4-1-2015

A Novel Root Based Arabic Stemmer

Mohammed N. Al-Kabi
Zarqa University

Saif A. Kazakzeh
Yarmouk University

Belal M. Abu Ata
Yarmouk University

Saif A. Al-Rababah
Al-albayt University

Izzat M. Alsmadi
Boise State University

Journal of King Saud University – Computer and Information Sciences (2015) 27, 94–103

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

A novel root based Arabic stemmer

Mohammed N. Al-Kabi [a], Saif A. Kazakzeh [b], Belal M. Abu Ata [b], Saif A. Al-Rababah [c], Izzat M. Alsmadi [d,*]

[a] Faculty of Sciences and IT, Zarqa University, P.O. Box 2000, 13110 Zarqa, Jordan
[b] CIS Department, IT & CS Faculty, Yarmouk University, 21163 Irbid, Jordan
[c] Information Systems Department, IT Faculty, Al-albayt University, Jordan
[d] Computer Science Department, Boise State University, Boise, ID 83725, USA

Received 26 December 2012; revised 7 December 2013; accepted 3 April 2014
Available online 21 March 2015

KEYWORDS

Natural Language Processing (NLP);
Computational intelligence;
Stemming;
Information retrieval

Abstract Stemming algorithms are used in information retrieval systems, indexers, text mining, text classifiers etc., to extract stems or roots of different words, so that words derived from the same stem or root are grouped together. Many stemming algorithms were built in different natural languages. Khoja stemmer is one of the known and widely used Arabic stemmers. In this paper, we introduced a new light and heavy Arabic stemmer. This new stemmer is presented in this study and compared with two well-known Arabic stemmers. Results showed that accuracy of our stemmer is slightly better than the accuracy yielded by each one of those two well-known Arabic stemmers used for evaluation and comparison. Evaluation tests on our novel stemmer yield 75.03% accuracy, while the other two Arabic stemmers yield slightly lower accuracy.

1. Introduction

Semitic languages are mainly used in the Middle East, and North Africa. The Arabic language is currently the most used Semitic language, since it is the native language for more than 290 million people Worldwide (Arabic language, 2015). These Semitic languages use the writing style from right to left. Most Semitic scripts use Abjad style. Abjad is a type of alphabet that omits some or all vowels. Not all Semitic languages use a cursive style (Abjad, 2012; Semitic languages, 2012) like the Arabic language (Arabic language, 2015). Semitic languages use non-concatenative (i.e. discontinuous) morphology to form words which represent a modified version of roots (Non-concatenative morphology, 2012; Semitic languages, 2012). Most of Semitic roots consist of three consonants (Triliteral) (Semitic languages, 2012). Affixes are used by Semitic languages. However, most of the words are formulated by vowels between the root consonants (Semitic languages, 2012). Therefore extracting the Semitic roots of different Semitic words is usually not a trivial process.

The official Arabic language also called Modern Standard Arabic (MSA) or Literary Arabic is widely used in schools, universities, academic establishments, newspapers, radio, TV stations, government agencies...etc. Arabic language is

* Corresponding author.
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.04.001

based on 28 letters, where the shapes of some of these letters are changed according to their location in the word. In addition, these letters can be joined together or written separately based on their location in the word. Several vowel diacritics are used especially in the holy Qu'ran and in classical poetry.

Not all Arabic words used in MSA are native Arabic words which are derived from Arabic three consonants' (i.e. Triliteral) origin. These include for example, the following Arabic words which lack authentic Arabic roots, since they are not derived from native Arabic roots while they are phonetically modified Arabic versions from their origins in other languages: (e.g. Television, "تلفاز، تلفزيون"), (Programmer, "مبرمج"), (Telephone, "تلفون"), (Computer, "كومبيوتر"), (Dictionary, "قاموس"), (Chemistry, "كيمياء"), (Physics, "فيزياء"), (Geography, "جغرافية"), (Lemon, "ليمون"), (Orange, "برتقال").

In natural languages it is normal to find a number of words derived from the same root or stem. Stemming is the process of extracting the root of each word, in order to treat a group of words that are derived from the same root as synonyms, since they suppose to refer to the same concept. However, in reality not all words which are derived from the same root may refer to the same concept. Stemming process is widely used in information retrieval, text mining, text classification. . .etc.

The following four Arabic words (Written, "مكتوب"), (Writings, "كتابات"), (Writer, "كاتب"), (Book, "كتاب") are derived from the same Arabic three consonants triliteral with origin verb (Wrote, "كتب"). They also refer to the same concept. Therefore stemming these four Arabic words is useful for some relevant tasks. On the other hand, the stemming of the following two Arabic words (accountant, "محاسب") and (computer, "حاسوب") which are derived from the same Arabic triliteral verb (counted, "حسب") shows that stemming is not beneficial, since these two Arabic words are not synonyms, and refer to two different concepts. Further, the following four Arabic Words: (Books, "كتب"), (Office, "مكتب"), (Library, "مكتبة"), (Writing, "كتابه") represent four different concepts that are derived from the same Arabic triliteral verb (Wrote, "كَتَبَ"). These examples show that Arabic stemming is not always straightforward where even if an automatic extraction tool is very accurate, when evaluating the semantics, some of the stemming activities are not relevant.

There are two types of stemming, the first type is light stemming which is used to remove affixes (i.e. prefixes and suffixes), while the second type is called heavy stemming (i.e. root stemming) which is used to extract the root of the words and include implicitly light stemming.

In this study, a novel Arabic stemming algorithm is proposed, implemented, and tested. The algorithm applies both the light and heavy (root) stemming techniques on Arabic words to extract the triliteral roots of words. Our Arabic stemming algorithm is not dictionary based. The conducted tests on this stemming algorithm reveal an accuracy of 75.03%. The results are compared with two Arabic stemmers described in previous research papers.

The rest of this article is organized as follows: Section 2 presents the related work, Section 3 presents the methodology adopted in this study, Section 4 presents experiments conducted to demonstrate the validity of the proposed algorithm. Section 5 presents an analysis and a comparison between our stemmer and two known Arabic stemmers. Finally Section 5 presents conclusion and future work.

