Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

BPR algorithm: New broken plural rules for an Arabic stemmer

Hamood Alshalabi [a,b], Sabrina Tiun [a,*], Nazlia Omar [a], Elham abdulwahab Anaam [a], Yazid Saif [c]

[a] Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Malaysia
[b] Sana'a University, Sana'a, Yemen
[c] Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia

ARTICLE INFO

Article history:
Received 2 January 2022
Revised 2 February 2022
Accepted 11 February 2022
Available online 22 February 2022

Keywords:
Arabic broken plural
Arabic root-based
Artificial intelligence
Arabic Root Extraction
Arabic morphological analyzer
Arabic corpus
Arabic text processing

ABSTRACT

One of the most important phases in text processing is stemming, whose aim is to aggregate all variations in a word into one group to aid natural language processing. The morphological structure of the Arabic language is more challenging than that of the English language; thus, it requires superior stemming algorithms for Arabic stemmers to be effective. One of the challenges is the irregular broken plural, which has been a problematic issue in Arabic natural language processing that affects the performance of Arabic information retrieval and other Arabic language engineering applications. Several studies have attempted to develop solutions to irregular plural problems, but the challenge remains, especially in extracting correct Arabic root words. In this paper, the broken plural rule (BPR) algorithm introduces new solutions to solve the problem in which an existing root-based method cannot extract correct roots by using their proposed rules. The BPR algorithm introduces several rules (main rules and subrules) to extract the correct roots of the Arabic irregular broken plural words. To evaluate the effectiveness of the BPR algorithm, we extracted roots from an Arabic standard dataset and applied the BPR algorithm as an enhancement to a root-based Arabic stemmer, ISRI. The obtained results from both evaluations showed encouraging results: (i) Only a few numbers of incorrect roots were stemmed on the large-sized Arabic word dataset. (ii) The enhanced root-based Arabic stemmer, ISRI + BPR, exhibited the best performance compared with the original ISRI stemmer and a well-known Arabic stemmer, ARLS 2. Thus, the proposed BPR algorithm has solved some of the irregular broken plural problems that eventually increase the performance of a root-based Arabic stemmer.

1. Introduction

In any natural language processing system, stemming is a common procedure. The importance of developing a competent stemmer is derived from the fact that stemming can pose a direct impact on the performance of any application in which it is used. Owing to its highly inflected and derivational structure, stemming Arabic terms is an extremely challenging task [10,21,22,28,29,30,32].

Aside from the problems in light and root-based Arabic stemming, broken plural (irregularities of plural) poses a challenge in Arabic stemming. This challenge is due to the irregular pattern of the standard Arabic plural words, which causes difficulties in extracting root words. In the Arabic language (singular, dual, and plural), plurals are divided into two categories: regular and irregular plurals. Plurals are formed by appropriate suffixation as in English: teacher: teachers ('معلم', 'معلمون'). The suffix masculine plural is formed through adding the suffix ('oun', 'ون') to the nominative suffix)'een', 'ين') in the accusative and genitive status.

The feminine plural is formed by attaching the suffix (at', 'ات) to the singular. Irregular or broken plurals are applied frequently to triliteral roots and formed by altering the singular as in English: tooth: teeth ('سنون' , 'اسنان'). Numerous nouns and adjectives have broken plurals (Haywood et al., 1965). In all states, singulars are affected by applying several different patterns that alter long vowels, (ي) ,(و) ,(أ), and (ى), within or outside of the framework of the consonants [24].

[*] Corresponding author.
E-mail address: sabrinatiun@ukm.edu.my (S. Tiun).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.02.006
1110-8665/© 2022 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University.

This paper describes the root-based development via the new broken plural rule (BPR) algorithm by introducing new rules and the subset rule algorithm. The preprocessing contains a new section, which is Arabized removal. The main objective of this study is to combine an Arabic stemmer with a new broken plural rules (BPR) algorithm supported by appropriate terms and conditions. The rest of this article is arranged as follows: Section 2 covers the relevant works, Section 3 contains the ISRI works, and Section 4 indicates the methods used in this study. Section 5 presents the experiments conducted to illustrate the validity of the suggested algorithm, and Section 6 shows the analysis and comparison of our proposed stemmer with two other Arabic stemmers. The conclusion is provided in Section 7.

2. Related works

The term selection for the Arabic language comprises four types of stemming techniques, namely, root-based stemming (heavy stemmer), statistical and hybrid stemmers, light-based stemmers, and artificial intelligence stemming. Each of these methods corresponds to a point on a scale of analysis. Root-based stemmers extract the root of an Arabic word via morphological analysis. Among the most prominent root-based stemmers is that of Khoja and Garside [28] that removes suffixes, infixes, and prefixes. It extracts the roots of words via pattern matching; nonetheless, several flaws and downsides exist, particularly with some words that represent 'broken plurals'. Taghva et al. [34] refined the algorithm developed by Khoja and Garside [28] through eliminating the necessity for a dictionary to extract the root. The ISRI stemmer [34] is considered one of the best algorithms for Arabic stemmers, which is included in the NLTK package. Syarief et al. [33] addressed the ISRI stemmer's weakness in comprehending words with two letters. The stemmer was upgraded by Al-Kabi et al. [6] through introducing additional rules and patterns. Pattern-based stemmer root extraction approaches, which do not require the use of a lexicon, have recently been created [6,31].

The stopword list is a list of words in a text that have no meaning. These words serve merely as syntactic operations and have no bearing on the topic content. They have two unique effects on NLP [12]. Multiple applications include information retrieval IR [18], text categorization [9], ontology construction [4], and many more NLP applications [14,13,3]. A new Arabic light stemmer was developed by Al-Lahham et al. [7] to amplify the precision of previous algorithms. In addition, Alshalabi et al. [11] recommended including a table of suffixes and prefixes to differentiate Arabic stems on the basis of word size instead of relying on morphological word patterns. This enhanced algorithm known as the Dlight Arabic stemmer uses word length to remove suffixes on the basis of the precise stage of stemming (double, quadriliteral, or triliteral roots).

