ISSN: 0067-2904

A Survey of Arabic Root Extraction Algorithms

Nisrean Jaber Thalji*
Department of Artificial Intelligence and Robotics, Faculty of Science and Information Technology, Jadara University, Irbed, The Hashemite Kingdom of Jordan

Received: 13/4/2022 Accepted: 25/3/2023 Published: 30/4/2024

Abstract

The importance of a survey of Arabic root extraction algorithms is that it can provide a comprehensive overview of the current state of the field and the various approaches used for this task. This can be helpful for academics and professionals working on Arabic natural language processing projects, since it enables them to comprehend the advantages and disadvantages of several approaches and select the one that is best suited to the task at hand. A survey can also help to pinpoint areas that require more research and can serve as a springboard for fresh studies in this area. In general, a survey can enhance the study of Arabic root extraction and improve the quality of Arabic natural language processing tools. Researchers can use this survey to compare the performance of several algorithms and select the best one for their particular application. In this survey, the majority of the root extraction algorithms for the Arabic language were studied with a focus on identifying their advantages and disadvantages, as well as their accuracy. The algorithms were classified into various approaches. The importance of root extraction in Arabic is explored, along with its uses and applications. Also, the issues that still require research and improvement in order to improve the efficiency of algorithms were highlighted. For instance, one issue that the algorithms encounter is the absence of a unified list that encompasses all roots, patterns, and affixes. Additionally, more rules are needed to regulate the process of root extraction.

Keywords: Arabic root extraction algorithm, Lemmatization, Natural Language Processing.

1. Introduction

The Arabic language is spoken by over 420 million people worldwide [1]. It is the official language in 26 countries, including Jordan, Iraq, Saudi Arabia, etc., and it is also widely spoken in many other countries in the Middle East and North Africa, in addition to being the language of the Quran and the language of Muslims throughout the world.

Arabic root extraction is a process used to find the root of a word in the Arabic language [2]. In Arabic, words are usually built from a set of two or more consonants called a "root." These roots form the basis of the word's meaning, and various affixes can be added to the root to create different forms of the word. For illustration, the root K-T-B (كتب) is the basis for words such as "kitab" (book), "kutub" (books), "maktab" (office), "maktaba" (library), "kataba" (he wrote), "katib" (writer), and "takhtobo" (you wrote). Each of these words is related to the idea of writing and books, but they all have different forms and meanings [1].

The Arabic root extraction process involves identifying a word root and then analyzing it to understand the meaning and structural function of the word. This process is useful for lexicography, lexicology, and computational linguistics studies.
Various approaches are used to process and analyze the Arabic word, which can be summarized in the following four approaches:

1.Root extraction: It refers to the process of identifying the root of a word in Arabic, which is typically a three-letter string. This is useful for understanding the meaning of a word, as many words in Arabic are derived from a common root. The Arabic language is distinguished by its roots and pattern systems, where all words are derived from a specific root. These root consonants are then arranged in patterns to form a range of related words [3]. An Arabic word originates from a root by attaching certain affixes in accordance with a regular set of word patterns [4]. The CV(C: consonant, V: vowel) provides an abstract illustration of the ordering of the root and short vowels (and some affixes) to form the stem [5].

2.Lemmatization: It is the procedure of reducing a word to its base form, typically the form of the word that is found in a dictionary [6]. This is useful for text analysis tasks such as information retrieval and text classification. It is necessary to get acquainted with some of the basic concepts used in root extraction processes to clarify and remove ambiguity [7]. The lemmatization process entails identifying each word of the text and linking it to a specific canonical word. In the Arabic language, the terms "lemma" and "root" are not always synonymous. The root of a word is the set of consonants that make up the word's core meaning, while the lemma is the base form of a word, which can be inflected to create different forms. The root is the foundation of the word, where it's derived from, while the lemma is the form that is used as a reference in dictionaries. In some cases, the lemma and root are the same, but in other cases, the lemma is derived from the root by adding prefixes, suffixes, or other affixes to change the word meaning or grammatical function, so it's different from the root. An example of a word where the lemma is different from the root is the word "جامعيات" (jameeyyat). This means "university degree holders" is derived from the root "ج-م-ع" (aa-ma-ja) but the lemma is "جامعي" (jameiyy) which means "university degree holder" [8].

3.Stemming: It is the process of reducing a word to its stem form, typically by removing prefixes and suffixes [7]. For example, in the word "يُشْرِبُ" (yushribu) which means "he drinks," the stem of this word is "شرب" (sh-r-b), which is the root of the word, and the inflectional suffixes "ي" (y) and "ُ" (u) have been removed. Another example is the word "الطالبات" (al-talabat), which means "the female students." The stem of this word is "طالب" (talab), the inflectional prefix "ال" (al), and the suffix "ات" (at) have been removed. The root of the word "الطالبات" (al-talabat) is "طلب" (T-L-B) [9].

4.Morphological analyzers: They are software tools that analyze the structure of words [10]. It can give several pieces of information about an Arabic word, including the root of the word, the stem of the word, and the inflectional patterns. The patterns are the grammatical markers, such as prefixes and suffixes, that indicate the tense, mood, person, gender, and number of the word. It can also provide the part of speech of the word, such as noun, verb, adjective, etc. In addition, the analyzer can generate the possible inflected forms of a word based on the root and the inflectional patterns. Moreover, if the morphological analyzer is integrated with a dictionary or a corpus, it can give the meaning of the word [8]. Figure 1**:** The differences between various approaches that are used to analyze the Arabic word illustrates the differences between these four approaches.

