International Journal of Information Science and Management
Vol. 20, No. 4, 2022, 85-100
https://dorl.net/dor/20.1001.1.20088302.2022.20.4.6.5

Original Research

The Effectiveness of Arabic Stemmers Using Arabized Word Removal

Hamood Al-shalabi
Lecturer, Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, Malaysia. Sana'a University, Sana'a, Yemen.
hmoud. shalabi@yahoo.com
hmoud.shalabi@siswa.ukm.edu.my
ORCID iD: https://orcid.org/0000-0003-3504-7186

Sabrina Tiun
Senior Lecturer, Universiti Kebangsaan Malaysia, Faculty of Information Science & Technology, Selangor, Malaysia.
Corresponding Author: sabrinatiun@ukm.edu.my
ORCID iD: https://doi.org/0000-0002-1134-973X

Nazlia Omar
Associate prof. Universiti Kebangsaan Malaysia, Faculty of Information Science & Technology, Selangor, Malaysia.
nazlia@ukm.edu.my
ORCID iD: https://doi.org/0000-0002-8173-8933

Kamal Ali Alezabi
Assistant Prof., Institute of Computer Science & Digital Innovation (ICSDI), UCSI University, Kuala Lumpur, Malaysia. Sana'a University, Sana'a, Yemen.
kamal@ucsiuniversity.edu.my
ORCID iD: https://orcid.org/0000-0002-5957-0732

Fatima N. AL-Aswadi
Lecturer, Faculty of Computer Science and Engineering, Hodeidah University, Hodeidah, Yemen. Universiti Sains Malaysia, Pulau Pinang, Malaysia.
fnsa15_com016@student.usm.my, fatima_aswadi@hoduniv.net.ye
ORCID iD: https://doi.org/0000-0001-5413-1207

Received: 20 May 2021
Accepted: 07 November 2021

Abstract

Other languages have influenced Arabic because of several factors, such as geographical nearness, trade communication, past Islamic conquests, science and technology, new devices, brand names, models, and fashion. As a result of these factors, foreign words are used in Arabic text and are known as Arabised words. Arabised words affect the Arabic natural language processing (NLP) task because identifying a correct stem or root from an Arabic word becomes more difficult. Therefore, a more efficient Arabic NLP can be developed if Arabised word removal is part of a pre-processing task. In this paper, we propose an algorithm for detecting and extracting Arabised words as a pre-processing task for an Arabic stemming task. This algorithm is a combination of lexicon-based and rule-based approaches. The lexicon list has been developed based on various sources of Arabic text resources, and the rule-based algorithm has been designed to cater to Arabised words with definite articles and use pattern matching on prefixes and suffixes. To evaluate the effectiveness of the proposed Arabised word removal algorithm on the Arabic NLP task, we use Arabised word removal as part of pre-processing in Arabic stemmers. Three Arabic stemmers are used in our evaluation, namely, light stemming, condition light and ARLS, on three types of Arabic standard datasets. Comparisons were made by measuring the performance of precision, recall and IFC on the stemmers with or without our Arabised word removal pre-processing. Results show that the performance on all the stemmers improves if Arabised word removal is

included as part of the stemming's pre-processing. Therefore, an efficient Arabic NLP application or task can be developed if Arabised word removal is included in the pre-processing stage for Arabic NLP application, mainly Arabic stemming.

Keywords: Arabised Word, Natural Language Processing, Arabised Words Removal, Arabic Text Pre-Processing, Arabic Stemming, Text Processing, Arabic Language.

Introduction

With the continuous growth of textual data on the web, the need for effective techniques and methods to handle and process this considerable amount of text has increased. This increased need has made natural language processing (NLP), speech recognition, text mining, and automated document classification promising research fields.

The stemming algorithm is a computational process that gathers all words that share the same stem and have some semantic relations (Paice, 1996). The main objective of the stemming process is to remove all possible affixes and thus reduce the word to its stem (Dawson, 1974). This process is generally used for document matching and classification by converting all likely forms of a word in the input document to the form in a reference document (Burden, 2000). Arabic stemming algorithms can be classified according to the desired level of analysis as either light-based algorithms (Larkey, Ballesteros & Connell, 2007) or rule-based algorithms (Khoja & Garside 1999). Light-based algorithms remove prefixes and suffixes from Arabic words, whereas rule-based algorithms reduce stems to roots (Xu, Fraser & Weischedel, 2002). Light-stemming refers to stripping off a small set of prefixes and/or suffixes without dealing with infixes or recognizing patterns and finding roots (Al-Sughaiyer & Al-Kharashi, 2004; Al Ameed et al., 2005).

Arabised words pose many challenges in stemming, such as the inability to apply standard rules of Arabic stemming because these words do not yet have primary Arabic roots and auto-recognition technology or methods. Moreover, when analyzed, incorrect roots for Arabic words belong to either stop or foreign words (Arabised words). The stemming algorithm cannot handle incorrect roots (Ghwanmeh, Kanaan, Al-Shalabi & Rabab'ah, 2009). Therefore, solving the Arabised word problem would improve the accuracy and performance of stemming algorithms.

Thus, we can classify Arabised words into two types:

1. The first type is foreign words that have no translation in Arabic but are written in Arabic (i.e., Mercedes [مرسيدس], iPhone [ايفون], and Samsung [سامسونج]).
2. The second type is foreign words that have a translation in Arabic (i.e., a radio [مذياعor راديو] and phone [هاتفor تلفون]).

The rest of this paper is organized as follows. Section 2 briefly reviews related work and the Arabic language morphology. Section 3 discusses the text collection used by the root extraction approach and briefly describes the construction of the lexicon list of Arabised words. Section 4 compares and discusses the linguistic approach and our proposed method for improving Arabised removal. Section 5 presents the evaluation criteria and experimental results. Finally, Section 6 and 7: discussion and concludes and discusses future work.

