methods like Leah *et al.* [6] stemmer classifies the affixes to four kinds of affixes: antefixes, prefixes, suffixes and postfixes that can be attached to words. Thus an Arabic word can have a more complicated form if all these affixes are attached to its root. The following example, Table 2, shows a sample of a word and its affixes [1]:

Table 2. A word and its affixes ليناقشوهم.

Antefix	Prefix	Core	Suffix	Postfix
لـ	يـ	ناقش	و	هم

So from the above example we see that if we could remove all affixes of a word then we will get the stemmed word which is not the root but basic word without any affixes and so we maintain the meaning of the word and improve the search effectiveness. This sounds to be a straightforward method by truncation of all possible affixes but a major problem in light stemming is that in many cases there is ambiguity. A particular sequence of letters may or may not play a role of affix [1], depending on the word. No morphological rules are currently available to determine the correct affixes.

In this paper we will introduce a method for detecting such an ambiguity and to find if a specific sequence is an affix or is part of the original word and thus we solve this ambiguity issue that may lead to a completely unexpected behavior. Also, we will provide a more complete list of all possible affixes.

3. Arabic Language Characteristics

Arabic is a very rich and complex language. Arabic has 28 characters and is written from right to left. Arabic language differs from English and European languages and the morphological representation of Arabic is rather complex because of the morphological variation and the agglutination phenomenon [1]. Arabic language is described as algebraic language which makes its morphological analysis process very difficult and hard. Arabic language is based on set of roots, so all nouns and verbs are generated from a set of roots which is about 11,347 root distributed as follow [8]:

- 115: Two character roots (and these roots have no derivations from them).
- 7198: Three character roots.
- 3739: Four character roots.
- 295: Five character roots.

These roots join with various vowel patterns to form simple nouns and verbs to which affixes can be attached for more complicated derivations. Patterns play an important role in Arabic lexicography and morphology. Each root can canonically combine with orthographically distinct patterns to form another words, for example, the root “لعب” is consisting of three characters root, the root “لعب” corresponds to the pattern “فعل” and the pattern preserves “ل” ،“ع” ،“ف” in the same order, where other letters can be added to form another pattern. For example, several patterns are derived from the base pattern “فعل” of the morpheme “لعب”. The pattern “مفعل” form the word “ملعب” by adding the letter “م” to the morpheme “لعب” [2].

4. The Proposed Rule-Based Light Stemmer

We are introducing a new method for stemming to solve many of the ambiguity problems related to light stemming, our method depends on set of possible affixes in which we only have a prefix and suffix, in our prefixes we combined all possible antefixes and prefixes to generate one complete list and in our suffixes we combined all possible suffixes and postfixes [2, 6], and we end up with the following list grouped by number of characters as shown in Tables 3 and 4.

Table 3. Arabic prefixes.

Prefix 1	ي ت ن ب ل
Prefix 2	ال لل سي ست سن كا فا با لي لت لن فت في فن
Prefix 3	وال بال فال كال ولل وسي وست وسن وسا ولا ولي ولت ولن
Prefix 4	وبال وكال

Table 4. Arabic suffixes.

Suffix 1	ه ة ك و ي ن ا ت
Suffix 2	ان ين ون ات هم هن ها كم كن نا وا تم تي تن ته يه ما يا تا تك

Now we will describe step by step how our algorithm works:

1. Before we stem any word first we match it against a set of all possible word patterns in Arabic, we get this list of patterns by combining the list of available patterns in Khoja stemmer with a list of word patterns provided by Marwan [8]. We also added a set of patterns to this list to have a complete list of Arabic word patterns. A sample list of patterns is as follows:
   فعّل فاعل افعل تفعل تفاعل انفعل افتعل استفعل تفعيل فعال افعال
   تفعل تفاعل انفعال افتعال افعلال استفعال مفعل مفاعل مفعل متفعّل
   متفعل متفاعل منفعل مفتعل مفعلّ مستفعل مفعول فعول مفعال فعّال

فعيـل أفعـل فعـلان فعـلاء فعلـى فواعـل مفاعيـل افاعـل فعيّـل يفتعـل
يستفعل تفتعل فعائل

Matching a word against our Arabic patterns list solves the problem of prefix/suffix sequence ambiguity, so if we have a word that starts with a possible prefix but before we truncate that prefix it matched one of the possible patterns then it's a valid word and this prefix sequence is part of the original word and we will not truncate it. For example the word "كامل" it starts with a possible suffix "كا" but its part of the original word so removing it will lead to the wrong word "مل" so we detect that it is part of the word since it matches the pattern "فاعل", thus we will not truncate it and return it as it is.