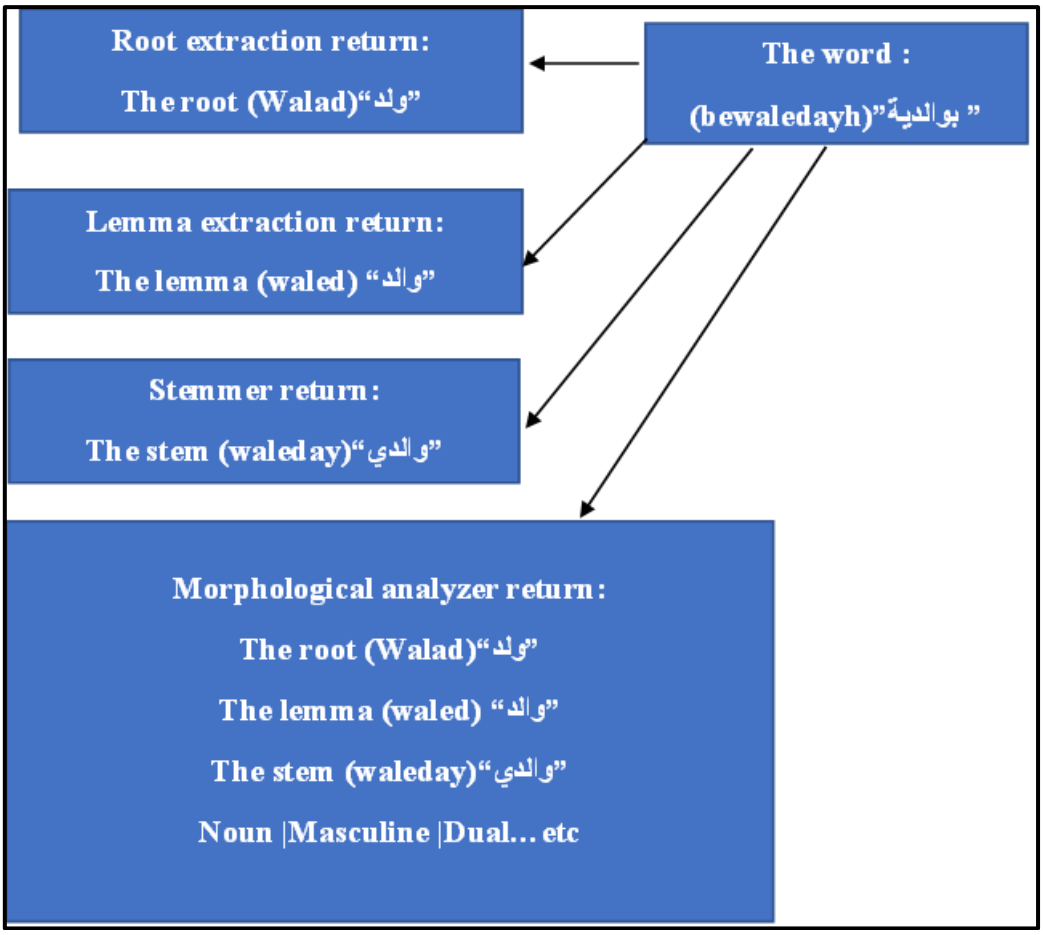

Figure 1: The differences between various approaches that are used to analyze the Arabic word

The word "بوالدية" (bewaledayh) is derived from the root "ولد" (walad); the stem is "والدي" (waleday); and the lemma is "والد" (waled). A morphological analyzer can give several pieces of information about the Arabic word other than the root, lemma, and stem. As the part of speech of the word is noun, it has the preposition "ب" (ba), the gender is masculine, the number is dual, etc. [8].

All of these approaches can be utilized to increase the accuracy of Arabic applications; such applications include:

• Part-of-speech tagging: This method is utilized to determine the grammatical function of each word in a sentence [11].

• Named entity recognition: This method is employed to recognize and categorize named entities such as individuals, organizations, and locations within a text [12].

• Text classification: This technique is used to classify text into predefined categories based on its content [13].

• Word segmentation: This technique is used to separate words in a text written in languages such as Arabic, Chinese, and Japanese that do not use spaces between words [8].

• Text summarization: This technique is used to generate a shorter version of a text that contains the main ideas or points [8].

• Information retrieval: Root extraction can be used to improve the accuracy of information retrieval systems by allowing them to better understand the meaning of words in a query [8].

• Machine Translation: Extracting roots can improve the performance of machine translation systems by allowing them to better understand the meaning of words in the source language and generate more accurate translations [14].
• Spell checking: Root extraction can be used to improve the accuracy of spell-checking systems by allowing them to identify correctly spelled words that are derived from the same root [11].
A survey of Arabic root extraction algorithm research is important for several reasons:
• Understanding the state-of-the-art: Surveying existing research allows one to understand the current state-of-the-art in Arabic root extraction algorithms and identify areas where further research is needed.
• Identifying research gaps: Surveying existing research can help identify gaps in the current body of knowledge and areas where more research is needed.
• Improving existing algorithms: By reviewing existing research, one can gain insight into how to improve existing algorithms or develop new algorithms that are more effective and efficient.
• Enhancing natural language processing (NLP) applications: Arabic root extraction algorithms are crucial for several NLP applications, such as text classification, information retrieval, machine translation, and text summarization.
• Language Specific: The Arabic language has a complex morphological structure, making it essential to have a thorough understanding of the language-specific methods and techniques used for root extraction.
• Cultural and historical context: the Arabic language has a rich cultural and historical context, and understanding the roots of Arabic words can provide deeper insight into these contexts.

In this research, a brief background on a number of root extraction algorithms is presented. Then, a comparison and discussion are made of the selected algorithms in terms of strengths and weaknesses, accuracy, data set, and method.

2. Literature review

Numerous studies have been conducted to review prior research in an effort to extract the root of the Arabic language or other languages and to carry out stem or morphological analysis. In this section, some of these studies are discussed.
El Sayed et al. [15] present a review of various techniques for extracting Arabic roots. The survey categorizes the Arabic root extraction algorithms into 20 separate methods. The main approaches utilized in these techniques are either morphological or statistical. The morphological analysis involves identifying the morphemes, affixes, patterns, and roots of a word, while the statistical approach calculates the semantic similarity or dissimilarity between words by examining shared substrings of a specific length. Both methods heavily rely on a dictionary as a reference for extracting or verifying the root. Statistical methods do not require prior language knowledge, predefined rules, or a vocabulary database, while morphological methods are more efficient for complex words not listed in dictionaries.
Hamza M. et al. [14] explore and analyze the research conducted on Arabic word root extraction and stemming algorithms. It provides a background and in-depth examination of several algorithms that extract the root of Arabic words in light, heavy, hybrid, leading, and Markovian forms. The paper also offers an overview of various stemming algorithms for extracting the root and stem of Arabic words and then compares and discusses a selection of these algorithms based on accuracy, dataset, and stemming method. The strengths and weaknesses of each algorithm are also evaluated.
Al-Sughaiyer and Al-Kharashi [16] present a survey that consolidates and organizes the information present in the literature to encourage further research and development in the field. Their work provides an introduction and overview of Arabic morphological analysis

techniques, classifying them into different categories. This research highlighted and summarized the weaknesses that previous algorithms suffered from.

Alothman and Alsalman [5] conduct a review of Arabic morphological analysis techniques from 2005 to 2019 and categorize them into a comprehensive and adaptable classification system. The paper evaluates the current Arabic morphological analyzers, reaches conclusions, and suggests future research directions for Arabic morphological analysis in order to advance the field and support new research. The findings of the study indicate that the two main methods for morphological analysis are linguistic lexicon-based and data-driven lexicon-based approaches, and all techniques found in the relevant literature fit into these two categories.

Mustafa et al. [17] aim to comprehensively examine the state of the art for stemming Arabic text. It covers a range of algorithms and discusses them in detail. The paper's main objective is to offer a clearer understanding of the various approaches and to lay the foundation for the development of a highly accurate and effective Arabic stemmer in the near future.