Literature Review

Many studies have been proposed and implemented using different techniques for the pre-processing stage for Arabic, especially the stop-words removal technique. This section explores prominent studies.

The stop-word list contains the words in a text with little meaning. Additionally, these words only serve as syntactic procedures but do not refer to the subject matter. These stop words affect NLP differently (Alshalabi, Tiun, Omar & Albared, 2013). multiple uses in many applications, such as information retrieval IR (Atwan, Mohd & Kanaan, 2013), text classification (Alhutaish & Omar, 2015; Alshalabi, Tiun, Omar, Al-Aswadi & Ali Alezabi, 2022), and ontology construction (AL-Aswadi, Chan & Gan, 2021), as well as in many other NLP applications as in Altawaier and Tiun (2016); Alshalabi, Tiun & Omar (2017); AL-Aswadi, Chan & Gan (2020); Ahmed, AL-Aswadi, and Noaman & Alma'aitah, (2022)

El-Khair (2006) measured the effect of three stop-word lists for Arabic information retrieval R: the general stop-word list, corpus-based list, and combined stop-word list. The study used popular weighting schemas and an Arabic newswire dataset from the Linguistic Data Consortium (LDC) and found that the overall performance of the general stop list was better than the other two lists. A similar study to that of El-Khair (2006) and Atwan et al. (2013) measured the effects of three stop-words lists with the light-stemming for Arabic information retrieval, namely, the general stop-word list, Khoja stop-words list, and combined stop-words list. They used a vector space model as the popular weighting scheme for examination. The idea of the study is to combine general and Khoja stop-words lists with light stemming from enhancing performance and comparing their effects on information retrieval. The Arabic newswire dataset from the LDC was used in this study, and the best performance was achieved with the combined stop-words list with light stemming.

Moreover, Al-Shalabi, Kanaan, Jaam, Hasnah and Hilat (2004) developed an algorithm to remove the Arabic stop-words based on a deterministic finite machine and created a stop list of more than 1000 words using several sources. One of these sources is a list that was compiled from other researchers' work, and the other source is the translation from English stop lists to Arabic. Then, the authors tested their developed system using 242 Arabic examples from the Proceedings of Saudi Arabian National Computer Conferences with 47897 words and a set from the holy Quran.

Furthermore, according to Al-Nashashibi, Neagu & Yaghi (2010), no standard of stemmer exists in the Arabic language. Many studies have been conducted for the Arabic language, such as studies on normalizing techniques, but they did not validate their results. Given no standard even for pre-processing steps, such as encoding, tokenization, and stop-words removal, a basic standard approach with the support of open-source tools will help to guide the research in this field. Additionally, Alshalabi et al. (2021) recommended including a suffixes and prefixes table to differentiate the Arabic stems based on word size instead of relying on morphological word patterns. This enhanced algorithm, known as the Dlight Arabic stemmer, uses word length to remove suffixes based on the precise stage of stemming (double, quadriliteral, or trilateral roots).

Moreover, Elbarougy, Behery and El Khatib (2020) discussed the importance of the pre-processing stage, which is considered a part of NLP. The pre-processing techniques are essential in the Arabic language, given the complexity of its structure. The pre-processing techniques seek to exemplify and enhance the performance of handling and processing the Arabic text. Pre-processing techniques include tokenization, standardization, stop-words

removal and structural analysis. The results show that the processing of Arabic text after using stop-words removal gives better results than keeping stop words in the text.

Few studies have investigated the removal of Arabised words, as in the study of Almusaddar (2014), which improved pre-processing for word stemming by introducing and using 100 Arabised words, which were collected manually as a list for the lexical dictionary. However, this list is insufficient because incorrect roots were analyzed and found to belong to either stop words or foreign (Arabised) words. The stemming algorithm cannot handle incorrect roots (Ghwanmeh, et al., 2009). Therefore, the Arabised word problem must be solved to improve the stemming algorithm's accuracy.

The Arabic language is one of the Semitic languages that have 28 letters. All letters are consonants, and the consonant is only written once, even if the same consonant occurs twice in a word. Our work focuses on enhancing the removal of Arabised words and testing it on three light-stemming approaches (Larkey et al., 2007; Al-Lahham, Matarneh and Hasan, 2018; Abainia, Ouamour and Sayoud, 2017). In the following sections, we will present our proposed method of removing Arabised words.

Materials and Methods

Data collection

The dataset in our study is collected from three data collections. The first collection is Al-Alwatan-2004. Its size is 14.4 MB, and the number of categories is six topics for parallel Arabic delicate corpus. It contains six dialects of modern standard Arabic, which includes more than 6000 sentences. The second one is the Al-Khaleej-2004 corpus. This corpus is prepared to achieve experiments on topic identification for the Arabic language. It is extracted from thousands of articles downloaded from an online newspaper. This corpus contains more than 5000 articles, which correspond to nearly 3 million words, and punctuation in this corpus is removed on purpose (for more information, check the works based on AL-Khaleej-2004 and Al-Watan-2004). Parallel Arabic punctuation is removed on purpose (for more information, check the results based on the Khaleej-2004 corpus (Abbas & Smaili, 2005). More details are presented in Tables 1 and 2.