2. In next step if the word didn't match any of the patterns then we need to truncate its prefix and suffix but before that we find the compatibility between the prefix and suffix where some suffixes could not be combined with certain suffixes in the same word and this also help us solving some ambiguity problems, for example the prefix "ال" may not be combined with the suffix "ك" so we cannot say "الكتابك" and thus if we have a word like "الكرنك" we will not remove the prefix and suffix which lead to the wrong word "كرن" but we will detect that the last character "ـك" is part of the original word and not a suffix and we will only remove the prefix "ال" which will lead to the correct stem "كرنك".
3. If the combination of the prefix and suffix is valid then we count the letters of the word after removing the prefix and suffix since Arabic words other than conjunctions like "في" ،"من" consists of at least 3 characters. if the number was larger or equal to 3 we remove prefix and suffix and return the truncated word but if the number of characters after truncation is less than 3 characters then we roll back since we know that we removed sequences that is part of the original word, and try to remove only the suffixes and count the number again if it is larger than 3 then we try to find if the word only has a dual or plural suffix like "ون" ،"ين" ،"ان" and return the truncated word. If the number after removing suffix was less than 3 then we roll back since again we know that this sequences is part of the original word, then we try to remove the prefix only. If the count is larger than 3 we return the truncated word else we return the original word as it is. For example if we have the word "ولدين" after removing the prefix and suffix we will end up with the wrong word "د" which has only one character so we roll back and remove only the suffix which is "ين" which returns "ولد" which has 3 characters and it is the correct stem so we return it.
4. Finally, we try to solve the problem of the plural form of irregular nouns in Arabic language, also called broken plural. In this case, a noun in plural takes another morphological form different from its initial form in singular.

We have a dictionary of patterns for this plural forms and their corresponding singular pattern as shown in the table below, so after getting the truncated word from the previous steps we match it against our irregular plural patterns if it matched one of them we return the corresponding singular pattern of the word. For example the word "مصانع" is a broken plural for the word "مصنع" so when we get this word we match it against our broken plural patterns dictionary and it will match the pattern "مفاعل" from which we get the singular pattern which is "مفعل".

Table 5. Singular and plural patterns.

Plural Pattern	Singular Pattern
مفاعل	مفعل
مفاعيل	مفعول
أفعال	فعل
فعلاء	فعل، فعال، فاعل، فعيل
فعال	فاعل
أفعل	فعل
أفعلة	فعيل، فعال
فواعل	فوعل، فاعل

In some cases a plural pattern may have more than one singular pattern like the pattern "فعلاء" it has the singular patterns "فعل"، "فعال"، "فاعل"، "فعيل" as shown in the table above, for example the word "أطباء" has a singular "طبيب" which is like "فعيل" but the word "عقلاء" has a singular "عاقل" which is like "فاعل", and the word "سمحاء" has a singular "سمح" which is like "فعل", and the word "جبناء" has a singular "جبان" which is like "فعال", so when we get such a plural pattern we will return all its possible singular forms. Some of the broken plural forms like "فعائل"، "فعالل"، "فعلة" has many singular patterns and has no rules to cover them and it is hard to deal with them without having a complete dictionary for these plurals.

5. Comparison

We took a sample terms list and tested it against a light stemming method and here we used Larky stemmer, root-extraction stemmer and here we used Khoja stemmer and our rule based stemmer, and the result were shown in Table 6.

From the above results we see that the light stemmer fails in many times in getting the correct stem of the word and in many words it produced a completely new word and sometimes a wrong word that doesn't exist in Arabic language and it didn't handle the broken plural forms, the root extraction stemmer as we see it produces a very general words (roots) that are far in their meaning from the original word, where our suggested stemmer produced the expected stems and removed all affixes effectively and it didn't remove them where they are part of the original word and it handles all the broken plural forms and generates the correct singular forms.

Table 6. A comparison between the three stemmers.

Term	Our Rule Based Stemmer	Light Stemmer (Larkey)	Root-Extraction Stemmer (Khoja)
كامل	كامل	امل	كمل
بلادي	بلد	بلاد	بلد
قراطيس	قرطاس	قراطيس	قراطيس
محامون	محامي	محام	حوم
مجانين	مجنون	مجان	جنن
ملاعب	ملعب	ملاعب	لعب
بحور	بحر	بحور	بحر
بخلاء	بخل، بخال، باخل، بخيل	بخلا	بخل
باستثناء	استثناء	ستثنا	ثني
بادرو	بادرو	درو	درأ
كوادر	كودر, كادر	كوادر	كدر
بشرية	بشر	بشر	بشر
كفلاء	كفل، كفال، كافل، كفيل	كفيلا	كفل
متفاهمون	متفاهم	فاهم	فهم
علي	علي	عل	علي
بسطاء	بسط، بساط، باسط، بسيط	بسطا	بسط
تقنين	تقنين	تقن	قنن
مباراة	مباراة	مباراة	برا
باستياء	استياء	ستيا	سيأ
فالح	فالح	ح	فلح
متمرن	متمرن	مرن	مرن

6. Conclusions and Results

The objective of stemming in general is to find the representative indexing form of a word and for Arabic as a highly inflected language, we require a good stemming for effective information retrieval, yet no standard approach to stemming has emerged but as we stated before there are two general methods that are used, either by extracting the root of the word like Khoja stemmer, or by just truncation of affixes like Larkey stemmer. The first method have many problems, first, the root dictionary requires maintenance to guarantee newly discovered words are correctly stemmed, second one is that in some cases it fails to remove the affixes of the word and thus fails to extract the root, for example the Khoja stemmer will fail to remove affixes in the words “تستغرق” and “ركبتيه” and so it will not stem them where they are respectively derived from the roots “غرق” and “ركب” [4]. The third and most important problem is that root-extraction stemmers are not useful in case of Arabic language from IR system point of view as we stated before and in many cases resulting in a new word which is the root that is very general and thus leading to a poor search effectiveness, for example the word “محامون” will be reduced to the root “حمى” where this root is very general and huge number of words can be derived from this root, so reducing all the forms that can be generated from that root to its basic form will result in a general index terms which has a serious effect on the quality of the results.

