IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 12, 2020, accepted May 25, 2020, date of publication June 1, 2020, date of current version June 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999259

Arabic Text Processing Model: Verbs Roots and Conjugation Automation

MOHAMED TAHAR BEN OTHMAN[1,2], (Senior Member, IEEE), MOHAMMED ABDULLAH AL-HAGERY[1,2], AND YAHYA MUHAMMAD EL HASHEMI[3]

[1]Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2]BIND Research Group, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[3]Department of Arabic Language and Literature, College of Arabic Language and Social Studies, Qassim University, Buraydah 51452, Saudi Arabia

Corresponding author: Mohamed Tahar Ben Othman (maathaman@qu.edu.sa)

This work was supported by the Deanship of Scientific Research, Qassim University, under Project 5347-coc-2018-2-14-S.

ABSTRACT The Natural Language Processing (NLP) is a process to automate the text or speech of Natural Languages. This automation is mainly conducted for Western languages. The Arabic Language got less focus in this area. This paper presents a Model to recognize an Arabic sentence. A new morphological model based on regular expressions is developed to recognize the Arabic verbs. A hash table containing all Arabic three-letters' root of verbs is implemented. The total number of Arabic verbs that are derived from three-letters' root size is 23090. The number of roots is 6104. A set of rules forming the Arabic grammar is used to derive and analyze the syntax of Arabic sentences. About 87% of the verbs represented in our regular expressions' engine are detected. Moreover, the sentences are also recognized. In several Surat of the Quran, only 9% of the detected verbs are false-positive (a non-verb declared as a verb), and 4% are considered false-negative (a verb is considered as a noun). This rate is mainly because we are not using vowels even that the Quran (our case study) is using them. The reason behind our decision is to be able to handle all Arabic texts, which mostly are not using vowels.

INDEX TERMS Arabic text processing, regular expression, root extraction, verbs root classification, data mining.

I. INTRODUCTION

NLP is a well-known domain that since several years keeping an area of research orientation. The complexity of this domain made progress going slowly as compared to the research in most of other domains. Moreover, between the different natural languages, the Arabic Language got less attention, and it is known that it has the most complex syntax, structure, and verb conjugation. The research in this direction becomes most needed to ease the discovery of extensive knowledge of words.

The Arabic text syntax and semantic are challenging because of the morphological features of the Arabic sentences and words [1]. The Arabic Language is rich and counted as the most complicated Language amongst the other languages. The sentence complexity is embedded in both syntax and its semantics. It also contains a massive number of vocabularies, including words synonyms, antonyms, and word roots as nouns or verbs.

This type of complexity is affecting on the understandability, analysis, automation of this Language. The word roots join with different vowel forms to constitute simple verbs and nouns. It can be generated using complicated methods in the grammatical derivation [2]. The Arabic Language comprises a diversity of forms of verbs. The conjugation of each verb depends on several factors.

There is a limited number of research works that concentrate on Arabic language automation and correction. It deals with word roots generation and syntax analysis, for instance, Abu-Errub *et al.* presented a technique to generate the trilateral Arabic word roots, based on a list of morphological weights and the roots of three consonants. They deal with the removal of the prefixes and the suffixes. The results show the usefulness of the proposed technique with a performance rate equal to 94% [3]. The main drawback of this research is that it does not mention the type and size of the input test.

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

An Arabic spelling correction system had been proposed to discover and remove or correct the spelling errors. The system is intended for the Arabic search in electronic dictionaries. It was evaluated on single-error queries, and the system performance was 28% better than the baseline. It is the first system for a spoken colloquial of the Arabic dialect [4]. As well, Azmi *et al.* designed an algorithm for Arabic spelling error detection and correction, where this method relies on dictionary search. The proposed algorithm introduced as a spell checker can detect the spelling errors and then suggests the real corrections for incorrect Arabic spelling entered by the user [5].

Also, a research work carried out by [6] developed a hybrid system based on two techniques: the correction model and the confusion matrix. The system automatically detects and corrects spelling errors written in Arabic. It contains a robust error confusion matrix. After the system evaluation, it found that the result outperforms other systems results.

Furthermore, Hassan [7] studied the writing problems of a sample of learners in regards to writing Arabic and English statements. The study identified three common errors in the statement structure, along with suggesting some practical solutions to minimize these errors.

On the other hand, a recent morphological analysis system has been developed [8], which analyzes the Arabic word surface patterns. The mechanism used to classify verbs' roots was based on a set of morphological rules. Later on, they construct a conjugated surface pattern database. The proposed system had been tested and evaluated by 4,000 verbs, and it only presented an error rate equal to 4%. This system performs roots' derivatives based on assumptions, and not the real Arabic used verb roots.

This research is focusing mainly on the automation process of Arabic roots generation. It aims at the development of a tool to process the Arabic text in a way to discover the syntax used in the input text and correct the errors. The tool can be used to automate the verb conjugation going from root to its conjugation and going back from conjugation to the root. It will also be used to detect syntax, provide the errors then help to use the correct syntax better. Later, this tool will be incorporated into a smart editor to help writing Arabic text.

The method includes a collection of Arabic verbs that are related to knowledge. It builds a tool using a database of knowledge. Moreover, it provides and implements a set of algorithms to cover verbs conjugation, root detection, algorithms evaluation, collections of the Arabic grammar rules, syntax, build the grammars, add the semantic actions of the syntax generation, and generate the syntax of an input sentence.

The rest of the paper is structured as follows; Section II introduces the literature review. Section III is describing the proposed model by the Arabic Language Grammar, the Verbs Database Construction (Verbs' Generation), and the Verb Recognition System. Furthermore, section IV highlights the experimentation and results, while section V demonstrates the conclusions and future works.

II. LITERATURE REVIEW

Since the last decade, many researchers were focusing on Arabic Language processing. Most researches in NLP were conducted toward the English language. The transition of the works done on the English to Arabic is not as trivial as it can be seen. The Arabic Language is the richest natural Language in syntax and semantics. Among these researches, we present the following related contributions.

Althobaiti *et al.* [9] built a Java-based library that consists of various tools for Arabic text processing. They presented a complete library to handle Arabic texts.

An Arabic morphology dataset was presented by [10]. They focused on the uniqueness of the root pattern phenomenon and studied the associative relationships between words meaning at a higher level and their possible occurrences. This approach can be viewed as an instantiated global root–pattern related to the morpho-phonetic items.

Some other works are given in [11], which studied Arabic text automation in a semantic approach. The work achieved by [12], built a classification method for the Arabic Language.

Moreover, there are some research works conducted on the Arabic root's extraction, for instance, a multi-objective method with a statistical method to separate the suggested Arabic roots. The results presented that the developed method improved the performance of extracting the Arabic roots [13]. Besides, Yousef *et al.* proposed an approach to improve the Arabic root extraction method for all words, according to the bi-gram technique. The performance of the proposed approach reached to 80% [14]. This approach succeeded with the vocalic roots.