Table1

Size of the Al-Khaleejj-2004 dataset

Topic	Number of documents	Corpus size (words)
International news	953	534075
Local news	2398	964862
Economy	909	417708
Sports	1430	550622
Total number	5690	2467267

Table 2

Size of the Al-Watan-2004 dataset

Topic	Number of documents	Corpus size (words)
Culture	2782	1393541
Religion	3860	3110640
Economy	3468	1456970
Local news	3596	1555594
International news	2035	856464

Topic	Number of documents	Corpus size (words)
Sports	4550	1420273
Total number	20291	9793482

The third is the TREC2002 collection (Gey & Oard, 2001). It contains 383872 documents, which are drawn from the archive of the Associated France Press (AFP) news wires. These documents are organized in files containing daily news from 1994 to 2000. The total number of words is 68197285. In the work of Al-Lahham et al. (2018), the query is made up of the header and the description of each topic.

Text Pre-Processing

-Text normalization and tokenization

In this step, all the irrelevant and noisy data are removed. The punctuation, stop-words, and numbers are considered noisy data. The process of eliminating such data can be described as follows:

Remove punctuation: Several special characters hinder the process of analyzing the text. A sample of such special characters is '! - ~ @ * ^ # & + $ % ='.

Remove numbers: All the numeric numbers will be removed. This numeric data refer to the indexing of words. Tokenising and noise removal: The sentence is split into words.

-Stop-words and Arabised words removal

The stop-word list contains the words in a text with little meaning. Additionally, these words only serve as syntactic procedures but do not refer to the subject matter. These stop words affect NLP differently (Alshalabi et al., 2013). They can affect the retrieval process because they have a quite high frequency and tend to reduce the impact of frequency variations between the less common words that ultimately influence the weighing process. Removing the stop words also changes the document length and affects the weighing process. They also affect the efficiency of text processing owing to their nature and the fact that they carry no meaning, which may result in a large amount of unproductive processing (Almusaddar, 2014; Bouzoubaa, Baidouri, Loukili & El Yazidi, 2009).

Moreover, the Arabised word is a foreign word loan from another language, such as Turkish, Persian, English, or French. Examples of such Arabised words are names of cars and brands, names of modern fashion and electronic devices, and the names of Gregorian calendar months. This study removes stop words according to the khoja list (Khoja & Garside, 1999).

Arabised Word Removal

-Lexicon-based Arabised word removal

First, a list of Arabised words is collected from many Arabic sources and dictionaries to build a lexicon dictionary. This list contains words written in Arabic letters but not of Arabic origin. The length of the Arabised list is 1751 words, and the Arabised words are collected manually from websites (Wikipedia, 2020). The Arabised lexicon list contains the names of currencies with their derivatives and the names of foreign countries with their capitals. It also has non-Arabic words, including the names of clubs, sports teams, celebrities, and economic terms. An example of the proposed Arabised lexicon-based is shown in Table 3.

Table 3
Example of Arabised lexicon

Arabised words	
Salon	صالون
Sofa	كنبه
Booth	كشك
Computer	كمبيوتر
Internet	انترنت

-Rule-based Algorithm for Arabised word removal

In addition to detecting and removing Arabised words using a lexicon-based approach, we propose a pattern-based approach to detect and extract Arabised words based on our proposed lists of prefixes and suffixes in Table 4. Detecting Arabised words using patterns of suffixes and prefixes merely match the prefixes and suffixes in Table 4 in an Arabic text. The word is detected as an Arabised word if a match is found.

Table 4
Lists for both suffixes and prefixes for the rule-based Arabised removal algorithm

prefixes	ليو', 'لانكا', 'لوجيا', 'جراف', 'يسيا', 'سيليا', 'رنت', ' ,اندا ' ,يزو , 'إريا', ' ,'ورو', 'لار', 'لوجي', 'اتور' 'فاي', , 'لاند', 'رالي', 'ستان', 'زياء', 'بايت', 'روز', 'لند, 'لند', , 'ديسك', 'راطي', 'راطيا 'غوليا','يليا' 'ماز', 'كيا', 'راطية', 'لينج', 'ناتور', 'بورو', 'ستيك', 'ستيكي', 'تيكية', 'تيكي', 'نيكي', 'نيكية', 'بادا', 'نسكو', 'لارين', 'ومبو', 'ومبي', 'ستريت', 'موتور', 'ديو', 'قاز', 'نيوم', 'روخ', 'اورد', 'دبليو']
suffixes	'ليو', 'لانكا','لوجيا','جراف','يسيا','سيليا','رنت','فاي', 'ورو', 'لار', 'لوجي', 'اتور', 'يزو' , 'إريا', 'اندا 'لاند', 'رالي', 'ستان', 'زياء', 'بايت', 'روز', 'لند, 'لند', , 'ديسك', 'راطي', 'راطيا', 'غوليا', 'يليا', 'ماز', 'كيا', 'راطية', 'لينج', 'ناتور', 'بورو', 'ستيك', 'ستيكي', 'تيكية', 'تيكي', 'نيكي', 'نيكية', 'بادا', 'نسكو', 'لارين', 'ومبو', 'ومبي', 'ستريت', 'موتور', 'ديو', 'قاز', 'نيوم', 'روخ', 'اورد','دبليو'

However, although Arabised lexicon-based and pattern-based prefixes and suffixes are used, these techniques are inefficient in detecting Arabised words without Arabic definite article prefix removal. The reason is that many Arabised words start with a definite Arabic article, such as 'بال' (by the), 'وال' (and the), 'ال' (of), ' ال' (the), 'و' (and), 'ل' (to), 'ك' (as) and 'ب' (in). For example, 'الكمبيوتر' (the computer), 'بفرنسا' (in France), and 'والسنجفوري' (Singaporean) are Arabised words with Arabic definite article prefixes. To solve this problem, we develop a rule-based Arabised removal algorithm that includes a rule to delete the definite Arabic article. Figure 1 shows this proposed rule-based algorithm (Algorithm 1) with definite article prefix deletion and pattern-based matching on prefixes and suffixes to detect and extract Arabised words.