2. Related Work

Several research papers and projects are proposed developing Arabic stemmers (e.g. Al-Shalabi and Evens, 1998; Khoja and Garside, 1999; Abu-Salem et al., 1999). There are many studies that present examples of Arabic Stemming algorithms and their effectiveness. Most of these studies claim an accuracy which exceeds 85%. It is impossible to verify these claims due to the lack of source codes and the datasets which were used in the testing process.

Chen and Gey (2002) study is not purely dedicated to the construction of Arabic stemming, since it aims to study English-Arabic cross-language retrieval (CLIR). Therefore the paper constructed two Arabic stemmers beside an Arabic stop word list. They used a simple program which is restricted to removing major Arabic prefixes: The ('definite article' (Alif-laam), الـ), and four plural suffixes: (Alif-taa, "ات"), (Alif-nuun, "ان"), (Waaw-nuun, "ون") and (Taa, "ة"). Then they built two stemmers, the first one is called MT-based Arabic stemmer, which uses online Ajeeb machine translation system to translate Arabic words to English. These words are then partitioned into groups or clusters, where each group of Arabic words has a common English stem. Next, the MT-based Arabic stemmer selects the shortest word in the cluster and considers it as an Arabic stem for all the Arabic words in the cluster. The second Arabic stemmer is called light stemmer, where its main task is to remove the top frequently used Arabic prefixes and suffixes. In their study Larkey et al. (2002) constructed and tested a number of Arabic light stemmers. Their tests showed that the effectiveness of information retrieval systems (IRSs) which use the best light stemmers yield much better effectiveness than those that use morphological stemmers attempting to find the Arabic root. They also concluded that using the best light stemmer within an IRS is better than avoiding stemming or using co-occurrence analysis to produce stem classes or using very light stemmers. Many think that light stemming is much easier and more accurate than heavy (root-based) stemming, since light stemming is restricted to strip off predetermined Arabic affixes (prefixes and suffixes) from Arabic words. In reality, in many situations the Arabic affix could be part of the root (e.g., (Governor, "والي")). Therefore the light stemmer should decide whether to remove the affix if it is really an affix, or to keep the affix if it is part of the Arabic root. Nwesri et al. (2005) exhibited in their study three novel techniques to remove Arabic prefixes (i.e. Arabic prepositions and conjunctions) from Arabic words inputted to their light stemmers. Those are Arabic light stemmers which could not be benchmarked with our new root-based stemmer.

Most of the Arabic words are derived from triliteral Arabic roots. However, there are very few quadri-literal Arabic roots relative to the number of triliteral Arabic Roots. Kanaan et al. (2004) presented a novel stemming algorithm dedicated to Arabic words derived from quadri-literal Arabic roots only and used a limited set consisting of 145 Arabic words. Stemmer of Kanaan et al., 2004 yields 95% accuracy. Our study is completely different Kanaan et al., 2004 study in data size which is much larger, and their study is restricted to Arabic words derived from quadri-literal Arabic roots, while this one designed for Arabic words is derived from triliteral Arabic roots.

Taghva et al. (2005) study presents the construction of a heavy (root-based) stemmer which does not rely on any dictionary of Arabic roots. Authors claimed that the effectiveness of their Arabic stemmer is equivalent to the known stemmer of Khoja and Garside (1999). In addition, they found that the ability of the root-based stemmer to find the right Arabic root is not an essential issue in monolingual Arabic information retrieval. Most of the studies related to Arabic stemmers are either based on a dictionary of Arabic roots or use a set of rules to identify the verb patterns of the Arabic words in order to find the Arabic roots. These stemmers are accurate but consume a lot of the computation resources of the computer system as Al-Serhan and Ayesh (2006) study claimed. Therefore those researchers present an Arabic stemmer based on neural networks, which is characterized by its efficiency and effectiveness. They also claimed that their novel stemmer capabilities are restricted to finding the root of Arabic words derived from triliteral Arabic roots. The stemmer is limited to Arabic words which consist of no more than five Arabic alphabets.

A novel Arabic morphological analysis method is presented by Al-Sughaiyer and Al-Kharashi (2006). The main idea of their novel method is based on verb pattern similarity of words derived from various Arabic roots. Their method is characterized by its simple computation, and its accuracy.

Another Arabic root-based stemmer is proposed by Al-Shalabi et al., 2007. Their stemmer is characterized by its capability to find Arabic triliteral, quadri-literal and penta-literal roots. The effectiveness of their stemmer is 95%, but their study does not refer to the dataset they used, and the stemmer is not offered online to the public. Therefore it is excluded from the benchmarking of this study.

Momani and Faraj (2007) presented another novel Arabic root-based stemmer to extract triliteral Arabic roots with a 73% accuracy using a dataset of more than 1500 Arabic words. They presented their Arabic stemmer with preliminary examples.

Similar to Porter stemmer popularity for English, Khoja stemmer (Khoja and Garside, 1999) got popular for Arabic stemming through many relevant citations. One of these attempts to improve over Khoja stemmer was presented by Kchaou and Kanoun (2008). They adopt two dictionaries of Arabic roots, one for normal Arabic roots, and the other dictionary is for radical Arabic roots, while Khoja stemmer was based only on one dictionary for Arabic roots. Authors tested their stemmer using 200,000 Arabic words, where they claimed 98% of accuracy.

Most of the research studies related to reducing Arabic words to their stems or roots in information retrieval and Natural Language Processing (NLP) concentrate on the construction of Arabic stemmers whether they are light or heavy (i.e. root-based) (Mustafa, 2002; AI-Sawadi and Khayat, 1996). In addition, lemmatization algorithms are used to obtain the roots, where lemmatizers are more robust than their counterparts since they depend on morphological analysis and vocabulary usage. Al-Shammari and Lin (2008a,b) presented the first Arabic lemmatizer. Authors claimed that their tests showed that their lemmatization algorithm is better than Khoja's Arabic root-based stemmer when these two different algorithms are used to cluster Arabic text documents.

Al-Shammari and Lin (2008a,b) present a new Arabic stemmer called Educated Text Stemmer (ETS). ETS is characterized by its efficiency, since it does not rely on any root dictionary, so it needs less storage space and needs less computational time relative to its counterparts. They also claim that their stemmer is effective and better than others, since for example it uses Arabic stop-words which are neglected by other stemmers to improve the extracted stems. In addition, ETS was capable to identify Arabic nouns and verbs.