Likewise, a proposed model was developed to identify the root of verbs by a software tool called *RootIT*, to overcome the problem of verb root generation without disambiguation [15]. Besides, an approach for Arabic root generation presented by [16], is a novel technique for Arabic NLP to generate the roots of the Arabic word. Also, Farwaneh, in [17], focused on the Levantine Arabic variety and created an account of a set of complex facts related to the inflection of sound verbs and non-sound verbs. The account distinguishes four levels of correspondence, (input-output), (output- output). Also, concentrated on the paradigmatic differences found in the inflection of sound verbs, his method concentrated on the stems of more than two consonants and on the non-sound verbs, whose stems comprise two consonantal realizations.

The Arabic sentence is syntactically ambiguous and complicated because of the deferent meaning of the word in the context and frequent usage of conjunctions, grammatical relationships, and other forms [18]. The Arabic Language is characterized by its complex morphology based on root-pattern schemes. The process of Arabic words' roots extraction is challenging, and it is an essential topic in NLP applications, for example, Information Retrieval, text analysis, machine translation, and speech tagging [3].

Besides, another study [19] discussed machine translation models and their current problems, and its lower accuracy, especially in the translation from the Arabic to Chinese as two complex languages. Besides, the researchers propose the best combination of factors that can help in the translation task within a proposed approach. On the other hand, Thalji *et al.* [20] worked on the Arabic Language, created a rule-based algorithm for roots extraction to eliminate the weaknesses of the previous methods. They used the corpus of ''Thalji'' for the testing process and comparison with other works.

Elazhary *et. al*. concentrated on automation tutoring to analyze the Arabic word root extraction [21]. They suggested an automated tutor that can be used in learning and teaching to help students to extract the appropriate roots of any Arabic word.

Yaseen and Hmeidi introduced an algorithm called Stemming Algorithm for roots extraction based on a set of rules and an Arabic roots file that contains the word roots. The algorithm keeps the affixes during the extraction steps, and the proposed algorithm is competitive with an accuracy equal to 84% [22]. As well, Mohammed [23] suggested a combined stemmer for Arabic words' root, which achieved an exploration ratio of roots equal to 99.08. Moreover, Zeroual *et al.* developed an efficient stemmer algorithm for Arabic text, which deals with the morphological characteristics of Arabic. They executed some experiments and evaluated the performance of the developed algorithm based on two styles of Arabic; Classical Arabic and Modern Standard Arabic. The outputs of the stemmer organized in three classes include the stem, a unique root, and a combined class from the root and stem [24].

In the Arabic Language, the word root has various patterns that can be matched by different algorithms. The Word-pattern matching algorithms are employed in extracting Arabic words' roots [25], [26]. According to this idea, a root is separated after canceling the affixes attached to the word. The root generation is achieved by comparing the corresponding pattern with the positions of the letters in the word.

Generally, two types of Arabic roots that can be classified according to the vowels [27]. The first is the vowel root that contains at least one vowel. The second is the base root, which does not contain any vowel. Besides, Blanchete *et al.* formalized a model for Arabic verbs based on pattern approach and the verb root. This model uses a linguistic classification method that identifies a group of morphological features. Their research work depends on two main parts: first, a dictionary, which contains patterns, lemmas, and roots and second, the generation of all potential verbs. The primary process concentrated on the roots classification and matching with patterns to give lemma. The output of this work introduced a dictionary that includes a big set of inflectional and derivational verbs styles [28].

Many techniques have been employed to find out the roots of Arabic words. Nevertheless, none of these methods have been approved as a slandered method because of the morphological richness of the Language [26]. Several methods of extracting the Arabic word roots have been presented [29], [30]. Some researchers rely on morphological rules to find out the trilateral word roots.

Many researchers use morphological rules to generate the roots of the Arabic word [3], the processes of roots extraction are not easy and very complex as a result of the multiple forms of the morphological formulas in Arabic words. Therefore, a technique was developed, which depends on some non-morphological rules combined with the statistical approach. This method proposed to reduce the word roots complexity process.

There is little research done on Arabic text processing, compared to other languages, especially colloquial text [12]. However, recently there is an increasing interest in this area of research. One of these researches modifies the classification of Arabic dialects by using metadata [31]. Although the difficulty of the Arabic text structures and with the absence of research in this field, there are limited contributions that have tried to analyze Arabic expressions by using different algorithms. However, these approaches have some restrictions.

For instance, El Kourdi *et al.* [32] classified a dataset of 1500 web documents written in Arabic, which were collected from the news channels and categorized into five classes by the Naïve Bayes algorithm. They got a low average accuracy equal to 69%. Most of the research contributions for Arabic text were achieved by lexical based categorization and classification using the machine learning algorithms [33]–[36], where, they have resources about standard Arabic. Also, Kamps *et al.* [37] developed a simple distance measure on WordNet to determine the semantic orientation of adjectives. It classifies the text by using a simple technique based on lexical relations. Also, [38] used WordNet to classify the English text based on the quantitative analysis of the glosses of terms that carries opinionated content that has a positive or a negative meaning. Moreover, in [39], the main features were extracted from Twitter data by using a simple lexicon-based approach.

The sentiment classification carried out by [40] used a supervised learning algorithm to identify the semantic orientation of the conjoined adjectives from a large corpus of conjunction constraints. While Turney [41] used an algorithm of machine learning to classify contents of reviews as recommended or not, which is predicted using the semantic orientation average of the expressions in a data set of reviews that encompass adjectives. The orientation of the semantic of a phrase is calculated based on the mutual information of the words and the given phrase ''excellent'' and ''poor'' using statistical techniques. Our proposed technique performs better as compared to most of the systems found in the literature. We reached more than 87% of the accuracy of recognized verbs.

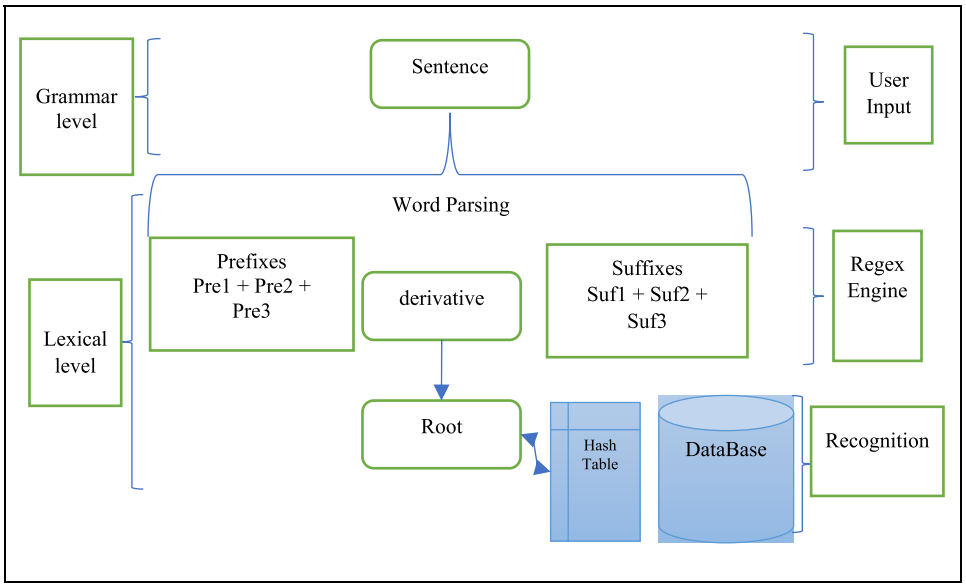


FIGURE 1. Proposed model architecture.

