New Arabic Root Extraction Algorithm

Nisrean Jaber Thalji[1], Emran Aljarrah[2], Roqia Rateb[3], Amaal Rateb Mohammad Al-Shorman[4]
Department of Artificial Intelligence and Robotics-Faculty of Science and Information Technology, Jadara University, Irbed, Jordan[1]
Department of Internet of Things-Faculty of Science and Information Technology, Jadara University, Irbed, Jordan[2, 4]
Department of Computer Science-Faculty of Science and Information Technology, Jadara University, Irbed, Jordan[3]

***Abstract*—This research presents a new algorithm for Arabic root extraction, which aims to improve the accuracy of Arabic Natural Language Processing Algorithms by addressing the weaknesses and errors of existing algorithms. The proposed algorithm utilizes a database, that includes a collection of roots, patterns, and affixes, to generate potential derivation roots of a word without eliminating affixes initially. By matching the derived word with patterns to identify potential roots, the proposed algorithm avoids the inaccuracies caused by eliminating affixes based on expectation methods. The study conducted a comparison of the proposed algorithm with three commonly used Arabic root extraction algorithms. The evaluation process is performed on three corpora. Results showed that the proposed algorithm achieved an average accuracy rate of 96%, which is significantly higher than the others.**

Keywords—Natural language processing; Arabic root extraction algorithm; Arabic applications; Arabic morphology; Text mining

I. Introduction

Arabic is a Semitic language with a rich history and culture, and it is spoken by over 420 million people worldwide. The Arabic language is characterized by a unique system of roots, where most words are derived from a three-letter root [1]. Therefore, the ability to accurately extract roots from Arabic words is crucial for understanding the language, conducting research, and developing natural language processing algorithms [2].

In recent years, there has been a growing interest in developing Arabic root extraction algorithms, particularly those based on datasets and rules. These algorithms rely on linguistic rules and datasets to extract the root of a given Arabic word. They have proven to be effective in extracting roots from a variety of Arabic texts, including classical literature, modern newspapers, and social media [3]. The importance of Arabic root extraction algorithms lies in their ability to improve natural languages processing tasks such as machine translation, text classification, and sentiment analysis. These algorithms also play a critical role in the development of Arabic language technologies, including spell checkers, search engines, and speech recognition systems [4]. Despite the challenges posed by the complexity of the Arabic language and its various dialects, researchers continue to explore new techniques and methods for Arabic root extraction. The development of these algorithms is essential for advancing the field of Arabic natural language processing and enhancing our understanding of the language [5].

To effectively extract the root from Arabic words, it is crucial to understand the fundamental concepts of root extraction in Arabic, which include roots, affixes, patterns, and derived words [6]. Until now, not all of the root words and affixes in the Arabic language have been identified. While Arabic scholars have discovered a significant number of them, there is still a need to uncover the remaining ones in order to obtain more precise outcomes when extracting the roots of Arabic words. Thalji et al. [7] have released an Arabic corpus containing 12,000 roots, 430 prefixes, 320 suffixes, 4,320 patterns, and 720,000 word-root pairs. One of the objectives of this paper is to thoroughly comprehend and examine the given corpus, and to extract significant information that can aid in the extraction of Arabic language roots.

This research introduces a new algorithm for Arabic root extraction in Natural Language Processing (NLP). The algorithm utilizes a database and pattern matching techniques to generate potential derivation roots without eliminating affixes initially, resulting in improved accuracy compared to existing algorithms. The evaluation conducted on three corpora demonstrates an average accuracy rate of 96%, highlighting the algorithm effectiveness in accurately extracting roots from Arabic words. This contribution has significant implications for Arabic NLP applications, enhancing the performance and reliability of tasks such as information retrieval, machine translation, and sentiment analysis. Overall, the paper presents a valuable addition to the field of Arabic language processing, advancing the accuracy and reliability of Arabic root extraction in NLP.