Ghwanmeh et al. (2009) present another Arabic root-based algorithm based on the Arabic morphological patterns. The capability of the proposed stemmer by Ghwanmeh et al. (2009) is restricted to native Arabic words that consist of four or more Arabic alphabets. However, this case was treated very well by Khoja Arabic stemmer (Khoja and Garside, 1999) as well in our new stemmer. The stemmer proposed in (Ghwanmeh et al., 2009) checks the length of the inputted Arabic word to determine whether to proceed with necessary steps to extract the Arabic root, or to leave the word as is, if the input length is less than 4. When the evaluated Arabic word is of length that exceeds 3, the algorithm starts with normalization of some its Arabic letters and then starts stripping off prefixes and suffixes. Afterward their proposed algorithm starts matching the extracted word with 81 Arabic triliteral verbs patterns (Forms, "الأوزان"). Those Authors tested their algorithm using a dataset of Arabic words extracted from a corpus of 242 abstracts from the proceedings of the Saudi Arabian national computer conferences. Tests of their Arabic stemmer yield an accuracy of 95%. (Ghwanmeh et al., 2009) stemmer is benchmarked in this study, the dataset they used is not adopted in this study since it is limited and restricted to computer-based topics, while ours include the used Arabic words which covers different aspects of our life.

Hmeidi et al. (2010) study exhibits a novel bigram-based Arabic stemming algorithm. Authors used two similarity measures (Manhattan and Dice). They tested their algorithm on the Holy Qu'ran and a corpus of 242 abstracts. They claimed that their algorithm was capable to extract triliteral, quadriliteral, pentaliteral, hexaliteral, and heptaliteral Arabic roots. Tests of their stemmer revealed that using bigram with Dice measure yields better Arabic roots than using bigram with Manhattan distance measure.

Abu Ata and Al-Omari, 2014 in 2014 paper proposed an Arabic stemmer dedicated to different Arabic dialects. They describe in their study a novel rule-based algorithm to extract stems from textual Arabic Gulf dialect.

Boubas et al. (2011) study exhibits a novel Arabic stemming algorithm which uses genetic algorithms and verbs pattern matching. This algorithm is based mainly on machine learning system and Arabic morphological rules or patterns. They produced an Arabic morphological analyzer capable to generate the Arabic root for any stream of Arabic words.

All our attempts to get all those stemmers listed in this section are failed to get more Arabic root-based stemmers to benchmarked with our new Arabic stemmer.

3. Methodology

In this study, a novel Arabic stemmer is presented to extract the trilateral Arabic roots from Arabic words derived from trilateral roots. The proposed stemmer is based on light and heavy (root-based) stemming methods. C#.NET language is used to implement our new proposed Arabic stemmer.

Afterward this algorithm is tested and its outputs are compared with the outputs of two other Arabic root-based stemmers (Khoja, (Khoja and Garside, 1999), and the stemmer proposed by Ghwanmeh et al. (2009)).

The comparison was restricted to two selected Arabic stemmers since most of the proposed Arabic stemmers presented previously in the literature are not offered to the public to be tested, except those of Khoja and Garside (1999), and Ghwanmeh et al. (2009). One of the authors in our paper was the developer of the stemmer in the second one (i.e. Ghwanmeh et al. (2009)). We checked Arabic stemming resources presented in (http://sites.google.com/site/nlp4arabic/), where Al-Stem Stemmer and Alex's version of Arabic Stemmer are Perl stemmers that were run and found that they cannot be used in this study, since they are light Arabic stemmers.

Light-based stemming algorithm is concerned with the removal of the affixes from the inputted words. Our new Arabic stemmer removes several predetermined Arabic affixes. Several examples of these Arabic affixes are shown in Table 1.

El-Affendi (2002) indicates in his study that the total number of Arabic roots is approximately 9464 roots. Triliteral Arabic roots constitute around 70% of the total number of Arabic roots, while 30% of the total number of Arabic roots is classified under quadri-literal Arabic roots. Sawalha and Atwell (2009) used in his study 2730 verb patterns and 985 noun patterns.

On the other hand the root-based stemming is based on comparing the Arabic word under consideration with Arabic triliteral verbs patterns (patterns, "الأوزان"), which are selected depending on the number of the letters in the word. By comparing the word to that specific verb (pattern, "وزن") we can derive the root of the word. Several examples of those Arabic triliteral verbs patterns (patterns, "الأوزان") are shown in Table 2.

Table 1 A sample list of Arabic affixes removed by our stemmer.

	1 Letter	2 Letters	3 Letters
Prefixes	(Alif, "ا"), (Waaw, "و"), & (Yaa', "ي")	(Alif-laam), "ال"), (Siin-nuun, "سن"), (Faa'-alif, "فا"), (Kaaf-taa, "كت"), & (Yaa-Alif, "يا")	(Waaw-alif-laam, "وال"), (Kaaf-alif-laam, "كال"),(Baa'-alif-laam, "بال"), & (Waaw-siin-taa, "وست")
Suffixes	(Yaa, "ي"), (Taa, "ت"), & (Laam, "ل")	(Haa'-nuun, "هن"), (Kaaf-nuun, "كن"), (Haa'-miim, "هم"), (Alif-taa, "ات") & (Taa-haa', "ته")	(Yaa-Alif-Taa, "يات"), (Kaaf-Miim-Alif, "كما"), & (Haa'-Miim-Alif, "هما")

Table 2 Arabic triliteral verbs' patterns.

Arabic word	Arabic verb pattern	Arabic triliteral verb
(School, "مدرسة")	(Maf'ala, "مفعلة")	(Studied, "درس")
(Forgiveness, "استغفار")	(Estefa'al, "استفعال")	(Forgave, "غفر")
(They are Looking, "ينظرون")	(Yaf'alon, "يفعلون")	(Looked, "نظر")

3.1. Stemming algorithms

This section introduces the algorithm of our proposed Arabic stemmer. Each inputted Arabic word to this stemmer has to proceed in three-phases. These three phases are described below. The first-phase is dedicated to removing Arabic affixes; second-phase is dedicated to identifying the verb pattern of each evaluated Arabic word, while the third-phase is dedicated to refining the proposed Arabic root. Fig. 1 exhibits the pseudo code of our proposed Arabic stemming algorithm to extract triliteral Arabic verbs.