In almost all studies that reviewed prior research about root extraction algorithms for the Arabic language, the authors focused on reviewing existing algorithms, recognizing their shortcomings, and offering potential solutions.

3. Arabic roots extraction algorithms

In this section, a majority of the studies that were conducted on extracting the root of the Arabic language were presented, and they were classified according to the approach they used. These algorithms can be grouped into three distinct approaches: the dictionary-based approach, the statistical-based approach, and the rule-based approach.

3.1 Dictionary-based approach

This approach for extracting the root of an Arabic word involves searching for the word in a root-word dictionary and, if it is found, returning its root. Another approach is using the database search method, where the algorithm checks if a given word matches any patterns stored in a database. If a match is found, the algorithm extracts the roots of the word and compares them to a list of roots stored in the database. If a match is found, the algorithm returns the matched roots as the result [18].

Alfedaghi and Al-Anzi [19] propose a database-based approach for finding the roots of Arabic words. These algorithms use two main lists: one for roots and another for patterns. The words are compared against the pattern list. If a match is found, the root is extracted using the matched pattern. The extracted root is then compared to the list of roots. If a match is found, the matched root is returned as the result. Figure 2 summarizes the description of the Alfedaghi and Al-Anzi algorithm.

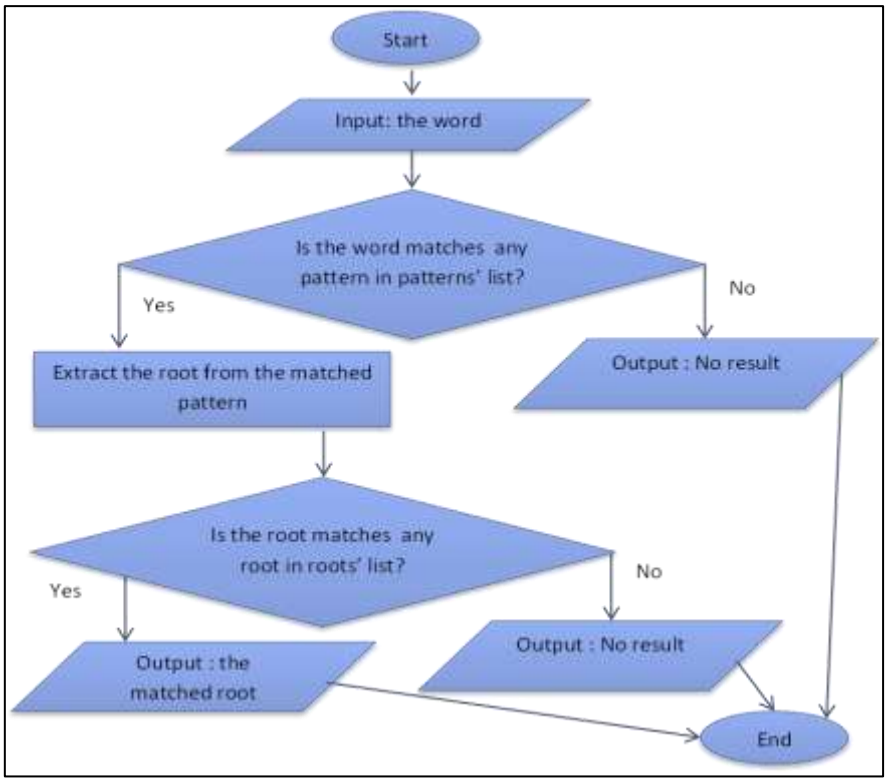

Figure 2: A description of the Alfedaghi and Al-Anzi algorithm

An advantage of this algorithm is that the database search is efficient and does not involve analyzing individual words. However, a drawback is that they require significant resources and precision in creating the necessary databases. Furthermore, they require updating if no pattern or root is found, and if multiple patterns match in the pattern list, they only return one root, whereas multiple roots should be returned to match the number of matched patterns [20].

3.2 Statistical based approach

To determine a word root form from an inflected word, these methods rely on statistical models. Al-Serhan et al. [21] present a statistical-based approach to finding the roots of Arabic words. The algorithm assigns a weight to each Arabic letter based on its position in the word and then uses mathematical equations to identify the root. Figure 1 presents the algorithm of the Al-Serhan et al. algorithm [21].

Inputs: Arabic word (string)
Outputs: Root of the Arabic word (string)

1. Assign weight values to each Arabic letter based on Table 1.
2. Set weight of each letter that is part of the input word, "سألتمونيها", to a nonzero value.
   - Set weight of a zero to the rest of the letters.
3. Assign letter order values to each letter in the input word based on Table 2.
   - For each letter in the input word, assign an order value based on its location in the word and the length of the word.
4. Remove stop words from the input word using a stop words removal module.
5. For each remaining letter in the input word, multiply its weight with its order value to get a product value.
6. Select the letters with the minimum product values and concatenate them in order to form the root of the input word.
7. Output the root.

Figure 1: The algorithm of Al-Serhan et al. algorithm

The algorithm defines weight values for each Arabic letter, as specified in Table 1. The letters "ة" and "ا" have a weight of 5. The letters "ئ" and "ي" have a weight of 3.5. The letters "ى", "و", and "ت" have a weight of 3. The letters "أ", "م", "ن", and "ا" have a weight of 2. The letters "س", "هـ", and "ل" have a weight of 1. The rest of the letters have a weight of 0.

Table 1: Weight values

Weight	Letters
5	ة, ا
3.5	ئ, ي
3	و, ى , ت
2	ن, م, أ, ا
1	هـ, س, ل
0	The rest of letters

The algorithm further assigns letter order values based on the length of the word and the location of the letter, as per Table 2.

Table 2: Order values

Word letter	Order values (if n even)	Order values (if n odd)
L(n+i)	(n/2+i)-0.5	(n/2+i)-1.5

A key advantage of this algorithm is that it is efficient and does not require word analysis. However, one of its limitations is that it does not differentiate between constant and non-constant letters in many cases and treats them with the same priority. Additionally, these approaches may struggle with dealing with the complexity and irregularities of the Arabic language, such as the many different forms of words and the numerous variations in spelling and pronunciation. Another limitation is that the results of the statistical approach may not be as accurate as rule-based approaches, especially when it comes to dealing with rare words or words with multiple possible roots. Also, statistical approaches may not provide a clear explanation for how the root was extracted, making it difficult to understand or improve the algorithm. The authors of this work did not measure the accuracy of their algorithm. However, it has been evaluated in other studies, such as the Alshawakfa et al. study [22], and found to have an accuracy of 14%.