III. PROPOSED MODEL

Our proposed Arabic Text Processing model uses the grammar rules of the Arabic Language. The Model is given in Fig. 1. There are different levels in this model:

Level 1: The Arabic text segment unit, which is the sentence. Parsing the sentence will narrow the recognition of the type of lexemes. Most of the words of type *NOUN* or *VERBS* cannot be recognized as such, mainly when they are vowel-less, except in their context in a sentence. Most of the researchers are looking only at the morphology level, and not considering the context (syntax) level.

Level 2: The lexical level, where the different parsed lexemes are matched to the forms of Arabic words using regular expression engine.

Level 3: Lexeme type recognition: one of our contributions at this level is the construction of a database containing all three-letters-root Arabic *Verbs*, all well-known special *Nouns,* and *Hurufs*.

In the rest of this section, we will present the main grammar rules representing the construction of a sentence in Arabic Language describing the main complexity and the different sentence forms of such a language.

A. ARABIC GRAMMAR

Fig. 2 gives a global idea about the Arabic Language Formal Grammar, which has been inspired from [42] and slightly enhanced, which we call here Arabic Context-Free Grammar (ACFG) that will be more tuned in the future work to represent as much as possible the sentences in an Arabic text. The lexemes that are represented by the tokens include; *Verb*, *Harf*, and *Noun.* These are the only lexemes kind in the Arabic Language that regroup all *Verbs*, *Hurufs,* and *Nouns*. The Verbs and their different forms are all known as well as *Hurufs*. Some Nouns are regrouped under some abstractions because they have some special effect in a sentence, and their forms may not change or may change in a particular way. The rest of the *Nouns* cannot be limited. For this reason, any lexeme that is not a *Verb* or *Harf* is a *Noun*.

This grammar is presented in its first form before it passes through different processing steps:

1) add semantic actions that are used to describe the syntax analysis
2) left factoring for the rules under the same name and starting with the same prefix

We denote the Non-Terminal Symbols the set of all rules names and the Terminal Symbols the set of all the tokens provided by the lexical analyzer for all lexemes (strings) in the user input including *Verb*, *Harf*, *Noun*, *VerbalTransformedParticle*, *AdjectiveParticle*, *Pronoun*, *Adverb* (Time, Place), and *Preposition*.

As ambiguity – the same sentence can have different meanings – in natural languages and particularly in the Arabic Language is frequently used, our model presents as much syntax analysis as it can find for the same sentence.

B. VERBS WITH THREE LETTERS' ROOT

The Verbs in Arabic Languages are classified in different ways based on the number of characters in their roots. The most used ones are three and four letters' roots. This study is focusing on three letters' roots. Table 1 is showing some verbs with their past derivatives. There are 15 verbs' past derivatives forms for three letters' root verbs in the Arabic Language. These derivatives forms are represented in Fig. 3. All Arabic verbs are described in detail in [43].

There are some special forms of the derivatives, which result from combining two letters or changing a letter by

TABLE 1. Verb' derivatives example.

Verb Root	فَعَلَ	فَعْلَ	فَعِلَ	فَعَّلَ	فَاعَلَ	أَفْعَلَ	تَفَعَّلَ	تَفَاعَلَ	اِنْفَعَلَ	اِفْتَعَلَ	اِفْعَلَّ	اِسْتَفْعَلَ	اِفْعَوْعَلَ	اِفْعَالَّ	اِفْعَوَّلَ
ء ب ء	أبأ	-	-	أبَّبَ	-	-	تأبَّبَ	-	-	اِئْتَبَّ	-	اِسْتَأبَّ	-	-	-
ء ب ب	أبَّ	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ء ت ب	-	-	-	أتَّبَ	أتَبَ		تأتَّبَ	-	-	-	-	-	-	-	-
ج د ل	-	-	جدِلَ	جَدَّلَ	جادَلَ	أجْدَلَ	تَجدَّلَ	تَجادلَ	اِجْدَلَ	اِجْتَدَلَ	-	-	-	-	-
ج ف ظ	-	-	-	-	-	-	-	-	-	-	-	-	-	اِجْفاظَّ	-
ج ل ذ	جلذ	-	-	-	-	-	-	-	-	-	-	-	-	-	اِجْلَوَّذَ
ج ل و	جَلا	-	-	جَلَّى	جالَّى	أجْلَى	تجلَّى	تجالَى	اِنْجَلَى	اِجْتَلَى	-	-	اِجْلَوْلَى	-	-

Sentence → NominalSentence | VerbalSentence
NominalSentence → [PreSubject] SubjectPhrase PredicatePhrase
PreSubject → OriginalParticle | TransformedParticle
OriginalParticle → NegativeParticle | InterrogativeParticle | Preposition
Transformed Particle → Verbal Transformed Particle | Adjective Particle
SubjectPhrase → Subject [ExpansionPhrase]
Subject→ Noun | Pronoun
PredicatePhrase → Predicate [ExpansionPhrase]
Predicate → Noun | DerivedNoun | Sentence | QuasiSentence
DerivedNoun → SubjectAdjective | ObjectAdjective | SuperlativeAdjective | VerbalAdjective
QuasiSentence → Preposition Noun | Adverb AnnexedNoun

VerbalSentence → VerbPhrase | VerbPhrase VerbalSubjectPhrase | VerbPhrase VerbalSubjectPhrase ComplementPhrase
VerbPhrase → TransitiveVerb | IntransitiveVerb | TransitiveVerb Object | TransitiveVerb Object AdjunctsPhrase | IntransitiveVerb AdjunctsPhrase
AdjunctsPhrase → Adjunct | [AdjunctsPhrase
Adjunct → QuasiSentence
VerbalSubjectPhrase → Subject [ExpansionPhrase] | LatentPersonal
ExpansionPhrase → Expansion | [ExpansionPhrase]
Expansion → Adjective | Coordinate | Annexed | CircumstantialObject | RelativeSentence
ComplementPhrase → ComplementElement [ComplementPhrase]
ComplementElement → Object | QuasiSentence | Expansion
Object → Noun | InterpretedInfinitive | NominalSentence
InterpretedInfinitive → InfinitiveParticle VerbalSentence
QuasiSentence → Preposition Noun | Adverb Annexed
Annexed → Noun
Adjective → DerivedNoun
Coordinate → Coordinator Noun | Coordinator Sentence
RelativeSentence → RelativeNoun RelativeSentenceBond
RelativeSentenceBond → VerbalSentence | NominalSentence

The *PreSubject* is a set of type *Hurufs*. It includes ['هل', 'أ', 'بل', 'و', 'ف', 'ثم', 'لا', 'لكن', 'حتى'] this list can be before the subject or before the verb. The following list is the *PreSubject* that may precede a Subject and not a Verb [‘من’, ‘إلى’, ‘عن’, ‘على’, ‘في’, ‘ب’, ‘ك’]

FIGURE 2. Arabic grammar (ACFG) [42].

TABLE 2. Verb' special forms 1.

Verb Root	Verbs in the past	Special forms
د ق ق	دقّ	‘فَعَّ’
ج ل و	جَلا	‘فَعَا’

TABLE 3. Verb' special forms 2.