The following subsections exhibit a detailed discussion of some of the essential steps shown in Fig. 1.

3.1.1. Removing Arabic affixes (prefixes and suffixes)

Arabic words that are used as inputs to this stemmer first have to be normalized. For example, the following three different shapes of the first Arabic alphabet (Alif, "ا, أ, إ, آ") will be normalized to (Alif, "ا").

In this phase it is essential for the proposed stemming algorithm to remove the real prefixes and suffixes. Consider the following Arabic word (Adults, "بالغون"), where the blind removal of the Arabic prefix (Baa'-alif-laam, "بال") will lead to a failure to find the right Arabic root (Reach, "بلغ"), since two letters are removed from the root (Reach, "بلغ").

Our proposed stemmer first attempts to identify Arabic affixes with different lengths as shown in Table 1, in order to remove these affixes. So in the first phase of our stemmer appropriate affixes are tested and eliminated from inputted Arabic words. Our stemmer removes the affixes after considering the length of the word and the length of the affix to control affix elimination process to yield better roots. For instance, consider the following three Arabic words: (The Reformers, "المصلحون"), (The Products, "المنتجات"), and (The Libraries, "المكتبات"). First our stemmer removes the definite article (Alif-laam), "ال") from those three words, and removes the suffix (Waaw-nuun, "ون") from (Reformers, "مصلحون"), and removes the suffix (Alif-taa, "ات") from (Products, "منتجات"), and (Libraries, "مكتبات"). Prefix and suffix removal converts the three Arabic words to the following Arabic words: (Reformer, "مصلح"), (Product, "منتج"), and (Office, "مكتب"). So the first phase of this stemmer yields affix free Arabic words. Moreover, we should notice that the semantic of the Arabic words (Reformer, "مصلح"), (Product, "منتج"), and (Office, "مكتب") is correct, which means that the removal of prefixes and suffixes was correct.

3.1.2. Arabic verb pattern identification

In the second phase the stemmer attempts to extract the correct Arabic root. The correctness of each extracted Arabic root by this stemmer is based mainly on identifying the right root pattern for the inputted Arabic word.

In this phase, we compare the output of the first phase to a set of verbs (patterns, "أوزان") in order to extract the right root. The main task in this phase is to identify the verb (pattern, "وزن") of the output of the first phase, by matching the output to a number of verbs (patterns, "أوزان") which have similar word lengths. Afterward a matching between the corresponding Arabic letters in the extracted word and pattern is conducted, where the following three Arabic letters (Faa', "ف"), (Ayn, "ع"), and (Laam, "ل") within the Arabic pattern

Input: A text file that contains the Arabic words

Output: Arabic Triliteral Verb/Verbs

1. Remove Arabic prefix(es) from each word
2. Normalize 3 shapes of (Alif, "أ، إ، آ") to (Bare Alif, "ا")
3. Remove suffix(es) from each word
4. Determine word length after removing affixes (prefixes and suffixes)
5. Identify Arabic patterns having same lengths to word length in step 4.
6. Compare each pattern identified in step 5 with extracted word from step 3
7. Select the closest pattern:
   a. Choose the pattern from the set of Arabic patterns having same lengths to word length which has the highest number of common Arabic letters with the Arabic word extracted from step 3.
   b. Determine the pattern which has the largest matching corresponding letters with the generated word from step 3 which is considered as the right pattern, where the corresponding Arabic letters within the extracted word from step 3 will not be compared with three Arabic letters (Faa', "ف"), (Ayn, "ع"), (Laam, "ل") within the pattern under consideration.
8. Eliminate all matched letters in step 7. The Arabic letters of the Arabic word extracted from step 3 which corresponds to the Arabic letters (Faa', "ف"), (Ayn, "ع"), and (Laam, "ل") in the selected pattern (found in step 7.a) are selected to constitute the extracted Arabic root.
9. Refine the extracted Arabic root by converting some of the Arabic letters.

Figure 1 Pseudo code of our proposed Arabic stemming algorithm.

are excluded from this similarity matching process. The pattern which achieved the highest matching will be considered by our algorithm.

The right Arabic root will successfully be extracted if the stemmer succeeds at this phase to identify the right verb (pattern, "وزن"). Fig. 2 presents the matching between the outputs of the previous phase (Reformer, "مصلح"), (Product, "منتج"), and (Office, "مكتب") and the verb pattern (''Maf'al", "مفعل"). The main task of this Arabic stemmer is to extract the three consonants (Triliteral) Arabic verb from which the original word is derived. All patterns used are derived from the Arabic trilateral verb (''Fa'ala", "فعل").

Identifying the right Arabic pattern for an Arabic word leads to extracting the right Arabic root by simply extracting the corresponding three Arabic letters within the preprocessed word to the following three Arabic letters (Faa', "ف"), (Ayn, "ع"), (Laam, "ل") in the identified pattern. In other words the Arabic root can be extracted simply by eliminating the matched Arabic letter/letters between the pattern and the extracted word from the first phase. As an example, our stemmer will identify the source (''Maf'al", "مفعل") as a verb pattern for the extracted Arabic words: (Reformer, "مصلح"), (Product, "منتج"), and (Office, "مكتب") from the previous phase. Thus in this case the extracted Arabic triliteral roots are: (Reformed, "صلح"), (Produced, "نتج"), and (Wrote, "كتب").