Algorithm 1: Rule-based Arabised word removal algorithm
Input: Arabic text Output: List of Arabised words

If (Length [word]>=5) and (word starts with ['بال'], ['وال'], ['لل'] or ['ال']),
remove prefixes from the word.
If (length [word]>=4) and (word starts with ['و'],('ل'), ('ك') or ('ب')),
remove prefixes from words.
If the word ends with any suffix in the list in Table 3,
return (word) as Arabised.
If the word starts with any prefix in the list in Table 3,
return (word) as Arabised.

Figure 1: A proposed rule-based algorithm for Arabised word removal

-Combination of lexicon-based and rule-based approaches for Arabised word removal

The Arabised word removal algorithm can be further improved by combining lexicon-based and rule-based approaches. Figure 2 presents the flowchart of the combination Arabised removal algorithm.

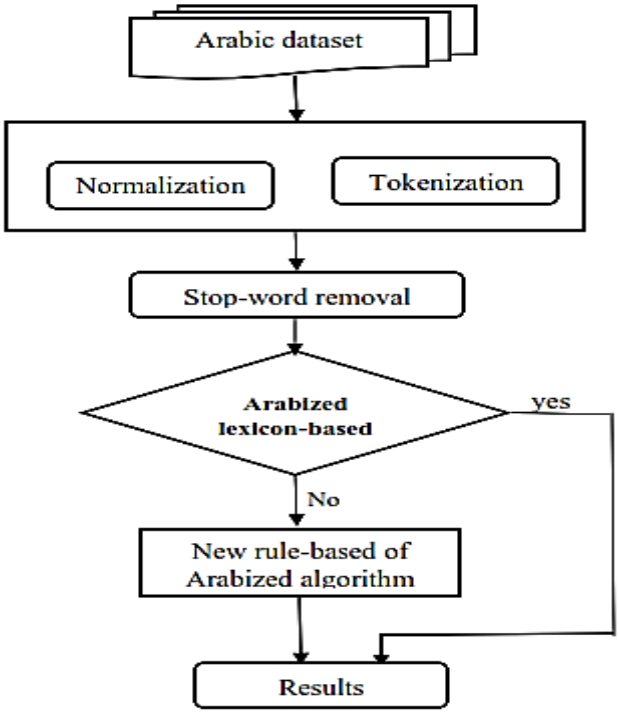

Figure 2: Flowchart diagram for the proposed Arabised words removal using a combination of lexicon-based and rule-based approaches

Also, Figure 3 presents the proposed algorithm for Arabised word removal that combines lexicons and rules.

Algorithm 2: Detecting Arabised words using lexicon-based and rule-based approach
Input: Arabic text Output: list of Arabised words
If (Length(word)>=5) and (word starts with ['بال'], ['وال'], ['لل'] or with ['ال']), remove the prefix from the word. else if (length(word)>=4) and (word starts with ['و'], ['ل'], ['ك'] or ['ب']), remove the prefix from the word. If a word is Arabised lexicon-based, return (word) as Arabised word. Else: If the word ends with any suffix list in Table 1, return (word) as Arabised. If the word starts with any prefix list in Table 1, return (word) as Arabised.

Figure 3: Proposed Arabised word detection or removal algorithm by combining lexicon-based and rule-based approaches

-Arabised word removal on Arabic stemmers

Figure 4 shows the flowchart diagram of the proposed Arabised word removal as pre-processing for Arabic stemming. Table 4 shows an example of the content of the Arabised lexicon list. Arabised word removal is a pre-processing stage to avoid the effect of Arabised words in Arabic stemming, which removes Arabised words from the text that needs to be stemmed. The following are the stages of Arabised word removal processing for the Arabic stemmer:

Stage 1: This stage aims to detect and remove Arabised words using the Arabised lexicon-based approach. The matched words from the dataset with the Arabised lexicon list are Arabised words and are removed.

Stage 2: This stage aims to remove Arabised words using the rules-based algorithm. If Arabised words are not in the Arabised lexicon, then the rules-based algorithm is used to detect and remove Arabised words. The rule-based algorithm of Arabised words includes a list of suffix and prefix patterns and the Arabic definite deletion rule.

To assess the effectiveness of the Arabised word removal algorithm, we select three Arabic stemmer approaches, namely, Larkey Light10 stemmer (Light10), Condition Light stemmer (Condlight) and Arlstem stemmer (Arlstem).