Verb Root	Verbs in the past	Special forms
د ع و	اِدَّعَى	'اِفْدَعَلَ'
ذ ر ع	اِذَّرَعَ	'اِفْدَعَلَ'
ز ج ر	اِزْدَجَرَ	'اِفْطَعَلَ'

another. The first form is due to getting the same letter in the second, and the third positions of the root. The second by having in the root one or two letters equal to ‘ا’, ‘ى’, or ‘و’ which are called weak letters (حروف العلة), or a vowel with a long sound. An example of these two forms is shown in Table 2. The third special form is when the letter’ ‘تَ’ is replaced by ‘دَ’, ‘طَ’, or ‘ذَ’ in the derivative ‘افتعل’ since it is easier to pronounce. Table 3 presents an example of this latter case.

In this paper, we are covering the first special form but not totally the second and the third as they have several forms depending on the context, which will be addressed in the future. These derivatives will change in the present tense to those given in Fig. 4.

In all of these derivatives (in the past and present), the ‘ف’, ‘ع’, ‘ل’ represent the first, second, and third letters of the root in the verb respectively. In an Arabic text, and depending on the pronoun and the tense, the verb is concatenated with prefixes given in Fig. 5, and suffixes are presented in Fig. 6.

As it is shown in Table 1, we can find no rule to extract the derivatives from a root. To recognize a verb, the root has to be extracted from the user lexeme, which has to be parsed, and the form of the used verb should also be a good derivative of the root. For this reason, in our model, a database is built for all three letters' root and their derivatives. A hash-table was used to find a root with its derivatives. As the 28 Arabic letters are used to construct roots, the chosen hash function is $28*\text{rank(a)}+28^2*\text{rank(b)}+28^3*\text{rank(c)}$, which gives a complexity of O(1). Where a, b, and c are the first, second, and third letters in the root, respectively. The rank is the rank of the letter in the list of the Alphabet (from 1 to 28). The algorithm building this hash-table is presented in Fig. 7. Table 4 presents the number of generated Arabic verbs per derivative per root's first letter.

The total number of 3-letters-root verbs in the Arabic Language is 23090. The number of roots is 6104.

Fig. 8 presents the number of verbs per root's first letter. The maximum number of verbs are those starting with the letter ‘ن’ whereas, the roots starting with ‘ظ’ and ‘ي’ have the least number of verbs.

```
['فَعَلَ', 'فَعُلَ', 'فَعِلَ', 'فَعَّلَ', 'فَاعَلَ', 'أَفْعَلَ', 'تَفَعَّلَ', 'تَفَاعَلَ', 'اِنْفَعَلَ', 'اِفْتَعَلَ', 'اِفْعَلَّ', 'اِسْتَفْعَلَ', 'اِفْعَوْعَلَ', 'اِفْعَالَّ', 'اِفْعَوَّلَ']
```

FIGURE 3. Verbs' past derivatives' forms.

```
['فَعَلُ', 'فَعَلُّ', 'فَعَوَّلُ', 'فَعَالُّ', 'فَعَوْعِلُ', 'سَتَفْعِلُ', 'فَعَلُّ', 'تَفَعَّلُ', 'تَفَاعَلُ', 'فَعِلُ', 'فَاعِلُ', 'فَعِلُ', 'فَعُلُ', 'فَعِلُ', 'فَعَلُ']
['فَعَوَّلُ', 'فَعَالُ', 'فَعْوَعَلُ', 'سَتَفْعَلُ', 'فَعَلُّ', 'فَتَعَلُ', 'نَفْعَلُ', 'تَفَاعَلُ', 'تَفَعَّلُ', 'فَاعِلَ', 'فَعَّلُ', 'فَعِلُ', 'فَعُلُ', 'فَعُلُ', 'فَعَلُ']
```

FIGURE 4. Verbs' present derivatives' forms.

TABLE 4. Count of generated Verbs.

	Derivatives														
First letter	فَعَلَ	فَعُلَ	فَعِلَ	فَعَّلَ	فَاعَلَ	أَفْعَلَ	تَفَعَّلَ	تَفَاعَلَ	اِنْفَعَلَ	اِفْتَعَلَ	اِفْعَلَّ	اِسْتَفْعَلَ	اِفْعَوْعَلَ	اِفْعَالَّ	اِفْعَوَّلَ
أ	176	18	101	114	31	77	113	17	2	52	1	47	0	0	0
ب	236	39	103	153	66	140	145	51	59	68	4	52	2	6	0
ت	73	2	41	45	19	61	20	10	3	17	0	12	0	0	0
ث	86	9	49	44	15	46	28	10	22	11	2	11	1	5	0
ج	204	25	87	129	61	117	99	44	47	85	4	37	2	2	5
ح	238	32	128	158	77	177	152	66	60	117	5	71	13	5	1
خ	189	25	109	136	57	130	116	53	61	95	6	41	12	5	1
د	205	12	88	117	54	112	75	39	63	31	4	21	4	9	0
ذ	88	9	35	37	11	41	25	12	19	6	0	17	1	1	0
ر	269	40	104	184	78	205	163	68	1	140	14	65	1	9	0
ز	156	8	54	89	28	87	80	20	31	49	9	11	0	11	0
س	187	29	97	163	68	145	117	48	73	64	4	27	3	4	0
ش	198	33	104	160	59	123	102	57	49	44	7	27	4	6	0
ص	126	16	52	85	41	89	70	42	32	36	8	19	4	10	0
ض	110	11	36	44	29	57	38	26	25	23	0	18	1	4	0
ط	123	10	50	78	18	69	56	20	26	28	3	20	2	0	0
ظ	19	1	9	10	8	15	10	6	1	6	0	5	1	0	0
ع	244	33	139	182	87	161	149	67	47	122	4	67	10	5	3
غ	117	8	79	84	37	89	77	35	34	58	3	32	6	4	0
ف	205	27	84	145	54	137	130	45	77	67	0	48	1	0	0
ق	215	29	107	156	57	148	146	48	64	110	4	47	3	5	1
ك	197	16	66	120	52	104	97	41	42	67	2	28	4	4	1
ل	204	11	102	111	62	116	114	42	1	97	2	29	0	1	0
م	253	31	84	120	72	134	141	56	42	102	5	34	0	6	0
ن	340	35	114	186	104	222	165	108	3	186	3	110	0	2	0
ه	175	5	62	103	49	79	101	41	54	60	1	33	2	1	0
و	239	37	94	152	77	180	150	56	6	56	0	90	0	2	0
ي	13	4	13	17	6	17	8	3	0	3	0	9	0	1	0

Fig. 9 shows the number of Verbs per root's derivatives. As all three-letters' verbs have a derivative 'فعل', it is the most used derivative, whereas, the derivatives containing the least number of verbs are 'اِفْعَلَّ', 'اِفْعَوْعَلَ' and 'اِفْعَوَّلَ'.