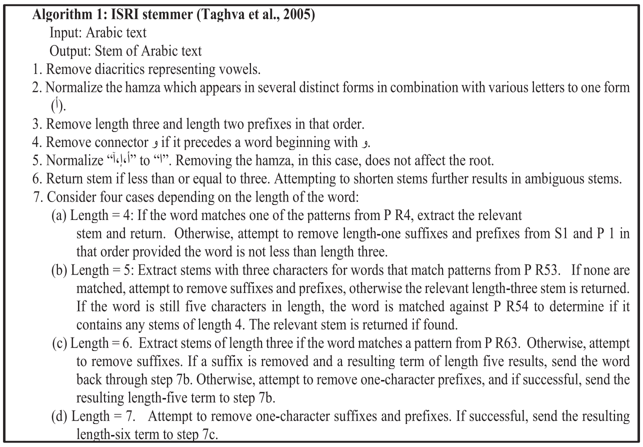
The Arabic language has a large vocabulary and morphological diversity. For given text length and kind, any word will occur less frequently than in English. In accordance with this hypothesis, Arabic datasets will have a larger degree of intrinsic sparseness than equivalent English datasets [23]). Broken plurals comprise ~ 10% of texts in large Arabic corpora [23] and ~ 14% of plurals [20]. For light-stemming algorithms, detecting broken plurals is essential. In modern standard Arabic, irregular (i.e., broken) plural identification is a concern for Arabic stemmers and language engineering applications. Their impact on IR performance has yet to be investigated. Broken plurals are formed by altering the singular as in English (e.g., foot – قدم, feet – اقدام) through the application of interdigitating patterns to stems; singular words cannot be recovered using standard affix stripping stemming techniques [24].

In this section, we refer to the most prominent study on root-based stemming [19]. To generate a stem, light stemming is confined to deleting a small number of Arabic prefixes and suffixes. In addition to extracting the right root, heavy steaming, also known as root-based stemming, entails automatically deleting Arabic prefixes and suffixes. The majority of original Arabic words are obtained from triliteral roots. The problem of handling irregular plural words in Arabic (sands,' رمال 'plural of ' رمل ' and souls, ' ارواح ' plural of ' روح ') is difficult for many of the proposed stemmers as they do not obey particular linguistic rules. For instance, Kchaou and Kanoun [27] developed an Arabic stemmer with two dictionaries that matches the input word with one dictionary for roots and the other for stems and consequently eliminates them. Aljlayl and Frieder [10] developed another stemmer that uses heuristic rules to remove unnecessary letters, such as, and then checks for the remaining word length to look for the extra prefixes and suffixes. To address this issue in a specific domain, some Arabic stemmers have been proposed. For instance, Abuata and Al-Omari [2] proposed a special Arabic stemmer for the Gulf dialect. To extract the triliteral stem from the input words, they used heuristic-based rules. Another study by El-Beltagy and Rafea [22] proposed a domain-specific light Arabic stemmer. The writers first delete the affixes from a word and then look up the remaining letters in the domain corpus text, which requires the user's assistance.

Moreover, a few works on Arabic morphology are worth to be highlighted, such as the work of Yousfi [35] and Fuad (2020). Yousfi [35] introduced a new system for morphological analysis using the surface patterns of the word to be analyzed. This approach requires classification of all Arabic verb roots by using the morphological rules of such patterns in the initial phase. The second phase is the construction of a data base of conjugated surface patterns. This system can analyze all Arabic verbs by decomposing the word to the prefixes, suffixes, and roots. This approach has been tested on a corpus of 4000 verbs, and the results were encouraging. Iazzi et al. [25] also introduced a system for the morphological analysis of the Arabic language. The system is based on the surface patterns of Arabic words. This work aimed to deal with Arabic derived nouns. It was based mainly on the building of a database for the surface patterns of such nouns. To deal with Arabic derived nouns, the article also referred to a previous study by Yousfi [35] for the analysis of Arabic verbs. This approach was tested against a corpus of 2400 Arabic words (400 verbs and 2000 derived nouns), and the obtained results were remarkable.

Fuad et al. (2020) evaluated four state-of-the-art Arabic morphological analyzers and found that identification of broken plural words, specifically the plural of paucity, is lacking. Therefore, Fuad et al. (2020) presented Qillah (paucity), a morphological extension that is built on top of other morphological analyzers and uses a hybrid rule- and lexicon-based approach to enhance the identification of plural of paucity. Two versions of Qillah were developed: one is based on the FARASA morphological analyzer (https://farasa.qcri.org/POS/), and the other is based on the CALIMA Star analyzer (https://github.com/CAMeL-Lab/camel_tools).

Broken or irregular plurals of adjectives and nouns, for example, would not be confused with their single forms [26]. Past tense verbs are distinguished from their present tense counterparts by internal distinctions and affixes [16,5,17,15]. However, none show any list for roots [1]. A light stemmer misses the correct stem of a word and a wrong word that does not exist in the Arabic language. A root-based stemmer also fails to stem the broken plural word form, resulting in extracted root words that deviate from the original word's meaning. We plan to refine a root-based stemmer to extract expected stems by using a new rule-based stemmer for broken plural words to solve this problem.



```
Algorithm 1: ISRI stemmer (Taghva et al., 2005)
   Input: Arabic text
   Output: Stem of Arabic text
1. Remove diacritics representing vowels.
2. Normalize the hamza which appears in several distinct forms in combination with various letters to one form
   (أ).
3. Remove length three and length two prefixes in that order.
4. Remove connector و if it precedes a word beginning with و.
5. Normalize "آ،إ،أ" to "ا". Removing the hamza, in this case, does not affect the root.
6. Return stem if less than or equal to three. Attempting to shorten stems further results in ambiguous stems.
7. Consider four cases depending on the length of the word:
   (a) Length = 4: If the word matches one of the patterns from P R4, extract the relevant
       stem and return.  Otherwise, attempt to remove length-one suffixes and prefixes from S1 and P 1 in
       that order provided the word is not less than length three.
   (b) Length = 5: Extract stems with three characters for words that match patterns from P R53.   If none are
       matched, attempt to remove suffixes and prefixes, otherwise the relevant length-three stem is returned.
       If the word is still five characters in length, the word is matched against P R54 to determine if it
       contains any stems of length 4. The relevant stem is returned if found.
   (c) Length = 6.  Extract stems of length three if the word matches a pattern from P R63.  Otherwise, attempt
       to remove suffixes. If a suffix is removed and a resulting term of length five results, send the word
       back through step 7b. Otherwise, attempt to remove one-character prefixes, and if successful, send the
       resulting length-five term to step 7b.
   (d) Length = 7.   Attempt to remove one-character suffixes and prefixes. If successful, send the resulting
       length-six term to step 7c.
```

Fig. 1. Pseudo code of the ISRI algorithm.