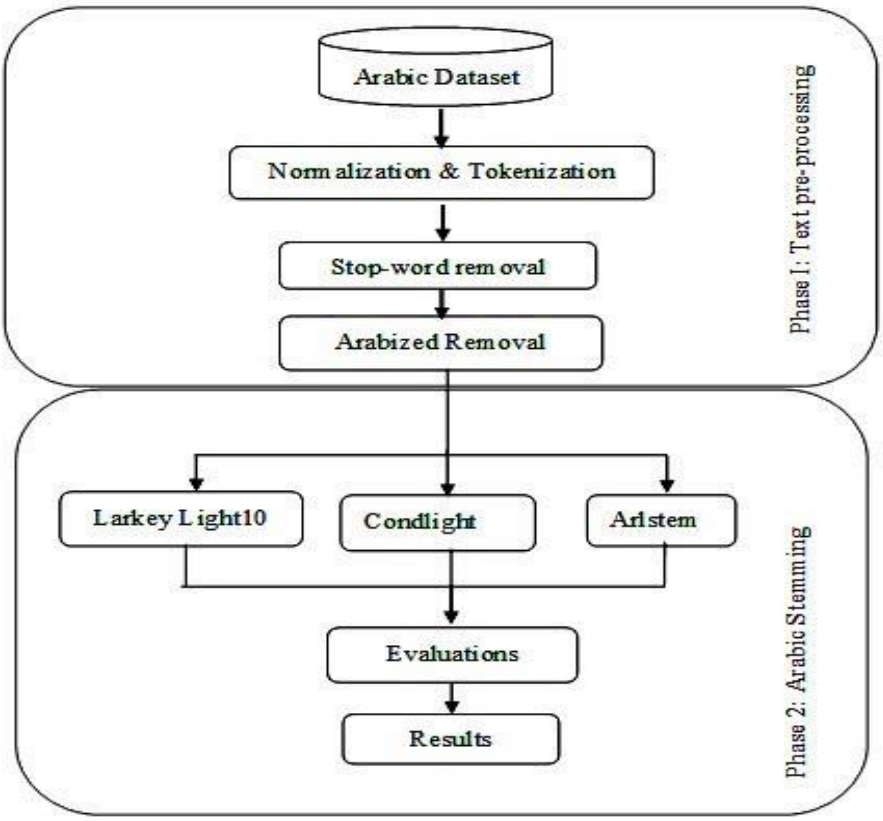

Figure 4: Implementation of proposed Arabised words removal on Arabic stemming

Evaluation metrics

The dataset that is used in the evaluation was the TREC2002 collection. In the evaluation, we used Precision (Eq. 1), Recall (Eq.2) and F-measure (Eq.3). Notably, these three metrics are the most used evaluation metrics in NLP applications and systems (Al-Kabi & Al-Mustafa, 2006; Al-Aswadi et al., 2020; Jabbar, Iqbal, Tamimy, Hussain & Akhunzada, 2020; Alshalabi et al., 2021).

$$\text{Precision} = \frac{\text{correct}}{\text{correct+incorrect}} \quad (1)$$

$$\text{Recall} = \frac{\text{correct}}{\text{correct+non-stem}} \quad (2)$$

$$\text{F-measure} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \quad (3)$$

Where the (Correct) (Incorrect) refers to the result of a word after stem by an algorithm if correct or incorrect root. Also, the 'non-stem' refers to the word that has not been processed by the algorithm or the word the algorithm can't be able to stem it. 'Precision' is the ratio between the numbers of correct roots and incorrect roots after stemmer. The 'Recall' is the ratio between the numbers of correct roots and non-stem words. Moreover, the Index compression factor (ICF) (Eq. 4) is also used as one of the evaluation metrics.

$$ICF = \frac{N-S}{N} \quad (4)$$

N is the number of unique words before stemming, and S is the number of unique stems after stemming.

Experimental results

In this study, two specific evaluations are carried out to assess the proposed methods of Arabised word removal. The first evaluates the effectiveness of the Arabised word removal method in detecting Arabised words in Arabic text corpora. The second evaluates the efficacy of the Arabised removal method in Arabic stemming.

Two standard datasets (text corpora) for the first experiment, Al-Khaleej-2004 and Al-Watan-2004 (Abbas, 2004), are used. The evaluation includes two methods that investigate the effectiveness of the proposed removal process in detecting and removing Arabised words, which may be found in both datasets. This experiment consists of two sub-experiments: (i) experiment 1.1 assesses the effectiveness of the Arabised lexicon-based method in detecting Arabised words, and (ii) experiment 1.2 assesses the effectiveness of the rules-based method in detecting Arabised words.

The second experiment assesses the effectiveness of Arabised word removal processing in Arabic stemming. We selected three Arabic stemmers for this experiment: Light10, Condlight and Arlstem. The descriptions of the three chosen Arabic stemmer algorithms have been given in the previous section. In this experiment, a comparison among selected Arabic stemmers is conducted with and without using Arabised word removal pre-processing. This comparison is to determine the effectiveness of our proposed Arabised removal processing for Arabic stemming. AFP_ARB (AFP_ARB 1994-2000) datasets are used to perform this experiment, namely

Experiment I: Arabised word detected and removal

Al-Watan-2004 and Al-Khaleej-2004 (Abbas, 2004) are used as test datasets in this experiment. Each dataset is pre-processed using normalization, tokenization, stop-words removal, and Arabised word detection and removal. The Arabised word removal process is done in two methods. Table 5 shows the output of Al-Khaleej-2004 after the pre-processing stages. Also, it shows the output from each pre-processing step based on the text categories in the Al-Khaleej-2004 text corpus.

Table 5
The number of stop-words and Arabised words detected in the Al-Khaleej-2004 dataset

Categories	Stop-words	Arabised words		percentage of Lexicon	percentage of Rule-based
		Lexicon-based	Rule-based		
Economy	88166	6152	21481	0.015	0.050
International	117925	5373	28145	0.010	0.050
Local	200434	3738	24619	0.004	0.030
Sport	112511	2550	17373	0.005	0.030

The result presents the number of extracted Arabised words from each text category using the Arabised removal pre-processing. The results shown in Table 5 show that the rule-based algorithm performed better than a lexicon-based algorithm. In the 'international' category, the rule-based algorithm extracts 28145 words against approximately (5%), whereas the dictionary-

based algorithm extracts 5373 words (around 1%). In the 'sport' category, the rule-based algorithm extracts 17373 words, approximately (3%), whereas the lexicon-based algorithm only extracts 2550 words (1%). In the 'economy' category, the lexicon-based algorithm extracts 6152 words, whereas the rule-based algorithm extracts 21481word; the rule-based algorithm increases approximately (5%) of extracted Arabised words compared to (2%) by the lexicon-based. In the 'local' category, the lexicon-based algorithm extracts 3738 words (~1%), whereas the rule-based algorithm extracts 24619 words, increasing by 3%. The 'religion' category contains large data, but the number of extracted Arabised words from the category is less than other categories with similar data sizes. The reason may be that most of the documents on 'religion' contain historical events, with writers keen on writing in standard Arabic and using fewer modern terms. Table 6 presents an example of Arabised words *extracted by the proposed Arabised removal algorithm in the Al-Khaleej-2004 dataset.*