II. Literature Review

The literature review summarizes current techniques used in Arabic root extraction, highlighting their strengths and weaknesses, with the goal of identifying gaps in current approaches and proposing areas for future research.

Alfredaghi and Al-Anzi [8] propose a data-base-oriented method for identifying Arabic roots using patterns and root lists. The method is efficient and does not require individual word analysis, but has some limitations such as only returning one root when multiple patterns match, and a relatively short list of patterns.

Al-Serhan et al. [9] propose a statistical approach for extracting Arabic roots based on assigning weights to letters and using mathematical equations. The approach is efficient and doesn't require word analysis, but has limitations such as difficulty in handling the complexities of the Arabic language, and may not provide as accurate results as rule-based approaches. Additionally, the lack of a clear explanation for root extraction makes it challenging to understand and improve the algorithm.

The Khoja and Garside algorithm [10] is a widely used rule-based approach to Arabic stemming that identifies and removes prefixes and suffixes from Arabic words to produce their root form. It follows a set of rules based on Arabic morphology to identify the prefixes and suffixes that can be stripped from a word to obtain its root. The algorithm is effective but has some limitations, such as mistakenly removing common prefixes and suffixes that are actually part of the root letters, missing certain rules, and providing only one solution for non-vocalized words.

Taghva, Elkhoury, and Coombs [11] developed a rule-based algorithm to extract roots, which aimed to improve Khoja and Garside's algorithm [10] by eliminating the need for a root list. The algorithm did not require a root list, which made the root extraction process faster, but it had limitations, such as ambiguity in affixes, providing only one solution for non-vocalized words, returning meaningless roots, and sometimes failing to extract roots for derived words that contain the "ابدال" rule. Cross-checking with the root list was necessary to minimize the number of erroneous roots.

Ghwanmeh et al. [12] developed a rule-based algorithm to extract the trilateral, quadrilateral, and pentaliteral roots of Arabic words. The algorithm removes affixes and matches the remaining word against a list of patterns. The algorithm still missed many roots, prefixes, suffixes, and patterns. It also faced challenges in dealing with affixes ambiguity problems and returned only one solution for non-vocalized words.

Alkabi [13] proposed a new algorithm for Arabic root extraction to improve the accuracy of Khoja and Garside's algorithm [10]. The new algorithm used additional patterns to supplement Khoja and Garside's patterns and was tested on MSA textual documents from Arabic newspapers websites. The results showed an increase in the accuracy of Khoja and Garside's algorithm from 71% to 76%. The study concluded that expanding the number of patterns can enhance the accuracy of root extraction algorithms.

The Word Substring Stemming Algorithm (WSS), proposed by Yaseen and Hmeidi [14], generates all possible substrings of a word and matches them with a list of known roots to extract Arabic roots. The algorithm achieved an accuracy of 83.9% when tested on the Holy Quran. The algorithm's main strength is its ability to extract all possible roots of derived words. However, it generates a large number of roots, most of which are not related to the original root, and it may mistakenly remove some prefixes and suffixes that are actually part of the root.

Boudchiche et al. [15] presented the second version of AlKhalil Morpho analyzer, a tool for Arabic text processing that analyzes the morphological and syntactic structure of the Arabic language, with improved accuracy and efficiency. The tool can be used for various natural language processing tasks and was evaluated using different methods such as coverage, speed, and an average number of suggested root forms and proposed word stems per word. However, the authors did not evaluate the precision, recall, and F-measure of the tool due to the unavailability of a corpus with all possible features for each word.

Atta and Al-Hmouz [16] introduced a rule-based approach to extract Arabic roots using a set of rules and a dictionary containing stop words, affixes, and roots. The algorithm was tested on a set of 480 proverbs in Standard Arabic and achieved an accuracy of 74.11%. The algorithm's strength lies in its attempt to enhance existing algorithms by suggesting new rules and changing the order of rules. However, the algorithm has limitations, such as limited applicability of rules and decreased accuracy as word length increases.