3.3 Rule-based approach

This approach relies on a set of predefined morphological rules to extract the root form of an inflected word. These rules are based on the linguistic structure of the Arabic language and are used to break down the inflected word into its root form. A rule-based approach for extracting the roots of Arabic words can utilize a database of words, roots, prefixes, suffixes, and patterns to help guide the algorithm in determining the root of a given word. This database can be used to store the rules, patterns, and words that the algorithm needs to apply during the root extraction process. The database can be used to store the words and roots that the algorithm has already processed as well as the rules and patterns that it needs to use in order to determine the root of a word [23].

Atta and Al-Hmouz [24] present a rule-based approach to finding the roots of Arabic words. The proposed algorithm used a set of rules to extract the roots and used a dictionary. The rules are based on two factors: the length of the word and the position of the letter in the word. The dictionary consists of stop words, affixes, and roots. The algorithm is tested on a set of proverbs

written in the standard Arabic language, which contains 480 proverbs with 2,493 words, including 1637 unique words. Figure 2 presents the phases of the Atta and Al-Hmouz algorithm [24]. The strength of this algorithm is that it tries to enhance the existing algorithms by changing the order of rules and suggesting new rules. Weaknesses in this algorithm are that the rules may apply to one set of words but not to another.

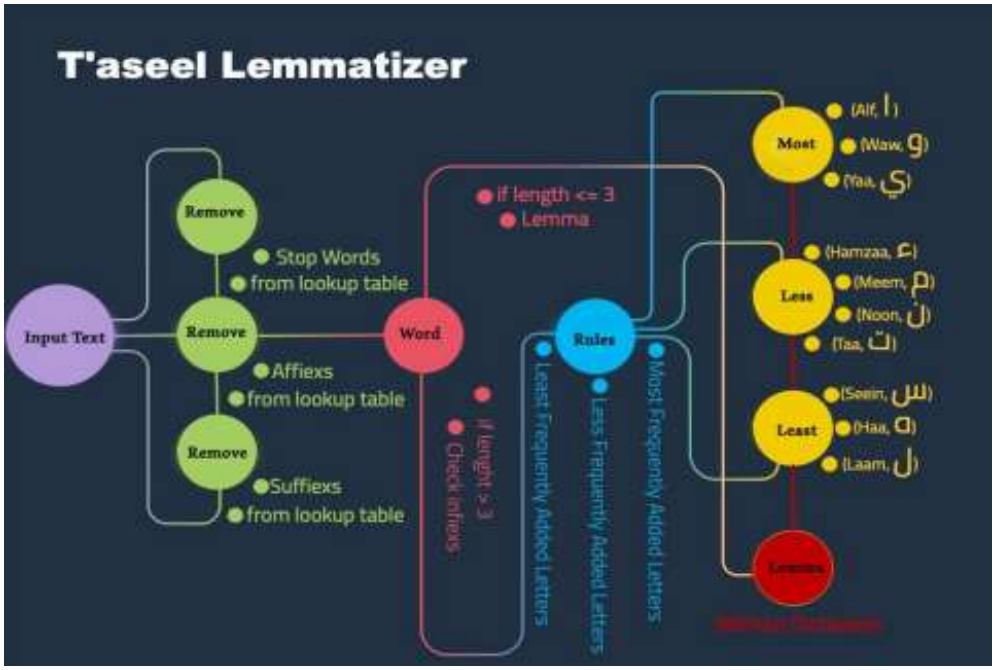

Figure 2: The phases of the Atta and Al-Hmouz algorithm

The accuracy of the algorithm decreases as the word length increases. The result of the algorithm was compared with the algorithm of Abainia et al. [10] and the algorithm of Taghva et al. [25]. The accuracy is 74.11%, and the performance is better than the algorithms of Abainia et al. and Taghva et al.

Sonbol et al. [26] developed a new algorithm for extracting Arabic roots that is rule-based and categorizes Arabic letters into five groups: constant letters, prefix letters, suffix letters, prefix-suffix letters, and extra letters. The general description of the algorithm is presented in Figure 3. The authors of the study tested their algorithm using two separate datasets. The first dataset was made up of 167162 word-root pairs, while the second dataset was a collection of 585 Arabic articles, which included 377793 words. The algorithm was found to be 96% accurate in its results. The new algorithm has several advantages over previous algorithms, one being that it categorizes Arabic letters into five groups, which reduces ambiguity in affix removal. Additionally, it incorporates new rules. However, there are also limitations such as missing many rules, such as when the root does not contain a constant letter, when the root does not start with a constant letter, and when the root contains only one constant letter. The algorithm also lacks many roots, prefixes, suffixes, and patterns and only provides one solution for non-vocalized words, disregarding other potential solutions. Al-Shawakfa et al. [22] evaluated the performance of the algorithm using their own corpus and found that it had an accuracy of 24%. This suggests that the algorithm is not very reliable, as its accuracy can vary significantly depending on the dataset used. Each algorithm is tailored to work best with the corpus it was designed for.

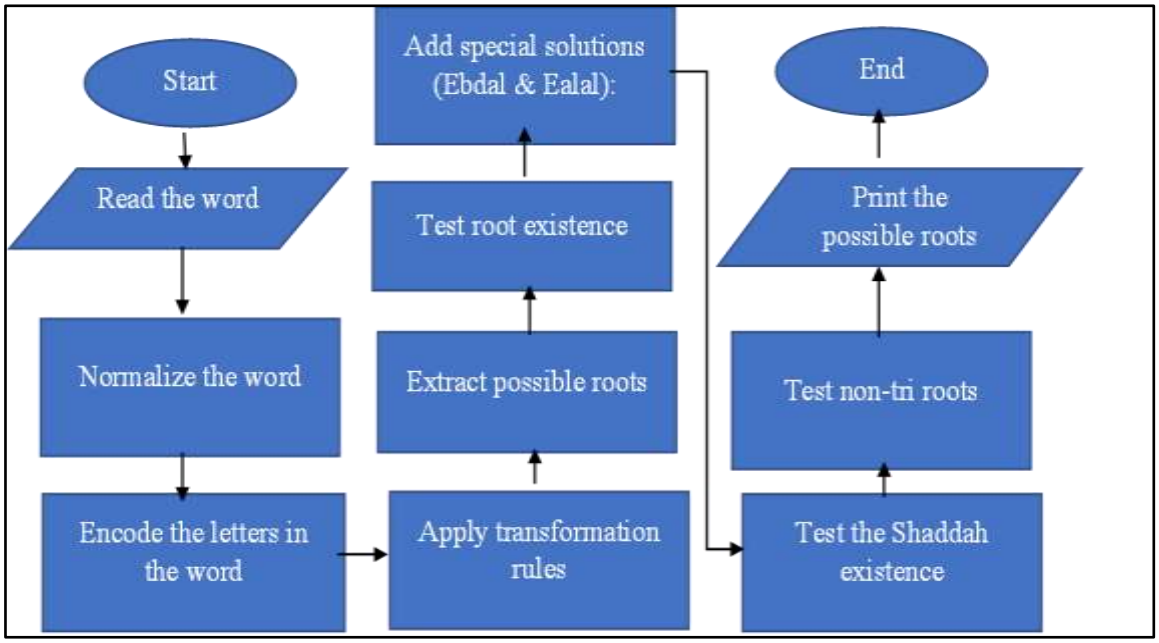

Figure 3: A general description of the Sonbol et al. algorithm

Boudchiche et al. [27] presented the second version of AlKhalil Morpho analyzer. The database of the first version is enhanced, and then the performance of the analyzer is enhanced. The algorithm for their algorithm is shown in Figure 4. AlKhalil Morpho Sys 2 is used to analyze and process Arabic text, specifically in terms of its morphological and syntactic structure. The tool is designed to be accurate and efficient and can be used for a variety of natural language processing tasks, such as text-to-speech synthesis, machine translation, and information retrieval.