Table 6
Example of Arabised words extracted by the Arabised word removal algorithm

Arabised words	
تكنولوجيا	Technology
انترنت	Internet
إلكتروني	Electronic
دبلوماسية	Diplomacy
دكتور	Doctor
فري	Free
فيزا	Visa
مليون	Million
ألمنيوم	Aluminium
اندرياس	Andreas
بنك	Bank

Table 7 presents the number of extracted Arabised words from each text category from the Al-Watan-2004 dataset. Table 7 shows that the lexicon-based removal algorithm produces only 10577 extracted words in the 'international' category. The proposed Arabised removal algorithm extracts 45934 words. In the 'sport' category, our proposed Arabised removal algorithm extracts 70092 words compared with the lexicon-based removal algorithm that detects 15280 Arabised words.

Table 7
The number of stop-words and Arabised words detected in the Al-Alwatan-2004 dataset

Categories	Stop-words	Arabised Words		Percentage of Lexicon	Percentage of Rule-based
		Lexicon-based	Rule-based		
Economy	282232	16681	64502	0.012	0.050
International	190869	10577	45934	0.003	0.010
local	299722	3892	33911	0.003	0.020
Sport	293250	15280	70092	0.010	0.050
Culture	317455	7893	45177	0.009	0.050
Religion	788844	3265	46131	0.002	0.030

In the 'economy' category, the lexicon-based removal algorithm extracts 16681 words,

whereas the rule-based Arabised removal algorithm extracts 64502 Arabised words. Moreover, the rule-based Arabised word removal algorithm extracts 33911 Arabised words, whereas the lexicon-based word removal algorithm extracts only 3892 Arabised words in the ‘international’ category. In the 'religion' category, the lexicon-based Arabised word removal algorithm extracts 3265 words, whereas the rule-based Arabised word removal algorithm extracts 46131 words according to the estimated percentage. Finally, in the ‘culture’ category, the rule-based algorithm extracts 45177 Arabised words, whereas our lexicon-based algorithm extracts only 7893 words.

Experiment II: A combination of Arabised removal processing in Arabic stemming

The second experiment uses the TREC2002 collection; a query is made up of the header and the description of each topic (Al-Lahham et al., 2018) (Larkey et al., 2007) (Abainia et al., 2017).

Table 8
Results of three Arabic stemmers in the AFB_ARB dataset with and without Arabised words removal pre-processing

Stemmers		Correct R	Incorrect R	All stems	Us
Light10	Without	15172131	38968066	34121520	145764
	With	13937575	34720998	31184433	140402
Condlight	Without	15788089	38352108	40188976	139195
	With	14782436	33876137	36335975	133659
ARLS	Without	18546892	35593305	41043699	158877
	With	17192233	31466340	37099529	152277

Table 9
Precision, recall, and F-measure of three Arabic stemmers in the AFB_ARB dataset with and without Arabised word removal pre-processing

Methods		Precision	Recall	F-measure	ICF
Light10	Without	28%	43%	34%	74%
	With	29%	44%	35%	75%
Condlight	Without	29%	53%	37%	75%
	With	30%	55%	39%	76%
ARLS	Without	34%	59%	43%	71%
	With	35%	60%	44%	72%

The experimental results in Table 8 are presented of Correct R, Incorrect R, all stems, and unique stems for three stemming approaches with/without our proposed new rules algorithm for Arabised words. The total number of words in the AFB_ARB dataset after stop-word removal is 54140197, and the number of extracted Arabised words by the proposed combined Arabised word removal algorithm is 5481624, showing that 10.12% of the words are Arabised words. The results show the enhancements of the three Arabic stemmers by applying the Arabised word removal algorithm. Also, the correct roots (Correct R) of the Light10 stemmer

are 15261324 before the removal of Arabised words and 14547147 correct roots after the Arabised word removal is applied. However, the high number of Correct R before Arabised removal does not mean the stemmer is better in performance.

Since Arabised words do not have Arabic base roots, According to problem statements, they cannot apply standard rules of Arabic stemming. Also, the stemming algorithms cannot handle the incorrect roots (Arabised words). As well as, in Table (8), the incorrect roots are reduced because they remove Arabised words that do not have corrected roots. On the other hand, correct roots are reduced after removing Arabised, as shown in the results, because the Arabised words result in a correct root after the stemmers' process. After all, the word after stem resembles a valid Arabic root, so the algorithm calculates it as a correct root, which is incorrect. We can call this process the misleading root that the Arabic algorithms overcame after deleting the Arabised words. This also explains the poor results in Precision, Recall, and ICF, but on the other hand, we can notice the quality after removing Arabised words through the results in unique stem (Us). It is the most important information retrieval process.