C. VERBS RECOGNITION

The verbs recognition in our model passes through different steps, which are presented in Fig. 10. The lexeme read from the user text is parsed by a regular expression engine, which tries to match the lexeme with all [[prefix]

```
prefix1 = ['', 'فَ', 'وَ']
prefix2 = ['', 'لِ', 'سَ', 'لَ']  #prefix for present only
prefix3 = ['ي', 'أ', 'ت', 'ن']  # prefix for present only
```

FIGURE 5. Verbs' prefixes (three levels).

prefix] derivative [suffix [suffix]] forms. The number of prefixes and suffixes in the Arabic Language may reach three for each. From the derivative, the root is extracted. Then the hash-table is indexed by the result of the hash-function of the root. If the derivative is represented in the code in the hash-table, then the lexeme is considered a verb. If all matching regular expressions do not lead to a verb, then the lexeme is not a verb. The regular expression engine is used by the lexical analyzer to check if a lexeme from the user input is a *Verb* after checking if it is a *Harf*. If it is neither *Harf* nor *Verb,* then it is considered as a *Noun*. Fig. 11 presents some regular expressions used in this engine.

IV. EXPERIMENTATION AND RESULTS

The Experiments and results demonstrate verb recognition and sentence recognition, each of which will be discussed in the following sub-sections.

Depending on the nouns: ['أَنَا', 'أَنْتَ', 'أَنْتِ', 'أَنْتُمَا', 'أَنْتُمْ', 'أَنْتُنَّ', 'هُوَ', 'هِيَ', 'هُمَا', 'هُمْ', 'هُنَّ', 'نَحْنُ'], the first suffix is used from the list ['تُ', 'تِ', 'تَ', 'تْ', 'تُمَا', 'تَا', 'ا', 'نَا', 'تُمْ', 'وا', 'نَ', ''']. Moreover, a pronoun may be attached as a second suffix (ضمائر متصلة): from ['نِي', 'كَ', 'كِ', 'هُ', 'هَا', 'هُمَا', 'نَا', 'كُمْ', 'هُمْ', 'هُنَّ', ''].

FIGURE 6. Verbs' suffixes (two levels).

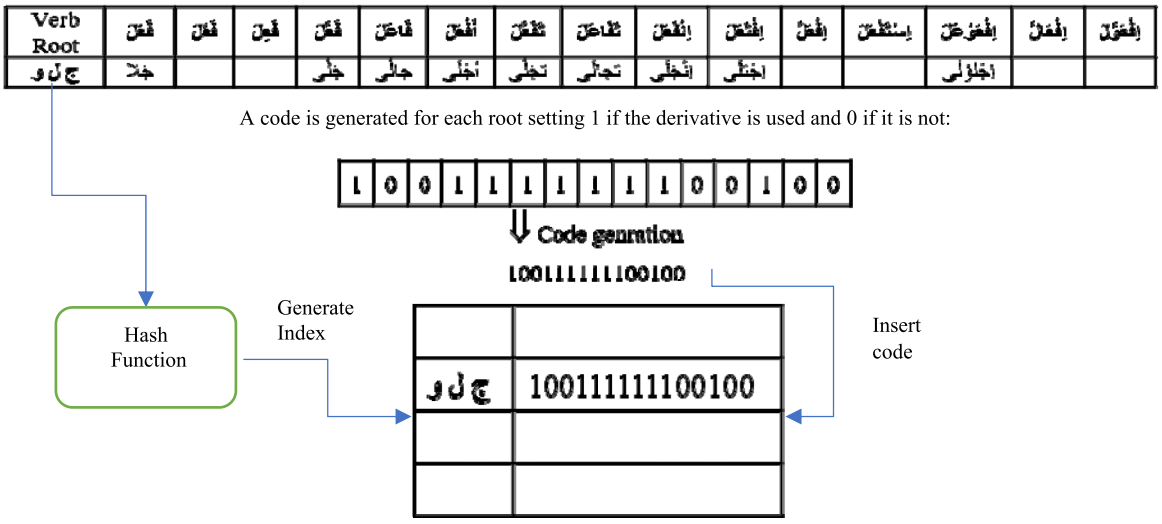


FIGURE 7. Hash-table verbs generation.

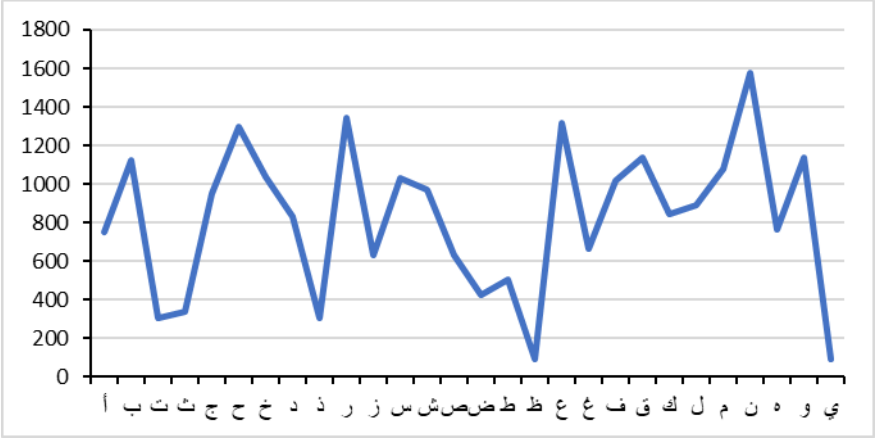


FIGURE 8. Verbs per roots' first letter.

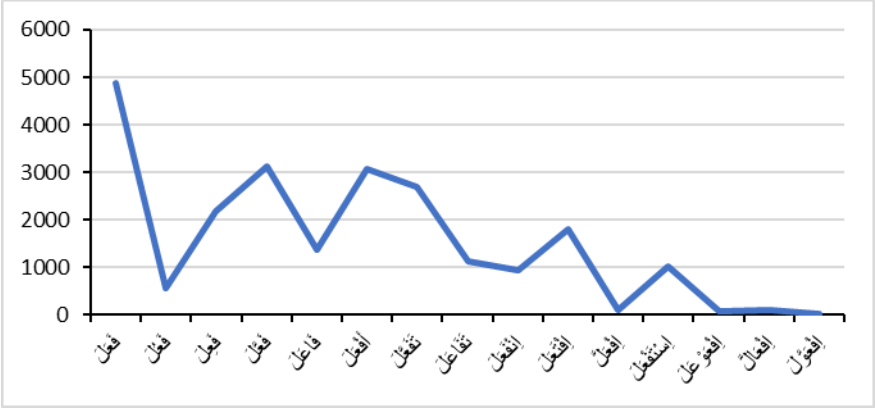


FIGURE 9. Verbs per roots' derivatives.