1. Pre-processing:
The text is divided into words, and kashida, diacritic marks, and non-Arabic characters are removed.
2. Segmentation:
The system analyzes each pre-processed word and breaks it down into proclitic, stem, and enclitic parts. It generates all possible segmentations and keeps only the valid ones.
3. Analysis of the stem:
For each valid segmentation, the system performs a four-step analysis of the stem:
3.1. The stem as an exceptional word.
3.2. The stem as a non-derived word.
3.3. The stem as a derived noun.
3.4. The stem as a verb.
4. Validation of results:
The results are validated through concordance checks between clitics and syntactic features, hamza allograph, and diacritic marks.
5. Display of results:
The analyzer displays all possible solutions associated with their morphosyntactic features for a given word.

Figure 4: The Algorithm of AlKhalil Morpho Analyzer

AlKhalil Morpho Sys 2 uses various evaluation methods to test the performance of the algorithm. Some common methods used to evaluate morphological analyzers include: Coverage refers to the proportion of words analyzed by the tool, while speed measures how many words can be analyzed per second. AN_Lemma represents the average number of suggested root forms per word, AN_Stem shows the average number of proposed word stems per word, and AN_Diac indicates the average number of potential vowelized forms per word, without the diacritical

mark on the final character. The developers use a combination of these methods to evaluate the performance of AlKhalil Morpho Sys 2, and they use a dataset of Arabic text as a test set. But the developers don't find the precision, recall, and F-measure. The precision is the percentage of words that are correctly analyzed among all the words that the system has analyzed. Whereas "recall" is the percentage of words that are correctly analyzed among all the words that should have been analyzed. F-measure is a combination of precision and recall; it is a measure of a system's accuracy and completeness [8]. The authors stated that in order to compute these metrics, a corpus is needed where each word is accompanied by a set of all its possible features. However, since such a corpus is not available as open source, it is not possible for them to compute these indicators [27].
Al-Kabi et al. [28] introduce a new algorithm for extracting Arabic roots by removing affixes and comparing the remaining word to a list of patterns. **Error! Reference source not found.** p resents the pseudocode for Al-Kabi et al. algorithm.

The algorithm was found to have an accuracy of 75.03% when tested on a dataset of 6081 word-root pairs. However, it also has limitations, such as missing roots, prefixes, suffixes, and patterns, as well as issues with affix ambiguity and only providing one solution for non-vocalized words.

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

Figure 5: The algorithm of Al-Kabi et al.

Thalji et al. [23] propose a new algorithm for the extraction of the root word from an Arabic word. The proposed algorithm is based on the morphological rules of the Arabic language, which are used to extract the root word from an inflected word. The flow chart of the proposed algorithm is shown in Figure 6.

The algorithm uses a set of morphological rules to break down the inflected word into its root form. The algorithm is based on a rule-based approach, which uses a set of morphological rules to extract the root form of the inflected word. The algorithm was tested on a large corpus of Arabic words and showed a high accuracy rate in extracting the root word. The newly proposed algorithm was compared to the Khoja algorithm [29], which is a reputable Arabic root extraction algorithm known for its high accuracy. The testing was done using the Thalji corpus, which is specifically designed for evaluating Arabic root extraction algorithms and contains

720,000 word roots. The results of the experiment showed that the Khoja algorithm had an accuracy of 63%, while the proposed algorithm had an accuracy of 94%.