In this phase, the system identifies all the verb patterns that have the same length as the resulted Arabic word from applying the first three steps of the above algorithm. Then, our stemmer starts matching the corresponding letters of the resulted word and each candidate verb pattern. The pattern which has the largest matching corresponding letters is one used by the stemmer to extract the Arabic root. As illustrated in Fig. 3, after finding the right pattern (''Maf'al", "مفعل") the system will eliminate similar letters except main letters. By

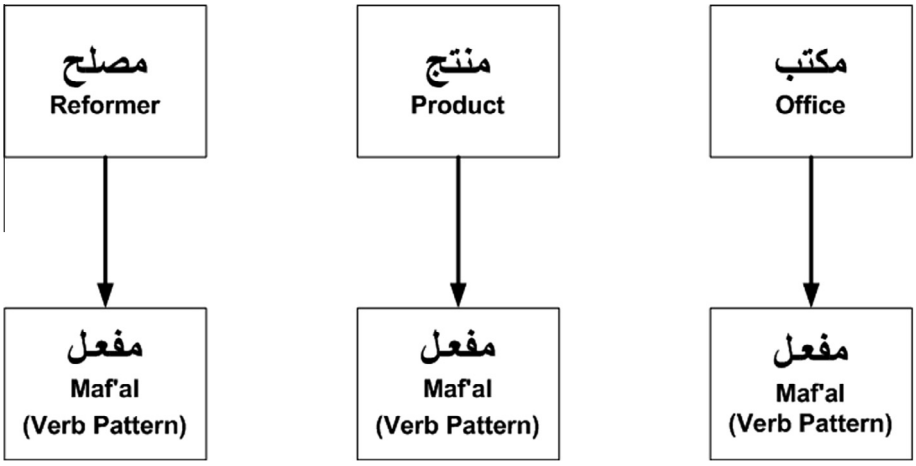

Figure 2 Root extraction process.

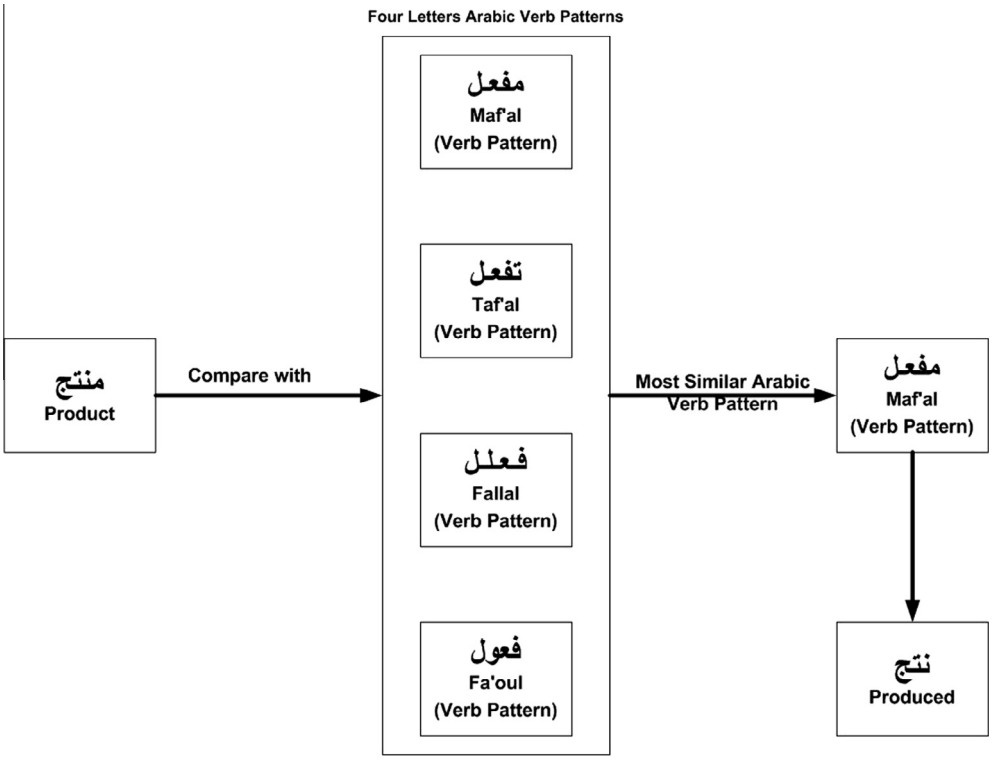

Figure 3 Arabic verb pattern matching and root extraction process.

doing so, the letter "م" will be removed, and the letters that correspond to the main letter will be returned in the same order as the source. The output of this process is then the root (produced, "نتج") for the evaluated word.

To show how our Arabic stemmer extracts the Arabic root of inputted Arabic words consider the 9-letter Arabic word (The Forgiveness, "الإستغفار"). The first problem that the stemmer has to remove the prefix, normalize some Arabic letters, remove suffix, and then determine the length of the Arabic word after stripping off prefixes and suffixes. Identify word length will lead to identify verb patterns with equal lengths. Select the appropriate pattern from the set of 7-letters patterns like (Ef'ta'le, "افتعالي"), (Estf'ta'le, "استفعال"), (Enf'a'le, "انفعالي"), etc. for the resulted word (Forgiveness, "إستغفار") after applying the first three steps. To solve this problem and select the appropriate pattern of this word the stemmer starts comparing corresponding Arabic letters in each of the candidate patterns and the input Arabic word (Forgiveness, "إستغفار").

Letter position	7	6	5	4	3	2	1
Pattern (Ef'ta'le, "افتعالي")	ي	ل	ا	ع	ت	ف	ا
Arabic word	ر	ا	ف	غ	ت	س	ا

The above comparison between the candidate pattern (Ef'ta'le, "افتعالي") and the resulted Arabic word (Forgiveness, "إستغفار") yields two matches at position 1 and 3. Similarly our stemmer starts another comparison between the candidate pattern (Enf'a'le, "انفعالي") and the resulted Arabic word (Forgiveness, "إستغفار") yields 1 match at position 1 as shown below:

Letter position	7	6	5	4	3	2	1
Pattern (Enf'a'le, "انفعالي")	ي	ل	ا	ع	ف	ن	ا
Arabic word	ر	ا	ف	غ	ت	س	ا

Similarly our stemmer starts another comparison between the candidate pattern (Estf'ta'le, "استفعال") and the resulted Arabic word (Forgiveness, "إستغفار") yields four matches at positions 1, 2, 3, and 6 as shown below:

Letter position	7	6	5	4	3	2	1
Pattern (Estf'ta'le, "استفعال")	ل	ا	ع	ف	ت	س	ا
Arabic word	ر	ا	ف	غ	ت	س	ا

Therefore the stemmer in such cases will select the Arabic pattern (Estf'ta'le, "استفعال"). Next the stemmer starts to extract non-matched Arabic alphabets from the resulted Arabic word (Forgiveness, "إستغفار"), and that means (Ghayn, "غ"), (Faa', "ف"), and (Raa', "ر") constitute the Arabic root (Forgive, "غفر").

