Arabic Word Stemming Algorithms and Retrieval Effectiveness

Tengku Mohd T. Sembok, Belal Abu Ata

Abstract**—**Documents retrieval in Information Retrieval Systems (IRS) is generally about retrieving of relevant documents pertaining to information needs. The more the system able to understand the contents of documents the more effective will be the retrieval outcomes. But understanding of the contents is a very complex task. Conventional IRS applies algorithms that can only approximate the meaning of document contents through keywords approach using vector space model. Keywords may be unstemmed or stemmed. When keywords are stemmed and conflated in retrieval process, we are a step forwards in applying semantic technology in IRS. Word stemming is a process in morphological analysis under natural language processing, before syntactic and semantic analysis. We have developed algorithms for Arabic stemming and incorporated it in our experimental system in order to measure retrieval effectiveness. The results have shown that the retrieval effectiveness has increased when stemming is used.

Index Terms**— —Information Retrieval, Natural Language Processing, Artificial Intelligence**

I. Introduction

Lovins [1] defines stemming algorithm as “a computational procedure which reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes”. For example, the words *group, groups, grouped, grouping,* or *subgroups* are reduced to the root *group*. Stemming algorithms play an important role in the fields of information retrieval and computational linguistics. In information retrieval, grouping words having the same root will increase the success with which documents can be matched against a query [2][3]. This research is to confirm that it is also apply to Arabic information retrieval systems. In computational linguistics, there is a need to identify linguistically correct roots, since the attached affixes provide information about the grammatical function of a word and thus help in the syntactic and semantic analysis of a sentence.

Many different approaches of stemming algorithms have been proposed, ranging from simple procedures that merely remove plurals, past and present participles to more sophisticated techniques that are based on morphological rules of specific languages [4]. The sophisticated algorithms involve iterative removal of affixes based on simple and context-sensitive morphological rules in order to achieve accurate results [5][6].

However, stemming algorithms for English language involve only in stripping of just the suffixes has been found to be sufficient for the purpose of information retrieval [7]. The usage of affixes in English (and similar languages) is far less complex than in languages such as Arabic where the stripping of suffixes alone would not be sufficient for retrieval purposes [8]. In Arabic, prefixes and infixed are used regularly in practice either in writing or speaking.

Stemming algorithm for Arabic words has been an important topic in Arabic information retrieval. Khoja [9] attempts to find roots for Arabic words by first removing prefixes and suffixes, and then tries to determine the root from the stripped words using a dictionary of root words. Light stemmers, such as the algorithm developed by Larkey [10], Darwish [11] and Chen [12] select some prefixes and suffixes to be truncated from the words and produce the stems. We envisage that the approach adopted by Khoja [9] is more appropriate in determining roots or stems, since the dominant present of infixes in Arabic words.

II. Arabic Language Stemming Algorithms

Approaches adopted by stemmers of other languages are not fully appropriate for the development of Arabic stemmer due to differences in the morphological structures peculiar to each of the languages. The main differences between Arabic and other languages as put forward by El-Sadany & Hashish [13] are as follows:

i. Arabic is one of Semitic languages which differ in structure of affixes from Indo-European type of languages such as English and French;

ii. Arabic is mainly roots and templates dependent in the formation of words;

iii. Arabic roots consonants might be changed or deleted during the morphological process;

iv. Arabic basically uses diacritics, such as (ـَ) *fatḥa* (a), (ـِ) *kasra* (i), (ـُ) *ḍamma* (u), (ـْ) *sukūn* (no vowel), instead of vowels.

Stemmers such as Porter’s algorithm are developed mainly to improve the retrieval performance of document retrieval systems. As a result, these stemmers do not make use of dictionary that checks for the correctness of the

Manuscript received April 5, 2013; revised April 15, 2013. This work was supported in part by the Malaysian Government Research Grant.

Tengku M. T. Sembok is with International Islamic University Malaysia, Gombak, and as advisor at Universiti Tenaga Nasional, and Multimedia University, Malaysia. (phone: +6012-337-3539; fax: +603-89260539;
e-mail: tmtsembok@gmail.com).