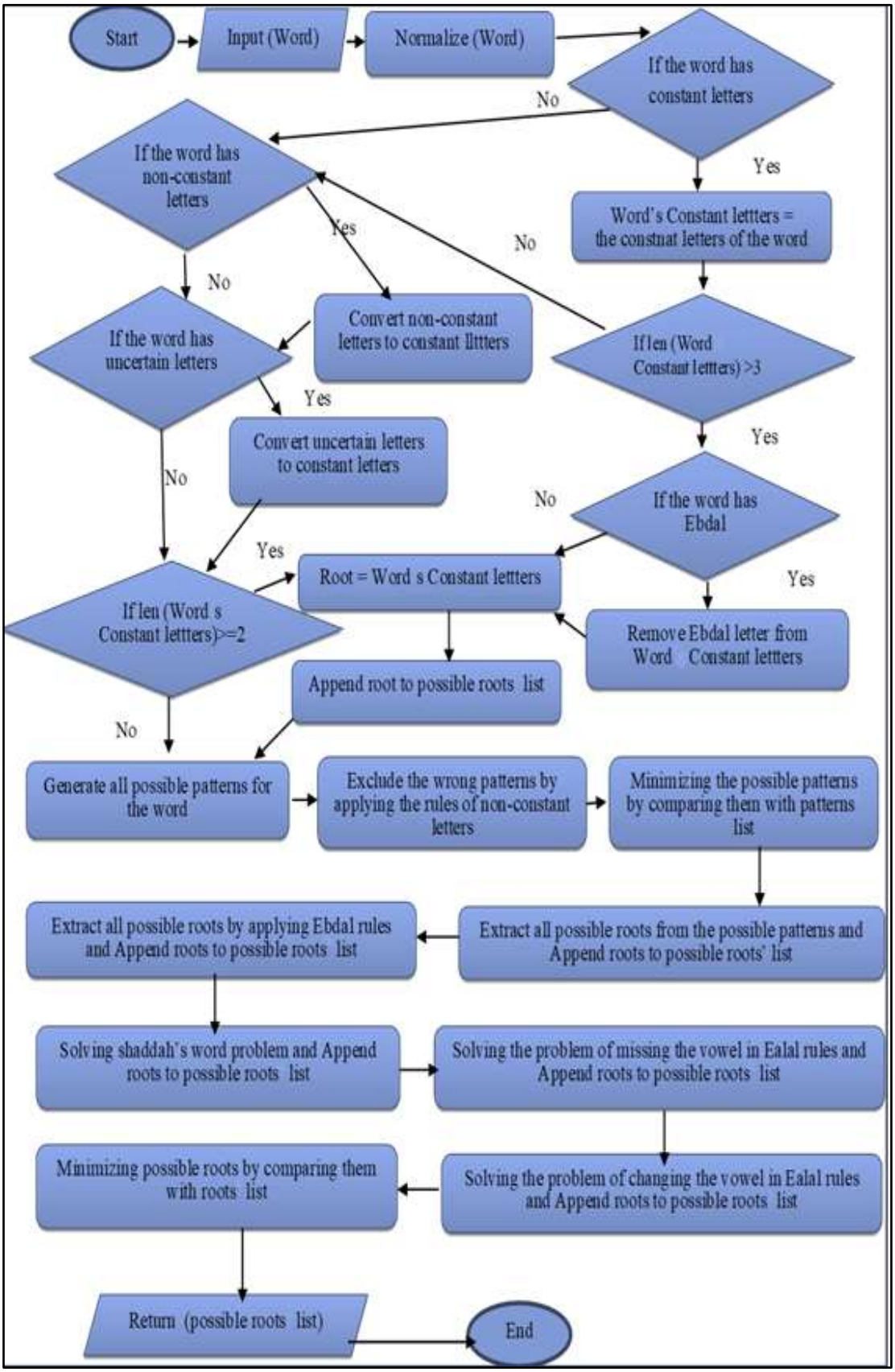

Figure 6: The flow chart of Thalji et al.'s algorithm

Khafajeh et al. [30] developed a hybrid algorithm for extracting the roots of Arabic words. Figure 7 shows their algorithm. They combine an optimization function with a set of non-morphological rules to improve the performance of the n-gram algorithm. The bigram approach was used to remove prefixes and suffixes.

Inputs:
A word in Arabic (word)
A list of potential roots (root_list)
Outputs:
The root of the input word (root)
Steps:

1. Divide the Arabic word into bilateral parts.
2. Calculate the number of unique bigrams in the word and assign it to variable A.
3. For each potential root in the list of roots:
   - Calculate the number of unique bigrams in the root and assign it to variable B.
   - Calculate the parts uniquely shared between the word and the root and assign it to variable C.
   - Calculate the degree of similarity between the word and the root and assign it to variable S: S=2C/(A+B)
4. Select the root with the highest value of S as the extracted root of the Arabic word
5. Return the output root.

Figure 7: Khafajeh et al. algorithm to extract Arabic root

The algorithm was tested using a dataset of over 6,000 unique words from 141 different roots. The results showed that the technique was able to accurately extract 99% of tripartite strong roots and 86% of tripartite vowel roots. However, the proposed algorithm has some limitations. For example, the rule stating that all extracted root letters should exist in the word and the rule stating that the extracted root letters should not exceed the number of letters in the word are not always true because vowel letters may be omitted during word derivation.
Alnaied et al. [31] propose a new algorithm for generating Arabic word stems, called Arabic Morphology Information Retrieval (AMIR). Figure 8 shows the model of AMIR for generating the roots [31].

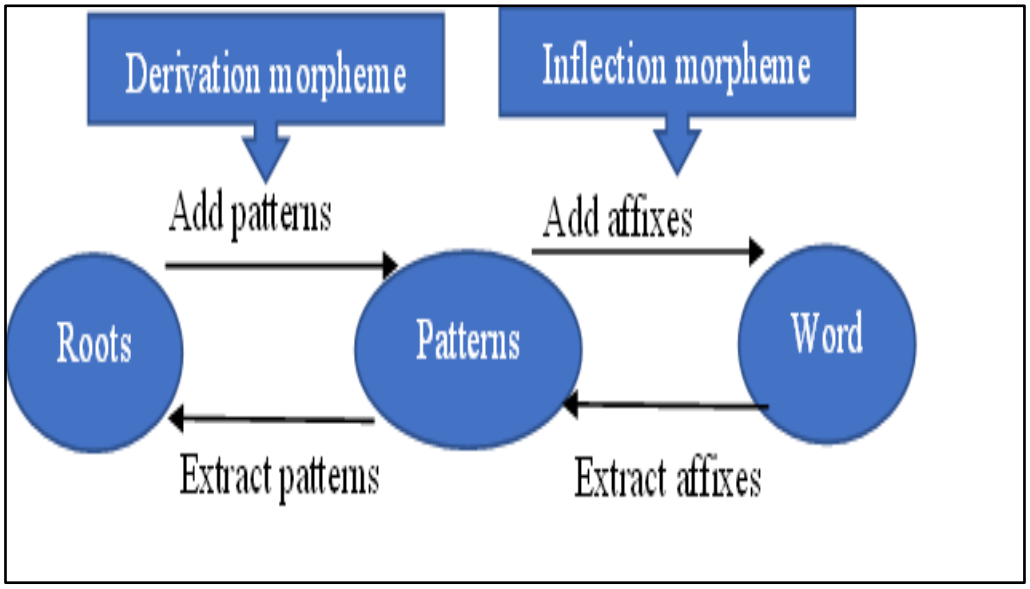

Figure 8: The AMIR Model for generate/extract root

This algorithm utilizes a set of rules that take into account the relationship between Arabic letters to identify the root or stem of words used as indexing terms in Arabic text search systems. To show the usefulness of the proposed algorithm, the authors highlight the benefits of the rules for various Arabic information retrieval systems. The performance of AMIR was evaluated by comparing it to LUCENE, FARASA, and a system that uses no stemmer in terms of mean average precision. The results showed that AMIR had a mean average precision of 0.34%, while LUCENE, FARASA, and the system without a stemmer had 0.27%, 0.28%, and 0.21%, respectively. The weaknesses of this stemmer include its inability to extract the root of many words, such as "أتكتبون," and its tendency to make errors in root extraction due to removing some letters that it believes to be extra while they are actually part of the root.

There are many rule-based algorithms available in the literature for Arabic root extraction. For instance, Thalji et al. [32] enhanced Khoja and Garside algorithm [29], while Thalji et al. [33] improved Sonbol et al. algorithm [26]. Kanaan et al. [34] presented an improved algorithm for extracting triliteral Arabic roots, while Ababneh et al. [35] built an effective rule-based light stemmer to enhance search effectiveness. Al-Kabi [36] improved Khoja stemmer [29]. Boudlal et al. [37] developed Alkhalil morpho sys1, a morphosyntactic analysis system for Arabic texts. Additionally, Yaseen et al. [38] presented an algorithm for extracting the roots of Arabic words without removing affixes, and Momani et al. [39] proposed an algorithm for extracting triliteral Arabic roots. Belal [40] also proposed a comprehensive processing approach for Arabic texts to extract their roots. The main principle of these algorithms is to either propose new rules for extracting the root of the Arabic language, rearrange previous algorithm rules, or increase the dictionary of data rules. But unfortunately, all of them did not use a unified corpus to test their algorithms. Each of them used aspecific corpus to test the algorithm on. And the results of the algorithms showed lower efficiency when tested on another corpus. Each approach has its own pros and cons, and the decision of which technique to implement is contingent on the specific purpose and the dataset in question.