Alnaied et al. [17] proposed a new method called Arabic Morphology Information Retrieval (AMIR) to generate Arabic word stems using a set of rules. AMIR outperformed other systems in terms of mean average precision, but it has some weaknesses, including its inability to extract the root of many words and its tendency to make errors in root extraction by removing some letters that are part of the root.

The research by R. Kanaan and G. Kanaan presents an improved algorithm for extracting triliteral Arabic roots [18]. In their work, Boudlal, Lakhouaja, Mazroui, and Meziane developed Alkhalil morpho sys1 [19], a morphosyntactic analysis system specifically designed for Arabic texts. The system provides detailed analysis and processing capabilities for Arabic linguistic features, aiding in various language processing tasks. In their research, Momani and Faraj proposed a novel algorithm for extracting tri-literal Arabic roots [20]. The algorithm introduces a new approach to accurately identify and extract the core tri-literal roots in Arabic words, contributing to Arabic language processing tasks. In the study by A. Belal [21], comprehensive processing techniques were developed for Arabic texts to extract their roots. The research focuses on providing robust methods for accurately identifying and extracting the roots of Arabic words, contributing to Arabic language analysis and processing tasks. Sonbol, Ghneim, and Desouki introduced a new approach for Arabic morphological analysis [22]. The study presents innovative methods and techniques for analyzing the morphological structure of Arabic language, contributing to the field of Arabic language processing and related applications. Hamza, Ahmed, and Hilal provide an overview of Arabic root extraction algorithms in the field of text mining [23]. The study explores various algorithms and techniques used for extracting roots from Arabic texts, highlighting their strengths and limitations. The survey serves as a comprehensive resource for researchers and practitioners interested in Arabic language processing and text mining.

Various algorithms for Arabic root extraction have been proposed in the literature, which aims to suggest new rules, change the order of rules, or increase the dictionaries of data rules. However, these algorithms were tested on specific corpora, and their efficiency decreased when tested on another corpus.

III. Methodology

In this research paper, the methodology section describes the procedures and techniques employed to conduct the study, providing a clear and concise explanation of the research dataset and methodology.

A. *The Dataset*

The research paper utilizes Thalji et al. dataset [7] and discusses the fundamental concepts of root extraction in Arabic. The dataset section is divided into three sections, namely roots, affixes, and patterns in Arabic known as "AWZAN".

1) Arabic roots: Arabic roots are the basic building blocks of words in the Arabic language [24]. The roots consist of two or more letters that combine to create a meaning. The roots can be sorted into categories based on their length and the type of letters they contain. The length-based roots are divided into five categories, and the second type of root is divided into two categories based on the type of letters they contain. This categorization helps provide insights into the morphology of the Arabic language and highlights the need for advanced root extraction algorithms to effectively handle the different types of roots. Upon examining Thalji et al.'s corpus [7], the roots can be sorted into categories based on their length and the type of letters they contain, as illustrated in Table I.

TABLE I. TYPES OF ARABIC ROOTS

Category	Sub Category	No of roots	Examples
Length-based root	2	480	(عض, "add", pride)
	3	8000	(درس, drasa, study)
	4	3112	(جمهر, jamher, mass)
	5	360	(غضنفر, adanfer, glory)
	6	48	(عنقفير, ankafeer, skillful)
Type of letters root	Vowel	3000	(يوم, yawem, day)
	Non-vowel	9000	(دفع, dfaa, pay)

2) Arabic affixes: Arabic prefixes are added at the beginning of a root word to modify its meaning or function, and can indicate various aspects of meaning [25]. Thalji et al.'s corpus includes 430 prefixes, with lengths ranging from one to six letters, and Table II provides statistical information on a subset of these prefixes. Understanding Arabic prefixes is important for comprehending the language and literature.