One of the cons of our new Arabic stemmers proposed in this study is its incapability to extract the correct Arabic roots from Arabic words whose lengths are less than 4 characters, and could not treat vowels properly in those short words. So the present version of our algorithm is incapable to extract the correct root from the following two Arabic words: (You see, "ترى"), (she saw, "رأت"), and output them as is. This problem should be considered in the enhancement of this stemmer in the future.

3.1.3. Root decision evaluation

Up to this phase, the system has the root of the given word. However, in Arabic language there are some letters that must be drawn differently when they come at the end or the middle of the word. These letters should be adjusted to have the word correctly displayed. For instance, the letter (Waaw with Hamza above, "ؤ") in the middle of the word should be transformed to (Alif, "أ"), while the same letter at the end of the word should be transformed to (Alif, "ى"). Applying this process will correct most of the generated roots.

4. Experimental analysis

As mentioned earlier, we have implemented our proposed stemming algorithm using C# .NET programming language. The system accepts a text file that includes the Arabic words and produces the roots of those words. Examples of the systems' output results are shown in Table 3. The following subsections show the test collection used to test our Arabic stemmer, beside the results of these tests.

4.1. The test collection

The research projects in this field lack a gold standard set to be used to carry benchmark tests of different Arabic stemmers. Therefore a dataset consisting of 6081 Arabic words derived from native Arabic triliteral verbs is constructed to evaluate our proposed Arabic stemming algorithm relative to the other two stemmers. Those include singular, dual, and plural Arabic words (nouns and verbs) which are derived from triliteral Arabic roots.

4.2. Results

The results of the tests on our novel algorithm yield an accuracy of 75.03% of the whole collection. We have compared the results of the tests on our proposed stemming algorithm with the results of tests on Ghwanmeh et al. (2009) Arabic stemmers using the same test collection. Those two stemmers (Khoja and Garside (1999) and Ghwanmeh et al. (2009)) yield accuracies of 74.03% and 67.40% respectively. Results of tests of our stemmer have slightly exceeded Khoja stemmer. Fig. 4 visualizes these results.

The accuracy of this Arabic stemmer may seem at a first glance lower than the accuracies of other Arabic stemmers reported in previous studies. One of these is: Ghwanmeh et al. (2009) which claims 95% accuracy, but within our study it yields an accuracy of 67.40%. This is due to differences in size and type of the datasets used to test these stemmers. Therefore there is a need to construct a standard Arabic dataset for Arabic stemmers to be used to benchmark different Arabic stemmers.

Table 3 Proposed stemmer' output results.

Inputted Arabic word	Number of letters	Extracted Arabic triliteral verb
(The noise, "الضجة")	5	(Noised, "ضجج")
(Welding, "ملتحمون")	7	(Welded, "لحم")
(The employments, "التعيينات")	7	(Employed, "عين")
(Will Send you, "سيرسلونكم")	9	(Sent, "رسل")

4.3. Analysis

In this section, an analysis of the effectiveness of our novel stemmer is conducted using 5176 Arabic words of different lengths. Major attributes for stemmers quality are the prediction accuracy of words' stems. Section 4.2 presents the overall output results of comparing the accuracy of the three Arabic stemmers (i.e. Khoja and Garside (1999), Ghwanmeh et al. (2009) and our proposed stemmers) under consideration. In this section the tests on the three Arabic stemmers will be conducted according to the length of the input Arabic word.

This section describes the comparison results in more details. Experiment is divided into different categories based on word length. The summary results are shown in bar charts in Fig. 5.

Our test collection has 677 words of four letters. The Khoja and Garside (1999) algorithm yields 69.2% accuracy, followed by our and Ghwanmeh et al. (2009) stemmers with 69.1% and 55.2% accuracies respectively. Fig. 5 presents the accuracy for the three stemmers to extract Arabic roots from four letters Arabic words. Also our test collection has 1071 Arabic words of five Arabic letters. Our algorithm yields 71.4% accuracy, followed by Khoja and Garside (1999) and Ghwanmeh et al. (2009) stemmers with 65.1% and 52.2% of accuracies respectively. Fig. 5 shows the five letters results for the three stemmers.

The test collection has 845 words of six letters. Our algorithm yields 71.8% accuracy, followed by Khoja and Garside (1999) and Ghwanmeh et al. (2009) stemmers with 71.4% and 63.3% of accuracies respectively. Fig. 5 shows the six letters results for the three stemmers. The test collection has 733 words of seven letters. The Khoja and Garside (1999) algorithm yields 84.3% accuracy, followed by our and Ghwanmeh et al. stemmers with 81.9% and 77.8% of accuracies respectively. Fig. 5 shows the seven letters results for the three stemmers. The test collection has 1850 words of eight letters. Our algorithm yields 77% accuracy, followed by Khoja and Garside (1999) and Ghwanmeh et al. (2009) stemmers with 76.5% and 75.5% of accuracies respectively. Fig. 5 shows the eight letters results for the three stemmers.

Fig. 5 shows that our stemmer effectiveness to extract Arabic triliteral roots from 5, 6 and 8 Arabic letters- words is better by overall results in comparison with the other two Arabic stemmers. Fig. 5 shows that Khoja and Garside (1999) stemmer effectiveness to extract Arabic triliteral roots from words of 4 and 7 letters is the best in terms of prediction accuracy. This clearly reveals that we still need to work on stemmer optimization to work with all word sizes.