3.4 Summary

In this section, the literature review studies for Arabic root extraction algorithms are summarized in

Table **3:** Summary of the Literature Review Studies for Arabic Root Extraction Algorithms. Efforts to improve the accuracy of Arabic root extraction often involve addressing the

challenges of affix ambiguity and incomplete lists of roots, patterns, prefixes, and suffixes. One way to enhance these algorithms is by refining the rules or expanding the available lists of roots, patterns, prefixes, and suffixes [40].

Table 3: Summary of the Literature Review Studies for Arabic Root Extraction Algorithms

Approach	Author and Year	Advantages/ Accuracy	Limitations
Dictionary-based	Alfedaghi and Al-Anzi (1989) [19].	The authors of these algorithms designed them to be straightforward and efficient and not necessitate a deep analysis of the words. They did not evaluate the accuracy of the algorithms. Additionally, later researchers did not evaluate these algorithms either due to the unavailability of the lists of patterns and roots that were used by the authors.	These algorithms are not effective when there is no match between the input word and the existing patterns or root lists. Additionally, if multiple patterns match the input word, the algorithms only return one root, when they should return as many roots as there are matched patterns. There is also a lack of information about the patterns used in these algorithms. Furthermore, these algorithms lack rules to guide the root extraction process if a word does not match any of the patterns or the extracted root does not match the list of existing roots.
Statistical-based	Alserhan, Al Shalabi and Kannan study (2003) [21].	One benefit of this algorithm is that it is quick and does not need extensive analysis of the words. The authors of this work did not measure the accuracy of their algorithm. However, it has been evaluated in other studies, such as Alshawakfa et al. study in 2010, and was found to have an accuracy of 14%.	The algorithm has a major weakness in that it treats non-constant letters with the same importance as constant letters, which can lead to inaccurate results. Additionally, these approaches may struggle with dealing with the complexity and irregularities of the Arabic language, such as the many different forms of words and the numerous variations in spelling and pronunciation.
Rules-based.	Sonbol et al. (2008) [26]	The study found that the algorithm had an accuracy rate of 96%. The new algorithm has several advantages over previous algorithms, one being that it categorizes Arabic letters into five groups, which reduces ambiguity in affix removal. Additionally, it incorporates new rules.	The algorithm proposed in the study has certain limitations, such as not accounting for situations where the root does not include a constant letter, does not begin with a constant letter, or only contains one constant letter. It also fails to cover a significant number of roots, prefixes, suffixes, and patterns and only offers one potential solution for non-vocalized words without considering other alternatives.
	M. Boudchiche et al. (2017) [27].	The selected dataset shows that the coverage accuracy of the algorithm is 99.31%. The developers of the algorithm have improved upon the previous version by increasing the size of the dictionary and implementing new rules.	The developers were unable to locate the values of precision, recall, and F-measure. When the dataset is altered, the outcome tends to be a decrease in performance. Enhancing the results requires the improvement of both the rules and the dictionary.
	N. Thalji et al. (2018) [23].	The results of the experiment showed that the proposed algorithm had an accuracy of 94%. The algorithm improved upon the previous ones by expanding the dictionary and introducing new rules.	The need to improve the rules and the dictionary is essential to enhancing the results.
	H. Atta, (2020) [24].	The accuracy is 74.11% on the selected dataset.	The efficiency of the algorithm decreases whenever the root contains two or more

		The algorithm tries to enhance the existing algorithms by reordering the rules and suggesting new rules.	nonconstant letters or whenever the word length increases. The rules are not always true for all words. In most cases, there are counterexamples.
	A. Alnaied et al. (2020) [31].	The study results showed that the AMIR algorithm had a mean average precision of 0.34%, while LUCENE, FARASA, and the system without a stemmer had 0.27%, 0.28%, and 0.21%, respectively. The AMIR algorithm uses a dictionary and a set of rules that take into account the relationship between Arabic letters to identify the root or stem of the words.	The weaknesses of this stemmer include its inability to extract the root of many words, such as "أتكتبون," and its tendency to make errors in root extraction due to removing some letters that it believes to be extra while they are actually part of the root.

Currently, the majority of methods for identifying the root of Arabic words are rule-based. These rules are used to determine the root of a derived word by utilizing a collection of patterns and affixes. The effectiveness of this approach can be influenced by the organization of the rules and the quantity of rules implemented. Additionally, this approach typically includes a preliminary step for identifying potential roots.

4. Conclusions

Based on the findings of this survey, several issues were identified. One of the biggest problems Arabic language processing experts are dealing with is the lack of a standardized corpus. The choice of particular corpora for testing purposes leads to variances in scope, word types, and usability for researchers. It is challenging to assess how well root extraction algorithms work because there isn't a reliable dataset or corpus. The dataset utilized can have a big impact on how well these algorithms perform. Furthermore, due to the complex nature of the Arabic language, the rules governing word formation are highly intricate. Moreover, the absence of a comprehensive list of patterns, affixes, and roots that can be used as a reliable list for developing effective root extraction algorithms for Arabic language processing made the research more difficult.

Based on the findings, several recommendations have been proposed. One of the most critical recommendations is to develop a standardized and reliable corpus that includes a diverse range of words and language patterns. That is the recommended solution to ensure fair testing of root extraction algorithms. By doing so, researchers can evaluate the performance of these algorithms more accurately and make more informed decisions about which algorithm to use for a particular task. Additionally, it is recommended to develop new rules to address abnormal cases and irregular words to improve the accuracy of root extraction algorithms. It is also recommended to create a comprehensive list of patterns, affixes, and roots that can be used as a reference for developing new root extraction algorithms because the effectiveness of the extraction process is strongly influenced by the quality of the list.

Based on the identified issues and recommendations, several areas of future work could be pursued to improve the Arabic root extraction process. One critical area of work is the development of a standardized and reliable corpus that includes a diverse range of words and language patterns. This would ensure fair testing of root extraction algorithms and enable researchers to evaluate their performance more accurately. Additionally, efforts could be made to develop new rules that address abnormal cases and irregular words, as well as to create a

comprehensive list of patterns, affixes, and roots that can be used as a reference for developing new root extraction algorithms.

References

[1] A. Nehar, D. Ziadi and H. Cherroun, "Rational kernels for Arabic root extraction and text classification," *Journal of King Saud University-Computer and Information Sciences,* vol. 28, no. 2, pp. 157-169, 2016.

[2] 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.