TABLE II. TYPES OF ARABIC PREFIXES

Prefixes (length-based)	Number	Examples
1	13	(ي, "yaa", the present tense for the male/s)
2	103	(ال, al, the)
3	146	(وال, wal, and the)
4	103	(ليست, leyasta, the present tense).
5	52	(كالمن, kaelmon, used for analogy)
6	13	(والاست, walest, used for noun)
Total	430	

The Arabic language has infixes, which are inserted within the root of a word to add more meaning. Thalji et al.'s corpus includes 11 Arabic infixes that are classified by length, ranging from one to two letters. It is important to note that a word may contain one or more of these infixes, or none at all. Table III shows a sample of these infixes and their corresponding statistics.

TABLE III. TYPES OF ARABIC INFIXES

Infixes Subcategory (length-based)	Number of prefixes	Prefixes
1	**4**	**ت, ي ,و, ا**
2	**7**	**وا, يا, او, يي, وي, يت, تا**

Arabic suffixes are added to the end of a root word to modify its meaning or grammatical function. They can indicate various aspects, including tense, aspect, voice, gender, number, and case. The study identifies 320 suffixes in the Arabic language, and Table IV shows the number of suffixes in each length category, along with examples for each type. The identification and understanding of Arabic suffixes are crucial for studying the Arabic language and literature. According to the same table, the length of three letters has the highest number of suffixes, with a total of 183. The length of four letters comes next with 94, and both the length of two and five letters have the same number of 40. The length of one letter has a total of 6, and finally, the length of six letters has only 3 suffixes.

TABLE IV. TYPES OF ARABIC SUFFIXES

Suffixes length	Number	Examples
1	6	(ت, "taa", indicates that the subject of the verb is a singular female)
2	40	(ات, at, indicates that the object is plural female)
3	132	(هما, homa, indicates that the object is dual)
4	94	(ناهم, nahom, indicates that the subject of the verb is a plural and indicates that the object is plural male).
5	40	(كموها, kamoha, indicates that the first object is a male plural and indicates that the second object is singular female).
6	3	(انيتان, entyan, indicates that the first object is a dual and indicates that the second object is dual also).
Total	320	

3) Arabic patterns: Arabic patterns are important for forming words in the Arabic language [26]. They consist of a sequence of letters added to a root to form a complete word with a specific meaning. The study identified 4320 Arabic patterns, ranging in length from three to twelve letters. The most common pattern length is seven letters, with 1296 patterns, followed by lengths of six and eight letters with almost equal numbers. The length of three letters has only one pattern, while the lengths of eleven and twelve letters have 43 patterns each, as shown in Table V. Understanding Arabic patterns is essential for mastering the language.

TABLE V. TYPES OF ARABIC PATTERNS

Pattern length	Number of patterns	Examples
3	1	(فعل, faal, indicates that one was doing something).
4	86	(يفعل, yafaal, indicates single male is doing something).
5	302	(فعلته, faalatho, indicates single female was doing something).
6	994	(الفعلة, alfealah, indicates singular noun).
7	1296	(يتفعلون, yatfaaloon, indicates plural male are doing something).
8	950	(يتفاعلان, yattafaalan, indicates dual male are doing something).
9	475	(واستفعلها, estafalaha, indicates single male was doing something and the object is femal).
10	129	(المفعولتان, almafolatan, indicates the object is dual female and it is noun).
11	43	(وافتعالاتهم, waeftealatehem, indicates that it is noun and plural female).
12	43	(والفوعلانيون, waltawalaneyoon, indicates that it is noun and plural male).
Total	4320	

B. Normalization

Before stemming, normalize the Arabic word by applying the following steps:

1) Remove diacritics punctuation and the Shadda.
2) Replace all distinct forms of Hamza with (أ).
3) Replace Madda (آ) with Hamza and Alef (أا).
4) Replace Alef Maksura (ى) with Alef (ا).

By applying these normalization steps, the algorithm ensures that all Arabic words are represented in a standardized form, which is essential for accurate and efficient root extraction.