4.4. Stemmer output analysis

Using stemmers leads to two types of errors (over-stemming and under-stemming). Over-stemming errors occur when words that refer to distinct concepts are stemmed to the same root. Consider the following two Arabic words (feet, "اقدام") and (Introduction, "مقدمة") which refer to two different concepts, but most probably stemmed to one Arabic triliteral verb (Presented, "قدم"). Under-stemming errors occur when

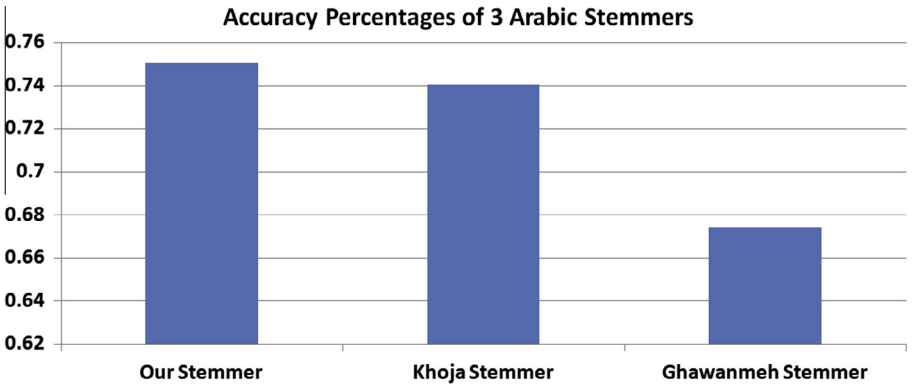

Figure 4 Stemmer results' comparison.

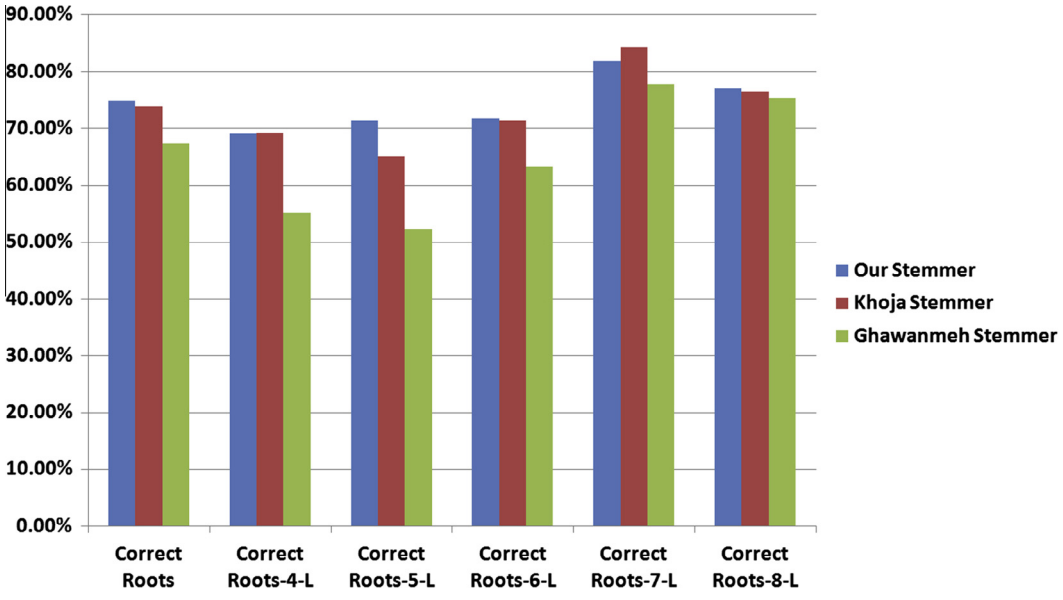

Figure 5 Variable length words' analysis.

words of the same concept are stemmed to different roots. Consider the following two Arabic words (mobile, "نقال") and (mobile, "جوال") which refer to the same concept, but they stemmed to two different Arabic triliteral verbs: (transferred, "نقل") and (Toured, "جال"). Table 4 shows examples of under-stemming (aka False Negative) errors of our stemmer.

Table 5 shows examples of over-stemming (False Positive) errors of our stemmer. Note that, under-stemming and over-stemming errors produced by our stemmer may or may not be produced by the other two stemmers. We can view these types of errors as a general Arabic language phenomena and not dependant on correct or wrong stemming.

Considering Table 4, we can easily notice that all the words have the suffix (for, "لـ") which is the cause of the wrong stemming cases, in which the algorithm removes the first letter of this suffix but not the second one since it is considered as an original letter in most cases. On the other side, in Table 5, the problem of over stemming (false positive) cases is caused by the same reason of removing an original letter in the comparison phase that considers a wrong shape to apply the heavy stemming on.

Table 4 Examples of under-stemming (False Negative) errors.

Inputted Arabic word	Our stemmer output (under-stemming errors)	Khoja stemmer output	Ghwanmeh stemmer output
(For the reports, "للتقارير")	لتقارير	قور	قرر
(For the consumer, "للمستهلك")	هلك	هلك	لمستهلك
(For the computers, "للحواسيب")	لحواسيب	للحواسيب	حسب
(And the control, "والسيطرة")	سيطر	طرة	سطر
(In the region, "بالمنطقة")	منطق	نطق	نطق
(The harshness, "القساء")	قساء	قسى	قسأ

Table 5 Examples of over-stemming (aka False Positive) errors.

Inputted Arabic word	Our stemmer output (over-stemming errors)	Khoja stemmer output	Ghwanmeh stemmer output
(His arguments, "محاوراته")	حار	حور	محاوراته
(The legacy, "التراثية")	راث	رثي	راث
(your listening, "استماعك")	معك	مبع	استماعك
(Vacations, "إجازات")	جزأ	أجز	جاز
(Coral, "مرجان")	رجن	رجن	مرج
(Recipes, "وصفات")	صفا	صفى	وصف

The two competitive Arabic stemmers Khoja and Garside (1999) and Ghwanmeh et al. (2009) yield lower accuracy than our Arabic stemmer when the lengths of input Arabic words are of: 4, 5, and 8 Arabic alphabets. The accuracies of our Arabic stemmer and Khoja and Garside (1999) Arabic stemmer are equivalent when the word length is 6. Ghwanmeh et al. Arabic stemmer shows low accuracy for 6-letters Arabic words. The Khoja and Garside (1999) Arabic stemmer yields better results for 7-letter input Arabic words relative to the other two Arabic stemmers.