Belal Abu Ata was with Universiti Kebangsaan Malaysia, Bangi, Malaysia. He is now teaching in a Jordanian university after returning from Malaysia.

resulted stems or roots. Improvement may be achieved by application of cross checking against a dictionary of root words. This approach of using a dictionary has shown some improvement for Malay stemming [6]. Arabic stemming is actually a process of morphological analysis applied for the word in order to extract the correct stem. The Arabic stemming approach adopted by most of the previous researchers is the iterative application of the following processes:

i. Striping of diacritics
ii. Striping of prefixes
iii. Striping of suffixes
iv. Determining the stem
v. Recoding the stem
vi. Verifying the stem with a dictionary of root words.

This approach was used by several researchers such as Khoja [9], El-Sadany & Hashish [13], Hilal [14], and Shahein & Youssef [15].

III. ARABIC WORD FORMATION

The grammatical system of the Arabic language is based on a root-and-pattern structure and considered as a root-based language with not more than 10,000 roots. A root in Arabic is the base verb form which is predominantly trilateral, and to a lesser extent, quadrilateral, pentaliteral, or hexaliteral. A root word generates derivative verbs and nouns by adding derivational affixes [16].

Affixes in Arabic are: prefixes, suffixes (or postfixes) and infixes (morphemes). Prefixes are attached at beginning of the words, where suffixes are attached at the end, and infixes are found in the middle of the words. For example, the Arabic word **الطالبات** (*altalibat*) which means *the female students,* consists of the elements as shown in Table 1**:** the root, the prefix, the suffix and the infix.

Table 1: Example of Arabic Affixes

Word	**root**	**prefix**	**suffix**	**infix**
الطالبات	*طلب*	*ال*	*ات*	*ا*
the female students	*a student (male)*	*the*	*Feminine indicator*	*noun formation*

In Arabic the definite article and some prepositions are written attached to the word. Thus, the definite article, *ال*, and some prepositions in Arabic are considered as a group of prefixes, besides the subject markers ا ,ي ,ت *(alif*, *ya*, and *ta*). Arabic allows up to three consecutive prepositions from this group to precede a word. For example, the word وبالوالدين (and with the children) which contains three prefixes (and و , with ب, the ال). Table 2 show the prefixes belong to this group.

IV. ARABIC SUFFIXES

There are fifteen suffixes in Arabic language which form a small set of suffixes compared to languages such as Malay and Slovene languages. However, Arabic allows up to three concurrent suffixes to be attached at the end of the word, for example, the word ضربناهم contains three prefixes (هم , ا, ن). Arabic suffixes are mostly made of attachable pronouns. Table 3 shows all the 15 Arabic suffixes and their meanings.

Table 2: Arabic Prefixes

Prefix	**Meaning**	**Exampl e**
ب	with, in, by	بالسيارة
ك	same as	كالدخان
س	will	سأذهب
و	and	ورجالهم
ال	the	النساء
أ	question marker	أأكلت
ف	then	فذهبوا
ل	to, because	لتنام

Table 3: Arabic Suffixes and Derivative Meanings

Suffix	**Derivative Meaning**	**Example**
ين	singular female	تلعبين
ان	male dual	يلعبان
و	plural male	ينمو
ه	singular feminine	ضربته
ك	addresser singular masculine	ضربك
ا	male dual	أكلا
ي	singular female	أكلتي
ن	plural	أكلن
ت	singular female	أكلت
ات	female plural	لاعبات
ون	absent male plural	يلعبون
وا	absent male plural	أكلوا
تم	addresser male	أكلتم
هم	absent male plural	ضربهم
كم	addresser male	ضربكم

V. RULE-BASED STEMMING ALGORITHM

There are few approaches adopted in the development of Arabic stemming algorithms. Among them are neural network approach [16][17], Support Vector Machine (SVM) application [18], and the rule based approach [19][20]. The first two approaches adopted machine learning technique to run the algorithms, and the last approach use the morphological rules to do the stemming. In this paper, the rule based approach is adopted to do the stemming. The rules are categorized into the following groups: prefixes, suffixes, and recoding. Besides the rules, templates of root words are used to generate possible roots for a given word. Lastly, a dictionary of root words is used to verify the validity of the root candidates.

The stemming algorithm is implemented in C language with Arabic support and having the following main modules:

- Prefix and suffix removal module
- Root generator and checking module
- Pattern generator and checking module
- Handler of double letters (تشديد) module
- Recoding module.