3. ISRI stemmer

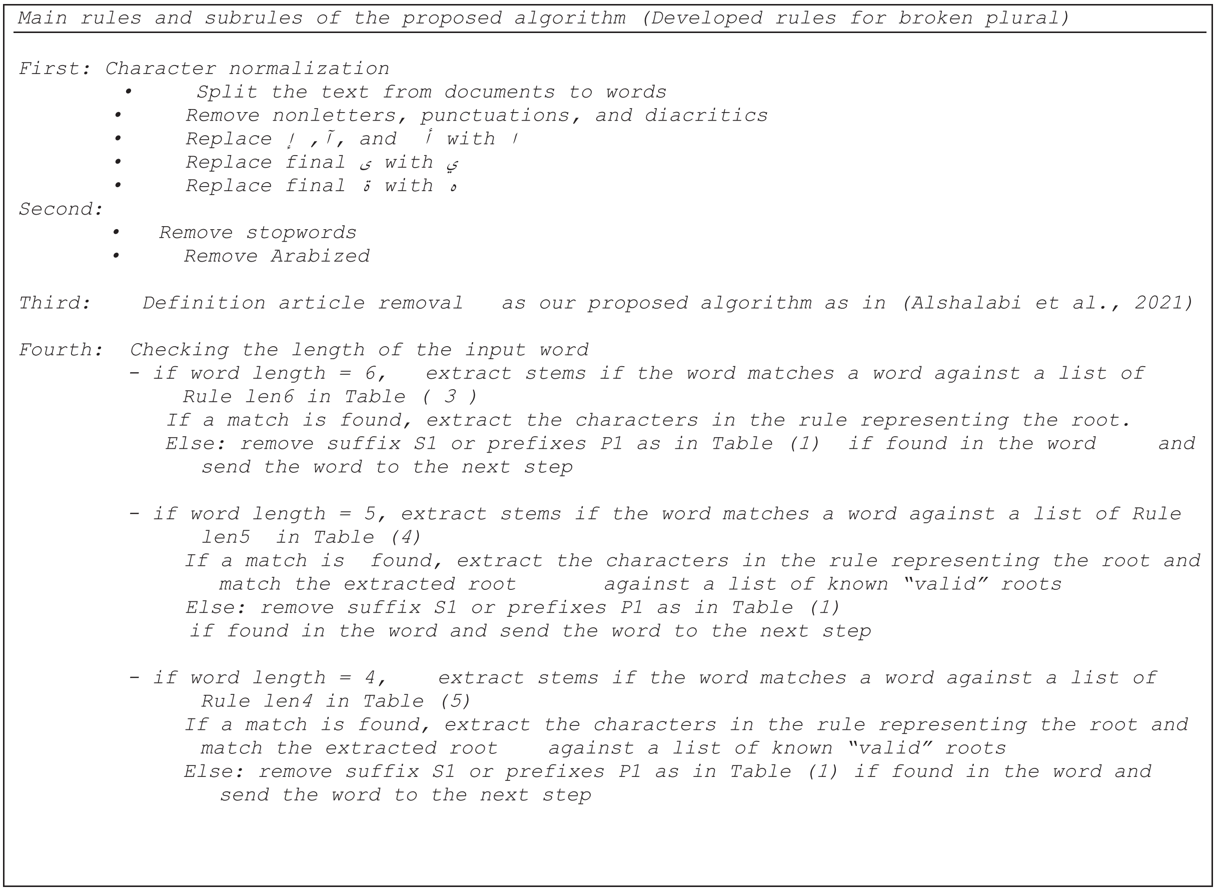
The ISRI stemmer introduces another simulation for linguistic process, which is similar to the Khoja stemmer [28,34]. It starts by normalizing input word, as well as discarding unrelated Arabic characters and diacritics. The normalization step involves unifying the various forms of Hamza to A (Alif ا), which is different from the Khoja stemmer [28]. This step is followed by several decisions to omit prefixes with three or more characters from the normalized word. Afterward, the word is mapped by ISRI to a group of patterns on the basis of its length, but potential suffixes are discarded in the absence of match. The stemming process ends when the word length is reduced by three or less characters. The main difference between Khoja and ISRI stemmers is that the latter never validates roots against any dictionary. ISRI only identifies a minimal word from an input word to be applied in (IR) tasks. Some drawbacks due to absence of dictionary are incorrect extracted root word and a root word consisting of unintelligible characters that disables further processing for linguistic purposes. Fig. 1 lists these steps. In step 7, longer words are trimmed to affixes with single character. Successful attempt leads to a comparison between the resultant shorter term and a range of patterns at varying levels to determine the matching pattern and extract the relevant term or the resulting word that is actually extremely short to serve as a viable stem.

Instances of outcomes, as reported by Taghva et al. [34] are as follows: sets of affixes are reflected in (P3, P2, P1, S3, S2, and S1), while (PR4, PR53, PR54, PR63, and PR64) refer to sets of Arabic suffixes and prefixes. Then, Syarief et al. [33] improved the weakness of the ISRI stemmer in processing words consisting of two letters. From the results of the experiment, the improvements increased the stemmer yield by 7.3%.

4. BPR algorithm: new rules for broken plural

The data used in this study is the standard dataset Text Retrieval Conference (TREC 2002). The pretreatment stage includes text normalization, stopword removal, Arabized detection and extraction, and definition article removal, which we have already mentioned in detail in our previous work on Dlight [11].

Most of the preceding studies [1,28,34,6,8,36] did not focus on root words that contain two letters. For example, the word love احباب in the weight of pattern افعال accords to past methods, the extracted root will be حبب, and this is the wrong root. The right



```
Main rules and subrules of the proposed algorithm (Developed rules for broken plural)

First: Character normalization
          •     Split the text from documents to words
         •     Remove nonletters, punctuations, and diacritics
         •     Replace إ ,آ, and  أ with ا
         •     Replace final ى with ي
         •     Replace final ة with ه
Second:
         •   Remove stopwords
         •     Remove Arabized

Third:    Definition article removal    as our proposed algorithm as in (Alshalabi et al., 2021)

Fourth:  Checking the length of the input word
         - if word length = 6,    extract stems if the word matches a word against a list of
               Rule len6 in Table ( 3 )
             If a match is found, extract the characters in the rule representing the root.
             Else: remove suffix S1 or prefixes P1 as in Table (1)  if found in the word     and
                send the word to the next step

         - if word length = 5, extract stems if the word matches a word against a list of Rule
                len5  in Table (4)
               If a match is  found, extract the characters in the rule representing the root and
                  match the extracted root       against a list of known "valid" roots
               Else: remove suffix S1 or prefixes P1 as in Table (1)
               if found in the word and send the word to the next step

         - if word length = 4,    extract stems if the word matches a word against a list of
                Rule len4 in Table (5)
               If a match is found, extract the characters in the rule representing the root and
                match the extracted root    against a list of known "valid" roots
               Else: remove suffix S1 or prefixes P1 as in Table (1) if found in the word and
                  send the word to the next step
```

Fig. 2. BPR algorithm for Arabic irregular broken plurals.