C. Pattern Generation

Generate all possible patterns that match the word. The following template is used for the possible patterns: <prefix> <ف> <infix> <ع> <infix><1ل4><ل3><ل2> <ل><suffix>. The format consists of different parts:

- <prefix>: This part represents the letters that come before the root of the word, like in the case of the pattern (المفعلات) for the word(المدرسات), the prefix is (ال). Also, the prefix part can be empty, like in the case of the pattern (فاعلات) for the word(دارسات).
- <ف>: This part corresponds to the first root letter, like in the case of the pattern (المفعلات) for the word (المدرسات), the <ف> letter is corresponding to the root letter(د). If the root consists of only two letters, the first root letter may not exist, in the case of the pattern (عل) of the word (قف), the initial letter of the original word (وقف) was removed, and this led to the deletion of the letter (ف) from the pattern, resulting in (عل).
- <infix1>: This part represents the infix letters that can appear between the first root letter and the second root letter, like in the case of the pattern (مفتعلون) for the word (منتشرون), the < infix1> letter that appears between the first root letter and the second root letter is (ت). This part may be empty, like in the case of the pattern (المفعلات) for the word(المدرسات), there is no infix letter between the first root letter and the second root letter.
- <ع>: This part represents the second letter of the root, like in the case of the pattern (المفعلات) for the word (المدرسات), the <ع> letter is corresponding to the root letter(ر). If the root consists of only two letters, the first root letter may not exist, in the case of the pattern (فل) of the word (قل), the second root letter of the original word (قول) was removed, and this led to the deletion of the letter (ع) from the pattern, resulting in (فل).
- <infix2>: This represents any infix letters that can appear between the second root letter and the third root letter, like in the case of the pattern (مفعول) for the word (منشور), the < infix2> letter that appears between the second root letter and the third root letter is (و). This part may be empty, like in the case of the pattern (المفعلات) for the word(المدرسات), there is no infix letter between the second root letter and the third root letter.
- <1ل>: This represents the third letter of the root, like in the case of the pattern (المفعلات) for the word (المدرسات), the <1ل> letter is corresponding to the root letter(س). If the root consists of only two letters, the third root letter may not exist, in the case of the pattern (افع) of the word (ارم), the third root letter of the original word (ارمي) was removed, and this led to the deletion of the letter (1ل) from the pattern, resulting in (افع).
- <2ل>: This represents the fourth letter of the root, like in the case of the pattern (متفعلل) for the word (متدحرج), the <2ل> letter is corresponding to the root letter(ج). If the root consists of three or two letters, the fourth root letter does not exist, in the case of the pattern (فاعل) of the word (دارس), the fourth root letter does not exist.
- <3ل>: This part represents the fifth letter of the root, like in the case of the pattern (فعللل) for the word (سفرجل), the <3ل> letter is corresponding to the root letter(ل). If the root consists of four or fewer letters, the fifth root letter does not exist, in the case of the pattern (فاعل) of the word (دارس), the fifth root letter does not exist.
- <4ل>: This part represents the sixth letter of the root, like in the case of the pattern (فعلللل) for the word (عنقفير), the <3ل> letter is corresponding to the root letter(ر). If the root consists of five or fewer letters, the sixth root letter does not exist, in the case of the pattern (فاعل) of the word (دارس), the fifth root letter does not exist.
- <suffix>: This part represents the letters that come after the root of the word, like in the case of the pattern (المفعلات) for the word(المدرسات), the suffix is (ات). Also, the suffix part can be empty, like in the case of the pattern (مفاعل) for the word(مدارس).

D. Remove Non-Patterns

Note that this approach generates all possible patterns that match the word, but not all of them will actually be valid Arabic patterns. Some of the generated patterns may need to be filtered. The identified patterns are matched against the set of patterns stored in the database. Any pattern that is not present in the database is eliminated.