A. Prefix and Suffix Removal Module

This module is designed to find and strip prefixes and suffixes from the given word. Though, the number of prefixes and suffixes are not many in Arabic, but their attachment rules to words are complicated. After some thorough study of the Arabic morphological structure and word formation, we came out with around 800 rules that cover both Arabic prefixes and suffixes attachment rules. The prefix and suffix rules are defined according to the following syntax:

1. Prefix rules: **prefix + *let(s)***
   where *let(s)* is a set of valid letters to follow the prefix.
   Example: أ + تأ
   which means أ *is considered a prefix if the next two letters are* تأ *such as in the word* أتأتي

2. Suffix rules: ***let(s)* + Suffix**
   where let(s) is a set of valid letters preceding the suffix.
   Example: ت + بب
   which means ت *is considered a suffix if the previous 2 letters are* بب *such as in the word* أحببت

Table 4 shows examples of prefixes and suffixes in the given words.

Table 4: Examples of Word Letters that match the Arabic Affixes

Word	**Letter(s)**	**Type of Affix**
فارس	ف	*Prefix*
لاعبون	ل	*Prefix*
بارد	ب	*Prefix*
بنات	ت	*Suffix*
متم	تم	*Suffix*
القرون	ون	*Suffix*

B. Root Generator and Checking Module

This module will try to find all the valid possible roots for a given word. The module will check for the root validity by using the hashing technique to search for it in the roots dictionary. This module will invoke the *Intensification Submodule* that checks for words of double letters in order to change it to the normal form.

C. Pattern Generator and Checking Module

This module will take the word to be stemmed and one possible root (generated by the root module) as an input and then derive a template that matches both of them. This process will be repeated for the entire possible root generated from the root generator module. The module will also check the resulted template for its correctness by matching it to a set of valid Arabic templates. An example of this is as follows:

- for the word فاسقين, some of the possible roots generated are فسق, فاس, سقي, قين, فاق , where the roots قين, فاق are ignored as they are not valid Arabic roots

- the templates for the remaining 3 roots are constructed with reference to the word فاسقين, the resulting templates and their validity in Arabic are shown in Table 5.

Table 5: Possible Templates for the Word فاسقين

Root	**Generated Template**	**Template Validity**
فسق	فاعلين	*valid*
فاس	فعلقين	*invalid*
سقي	فافعلن	*invalid*

D. Handling Double Letter Module

There are many Arabic words and roots with *double* letters, which means that two similar adjacent letters are combined into one letter. This module will check for such words and its root and reconstruct the word by adding that letter. This will help in obtaining the correct root. Examples of words with *Intensification* are shown in Table 6.

Table 6: Examples of Arabic words with *Intensification*

Root	**Generated Template**	**Template Validity**
فسق	فاعلين	*valid*
فاس	فعلقين	*invalid*
سقي	فافعلن	*invalid*

E. Recoding Module

The *recoding module* main concern is to change some of the letters to their correct form. These changes will probably occur during the process of template formation when a word is formed from a root. Some letters may be dropped, changed or replaced by other letters. Table 7 lists some of the most recoded Arabic letters.

F. Stemming Algorithm Flowchart

The flow chart of the stemming algorithm is shown in Figure 1. The stemming process begins by processing a word and trying to find its correct stem. In case the word does have a correct stem, then the word without its affixes will be returned.

Table 7: Examples of Letter Recoding for Arabic Words

Word	**Recoding Rule (from→ to)**	**Word after Recoding**

هزئ	ئ → ؤ	هزؤ
	ئ → أ	هزأ
نبئ	ئ → أ	نبأ
خطئ	ئ → أ	خطأ
خسئ	ئ → أ	خسأ
صبئ	ئ → أ	صبأ
سيئ	ئ → أ	سيأ
نبء	ء → أ	نبأ
دنى	ى → ا	دنا
تؤمن	ؤ → أ	تأمن
يؤمر	ؤ → أ	يأمر
يؤخذ	ؤ → أ	يأخذ
ؤمر	ؤ → أ	أمر
راد	ا → و	رود
حيا	ا → ي	حيي

The stemming algorithm will take as input an Arabic word (not a stop word), and the output will be the extracted root (or stem). In cases where the algorithm cannot find a root for the specific word, the word itself will be taken as a root. Such cases are few and it depends on the quality of the algorithm proposed.