Table 1
List of prefixes(P1) and suffixes(S1) to be removed.

Set	Description	List
P 1	length one prefixes	ل ,ب, ف ,س, و, ي ,ت, ن ,ا
S1	length one suffixes	ة , ه , ي , ك ,ت, ا, ن

root is حب, the word وداد in accordance with the pattern فعال root will be ودد, and the right root is ود. Therefore, we build subrules and develop a subrule algorithm to solve these problems to find correct root words.

Furthermore, we introduce new solutions to solve the problem in which an existing root-based (Khoja's) method cannot extract correct roots by using the main rule and subrules. For example, in Table 5, Procedure len5 in Rule (6) states that the main rule is "If the word ends with (اء), then three subrules will apply the first subrule (if first char 'ا' and fourth char 'ي'). For example, for the words ('اسوياء','ابرياء'(, the correct roots are ('سوي' ,'بري'). In the second subrule "if first char 'ا' and fourth char (Not 'ي')," such as " أخلاء أطباء أعزاء ", then the root of this word will be two letters, and the new rule result will be " خل' 'عز 'طب'; else, root = first char + second char + third char as the third subrule. For example, the words (بخلاء سعداء) will have the correct roots as ("بخل, سعد ").

In this part, more patterns are developed to extract correct roots by using Khoja's method. New patterns are also added to the Khoja list; thus, we have access to a comprehensive array of Arabic word patterns. This research proposes an algorithm broken plural rule, as shown in Fig. 2. A sample list of proposed new patterns expected from this study is shown in Table 2. To begin our description of our stemmer, we define the sets of prefixes (P1) and suffixes (S1) in Table 1.

A detailed discussion of the algorithm offers a broad review of the stemmer. Stemming is performed in the following order (with more details shown in Tables 3–5).

As a summary of this algorithm, the normalization step in the beginning starts with the letters' normalization by splitting the text from the documents to words; removing nonletters, punctuations, and diacritics; replacing إ ,آ, and أ with ا; replacing final ى with ي; and replacing final ة with ه. We use the normalization step suggested by Larkey et al. [29]. The next step entails the stopword and Arabized removal as in [11] because these words function merely as syntactic operations and have no bearing on the topic matter. These stopwords have dissimilar impacts on NLP. In the third step, the elimination of definition articles as a distinct stage is conducted as a new concept as in [11]. The concept is that we separate the definition articles from the prefixes because, in most circumstances, when the definition article is removed from the word, the stemmer will most likely yield the proper root directly, thus avoiding further word processing that leads to the wrong root. For example, in the word العب (I am play), elimination of definition articles ال in accordance with root-based approaches will lead to a wrong root, and the word will be (عب). By contrast, in our algorithm on definition article removal, for the word (العب), the rule is as follows: if length (word) >= 4 and a word starts with (ل or ا), then remove (ا). In this case, the word (العب) will be (لعب) game , which is the correct root. Further processing for suffix and prefix removal is not needed. Lastly, depending on the length of the input word, we insert several rules as main rules and subrules in the final stage.

Table 2
Patterns for different lengths.

Pattern 6	فعاليل	مفاعيل	افاعيل	فعلائي	افعائي	افعلاء			
Pattern 5	فواعل	فعاله	تفاعل	يفاعل	مفاعل	كفاعل	بفاعل	فعالي	فعائل
	فعالل	افعلاء	فعلاء	افعال	فاعول	افعله			
Pattern 4	فعله	فعال	افعل						

Table 3
Procedures for broken plural words len6.

R-No	Procedures len6	Main rule	Prefixes	middle	suffixes	subrule	example
1	if (fifth char is 'ي') and (third char is 'ا')	مفاعيل افاعيل فعاليل	'م','ت'	'ا','ي'		if (first char is 'م' or 'ت'): root = second char and third char and sixth char else root = first char and second char and fourth char and sixth char	تقاطيع مقاتيل مشاريع قطع قتل شرع = صناديق =صندق
2	if first char is 'ا' and last chars 'اء'	Root = remove prefix's and suffixes	'ا'		'اء'	Root = remove prefix and suffix chars	انبياء اولياء اصفياء = انقياء,ولي نبي صفي
3	if first char is 'م' and second char is 'ت') and (fifth char is 'ئ') and (fourth char is 'ا')	Root = third char and fifth char and sixth char	'م'	'ا','ئ','ت'		Root = third char + fourth char + last char	متفائل= ,فال
4	if (first char any 'ب','ك','ل') and (fifth char any 'ي','ئ') and (fourth char is 'ا')	Root = second char and third char and sixth char	'ب','ك','ل'	"',ا ي','ئ"		Root = second char + third char + last char	كفعائل بفعائل لفعائل : = فعل
5	If the third char is 'ا' and fifth char 'ي' and sixth char 'ن'	Root = first char + second char + fourth char+ 'ا'+sixth char		ا،ي	ن		شياطين رياحين سلاطين=شيطان ريحان
6	if first char 'ا' and third char 'ا' and fifth char 'ي')	Root = second char + fourth char + sixth char	ا	ا،ي			افاعيلفعل =
7	if fourth char 'ا' and suffixes are ('ئي') ':		ا	ا	ئي	if first char, not 'ا' :Root = first char + second char +'ي'+third char if first char is 'ا' :Root = second char + third char	فعلائي عملائي =عميل احبائي اعزائيnew root 2 length حب عز

Table 4
Procedures for broken plural words len5.