Discussion

Arabic stemmer faces many challenges; one of these challenges is the foreign words (Arabised words) that are overwhelmingly mixed with the Arabic language (Al-Shbiel 2017). These Arabised words negatively affect some aspects of Arabic language analysis, including extracting Arabic roots (Al-Kabi & Al-Mustafa, 2006). Thus, an efficient algorithm for recognizing Arabised words expressed through prefixes and suffixes, which come from the basic idea of the light stemmer, is needed. The Arabised word removal approach can be lexicon-based or rule-based. With a lexicon-based approach, the algorithm matches the words from an Arabic text with a list of Arabised words. If a match is found, the words are identified as Arabised. However, unseen Arabised words cannot be handled using the lexicon-based approach. Thus, a rule-based approach is required. In our study, the rule-based approach of detecting Arabised words is done by matching suffixes and prefixes and deleting the definite Arabic article. The rule-based approach overcomes the limitation of the lexicon-based approach. The increase of extracted words by the rule-based approach compared with the lexicon-based approach is shown in Tables 8 and 9.

As we defined previously, Arabised words are words that were originally foreign words that are written in Arabic letters in Arabic texts. Thus, stemming Arabised words sometimes results in correct roots, but they are, in fact, wrong roots. In other words, stemming the Arabised word can waste time. Therefore, removing Arabised words in the pre-processing stage increases the overall efficiency of Arabic stemmer algorithms, which is our study's motivation.

However, though the Arabised removal process is needed during the pre-processing stage of Arabic stemmer, it is hard to see works on Arabised removal processing. The most similar work on Arabized word removal was the work of Almusaddar (2014). Almusaddar (2014) improved a pre-processing stage of Arabic word stemming by introducing and using 100 Arabised words. The Arabised words were collected manually as a list for the lexical dictionary and utilized like a stopword list before stemming words. Almusaddar (2014) used Larkey's stemming algorithm to evaluate the effectiveness of their Arabised word removal, similar to our study. However, we cannot compare their results with the results of our work since Almusaddar (2014) has used a simple evaluation with ten random query documents and a simple error percentage as a measurement metric.

Our work is more advanced than Almusaddar (2014) since our Arabised lexicon list is larger, totaling 1751 words. In addition, we also proposed a rule-based Arabized word removal to remove any words that are supposed to be Arabised but not in the Arabised word list. Therefore, combining a large set of Arabized words and an efficient rule-based Arabised removal algorithm, one can perceive that our work on Arabized removal is more efficient than the previous work.

Conclusions

With the continuous development of technology, new terms, such as names of car models, mobiles, and modes of fashion, affect the Arabic language. These new terms enter the Arabic language as Arabised words. These Arabised words have a negative effect on processing Arabic data either for retrieving documents or stemming the correct roots for Arabised words according to specific rules. As mentioned (Al-Kabi & Al-Mustafa, 2006), Arabised words negatively affect Arabic text processing or NLP. Thus, in this study, we propose an improved approach to removing Arabised words and prove the necessity of applying Arabised word removal as pre-processing in Arabic NLP. All the Arabic stemmers improved with the increase in precision, recall, and ICF during Arabised word removal, proving that Arabised word removal pre-processing enhanced Arabic NLP application.

Although the results of our proposed Arabised removal algorithm are promising, the algorithm has some limitations. That is, some of the prefixes and suffixes of the Arabised words have the same prefixes and suffixes of the original Arabic words. This challenge can be addressed through a deep understanding of Arabic when constructing comprehensive and appropriate rules for recognizing Arabic root words. Having specific and proper rules leads to developing a more efficient Arabised removal method. We present a new algorithm for detecting and extracting Arabised words by combining lexicon-based and rule-based Arabised word removal approaches. We further prove the efficiency of the Arabised word removal method and the necessity of Arabised word removal as part of pre-processing in Arabic NLP application. By comparing Arabic stemming with and without Arabised word removal, the performance of the three algorithms in removing Arabised words achieved better-stemming results. In a future study, we aim to enhance this proposed technique and algorithm to handle the limitation of the present study.

Acknowledgment

Universiti Kebangsaan Malaysia partially funds this project under the research code GUP-2020-063.

Reference

Abainia, K., Ouamour, S., & Sayoud, H. (2017). A novel robust Arabic light stemmer. *Journal of Experimental & Theoretical Artificial Intelligence*, 29(3), 557-573. https://ui.adsabs.harvard.edu/link_gateway/2017JETAI..29..557A/doi:10.1080/0952813X.2016.1212100

Abbas, M. (2004). Arabic Corpora. Retrieved from https://sites.google.com/site/mouradabbas9/corpora

Abbas, M., & Smaili, K. (2005, September). Comparison of topic identification methods for Arabic language. *In Proceedings of International Conference on Recent Advances in Natural Language Processing, RANLP* (pp. 14-17).

Ahmed, I. A., AL-Aswadi, F. N., Noaman, K. M. & Alma'aitah W. Z. (2022). Arabic knowledge graph construction: A close look in the present and into the future. *Journal of King Saud University-Computer and Information Sciences*, 34(9), 6505-6523. https://doi.org/10.1016/j.jksuci.2022.04.007

Al-Aswadi, F.N., Chan, H.Y. & Gan, K.H. (2020). Automatic ontology construction from text: a review from shallow to deep learning trend. *Artificial Intelligence Review*, 53 (6), 3901-3928. https://doi.org/10.1007/s10462-019-09782-9

AL-Aswadi, F.N., Chan, H.Y., Gan, K.H. (2021). Extracting Semantic Concepts and Relations from Scientific Publications by Using Deep Learning. In: Saeed, F., Mohammed, F., Al-Nahari, A. (eds) *Innovative Systems for Intelligent Health Informatics. IRICT 2020. Lecture Notes on Data Engineering and Communications Technologies*, vol 72. Springer, Cham. https://doi.org/10.1007/978-3-030-70713-2_35

Al-Kabi, M. & R. Al-Mustafa (2006). Arabic root based stemmer. *proceedings of the international Arab conference on information technology*. Jordan. Retrieved from https://www.acit2k.org/ACIT2006/Proceeding/139.pdf

Al-Lahham, Y.A., Matarneh, K., Hasan, M. (2018). Conditional arabic light stemmer: Condlight. *The International Arab Journal of Information Technology*, 15 (3A), 559–564.