Table 10 shows the number of errors obtained by our stemming algorithm compared to the results obtained by Al-Omari's algorithm. Hence, we can conclude that our algorithm does performance better than Al-Omari's algorithm. Table 11 shows all the 21 words that have been stemmed wrongly and the types of errors for each word. Table 12 shows the distribution of unique errors in Quran data collection.

There are a total of 21 unique errors as shown in Table 10. These errors are classified into 5 groups, namely, *understemming, over-stemming, spelling, unchanged*, and *others*. The names of the groups describe the type of errors. Understemming and overstemming indicate that the resultant stems are under stemmed or over stemmed. The group *spelling* indicates there is one letter in the resulted stem that is different from the correct root. The unchanged group indicates that the resultant stem is the same as the original word which is not the correct root. Others indicate other types of stemming errors.

VI. EXPERIMENTS

The database collection for the experimental retrieval system consists of the Quran collection which contains 6236 documents or verses. The Quran consists of 114 chapters where every chapter contains variable lengths of verses in it as can be seen in Table 8. The number of queries used in the experiment is 36 and their characteristics are shown in Table 9.

TABLE 8: The Quran's chapters with their corresponding total number of verses

*	#	*	#	*	#	*
1	7	**20**	135	**39**	75	**58**
2	286	**21**	112	**40**	85	**59**
3	200	**22**	78	**41**	54	**60**
4	176	**23**	118	**42**	53	**61**
5	120	**24**	64	**43**	89	**62**
6	165	**25**	77	**44**	59	**63**
7	206	**26**	227	**45**	37	**64**
8	75	**27**	93	**46**	35	**65**
9	129	**28**	88	**47**	38	**66**
10	109	**29**	69	**48**	29	**67**
11	123	**30**	60	**49**	18	**68**
12	111	**31**	34	**50**	45	**69**
13	43	**32**	30	**51**	60	**70**
14	52	**33**	73	**52**	49	**71**
15	99	**34**	54	**53**	62	**72**
16	128	**35**	45	**54**	55	**73**
17	111	**36**	83	**55**	78	**74**
18	110	**37**	182	**56**	96	**75**
19	98	**38**	88	**57**	29	**76**
58	22	**77**	50	**96**	19	**58**
59	24	**78**	40	**97**	5	**59**
60	13	**79**	46	**98**	8	**60**
61	14	**80**	42	**99**	8	**61**
62	11	**81**	29	**100**	11	**62**
63	11	**82**	19	**101**	11	**63**
64	18	**83**	36	**102**	8	**64**
65	12	**84**	25	**103**	3	**65**
66	12	**85**	22	**104**	9	**66**
67	30	**86**	17	**105**	5	**67**
68	52	**87**	19	**106**	4	**68**
69	52	**88**	26	**107**	7	**69**
70	44	**89**	30	**108**	3	**70**
71	28	**90**	20	**109**	6	**71**
72	28	**91**	15	**110**	3	**72**
73	20	**92**	21	**111**	5	**73**
74	56	**93**	11	**112**	4	**74**
75	40	**94**	8	**113**	5	**75**
76	31	**95**	8	**114**	6	**76**

* - Chapter number.
# - Total number of verses.

Quantitative characteristics	**Before stop wording**	**After stop wording**
Number of queries	36	36
Total number of terms in the queries	387	238
Average number (median) of terms per query	10	6
Maximum number of terms in a query	22	17
Minimum number of terms in a query	2	1

TABLE 9: The main characteristics of the set of queries before and after deletion of stop words

An experiment to compare retrieval effectiveness of using conflation method, stemming of Arabic words, has been carried out and found that it performs better than nonconflation method. The graph in Figure 2 shows the analysis of performance based on recall-precision measurement between the nonconflation methods and conflation methods using our stemmer (Abu Ata's) and Al-Omari's stemmer. In terms of retrieval effectiveness, our stemmer performs better than that of Al-Omari's. Both stemmers perform better than nonconflation method.