R-No	Procedures / len5	**Main rule**	**prefix's**	**middle**	**suffixes**	**subrule**	example
1	if word. End with ('يه') and star with ('ا')	Word = remove prefix's and suffixes + 'ا'	ا		ية		اغنية غنا
2	if third chars 'ا'	Root = first char + fourth char + fifth char		ا			فواعل
3	If second char 'و' and third char 'ا'		ت،ي،م،ن	و،ا		if first char 'ت' or 'ي' or 'ن' or 'م' Root = second char + fourth char + fifth char Else: Root = first char + fourth char + fifth char	تواصل يواصل# فواعل جمع التكسير

4	If fourth char 'ئ' and third char 'ا'		م	ا ،و،		if second char 'و' and first char 'م' Root = first char + third char + fifth char
						if first char 'م' and second char Not !='و'
						Root = second char + third char + fifth char
						if first char Not 'م' and second char Not !='و' and fifth char Not'ل'
						Root = first char + second char + sixth char
						If first char Not 'م' and second char Not 'و' and fifth char 'ل' Root = first char + second char + 'ي' + fifth char
							موائد
							مكائد
							حقائب
							قبائل فصائل قبيل

5	If third char 'ا':		م	ا،	ه	if first char 'م'
						Root = second char + fourth char + fifth char
						if first char Not 'م' and fifth char 'ي':
						Root = first char + second char + fourth
						if fifth char 'ه' and first char 'م'
						Root = second char + third char + fourth char
						else Root = first char + second char + fourth char + fifth char
							مدامع منافع
							حضاري
							معانه مغاره
							سنابل =سنبل

6	If fourth char 'ا' and fifth char 'ء'					if first char 'ا' and (fourth char 'ي')
						Root = second char + third char + fourth charr
						if first char 'ا' and fourth charr (Not 'ي')
						Root = second char + third char
						else: Root = first char + second char + third char
							ابرياء اسوياء
							اجلاء اطباء اعزاء
							بخلاء = بخل

7	if fourth char 'ي' and first char 'ت'	Root = second char + third char + fifth char	ت	ي			تفعيل تحويل
8	if fourth char 'و' and second 'ا'	Root = first char + third char + fifth char					فاعول
9	if fifth char('ي' or 'ى') and third char 'ا':		ي ، م، ا، ،ت		ي، ى	if first char ('ت' or 'ا' or 'ي' or 'م' Root = remove suffix and prefix Else Root = first char + second char + fourth char	معالي تعالى مقالي ينادي صحارى
10	if fourth char 'ا' and(first char 'ا') :	Root = second char + third char + fifth char					افعال املاك# امراض اغصان
11	If first char 'ا') and fifth char 'ه' or ه') and third char Not 'ا'	Root = remove prefix and suffix chars	ا		ه		افعله ازمنه ارغفه ادمغه

5. Evaluation of the BPR algorithm

5.1. Effectiveness of the BPR algorithm

We should evaluate the effectiveness of the BPR algorithm before it is applied to our proposed or any Arabic stemmer. Thus, two types of investigations are conducted. One is to evaluate the effectiveness of the BPR algorithm stemming words through finding the number of incorrect and correct stemmed words. The other is to apply BPR to a well-known Arabic stemmer, the ISRI stemmer, and evaluate how effective the BPR algorithm is through comparing the original ISRI stemmer and the enhanced ISRI (ISRI + BPR). The following is the detailed explanation on how we apply the ISRI stemmer by using our proposed BPR algorithm and expect an improvement in ISRI's performance.

5.2. ***ISRI + BPR:*** *Improved ISRI stemmer using BPR algorithm*

The ISRI stemmer [34] presents another simulation for the linguistic process that is comparable to the Khoja stemmer [28]. It begins by eliminating diacritics and nonrelated Arabic characters from the input word [28]. The normalized word is then mapped to a group of patterns on the basis of its length after a sequence of judgments to eliminate probable prefixes with three or less characters. If no match is found, ISRI looks for possible matches within a group's patterns. When the input word's remaining length is three or fewer characters, the stemming process should end. ISRI does not validate roots against any form of dictionaries, which is another significant distinction from the Khoja stemmer. ISRI is more concerned with locating the smallest possible representation of an input word to retrieve information. The absence of a lexicon has various negative consequences, such as the fact that the extracted roots are not always valid; the root could be a meaningless string of letters. Roots would be untrustworthy for subsequent processing, particularly in language tasks. Stemming proceeds as shown in Fig. 3, and we call the new stemmer as the (ISRI + BPR) stemmer.

The main variance between the Khoja and ISRI stemmers is that the latter never validates roots against any dictionary. The ISRI stemmer only identifies minimal words from an input word to be applied in (IR) tasks. Some drawbacks due to the absence of dictionary are incorrect extracted root words and root words consisting of unintelligible characters that disable further processing for

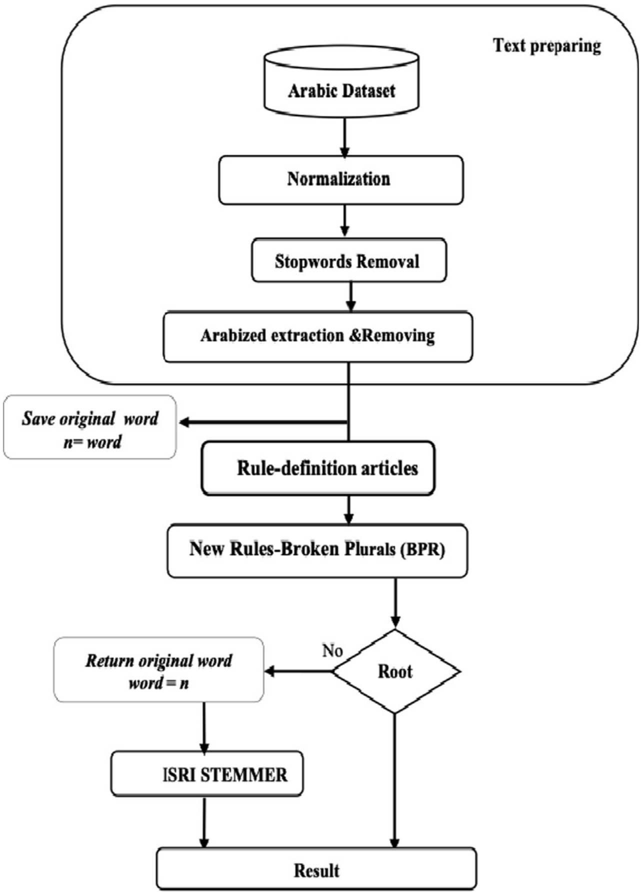


Fig. 3. Diagram for developing the ISRI + BPR stemmer.

linguistic purposes. This method has a shortcoming in extracting the roots of irregular plural words. To solve this problem, the BPR algorithm is the solution. That is, the improvement of the ISRI stemmer by using the BPR algorithm can be used as a benchmark on whether the BPR algorithm really works. Fig. 3 shows the diagram of the enhanced ISRI (ISRI + BPR) stemmer.