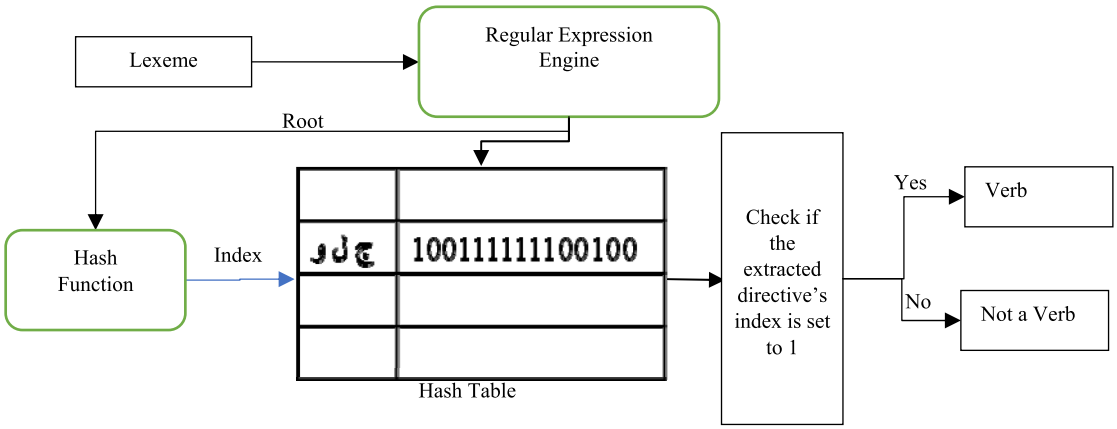


FIGURE 10. Verb recognition flowchart.

Regular expression	Meaning
`re.sub('[فعل]','.',v3[i])`	Replace the letters 'ف', 'ع', and 'ل' by a dot in all verbs' derivatives forms.
`re.sub('[]','',s)`	Remove all vowels from the lexeme s. Note that at this stage of the research, we are considering vowels in only a few cases.

`For all member in prefix1+prefix2+prefix3:`
`  For all verbs derivatives forms:`
`  Construct the regular expression Regex.`
`  For all combined suffixes:`
`  Concatenate the suffix to Regex.`
	Construction of every regular expression for all verb derivatives forms after adding the prefixes and the suffixes. Note that the prefixes and suffixes are depending on the use of the past, present, and future. The future tense is only considered by adding the prefix 'سَ'. Concatenating 'سَ' with a present verb gives the future tense.

`Regex.match(s)`	Match the regular expression Regex with the lexeme s.

FIGURE 11. Some regular expressions and their meanings.

A. VERB RECOGNITION

The input to our framework was composed of some Surat from the Quran. Fig. 12 shows the number of 3-letters-root's verbs that we recognized from some Surat of the Quran.

Globally, there are only four used verbs' derivatives, at a certain level, in these Surat: فَعَلَ, which is the most used in all Surat, فَعَّلَ, فَاعَلَ, and أَفْعَلَ. This goes with the fact that these derivatives have the maximum number of roots in the Arabic language.

We denote by false-positive the fact that a lexeme, which is not a verb, is recognized as a verb. Similarly, by false-negative, we describe the fact that a verb is not recognized as such. Table 5 gives the false-positive and false-negative of verbs recognition. The average false-positive rate is around 9%. The average false-negative rate is around 4%, which is due to some added rules to eliminate the nouns that have verb roots. These rates can be considerably reduced while taking into account the vowels. Moreover, and for the Arabic texts

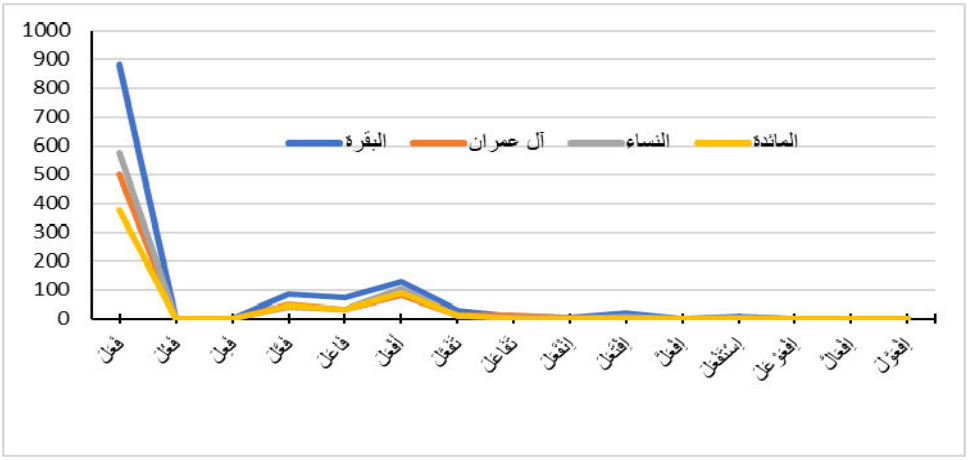


FIGURE 12. 3-letters-root Verbs' recognition.

فَأَسْقَيْنَاكُمُوهُ وَمَا أَنْتُمْ لَهُ بِخَازِنِينَ										Input
بِخَازِنِينَ	له	أَنْتُمْ	ما	و	ه (suffix 3)	كمو (suffix 2)	نا (suffix 1)	أسقى (أَفْعَلَ)	فَ (prefix)	Lexical Analyzer Level
Noun	PreNoun	Noun	PreNoun	PreNoun	Pronoun	Pronoun	Noun	Verb (past: root س ق ي 100111111101000)	PreVerb	
A token is a structure that contains all needed information to help the parser decide of the syntax analysis										
Nominal Sentence					Verbal Sentence					
خبر	جار ومجرور	مبتدأ	حرف نفي	حرف عطف	مفعول به ثان	مفعول به أول	فاعل	فعل ماضي	حرف استئناف	First analysis / Parser Level
خبر ما	جار ومجرور	اسم ما	فعل ناسخ	حرف عطف	مفعول به ثان	مفعول به أول	فاعل	فعل ماضي	حرف استئناف	Second analysis / Parser Level

FIGURE 13. Sentence syntax analysis.

TABLE 5. False-positive and alse-negative rate for different Surats.

Surat	Total recognized verbs	False-positive rate	False-negative rate
البقرة	1250	8.4	4.2
آل عمران	710	9.7	3.8
النساء	789	9.2	4.1
المائدة	570	8.9	3.6

without vowels, the context (the sentence at the syntax level) will help to reduce the rate.

B. SENTENCE RECOGNITION

A verb in the form [[prefix] prefix] derivative [suffix [suffix]] is, in fact, a sentence or a part of it. By parsing this lexeme, the sentence syntax is recognized. In the same way, the *Hurufs* and *Nouns* may have prefixes and suffixes and will be treated using regular expressions. While this latter is easy for the *Hurufs*, it is not trivial for Nouns. This will be focused on future work.

Using the grammar presented in Fig. 2, the parser is built to recognize and analyze the syntax of an Arabic sentence. If there is an ambiguity, all different recognized forms of syntax analysis are given. Fig. 13 gives an example of a sentence syntax analysis.

Table 6 presents the comparison of the results between different techniques for Arabic root extraction. Even if the input is not the same, our proposed model outperforms other methods.

Most of the recent researches concentrated on the morphological level extracting roots from Arabic words with a recognition performance between 94% and 96%. Compared to these researchers our proposed model is as good as the declared approaches since it achieved 87% verbs recognition and 9% of false-positive (non-verbs) rate, which brings the total to 96%. Our model is close to what is proposed in [22] as they use a file containing some Arabic roots while we are building a hash-table that contains all letters' root verbs. Similarly, they are using what they call Arabic rules while we are describing the verbs' derivatives and their potential prefixes and suffixes using a regular expression. Moreover, our proposed model uses Arabic grammar to recognize a sentence and give its syntax analysis. On the other hand, [3] and [8] claim to have a root extraction performance rate of 94% and 96% respectively. However, the first did not mention the input type and size, while in the case of the second one, the test is done over 4000 verbs not included in the normal Arabic text, which adds complexity to distinguish between verbs and nouns.