Table 10: Results of the Experiments for 10 Quran Chapters

Number of words wrongly stemmed			
		Ours	**Al-Omari's**
Chapter 1	684+	9	25
	126~	6	15
Chapter 2	450+	8	23
	83~	6	17
Chapter 3	444+	6	24
	100~	4	20
Chapter 4	462+	9	27
	103~	5	19
Chapter 5	202+	6	15
	56~	3	12
Chapter 6	278+	5	19
	48~	2	17
Chapter 7	341+	9	29
	73~	4	19
Chapter 8	299+	5	12
	73~	4	10
Chapter 9	341+	4	14
	88~	3	9
Chapter 10	181+	4	7
	46~	2	4
	3682+	**65**	**195**
Totals	**796~**	**39**	**142**
	330*	**21**	**85**

Keys:
+ Total number of all words in the chapter
~ Total number of unique words in the chapter
*Total number of unique words in all the chapters

Table 11: Stemming Errors on Ten Chapters of the Quran

Word	**Actual Root**	**Resulting Root**	**Error Type**
ربه	ربب	ربه	unchanged
موتهـ	موت	موة	spelling
الظا	ظنن	ظان	spelling
الريا	ريح	راح	spelling
بالبا	بطل	اطل	spelling
وبار	برك	ارك	spelling
فويل	ويل	يل	overstemming
الفلك	فلك	لك	overstemming
ليبلو	بلو	بل	overstemming
بآل	بآل	آل	overstemming
بأس	بأس	أس	overstemming
بوالد	ولد	ديه	others
	وقى	قا	others
تنزيـ	نزل	زيل	others
كرها	كره	رها	others
والف	فسق	سوق	others
المبيـ	بين	مبي	others
فتبينـ	بين	تبي	others
مبين	بين	مبي	others
بالهم	بآل	هم	others
فأتنا	أتي	تنا	others

Table 12: Distribution of Errors in Quran Data Set

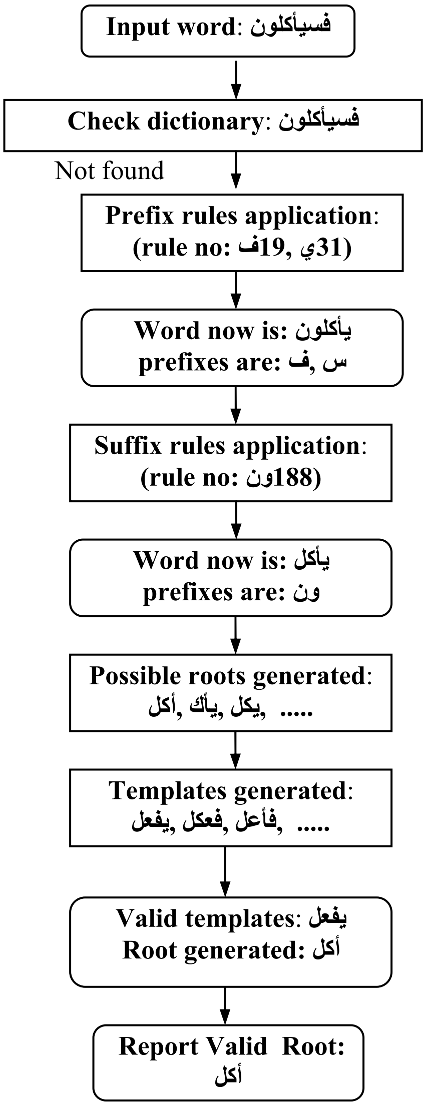
Error Type	**Number (%)**
Overstemming	5 (23.8%)
Understemming	0 (0 %)
Unchanged	1 (4.7%)
Spelling	5 (23.8%)
Others	10(47.6%)

VII. CONCLUSION

Our experiments have shown that our stemming algorithm performs better than that of Al-Omari [21]. Could it be improved further? Our analysis suggests that most of the errors are due to the order in which the stemming rules are applied, and we are currently considering ways in which this ordering can be best applied.

In terms of retrieval effectiveness, both stemming algorithms perform better than non-conflation method. This experiment conforms to the result obtained by experiments done on English language.