This section presents our rule-based development by using new BPRs (ISRI + BPR). The algorithm entails many steps. The preprocessing step involves normalization, stopword removal, and Arabized removal. The next step involves the removal of definition articles and the application of the new BPR, followed by the ISRI stemmer.

5.3. Evaluation of the BPR algorithm on the Arabic TREC dataset

In this study, a few specific evaluations are performed to assess the proposed BPR algorithm. In this evaluation, the stem refers to the word that has been processed by the algorithm. The stem may be a correct root (Correct R) or an incorrect root (Incorrect R). 'Un' refers to the unique stem of all stems, and the nonstem refers to the word that has not been processed by the algorithm. Moreover, all stem denotes all the words that the algorithm has been able to process or stem. Thus, to evaluate the effectiveness of the BPR algorithm beforehand, two types of investigations are conducted. The first is to evaluate the effectiveness of the BPR algorithm stemming words through finding the number of incorrect and correct stemmed words. The second is to apply BPR to the ISRI stemmer and evaluate how effective the BPR algorithm is through comparing the original ISRI stemmer with the enhanced ISRI (ISRI + BPR).

On the basis of the experimental results, the effectiveness of the BPR stemmer have been proved. In the first experiment, the number of incorrect and correct stem words stemmed by the BPR stemmer is determined. From Table 6, the BPR stemmer stems the words on the TREC2002 dataset correctly for almost ¾ of the entire dataset, and only 1/3 are incorrect. This result indicates the ability of BPR to stem words correctly, although the size of the given dataset is large. To be specific (Table 6), BPR produces 7,788,877 stems from the TREC2002 dataset that has 42,127,701 words in total. A total of 5,661,488 stem words are correctly stemmed (Correct R), which are 73% of the total stem words (all stems); only 2,127,389 (or 27%) are incorrect stem words (Incorrect R). In accordance with the result (Table 6), BPR shows the ability to stem words with high performance.

Furthermore, evaluation of the BPR algorithm is performed through applying the BPR algorithm to an existing, well-known Arabic stemmer, i.e., the ISRI stemmer [34]. If the BPR algorithm can enhance the performance of the original ISRI stemmer, it will

Table 6
Output result of the BPR stemmer.

Method	Correct R	Incorrect R	all stems
BPR	5,661,488	2,127,389	7,788,877

Table 7
Correct R, Incorrect R, all stems, and unique stems for ISRI + BPR with two algorithms.

	Correct R	Incorrect R	all stems	Un
ARLS 2	23,325,319	30,814,878	42,454,078	152,358
ISRI	38,552,181	15,588,016	45,281,186	56,990
ISRI + BPR	**41,276,395**	**11,220,651**	**41,282,107**	**55,839**

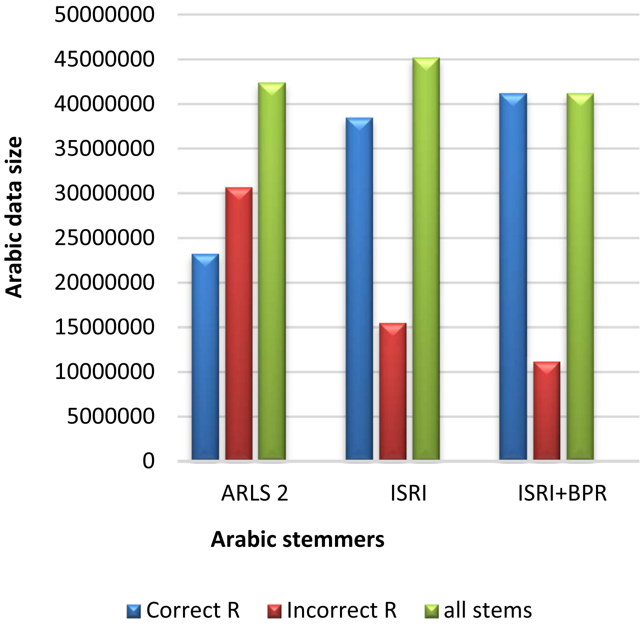


Fig. 4. Comparison of ISRI + BPR's performance with the performance of two well-known Arabic stemmers.

Table 8
Precision, recall, and ICF comparison of two well-known Arabic stemmers and ISRI + BPR.

	Precision(%)	Recall(%)	F-measure(%)	ICF(%)
ARLS 2	43	67	52	72
ISRI	71	81	76	89
ISRI + BPR	**79**	**85**	**82**	**90**

Table 5
Procedures for broken plural words len4.

R-No	Procedures len4	Main rule	**prefix's**	**middle**	**suffixes**	Subrule	example
1	if first char 'ا'	Root = remove prefix	'ا'				انفس أعين #
2	If third char 'ا'	Root = remove middle		ا			فعال): قتال

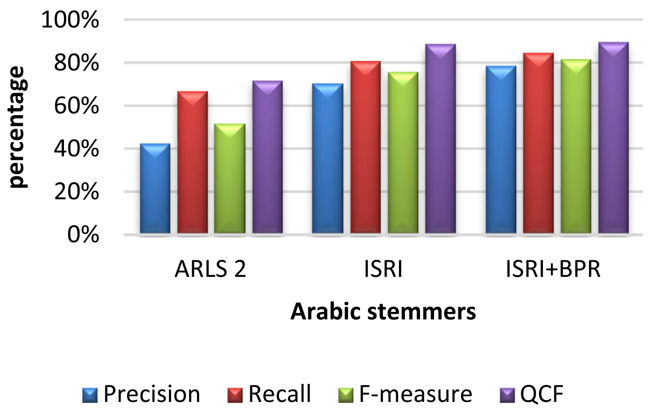


Fig. 5. Comparison of ISRI + BPR's performance with the performance of two well-known Arabic stemmers.

further confirm the ability to stem words with high performance. The results in Table 7 show an evident improvement in extracting correct roots. The BPR algorithm also achieves better results in reducing wrong roots, given that it can analyze the largest number of words that are subject to ISRI + BPR improvement, which enables the algorithm to extract a smaller number of unique queries.

Moreover, ISRI + BPR increases the number of correct roots by approximately 17,951,076 more than ARLS 2 does and 2,724,214 more than the original ISRI stemmer does, as shown in Fig. 4. The F-measure result also indicates a marked improvement.