E. Extract the Roots

Once the potential patterns of the word have been determined, the next step is to identify the root of the word, which can be done by following these instructions:

- The initial letter of the root is the one that corresponds to <ف>. Like in the case of the pattern (المفعلات) for the word (المدرسات), the <ف> letter is corresponding to the root letter(د).
- The second letter of the root is the one that corresponds to <ع>. Like in the case of the pattern (المفعلات) for the word (المدرسات), the <ع> letter is corresponding to the root letter(ر).
- The third letter of the root is the one that corresponds to <ل1>. Like in the case of the pattern (المفعلات) for the word (المدرسات), the <ل1> letter is corresponding to the root letter(س).
- The third letter of the root is the one that corresponds to <ل2>. Like in the case of the pattern (متفعلل) for the word (متدحرج), the <ل2> letter is corresponding to the root letter(ج).
- The third letter of the root is the one that corresponds to <ل3>. Like in the case of the pattern (فعللل) for the word (سفرجل), the <ل3> letter is corresponding to the root letter(ل).
- The third letter of the root is the one that corresponds to <ل4>. Like in the case of the pattern (فعلللل) for the word (عنقفير), the <ل3> letter is corresponding to the root letter(ر).

IV. Result and Discussion

This section presents the study findings by describing the testing and comparison of the algorithm with different algorithms on various datasets.

A. Testing the Algorithm in Different Datasets

The suggested algorithm is tested on three different corpora, and its effectiveness is evaluated using precision, recall, and F1 score metrics based on root length and type. Thalji et al. corpus, which includes 720,000 word-root pairings, is divided into five categories based on root length and two types based on letter type. Alshawakfa et al.'s corpus [27], comprises 27,628,821 word-root pairs. The corpus includes only trilateral roots and is divided into two types of roots based on root letter type: vowel roots (VR) and non-vowel roots (NVR). Alkabi et al.'s corpus [26], which includes 6081 word-root pairs. This corpus is distributed into two root types based on their length, namely TR and QR, and two types of roots based on their letter type: VR and NVR. Table VI summarizes the average results across all corpora and compares the performance of the algorithm on three different datasets. In cases where a particular root type is not listed in a corpus, such as BR in Alshawakfa et al.'s corpus, the column is marked with a "-". The table indicates that the proposed algorithm consistently produces high values for precision, recall, and F1 score, regardless of the type of corpus used.

TABLE VI. Summary of the System Evaluation using Three Different Corpora

Roots type	Precision			Recall			F-Measure		
	Thalji	Alshawakfa	Alkabi	Thalji	Alshawakfa	Alkabi	Thalji	Alshawakfa	Alkabi
2	82	-	-		-	-	84	-	-
3	81	88	82	93	100	95	87	94	88
4	87	-	86	92	-	88	89	-	87
5	90	-	-	92	-	-	91	-	-
6	93	-	-	85	-	-	89	-	-
Avg.	87	88	84	90	100	92	88	94	88
VR	70	77	71	90	100	82	79	87	76
NVR	90	97	85	94	100	97	92	99	90
Avg.	80	87	78	92	100	90	85	93	83

B. Comparing the Algorithm with other Root Extraction Algorithms

To evaluate the accuracy of the proposed algorithm, a comparison is made with three other Arabic root extraction algorithms, namely Khoja and Garside's algorithm [10], Sonbol et al.'s algorithm [22], and Alkabi et al.'s algorithm [26]. These algorithms are established and respected methods in Arabic language applications and have demonstrated high accuracy performance in their respective studies. The comparison is conducted on the Thalji et. al.'s corpus, Alshawakfa et al.'s corpus, and Alkabi et al.'s corpus, and the results are presented in Table VII.

TABLE VII. Accuracy Performance of the Four Algorithms

The algorithm	**Corpus type**			**Average**
	Thalji's corpus	**Alshawakfa et al.'s corpus**	**Alkabi et al.'s corpus.**	
Khoja and Garside's algorithm	63%	34%	74%	57%
Sonbol et al.'s algorithm	68%	24%	65%	52%
Alkabi et al.'s algorithm	70%	35%	76%	60%
The proposed algorithm	92%	100%	95%	96%

The proposed algorithm for Arabic root extraction outperformed previous algorithms in terms of accuracy according to Table VI and VII. The proposed algorithm returns all potential roots for non-vocalized words and includes all types of roots, not just one specific type. Previous algorithms returned only one root for non-vocalized words and ignored other possible solutions, resulting in a decrease in accuracy. Additionally, previous algorithms suffered from the reduction of the content of the lists used, ambiguity problems, and the inability to extract roots for words without any consonant letters. Overall, the proposed algorithm had the highest accuracy rate compared to previous algorithms.