The second method is more useful from IR system point of view considering Arabic language, but again the available techniques have two major problems, the first one is in many cases these stemmers truncate a sequence of characters that matches one of the affixes but it is actually part of the original word and this will lead to a completely new word for example Larkey stemmer will truncate the word “ولدين” to “لد” which has no meaning in Arabic [10], the second problem is that they are not dealing with the plural form of irregular nouns in Arabic language, so in many cases it fails to group words that have the same meaning in one reduced form. In our rule-based light stemmer we are considering the second method but we introduced a new algorithm as described before that uses a set of rules to determine if a certain sequence of characters is part of the original word or not and this helped us solving some ambiguity problems, also we introduced a way for handling the majority of broken plural forms and reducing them to their singular form as we described in details before and this helped us grouping words of the same meaning in a common form.

References

[1] Abdusalam N., Seyed T., and Falk S., “Stemming Arabic Conjunctions and Prepositions,” *in Proceedings of the 12th international conference on String Processing and Information Retrieval*, Heidelberg, pp. 206-217, 2005.

[2] Aitao C., “Building an Arabic Stemmer for Information Retrieval,” *in Proceedings of the Eleventh Text Retrieval Conference*, Berkeley, pp. 631-639, 2003.

[3] Hayder A., Shaikha A., Amna A., Khadija A., Naila A., Noura A., and Shaikha A., “Arabic Light Stemmer: A New Enhanced Approach,” *in Proceedings of Software Engineering Department,* UAE University, Dubai, pp. 1-9, 2005.

[4] Kazem T., Rania E., and JeÌrey C., *Arabic Stemming Without A Root Dictionary*, Information Science Research Institute, USA, 2005.

[5] Khoja S. and Garside R., Stemming Arabic Text, available at: http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps, last visited 1999.

[6] Leah L., and Lisa B., and Margaret C., *Light Stemming for Arabic Information Retrieval*, University of Massachusetts, Springer, 2007.

[7] Leah L. and Margaret C., “Arabic Information Retrieval at UMass in TREC-10,” *in Proceedings of the 10th Text REtrieval Conference*, Gaithersburg, pp. 562-570, 2001.

[8] Marwan B., *Arabic Language Processing in Information Systems*, Springer, 2004.

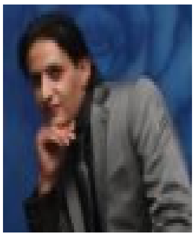
[9] Mohammed A. and Ophir F., "On Arabic Search: Improving the Retrieval Effectiveness Via A Light Stemming Approach," *in Proceedings of Computer Technology Department, Riyadh College of Technology*, USA, pp. 340- 347, 2002.

[10] Youssef K. and Jian-Yun N., "Effective Stemming for Arabic Information Retrieval," *in Proceedings of the Challenge of Arabic for NLP/MT Conference*, UK, pp. 68-74, 2006.

Mohammad Ababneh recived his PhD in computer engineering, from Cairo University, Egypt in 2000. He is an associate professor of computer engineering, he is an instructor in Computer Information Systems, Balqa Applied University, Jordan. Worked in different universities in Jordan. Teaching different subjects like, microprocessors, computer organization, computer architecture, image processing, algorithms, compilers, programming, etc., he also held administration positions as a vice dean, dean assistant and head of department. Published around 16 papers in different computer disciplines.

Riyad Al-Shalabi recived his PhD in computer science, Illinois Institute of Technology, Chicago, IL, USA 1996. He is a full professor of computer science, he is an instructor in Computer Information Systems, University of Banking and Financial Sciences, Jordan. He served in several universities where he served as a chairman in Computer Science Department at Yarmouk university, he had many consultancy assignments in the areas of education in Jordan. Currently, he is interested in the area of arabic natural language processing and arabic information retrieval.

Ghassan Kanaan recived his PhD in computer science, Illinois institute of technology, Chicago, IL, USA in 1997. He is an associate professor of computer science, he is an instructor in Computer Information Systems, Arab Academy for Banking and Financial Sciences, Jordan. He served in several universities where he served as a chairman in Computer Information Systems Department at Yarmouk University, he had many consultancy assignments in the areas of education in Jordan and. He is interested in database systems, currently he is interested in the area of arabic natural language processing and arabic information retrieval.



Alaa Al-Nobani received his MSc degree in computer science and his BSc in information technology form Al-Balqa University, Jordan in 2008, 2004, respectively. Currently, he is a technical and products manger at IKoo Media-Jabber Intermet Group, United Arab Emirates.