The results in Table 8 and Fig. 5 show that the ISRI + BPR stemmer increases the stem precision by 36% more than ARLS 2 does and 08% more than the original algorithm, ISRI, does. ISRI + BPR increases the stem F-measure by approximately 30% higher than ARLS 2 does and approximately 06% higher than the ISRI stemmer does.

6. Discussion

In general, the goal of stemming is to discover a word's representative indexing form. For effective information retrieval in Arabic, which is a heavily inflected language, we need good stemming. Currently, no standard strategy for stemming Arabic words exists. However, two broad ways are used to stem Arabic words: extracting the word's root, such as the Khoja stemmer, or simply truncating affixes, such as the Larkey stemmer. The former method has numerous flaws. First, the root dictionary must be maintained to ensure that newly found words are appropriately stemmed. Second, it fails to remove the affixes of words in some circumstances and therefore fails to extract the root. For example, the root-based technique stemmer will fail to remove affixes in words, as shown in Table 9; hence, it will not stem them when they come from the roots. The third and most serious issue is that, from the perspective of an information retrieval system, root extraction stemmers are useless in the Arabic language. In many cases, their adoption results in a new word that is very general, leading to poor search efficiency. A comparison of how different Arabic words are stemmed using different stemmers is shown in Table 9. The table shows an example of the correct root extracted by the ISRI + BPR stemmer against the incorrect stemmed words by the two well-known Arabic stemmers, i.e., ARLS 2 and ISRI. BPR enhances the ISRI approach due to the strength and flexibility of the proposed rules; the rules contain many sub-options (subrules) to stem the root correctly. In addition, the BPR algorithm is able to extract the correct root of the words that the previous algorithms are unable to extract. For instance, the word 'boxes' (صناديق) is incorrectly processed by ISRI and ARLS 2 but correctly stemmed by BPR into the word 'encase' (صندق). BPR can also extract roots that contain two letters (as shown in Table 4, in the main rule 6 subrule 2, the words اخلاء, اطباء, اعزاء will have two-letter root words, which are عز، طب، خل). To sum up, BPR has wider capability in extracting roots because it can extract quintain, quatrain, tercet, and couplet roots like no other algorithm had done before.

Table 9
Example of stemmed words by ISRI + BPR, ISRI, and ARLS 2.

ISRI + BPR	ISRI	ARLS 2	Words
شرع	مشاريع	مشاريع	مشاريع
صندق	صناديق	صناديق	صناديق
نبي	بيء	نبيء	انبياء
ولي	ليء	وليء	اولياء
صفي	صفياء	صفيء	اصفياء
نقي	قيء	نقيء	انقياء
فأل	فائل	متفائل	متفائل
فعل	كفعائل	كفعائل	كفعائل
فعل		بفعائل	بفعائل
فعيل	فعل	فعائل	لفعائل
شيطان	شيط	شياط	شياطين
سلطان	سلط	سلاط	سلاطين
فعل	اعل	افعيل	افاعيل
فعيل	علئ	فعلائ	فعلائي
عميل	عملائ	عملائ	عملائي

7. Conclusion

To solve the problem of stemming broken plural words, which is considered a major weakness of root-based methods for extracting roots from Arabic words, we designed and developed new rules for stemming BPR in accordance with accurate Arabic grammar. The first section describes the proposed algorithm for developing new BPRs, with the new rules as the latest contribution to enhancing Arabic root-based approaches by overcoming their weaknesses and shortcomings. The ISRI method was improved by applying the BPR algorithm to the original ISRI stemmer, and we named it ISRI + BPR stemmer. The experimental results from the first investigation showed that the BPR development had achieved remarkable success, with an encouraging number of correct extracted roots by using the BPR algorithm on a well-known Arabic dataset. In the second investigation, the results obtained by ISRI + BPR were the best compared with the benchmark Arabic stemmers: the original ISRI stemmer and ARLS 2. Both results indicated an evident increase in the percentage of correct roots and a reduction in incorrect roots in terms of precision, recall, F-measure, and ICF measures. Thus, we can conclude that the proposed BPR algorithm has solved some of the irregular broken plural problems, which eventually helped in improving the performance of a root-based Arabic stemmer. In the future work, we will extend this approach to all Arabic derived names..

Conflicts of interest statement

All authors declare that they have no conflicts of interest.

Acknowledgements

This research was partially funded by the Ministry of Higher Education Malaysia under research code: FRGS/1/2020/ICT02/UKM/02/1.