Al-Nashashibi, M. Y., Neagu, D., & Yaghi, A. A. (2010, November). Stemming techniques for Arabic words: A comparative study. In *2010 2nd International Conference on Computer Technology and Development* (pp. 270-276). IEEE.

Al-Shalabi, R., G. Kanaan, J. M. Jaam, A. Hasnah & E. Hilat (2004). Stop-word removal algorithm for Arabic language. *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications.* pp. 545.

Al-Sughaiyer, I. A. & I. A. Al-Kharashi (2004). Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3),189-213. https://doi.org/10.1002/asi.10368

Al Ameed, H., Al Ketbi, S., Al Kaabi, A., Al Shebli, K., Al Shamsi, N., Al Nuaimi, N., & Al Muhairi, S. (2005, September). Arabic light stemmer: A new enhanced approach. In *The Second International Conference on Innovations in Information Technology (IIT'05)* (pp. 1-9).

Alhutaish, R. & N. Omar (2015). Arabic text classification using k-nearest neighbour algorithm. *The International Arab Journal of Information Technology, (IAJIT)* 12(2), 190-195.

Almusaddar, M. 2014. *Improving Arabic light stemming in information retrieval systems*. MSC Thesis. Computer Engineering Department, Faculty of Engineering, Research and Postgraduate Affairs, Islamic University, Gaza, Palestine. Retrieved from https://library.iugaza.edu.ps/thesis/112883.pdf

Alshalabi, H., S. Tiun, N. Omar, F. N. Al-Aswadi & K. Ali Alezabi (2022). Arabic light-based stemmer using new rules. *Journal of King Saud University - Computer and Information Sciences*, 34(9), 6635-6642. https://doi.org/10.1016/j.jksuci.2021.08.017

Alshalabi, H., Tiun, S., Omar, N. & Albared, M. (2013). Experiments on the use of feature selection and machine learning methods in automatic Malay text categorization. *Procedia Technol*, 11, 748-754. https://doi.org/10.1016/j.protcy.2013.12.254

Alshalabi, H. A., S. Tiun & N. Omar (2017). A comparative study of the ensemble and base classifiers performance in Malay text categorization. *Asia-Pacific Journal of Information Technology and Multimedia* 6(2), 53-64. Retrieved from http://journalarticle.ukm.my/11854/1/19180-65874-1-PB.pdf

Altawaier, M. M. & Tiun, S. (2016). Comparison of machine learning approaches on arabic twitter sentiment analysis. *International Journal on Advanced Science, Engineering and Information Technology* 6(6), 1067-1073. http://dx.doi.org/10.18517/ijaseit.6.6.1456

Atwan, J., Mohd, M. & Kanaan, G. (2013, August). Enhanced arabic information retrieval: Light stemming and stop words. In *International Multi-Conference on Artificial Intelligence Technology* (pp. 219-228). Springer, Berlin, Heidelberg.

Bouzoubaa, K., Baidouri, H., Loukili, T. & El Yazidi, T. (2009). Arabic stop words: Towards a generalisation and standardisation. In *the 13th International Business Information Management Association Conference IBIMA*. Marrakech, Morocco.

Burden, P. (2000). Stemming algorithms and their use. [online]. http://www.scit.wlv.ac.uk/. Retrieved from: http://www.scit.wlv.ac.uk/seed/docs/mypapers/stemalg.html

Dawson, J. (1974). Suffix removal and word conflation. *ALLC bulletin*, 2(3), 33-46.

El-Khair, I. A. (2006). Effects of stop words elimination for Arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences*, 4(3), 119-133. https://doi.org/10.48550/arXiv.1702.01925

Elbarougy, R., Behery, G. & El Khatib, A. (2020). A proposed natural language processing preprocessing procedures for enhancing arabic text summarization. In *Recent Advances in NLP: The Case of Arabic Language* (pp. 39-57). Springer, Cham.

Gey, F. C. & Oard, D. W. (2001, November). The TREC-2001 Cross-Language Information Retrieval Track: Searching Arabic Using English, French or Arabic Queries. In *TREC* (Vol. 2001).

Ghwanmeh, S., Kanaan, G., Al-Shalabi, R. & Rabab'ah, S. (2009, August). Enhanced algorithm for extracting the root of Arabic words. In *2009 sixth international conference on computer graphics, imaging and visualization* (pp. 388-391). IEEE.

Jabbar, A., Iqbal, S., Tamimy, M. I., Hussain, S. & Akhunzada, A. (2020). Empirical evaluation and study of text stemming algorithms. *Artificial Intelligence Review*, 53(8), 5559-5588. https://doi.org/10.1007/s10462-020-09828-3

Khoja, S. & Garside, R. (1999). S*temming Arabic text*. Lancaster, UK, Computing Department, Lancaster University.

Larkey, L. S., Ballesteros, L. & Connell, M. E. (2007). Light stemming for Arabic information retrieval. In *Arabic computational morphology* (pp. 221-243). Springer, Dordrecht.

Paice, C. D. (1996). Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science* 47(8), 632-649. https://doi.org/10.1002/(SICI)1097-4571(199608)47:8%3C632::AID-ASI8%3E3.0.CO;2-U

Wikipedia (2020). List of circulating currencies. Retrieved from https://en.wikipedia.org/wiki/List_of_circulating_currencies

Xu, J., Fraser, A., & Weischedel, R. (2002, August). Empirical studies in strategies for Arabic retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 269-274).