TABLE 6. Comparison of the results with our proposed method and state-of-the-art.

Method	Description	Roots recognition rate
Morphological Analysis [3]	Based on a list of morphological weights and the roots of three consonants, the proposed solution deals with the removal of the prefixes and the suffixes.	94%
Morphological analysis system [8]	The mechanism used to classify verbs' roots based on a set of morphological rules. The solution is evaluated by 4000 verbs.	96%
Bi-gram [13], [14]	The multi-objective method with a statistical method to separate the Arabic roots suggested in [13] is improved in [14] using the bi-gram technique.	80%
Translation Model [19]	Madamira system translation model from Arabic to Chinese.	96.2%
Rule-based root extraction [20]	Provided a Root extraction algorithm based on Arabic roots' rules investigation.	94%
Heavy/light Stemmer [24]	Stemmer algorithm based on Arabic's morphological features.	96.9%
Stemming Algorithm [22]	Based on a set of rules and a file containing Arabic roots, the authors extract roots from input texts.	84%
Naïve Bayes algorithm [32]	Classification of a set of 1500 documents into 5 classes.	69%
Our Proposed Method	We used regular expressions to divide the string into prefix, verb (mapped into a derivative from), and suffix. Extract the root from the verb's derivative and get the index in the hash-table to check if the verb's derivative exists.	87%

V. CONCLUSION

To be able to derive and know the syntax analysis of an Arabic sentence, we proposed a new model based on regular expressions and Arabic grammar rules. We generate all verbs with three-letters' root and their derivatives and build a hash-table with access complexity of O(1). A lexical analyzer is reading, slicing, and returning the token for each input word to the parser that is checking the grammar rules and producing the analysis. All verbs represented in the regular expression engine are detected, among which only 9% are false-positive. A false-negative rate of 4% represents the verbs that are rejected because they are considered as Nouns. Our model performance is as good as claimed by most of the researches using a stemming process for root extraction when counting the total root recognition (87% verbs and 9% of false-positive). As Arabic text processing needs more deep analysis than root recognition, our model is covering such need, by attempting to recognize verbs over nouns, which is not an easy task in the Arabic Language and needs further future work. The grammar is used to recognize Arabic sentences and get the syntax analysis. Although our results are promising, some verbs forms were not taken into consideration like imperative verbs, verbs with weak letters (vowel with long sound 'ا', 'ى', and 'و'). These will be considered in future work. The same will be for more refining the grammar to include more granularity in the syntax.

ACKNOWLEDGMENT

The authors gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, on the material support for this research under the Project number: 5347-coc-2018-2-14-S during the academic year 1439 AH / 2018 AD.

REFERENCES

[1] M. Beseiso, A. R. Ahmad, and R. Ismail, "An Arabic language framework for semantic Web," in *Proc. Int. Conf. Semantic Technol. Inf. Retr.*, Jun. 2011, pp. 7–11.

[2] M. Poprat and D. Jena, "Building a biowordnet by using wordnet's data formats and wordnet's software infrastructure—A failure story," *Softw. Eng., Test., Qual. Assurance Natural Lang. Process.*, vol. 4, no. 6, pp. 31–39, 2008.

[3] A. Abu-Errub, A. Odeh, Q. Shambour, and O. A. Hassan, "Arabic roots extraction using morphological analysis," *Int. J. Comput. Sci. Issues*, vol. 11, no. 2, pp. 128–134, 2014.

[4] C. A. Rytting, P. Rodrigues, T. Buckwalter, D. M. Zajic, B. Hirsch, J. Carnes, N. Lynn, S. C. Wayland, C. Taylor, J. White, and C. Blake, III, "Error correction for Arabic dictionary lookup," in *Proc. 7th Int. Conf. Lang. Resour. Eval.*, 2010, pp. 263–268.

[5] A. M. Azmi, M. N. Almutery, and H. A. Aboalsamh, "Real-word errors in Arabic texts: A better algorithm for detection and correction," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1308–1320, Aug. 2019.

[6] H. M. Noaman, S. S. Sarhan, and M. A. A. Rashwan, "Automatic Arabic spelling errors detection and correction based on confusion matrix-noisy channel hybrid system," *J. Theor. Appl. Inf. Technol.*, vol. 40, no. 2, pp. 54–64, 2016.

[7] E. L. M. Hassan, "The impact of standard Arabic verb phrase structure on Moroccan EFL learners' writing," *J. Humanities Social Sci.*, vol. 24, no. 1, pp. 60–67, 2019.

[8] A. Yousf, "The morphological analysis of Arabic verbs by using the surface patterns," *Int. J. Comput. Sci. Issues*, vol. 7, no. 3, pp. 33–36, 2010.

[9] M. Althobaiti, U. Kruschwitz, and M. Poesio, "AraNLP: A Java-based library for the processing of Arabic text," in *Proc. 9th Int. Conf. Lang. Resour. Eval.*, 2014, pp. 4134–4138.

[10] B. Haddad, A. Awwad, M. Hattab, and A. Hattab, "Associative root–pattern data and distribution in Arabic morphology," *Data*, vol. 3, no. 2, pp. 1–17, 2018.

[11] H. M. Alghamdi, A. Selamat, and N. S. Abdul Karim, "Arabic Web pages clustering and annotation using semantic class features," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 26, no. 4, pp. 388–397, Dec. 2014.

[12] W. Alabbas, H. M. Al-Khateeb, and A. Mansour, "Arabic text classification methods: Systematic literature review of primary studies," in *Proc. 4th IEEE Int. Colloquium Inf. Sci. Technol. (CiSt)*, Oct. 2016, pp. 361–367.

[13] H. Khafajeh, N. Yousef, and M. Abdeldeen, "Arabic root extraction using a hybrid technique," *Int. J. Adv. Comput. Res.*, vol. 8, no. 35, pp. 89–95, Mar. 2018.

[14] N. Yousef, A. Abu-Errub, A. Odeh, and H. Khafajeh, "An improved Arabic word's roots extraction method using N-gram technique," *J. Comput. Sci.*, vol. 10, no. 4, pp. 716–719, 2014.

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

[16] M. O. Hegazi, "An approach for Arabic root generating and lexicon development," *Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 1, pp. 9–15, 2016.

[17] S. Farwaneh, "Non-sound' verb inflection in Arabic: Allomorphic variation and paradigmatic uniformity," *Morphology*, vol. 30, no. 1, pp. 61–89, Feb. 2020.

[18] A. Abu and E. Hanandeh, "Developing a transition parser for the Arabic language," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 9, pp. 173–175, 2016.

[19] F. Aqlan, X. Fan, A. Alqwbani, and A. Al-Mansoub, "Improved Arabic–Chinese machine translation with linguistic input features," *Future Internet*, vol. 11, no. 1, p. 22, Jan. 2019.

[20] N. Thalji, N. Adilah, W. Bani, 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.

[21] H. Elazhary, A. Alharthi, E. Balkhi, G. Aljahdali, D. Zagzoog, and A. Alkhammsh, "Automated tutoring of Arabic word root extraction," *Int. J. Sci. Eng. Res.*, vol. 6, no. 7, pp. 687–691, Jul. 2015.