References

[1] Ababneh M, Al-Shalabi R, Kanaan G, Al-Nobani A. Building an effective rule-based light stemmer for Arabic language to improve search effectiveness. Int Arab J Inf Technol (IAJIT) 2012:9.
[2] Abuata B, Al-Omari A. A rule-based stemmer for Arabic Gulf dialect. J King Saud Univ-Comput Inf Sci 2015;27(2):104–12.
[3] AL-Aswadi FN, Chan HY, Gan KH. Extracting Semantic Concepts and Relations from Scientific Publications by Using Deep Learning. arXiv preprint arXiv:2009.00331. (2020).
[4] AL-Aswadi FN, Chan HY, Gan KH. Extracting Semantic Concepts and Relations from Scientific Publications by Using Deep Learning. Cham: Springer International Publishing; 2021. p. 374–83.
[5] Al-Fuqaha A, Kountantis D, Cooke S, Elbes M, Zhang J. A genetic approach for trajectory planning in non-autonomous mobile ad-hoc networks with qos requirements. In: 2010 IEEE Globecom Workshops. IEEE; 2010. p. 1097–102.
[6] Al-Kabi MN, Kazakzeh SA, Abu Ata BM, Al-Rababah SA, Alsmadi IM. A novel root based Arabic stemmer. J King Saud Univ-Comput Inf Sci 2015;27(2):94–103.
[7] Al-Lahham YA, Matarneh K, Hasan M. Conditional Arabic light stemmer: condlight. Int Arab J Inf Technol 2018;15:559–64.
[8] Al-Omari A, Abuata B, Al-Kabi M. Building and benchmarking new heavy/light Arabic stemmer. The 4th International conference on Information and Communication systems (ICICS 2013), 2013.
[9] Alhutaish R, Omar N. Arabic text classification using k-nearest neighbour algorithm. Int Arab J Inf Technol (IAJIT) 2015;12:190–5.
[10] Aljlayl M, Frieder O. On Arabic search: improving the retrieval effectiveness via a light stemming approach. In: Proceedings of the eleventh international conference on Information and knowledge management. p. 340–7.
[11] Alshalabi H, Tiun S, Omar N, Al-Aswadi FN, Ali AK. Arabic light-based stemmer using new rules. J King Saud Univ - Comput Inf Sci 2021. doi: https://doi.org/10.1016/j.jksuci.2021.08.017.
[12] Alshalabi H, Tiun S, Omar N, Albared M. Experiments on the use of feature selection and machine learning methods in automatic Malay text categorization. Procedia Technol 2013;11:748–54.
[13] Alshalabi HA, Tiun S, Omar N. A comparative study of the ensemble and base classifiers performance in Malay text categorization. Asia-Pasific J Inf Technol Multimedia 2017;06(02):53–64.
[14] Altawaier MM, Tiun S. Comparison of machine learning approaches on Arabic twitter sentiment analysis. Int J Adv Sci, Eng Inf Technol 2016;6:1067–73.
[15] AlZu'bi S, Hawashin B, EIBes M, Al-Ayyoub M. A novel recommender system based on apriori algorithm for requirements engineering. In: 2018 fifth international conference on social networks analysis, management and security (snams). IEEE; 2018. p. 323–7.
[16] AlZu'bi S, Jararweh Y, Al-Zoubi H, Elbes M, Kanan T, Gupta B. Multi-orientation geometric medical volumes segmentation using 3d multiresolution analysis. Multimedia Tools Appl 2019;78(17):24223–48.
[17] AlZubi S, Islam N, Abbod M. Enhanced hidden markov models for accelerating medical volumes segmentation. In: 2011 IEEE GCC Conference and Exhibition (GCC). IEEE; 2011. p. 287–90.
[18] Atwan J, Mohd M, Kanaan G. Enhanced Arabic information retrieval: Light stemming and stop words. In: International Multi-Conference on Artificial Intelligence Technology. Springer; 2013. p. 219–28.
[19] Bi Y, Bhatia R, Kapoor S. Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys). Springer Nature; 2019.
[20] Boudelaa S, Gaskell MG. A re-examination of the default system for Arabic plurals. Language Cognit Processes 2002;17(3):321–43.
[21] Chen A, Gey F. Building an Arabic stemmer for information retrieval. TREC. 2002:631–9.
[22] El-Beltagy SR, Rafea A. An accuracy-enhanced light stemmer for arabic text. ACM Trans Speech Language Process (TSLP) 2010;7:1–22.
[23] Goweder A, De Roeck A. Assessment of a significant Arabic corpus, Arabic NLP Workshop at ACL/EACL. (2001).
[24] Goweder A, Poesio M, De Roeck A, Reynolds J. Identifying Broken Plurals in Unvowelised Arabic Tex. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. p. 246–53.
[25] Iazzi S, Yousfi A, Bellafkih M, Aboutajdine D. Morphological analyzer of Arabic words using the surface pattern. Int J Comput Sci Issues (IJCSI) 2013;10:254.
[26] Kanan T, Sadaqa O, Almhirat A, Kanan E. Arabic light stemming: A comparative study between p-stemmer, Khoja stemmer, and light10 stemmer. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). IEEE; 2019. p. 511–5.
[27] Kchaou Z, Kanoun S. Arabic stemming with two dictionaries. In: 2008 International Conference on Innovations in Information Technology. p. 688–91.
[28] Khoja S, Garside R. Stemming Arabic text. Lancaster, UK: Computing Department, Lancaster University; 1999.
[29] Larkey LS, Ballesteros L, Connell ME. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. p. 275–82.
[30] Larkey LS, Ballesteros L, Connell ME. Light stemming for Arabic information retrieval, Arabic computational morphology. Springer; 2007. p. 221–43.
[31] Nehar A, Ziadi D, Cherroun H. Rational kernels for Arabic root extraction and text classification. J King Saud Univ-Comput Inf Sci 2016;28(2):157–69.
[32] Nwesri AF, Tahaghoghi SM, Scholer F. Stemming Arabic conjunctions and prepositions. In: International Symposium on String Processing and Information Retrieval. Springer; 2005. p. 206–17.
[33] Syarief MG, Kurahman OT, Huda AF, Darmalaksana W. Improving Arabic Stemmer: ISRI Stemmer. In: 2019 IEEE 5th International Conference on Wireless and Telematics (ICWT). IEEE; 2019. p. 1–4.
[34] Taghva K, Elkhoury R, Coombs J. Arabic stemming without a root dictionary, Information Technology: Coding and Computing. In: ITCC 2005. International Conference on, IEEE. p. 152–7.
[35] Yousfi A. The morphological analysis of Arabic verbs by using the surface patterns. IJCSI Int J Comput Sci Issues 2010;7:11.
[36] Zeroual I, Boudchiche M, Mazroui A, Lakhouaja A. Developing and performance evaluation of a new Arabic heavy/light stemmer. In: Proceedings of the 2nd international Conference on Big Data, Cloud and Applications. p. 17.

Dr Alshalabi Hamood is currently a lecturer at Sana'a University, Yemen. He holds his PhD from the Universiti Kebangsaan Malaysia, Malaysia (UKM). His research focuses in Natural Language Processing on Text Processing, and Image Processing.

Dr Sabrina Tiun is a senior lecturer at the Faculty of Information Science and Technology (FTSM) in Universiti Kebangsaan Malaysia (UKM). Her range of research interests are from Natural Language Processing, Computational Linguistic, Speech Processing to Social Science Computing. Currently, she is the CAIT Teaching & Learning Coordinator at the FTSM, UKM.

Assoc. Professor Dr Nazlia Omar is an Associate Professor at the Centre for AI Technology (CAIT), Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM). She holds her PhD from the University of Ulster, UK. Her main research interest is in the area of Natural Language Processing and Computational Linguistics. She is currently the head of Asian Language Processing Lab (ASLAN) under the Centre of Artificial Intelligence Technology (CAIT).

Dr Elham anaam. She holds her PhD from Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM). Her research focuses on E-CRM adoption and Multi criteria decision making.

Yazid Saif is currently a research assistant at Tun Hussien Onn (UTHM). Faculty of mechanical engineering. He is a PhD a student at the Universiti Tun Hussien Onn Malaysia, (UTHM). His research focuses in measuring models and detecting features based on Image Processing, and his concern with Artificial Intelligent focusing on the convolutional neural network.