5. Conclusion and future work

In this work, we proposed, developed and evaluated a new Arabic stemmer. Three main processing phases were applied to generate Arabic roots from words. Phase 1 is responsible for removing prefixes and suffixes, Phase 2 is responsible for comparing output to standard word sources or shapes, and phase 3 is responsible for correcting the extracted root. Preliminary experimental results indicated an acceptable accuracy for roots' prediction. We compared our stemmer with two Arabic stemmers, where the same dataset is used. Results showed that our algorithm is better in terms of accuracy in most cases (of different word lengths) in comparison with the other two Arabic stemmers.

We plan to enhance the effectiveness of this stemmer in the future, by trying to accomplish the following: Check the conformance between the removed prefixes and the removed suffixes. Actually, there are some cases in which, removing a certain suffix led us to ignore some other prefixes and not to remove them, and vice versa. Solving this problem may lead to enhancing the effectiveness of our stemmer. Also our stemmer capability is restricted to the extraction of Arabic triliteral roots, and it fails for example to extract quadriliteral roots (i.e. roots with four consonants), so enhanced version should be prepared. Also next version of this should be capable to extract Arabic roots from 2-letters and 3-letters Arabic words, and should be capable to deal with vowels on these short words.

References

Abjad, (2012).“Wikipedia, the free encyclopaedia”. <http://en.wikipedia.org/wiki/Abjad>.

Abu Ata, B., Al-Omari, A. (2014). A Rule-Based Stemmer for Arabic Gulf Dialect. Journal of King Saud University - Computer and Information Sciences (JKSU). (Submitted).

Abu-Salem, H., Al-Omari, M., Evens, M., 1999. Stemming methodologies over individual query words for an Arabic information retrieval system. J. Am. Soc. Inf. Sci. (JASIS) 50 (6), 524–529.

AI-Sawadi, A.D., Khayat, M.G., 1996. An end-case analyzer of arabic sentences. J. King Saud Univ. – Comput. Inf. Sci. (JKSU) 8, 21–52.

Al-Serhan, H., Ayesh, A., 2006. A triliteral word roots extraction using neural network for Arabic, In: The 2006 International Conference on Computer Engineering and Systems, pp. 436–440.

Al-Shalabi, R., Kanaan, G., Ghwanmeh, S., Nour, F. M., 2007. Stemmer algorithm for Arabic words based on excessive letter locations. In: 4th International Conference on Innovations in Information Technology (IIT ‘07), pp. 456–460.

Al-Sughaiyer, Imad A., Al-Kharashi, Ibrahim A., 2006. Rule parser for Arabic stemmer. Lect. Notes Comput. Sci. 2448 (2006), 11–18. http://dx.doi.org/10.1007/3-540-46154-X_2.

Arabic language, (2015).“Wikipedia, the free encyclopaedia”. <http://en.wikipedia.org/wiki/Arabic_language>.

Boubas, A., Lulu, L., Belkhouche, B., Harous, S., 2011. “GENESTEM: A novel approach for an Arabic stemmer using genetic algorithms”. In: International Conference on Innovations in Information Technology (IIT 2011), pp. 77–82.

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.

Eiman Al-Shammari, Jessica Lin, 2008. A novel Arabic lemmatization algorithm. In: Proceedings of the second workshop on Analytics for noisy unstructured text data (and 08). pp. 113–118.

Eiman Al-Shammari, Jessica Lin, 2008. Towards an error-free Arabic stemming. In: Proceedings of the 2nd ACM workshop on Improving non English web searching (iNEWS ‘08). pp. 9–16.

El-Affendi M. A., 2002. An LVQ connectionist solution to the non-determinacy Problem in Arabic morphological analysis: a learning hybrid algorithm. Natural Language Engineering, 8(1), pp. 3–23. Cambridge University Press.

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.

Hmeidi, Ismail I., Al-Shalabi, Riyad F., Al-Taani, Ahmad T., Najadat, Hassan., Al-Hazaimeh, Shaker A., 2010. A novel approach to the extraction of roots from Arabic words using bigrams. J. Am. Soc. Inf. Sci. Technol. (JASIS) 61 (3), 583–591.

Kanaan, G., Al-Shalabi, R., Jaam, J.M., Al-Kabi, M.N., Hasnah, A., 2004. A new stemming algorithm to extract quadri-literal Arabic roots. In: Proceedings of International Conference Information and Communication Technologies: From Theory to Applications, pp456 - 460.

Kchaou, Z., Kanoun, S., 2008. Arabic stemming with two dictionaries. In: International Conference on Innovations in Information Technology (IIT 2008), pp. 688–691.

Khoja, S., Garside, R., 1999. Stemming Arabic Text. Computing Department, Lancaster University, Lancaster, UK.

Larkey, L., Ballesteros, L., Connell, M.E., 2002. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In: SIGIR’02, Tampere, Finland, pp. 275–282

Momani, M., Faraj, J., 2007. A novel algorithm to extract tri-literal Arabic roots. In: International Conference on Computer Systems and Applications (AICCSA ‘07), pp. 309–315.

Mustafa, S.H., 2002. A relational approach to the design of an Arabic lexical database. J. King Saud Univ. – Comput. Inf. Sci. (JKSU) 14, 1–23.

Non-concatenative morphology. (2012).“Wikipedia, the free encyclopaedia”. <http://en.wikipedia.org/wiki/Nonconcatenative_morphology>.

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

Riyad Al-Shalabi, Martha Evens, 1998. “A Computational Morphology System for Arabic”. Computational Approaches to Semitic Languages Workshop, COLING 98, Montreal, Canada. 58–65.

Sawalha, M., Atwell, ES. (2009). Linguistically Informed and Corpus Informed Morphological Analysis of Arabic. In: Proceedings of the 5th Corpus Linguistics Conference (CL2009). University of Liverpool, UK. Lancaster University, University Centre for Computer Corpus Research on Language, University of Liverpool. http://ucrel.lancs.ac.uk/publications/cl2009/,.pp. 1258–1265.

Semitic languages, (2012). “Wikipedia, the free encyclopaedia”. <http://en.wikipedia.org/wiki/Semitic_languages>.

Taghva, K., Elkhoury, R., Coombs, J., 2005. Arabic stemming without a root dictionary. In: International Conference on Information Technology: Coding and Computing (ITCC 2005), pp. 152–157.