The proposed algorithm has limitations in dealing with derived words that consist of only one letter, as they are not considered in the algorithm. These derived words originate from weak roots with three letters, wherein the weak letters are omitted during the derivation process. Moreover, the algorithm's tendency to generate all possible roots can result in confusion when attempting to identify the specific root required, as it primarily focuses on individual derived words rather than considering the context of complete meaningful sentences or paragraphs. Future work can address these limitations by enhancing the algorithm to handle one-letter derived words and incorporating techniques such as natural language processing or linguistic analysis to improve its contextual understanding and accuracy in root extraction.

V. CONCLUSION

This study examined the algorithms used to extract Arabic word roots and found that their accuracy is affected by the lack of comprehensive lists and essential rules. Previous research focused on trilateral roots and ignored other types, resulting in incorrect results for non-vowel roots, which make up 75% of all roots. Weak roots were also found to be a major cause of failure in previous Arabic root extraction algorithms due to their numerous irregular cases.

This study proposes a new algorithm for extracting Arabic word roots and compares its accuracy with three commonly used algorithms. The proposed algorithm generates all potential derivation roots of a word without eliminating affixes first, which is different from previous algorithms. The study uses Thalji et al.'s corpus to utilize the maximum amount of content available in the lists. The proposed algorithm achieves an average accuracy of 96%, which is significantly higher than the accuracy of the other three algorithms.

REFERENCES

[1] N. Thalji and S. Alhakeem, "Developing an effective light stemmer for Arabic language information retrieval," International Journal of Computer and Information Technology, vol. 5, no. 1, pp. 55-59, 2016.

[2] N. K. Masrei, "An innovative automatic indexing method for Arabic text,". M.S. thesis, Dept. Comput. Sci., Lebanese Am. Univ., Lebanon, 2020.

[3] N. Thalji, N. Hanin, W. Bani-Hani, S. Al-Hakeem and Z. Thalji, "A novel rule-based root extraction algorithm for Arabic language," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 10, pp. 120-128, 2018.

[4] M. S. S. Sawalha, Open-source resources and standards for Arabic word structure analysis : Fine grained morphological analysis of Arabic text corpora, Leeds, UK: University of Leeds, 2011.

[5] N. Thalji, N. Hanin, Z. Thalji, W. Bani Hani and S. Al-Hakeem, "Towards improving rule-based Arabic root extraction algorithm for non-vocalized text," Int. J. Comput. Inf. Technol., vol. 7, no. 6, pp. 235-242, 2018.

[6] N. Thalji, N. Hanin, Z. Thalji and S. Al-Hakeem, "Enhancing the accuracy of Sonbol’s Arabic root extraction algorithm," Jordan. J. Comput. Inf. Technol., vol. 4, no. 3, pp. 159-174, 2018.

[7] N. Thalji, A. Hanin, Y. Yacob and S. Al-Hakeem, "Corpus for test, compare and enhance Arabic root extraction algorithms," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 5, pp. 229-236, 2017.

[8] S. Al-Fedaghi and F. Al-Anzi, "A new algorithm to generate Arabic root-pattern forms," in in Proc. 11th Nat. Comput. Conf. Exhib., Dhahran, Saudi Arabia, 1989, pp. 04-07.

[9] H. Al-Serhan, R. Al-Shalabi and G. Kannan, "New approach for extracting Arabic roots," in in Proc. 2003 Arab Conf. Inf. Technol., Egypt, 2003, pp. 42-59.

[10] S. Khoja and R. Garside, "Stemming Arabic text," Ph.D. dissertation, Computing Department, Lancaster University, Lancaster, UK, 1999.