Figure 1: The steps to find the stem for the word: **فسيأكلون:**


Input word: فسيأكلون
Check dictionary: فسيأكلون
Not found
Prefix rules application: (rule no: ف19, ي31)
Word now is: يأكلون
prefixes are: ف, س
Suffix rules application: (rule no: ون188)
Word now is: يأكل
prefixes are: ون
Possible roots generated: أكل, يأك, يكل, ...
Templates generated: يفعل, فعكل, فأعل, ...
Valid templates: يفعل
Root generated: أكل
Report Valid Root: أكل

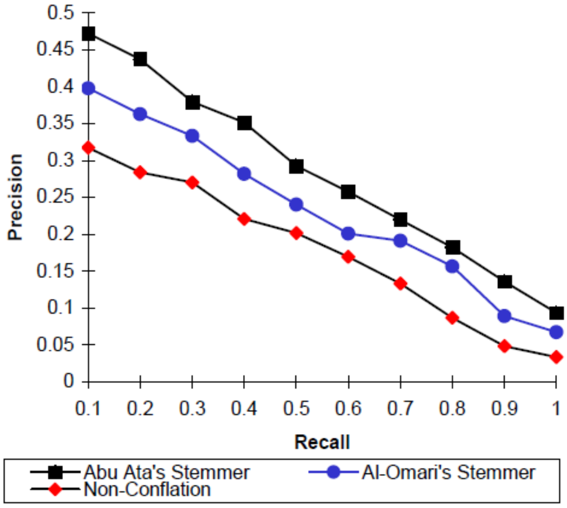


Fig. 2 Average Recall-Precision Graph for conflation and nonconflation methods on Arabic Texts

REFERENCES

[1] Lovins, J.B. Development of A Stemming Algorithm. *Mechanical Translation and Computational Linguistics*. 1968. **11**(1-2): 22-31.

[2] Harman, D. 1991. How Effective is Suffixing. *Journal of American Society for Information Science*. 42(1): 7-15.

[3] van Rijsbergen, C. J. *Information Retrieval*. London: Butterworths. 1979.

[4] Sembok, T. M.. T, M. Yusoff & F. Ahmad. A Malay Stemming Algorithm For Information Retrieval, *Proceedings of the 4th International Conference and Exhibition on Multi-Lingual Computing*. 1994.

[5] Lennon , M. An Evaluation of Some Conflation Algorithms for Information Retrieval. *Journal of Information Science*. 1981. **3**: 177-183.

[6] Ahmad, F., Mohammed Yusoff, Sembok, T.M.T. Experiments with A Malay Stemming Algorithm, *Journal of American Society of Information Science*. 1996.

[7] Porter, M. F. An algorithm for suffix stripping'. *Program, 14*, 130-137. 1980.

[8] Al-Kharashi, I.A. & Evens, M.W. Comparing words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System. *Journal of the American Society for Information Science*. 1994. **45**(8): 548-560.

[9] Khoja, Shereen. Stemming Arabic Text. http://zeus.cs.pacificu.edu/shereen/research.htmLarkey (2001),

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

[11] Darwish, K. Building a Shallow Morphological Analyzer in One Day. ACL Workshop on Computational Approaches to Semitic Languages.

[12] Chen, A., Gey, F. Building an Arabic Stemmer for Information Retrieval. TREC-2002.

[13] El-Sadany, T.A., Hashish, M.A. An Arabic Morphological System, IBM System Journal. 1989. vol.28(4). Pp 600-612.

[14] Hilal, Y. 1990. Automatic Processing of Arabic Language and Applications. *Proceedings of the Arabic Language Processing Using Computer Conference*. 1990. pp 213-219.

[15] Shahein, H. I. & Youssef, S.A. A Model for Morphology As A Production System. *Proceedings of the Second Cambridge Conference on Bilingual Computing in Arabic and English*. 1990.

[16] Alserhan,H.M. & Ayesh, A.S. An Application of Neural Network for Extracting Arabic Word Roots. *Proceedings of the 10th WSEAS International Conference on COMPUTERS*. 2006.

[17] Mesleh, A. Support Vector Machines Based Arabic Language Text Classification System: Feature Selection Comparative Study, *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*. 2007.

[18] Shquier, Mohammed M. Abu; Sembok, Tengku Mohd T. Word Agreement and Ordering in English-Arabic Machine Translation. *Proceedings of ITSim 2008: International Symposium on IT*. 2008. *Volume 1, 26-28.*

[19] Awajan A. 2003. A Rule-Based Morphological Analyzer of Arabic Word. WSEAS *Transactions on Computers. No. 2. Volume 2.*

[20] Ghwanmeh, S. Effect of Excessive Letter Location in Arabic Lexical Items: A Stemmer Algorithm Approach. WSEAS Transaction. 2011.

[21] Al-Omari, H. *ALMAS: An Arabic Language Morphological Analyzer System*. National University of Malaysia. Bangi, Selangor. 1994.