[3] A. Setiyadi, A. Anhar and H. Anwar, "Existence of Arabicization methods for naturalising contemporary technical vocabularies into the Arabic language," *Journal of Research and Innovation in Language,* vol. 4, no. 3, pp. 309-319, 2022.

[4] G. Lebbos, "Arabic information extraction methods a survey," *London Journal of Research of Engineering Research,* vol. 19, no. 2, pp. 11-28, 2019.

[5] A. Alothman and A. Alsalman, "Arabic morphological analysis techniques," *International Journal of Advanced Computer Science and Applications,* vol. 11, no. 2, pp. 214-222, 2020.

[6] E. Al-Shammari and J. Lin, "A novel Arabic lemmatization algorithm," in *Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data ACM,* Singapore, 2008, pp. 113-118.

[7] B. Azman, "Root identification tool for Arabic verbs," *IEEE Access,* vol. 7, pp. 45866-45871, 2019.

[8] Majdi Shaker Salem Sawalha, "Open-source resources and standards for Arabic word structure analysis : Fine grained morphological analysis of Arabic text corpora," *PhD thesis, University of Leeds,* 2011.

[9] 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.

[10] 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.

[11] I. Al-Sughaiyer and I. Al-Kharashi, "Arabic morphological analysis techniques: A comprehensive survey," *Journal of the American Society for Information Science and Technology,* vol. 55, no. 3, pp. 189-213, 2004.

[12] T. Kanana, S. Ayouba, E. Saifb, G. Kanaanb, P. Chandrasekara and A. Foxa, "Extracting named entities using named entity recognizer and generating topics using latent dirichlet algorithm for Arabic news articles," in *Proceedings of the International Computer Sciences and Informatics Conference (ICSIC2016),* Amman, 2016, pp. 51-60.

[13] A. Yousif, V. Samawi, I. Elkabani and R. Zantout, "The effect of combining different semantic relations on Arabic text classification," *World of Computer Science and Information Technology Journal,* vol. 5, no. 1, pp. 112-118, 2015.

[14] 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.

[15] M. El Sayed, G. Lebbos and H. Hajjar, "Arabic information extraction methods: A survey," *London Journal of Engineering Research,* vol. 19, no. 2, pp. 11-28, 2019.

[16] I. Al-Sughaiyer and I. Al-Kharashi, "Arabic morphological analysis techniques: A comprehensive survey," *Journal of the American Society for Information Science and Technology,* vol. 55, no. 3, pp. 189-213, 2004.

[17] M. Mustafa, A. Salah Eldeen, S. Bani-Ahmad and A. Elfaki, "A comparative survey on Arabic stemming: Approaches and challenges," *Intelligent Information Management,* vol. 9, no. 2, pp. 39-67, 2017.

[18] A. Ababneh, L. Joan and Q. Xu, "Arabic information retrieval: A relevancy assessment survey," in *25th International Conference On Information Systems Development*, Poland, 2016, pp. 345-357.

[19] S. Al-Fedaghi and F. Al-Anzi, "A new algorithm to generate Arabic root-pattern forms," in *Proceedings of the 11th National Computer Conference and Exhibition*, Dhahran, Saudi Arabia, 1989, pp. 04-07.

[20] H. Aljaloud, M. Dahab and M. Kamal, "Stemmer impact on Quranic mobile information retrieval performance," *International Journal of Advanced Computer Science and Applications,* vol. 7, no. 12, pp. 135-139, 2016.

[21] H. Al-Serhan, R. Al-Shalabi and G. Kannan, "New approach for extracting Arabic roots," in *Proceedings of the 2003 Arab Conference on Information Technology*, Egypt, 2003, pp. 42-59.

[22] E. Al-shawakfa, A. Al-Badarneh, S. Shatnawi, K. Al-Rabab'ah and B. Bani-Ismail, "A comparison study of some Arabic root finding," *Journal of the American Society for Information Science and Technology,* vol. 61, no. 5, pp. 1015-1024, 2010.

[23] N. Thalji, N. Hanin, W. Bani-Hani, S. Al-Hakeem and Z. Thalji, "A novel rule-based root extraction algorithm for Arabic language," *International Journal of Advanced Computer Science and Applications,* vol. 9, no. 10, pp. 120-128, 2018.

[24] H. Atta and A. Al-Hmouz, "Enhanced Arabic root-based lemmatizer," *M.S. thesis, Department of Computer Science, Middle East University*, Amman, 2020.

[25] K. Taghva, R. Elkhoury and J. Coombs, "Arabic stemming without a root dictionary," in *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, Las Vegas, NV, USA, 2005, pp. 152-157.

[26] 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.

[27] M. Boudchiche, A. Mazroui, M. Bebah, A. Lakhouaja and A. Boudlal, "AlKhalil morpho sys 2: A robust Arabic morpho-syntactic analyzer," *Journal of King Saud University-Computer and Information Sciences,* vol. 29, no. 2, pp. 141-146, 2017.

[28] M. Al-Kabi, S. Kazakzeh, B. Abu-Ata, S. Al-Rababah and I. Alsmadi, "A novel root based Arabic stemmer," *Journal of King Saud University-Computer and Information Sciences,* vol. 27, no. 2, pp. 94-103, 2015.

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

[30] H. Khafajeh, N. Yousef and M. Abdeldeen, "Arabic root extraction using a hybrid technique," *International Journal of Advanced Computer Research,* vol. 8, no. 35, pp. 90-96, 2018.

[31] A. Alnaied, M. Elbendak and A. Bulbul, "An intelligent use of stemmer and morphology analysis for Arabic information retrieval," *Egyptian Informatics Journal,* vol. 21, no. 4, pp. 209-217, 2020.

[32] 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," *International Journal of Computer and Information Technology*, vol. 7, no. 6, pp. 235-242, 2018.

[33] N. Thalji, N. Hanin, Z. Thalji and S. Al-Hakeem, "Enhancing the accuracy of Sonbol's Arabic root extraction algorithm," *Jordanian Journal of Computers and Information Technology,* vol. 4, no. 3, pp. 159-174, 2018.

[34] 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.

[35] M. Ababneh, R. Al-Shalabi, G. Kanaan and A. Al-Nobani, "Building an effective rule-based light stemmer for Arabic language to improve search effectiveness," *Int. Arab J. Inform. Technol.,* vol. 9, no. 4, pp. 368-372, 2012.

[36] 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.

[37] 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.

[38] Q. Yaseen and I. Hmeidi, "Extracting the roots of Arabic words without removing affixes," *Journal of Information Science,* vol. 40, no. 3, pp. 376-385, 2014.

[39] 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.

[40] N. Thalji, A. Hanin, Y. Yacob and S. Al-Hakeem, "Corpus for test, compare and enhance Arabic root extraction algorithms," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 5, pp. 229-236, 2017.

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