[11] K. Taghva, R. Elkhoury and J. Coombs, "Arabic stemming without a root dictionary," in in Proc. Int. Conf. Inf. Technol.: Coding and Computing (ITCC'05) - Volume II, Las Vegas, NV, USA, 2005, pp. 152-157.

[12] S. Ghwanmeh, S. Rabab'ah, R. Al-Shalabi and G. Kanaan, "Enhanced algorithm for extracting the root of Arabic words," in Proc. Sixth Int. Conf. Comput. Graphics, Imaging and Visualization, pp. 388-391, 2009.

[13] M. Al-Kabi, "Towards improving Khoja rule-based Arabic stemmer," in IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, Amman, 2013, pp. 1-6.

[14] Q. Yaseen and I. Hmeidi, "Extracting the roots of Arabic words without removing affixes," J. Inf. Sci., vol. 40, no. 3, pp. 376-385, 2014.

[15] M. Boudchiche, A. Mazroui, M. Bebah, A. Lakhouaja and A. Boudlal, "AlKhalil morpho sys 2: A robust Arabic morpho-syntactic analyzer," J. King Saud Univ. Comput. Inf. Sci., vol. 29, no. 2, pp. 141-146, 2017.

[16] H. Atta and A. Al-Hmouz, "Enhanced Arabic root-based lemmatizer," M.S. thesis, Dept. Comput. Sci., Middle East University, Amman, 2020.

[17] A. Alnaied, M. Elbendak and A. Bulbul, "An intelligent use of stemmer and morphology analysis for Arabic information retrieval," Egypt. Inform. J., vol. 21, no. 4, pp. 209-217, 2020.

[18] R. Kanaan and G. Kanaan, "An improved algorithm for the extraction of triliteral Arabic roots," European Scientific Journal, vol. 10, no. 3, pp. 346-355, 2014.

[19] A. Boudlal, A. Lakhouaja, A. Mazroui and A. Meziane, "Alkhalil morpho sys1: A morphosyntactic analysis system for Arabic texts," in International Arab Conference on Information Technology, New York, 2010, pp. 1-6.

[20] M. Momani and J. Faraj, "A novel algorithm to extract tri-literal Arabic roots," in 2007 IEEE/ACS International Conference on Computer Systems and Applications, Amman, 2007, pp. 309-315.

[21] A. Belal, "Comprehensive processing for Arabic texts to extract their roots," Iraqi Journal of Science, vol. 60, no. 6, pp. 1404-1411, 2019.

[22] R. Sonbol, N. Ghneim and M. Desouki, "Arabic morphological analysis: A new approach," in In 3rd International Conference on Information and Communication Technologies: From Theory to Application, Damascus, Syria, 2008, pp. 1-6.

[23] M. Hamza, T. Ahmed and A. Hilal, "Text mining: A survey of Arabic root extraction algorithms," International Journal of Advanced and Applied Sciences, vol. 8, no. 1, pp. 11-19, 2021.

[24] K. Abainia, S. Ouamour and H. Sayoud, "A novel robust Arabic light stemmer," Journal of Experimental and Theoretical Artificial Intelligence, vol. 29, no. 3, pp. 557-573, 2017.

[25] H. Alshalabi, S. Tiun, N. Omar, F. AL-Aswadi and K. Alezabi, "Arabic light-based stemmer using new rules," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 9, pp. 6635-6642, 2022.

[26] M. Al-Kabi, S. Kazakzeh, B. Abu-Ata, S. Al-Rababah and I. Alsmadi, "A novel root based Arabic stemmer," J. King Saud Univ. - Comput. Inf. Sci., vol. 27, no. 2, pp. 94-103, 2015.

[27] E. Alshawakfa, A. Al-Badarneh, S. Shatnawi, K. Al-Rabab’ah and B. Bani-Ismail, "A comparison study of some Arabic root finding," J. Am. Soc. Inf. Sci. Technol., vol. 61, no. 5, pp. 1015-1024, 2010.