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

[23] R. Mohammed, "New Arabic stemming based on Arabic patterns," *Iraqi J. Sci.*, vol. 57, no. 3, pp. 2324–2330, 2016.

[24] I. Zeroual, M. Boudchiche, A. Mazroui, and A. Lakhouaja, "Developing and performance evaluation of a new Arabic heavy/light stemmer," in *Proc. 2nd Int. Conf. Big Data, Cloud Appl.*, Mar. 2017, pp. 1–6.

[25] I. A. Al-Sughaiyer and I. A. Al-Kharashi, "Arabic morphological analysis techniques: A comprehensive survey," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 55, no. 3, pp. 189–213, 2004.

[26] S. Al-Fedaghi and F. Al-Anzi, "A new algorithm to generate Arabic root-pattern forms," in *Proc. 11th Nat. Comput. Conf. Exhib.*, 1989, pp. 391–400.

[27] J. Wightwick and M. Gaafar, *Arabic Verbs & Essentials of Grammar*, 3rd ed. New York, NY, USA: McGraw-Hill, 2018.

[28] I. Blanchete, M. Mourchid, S. Mbarki, and A. Mouloudi, "Formalizing Arabic inflectional and derivational verbs based on root and pattern approach using NooJ platform," in *Formalizing Natural Languages With NooJ and Its Natural Language Processing Applications. NooJ* (Communications in Computer and Information Science), vol. 811, S. Mbarki, M. Mourchid, and M. Silberztein, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-73420-0_5.

[29] A. Alsaad and M. Abbod, "Arabic text root extraction via morphological analysis and linguistic constraints," in *Proc. UKSim-AMSS 16th Int. Conf. Comput. Modeling Simulation*, Mar. 2014, pp. 125–130.

[30] A. H. Fatma, "Towards a new approach for Arabic root extraction: Exploit relations between the word letters and their placement in the word for Arabic root extraction," *Comput. Sci.*, vol. 14, no. 3, p. 327, 2013.

[31] W. Alabbas, H. M. Al-Khateeb, A. Mansour, G. Epiphaniou, and I. Frommholz, "Classification of colloquial Arabic tweets in real-time to detect high-risk floods," in *Proc. Int. Conf. Social Media, Wearable Web Anal. (Social Media)*, Jun. 2017, pp. 1–8.

[32] M. El Kourdi, A. Bensaid, and T.-E. Rachidi, "Automatic Arabic document categorization based on the Naïve Bayes algorithm," in *Proc. Workshop Comput. Approaches Arabic Script-Based Lang. (Semitic)*, vol. 4, 2004, pp. 51–58.

[33] M. Oakleaf, "Writing information literacy assessment plans: A guide to best practice," *Commun. Inf. Literacy*, vol. 3, no. 2, pp. 80–90, 2009.

[34] M. Abdul-Mageed, S. Kübler, and M. Diab, "SAMAR: A system for subjectivity and sentiment analysis of Arabic social media," in *Proc. 3rd Workshop Comput. Approaches Subjectivity Sentimental Anal.*, Jul. 2012, pp. 19–28.

[35] A. Shoukry and A. Rafea, "Sentence-level Arabic sentiment analysis," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2012, pp. 546–550.

[36] L. Albraheem and H. S. Al-Khalifa, "Exploring the problems of sentiment analysis in informal Arabic," in *Proc. 14th Int. Conf. Inf. Integr. Web-Based Appl. Services (IIWAS)*, vol. 12, 2012, p. 415.

[37] J. Kamps, M. Marx, R. J. Mokken, and M. de Rijke, "Using wordnet to measure semantic orientations of adjectives," in *Proc. 4th Int. Conf. Lang. Resour. Eval.*, 2004, pp. 1115–1118.

[38] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, vol. 5, 2005, p. 617.

[39] P. Palanisamy, V. Yadav, and H. Elchuri, "Serendio: Simple and practical lexicon based approach to sentiment analysis," in *Proc. 2nd Joint. Conf. Lexical Comput. Semantics (SEM), 7th Int. Work. Semant. Eval. (SemEval)*, vol. 2, 2013, pp. 543–548.

[40] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proc. 8th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 1997, pp. 174–181.

[41] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 417–424.

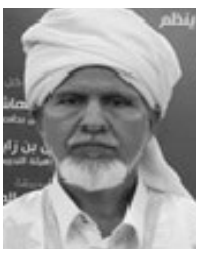
[42] K. J. Al Daimi and M. A. Abdel-Amir, "The syntactic analysis of Arabic by machine," *Comput. Humanities*, vol. 28, no. 1, pp. 29–37, Jan. 1994.

[43] A. S. Almahubi. (2017). *Verbs of Arabic Language*. [Online]. Available: http://www.alukah.net

MOHAMED TAHAR BEN OTHMAN (Senior Member, IEEE) received the Ph.D. degree in computer science from the National Institute of Polytechnic of Grenoble (INPG), France, in 1993, the master's degree in computer science from the École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG), in 1989, and the Senior Engineer Diploma degree in computer science from the Faculty of Sciences of Tunis.

He worked as a Postdoctoral Researcher at the Laboratoire de Génie Logiciel (LGI), Grenoble, France, from 1993 to 1995, and the Dean of the Faculty of Science and Engineering, University of Science and Technology, Sana'a, Yemen, from 1995 to 1997. He was a Senior Software Engineer with Nortel Networks Corporation, Canada, from 1998 to 2001, and an Assistant Professor with the Computer College, Qassim University, Saudi Arabia, from 2002 to October 2010, and also an Associate Professor. He has been a Professor of computer science, since November 2017. His research interests include data mining, artificial intelligence, information security, and bioinformatics.

MOHAMMED ABDULLAH AL-HAGERY received the B.Sc. degree in computer science from the University of Technology, Baghdad, Iraq, in 1994, the M.Sc. degree in computer science from the University of Science and Technology (USTY), Sana'a, Yemen, in 1998, and the Ph.D. degree in computer science and information technology (software engineering) from the Faculty of Computer Science and IT, Universiti Putra Malaysia (UPM), in November 2004. He was the Head of the Computer Science Department, College of Science and Engineering, USTY, from 2004 to 2007. Since 2007, he has been a Staff Member with the Department of Computer Science, College of Computer, Qassim University, Saudi Arabia. He was appointed as the Head of the Research Centre, Computer College, and a Council Member of the Scientific Research Deanship, Qassim University, from September 2012 to October 2018. He is currently an Associate Professor. He has published more than 25 research articles in various international journals. He is teaching the master's degree students and a supervisor of four master's thesis. He is a Jury Member of several Ph.D. and master's thesis and an internal and external examiner in the field of his specialization.



YAHYA MUHAMMAD EL HASHEMI received the Ph.D. degree in applied linguistics from Cheikh Anta Diop University, Dakar. Since 2014, he has been a Professor with the Department of Arabic Language, College of Arabic Language and Social Studies, Qassim University. He is currently a Professor of linguistics with the Faculty of Arts and Humanities, University of Nouakchott, and a Researcher of applied linguistics and jurisprudence.