[Egyptian Informatics Journal 21 (2020) 209–217](https://doi.org/10.1016/j.eij.2020.02.004)

Contents lists available at [ScienceDirect](http://www.sciencedirect.com/science/journal/11108665)

Egyptian Informatics Journal

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com/)

[](http://crossmark.crossref.org/dialog/?doi=10.1016/j.eij.2020.02.004&domain=pdf)An intelligent use of stemmer and morphology analysis for Arabic information retrieval

Ali Alnaied [a](#_bookmark0),[⇑](#_bookmark3), Mosa Elbendak [b](#_bookmark1), Abdullah Bulbul [c](#_bookmark2)

a *Department of Electrical and Computer Engineering, Ankara Yildirim Beyazit University, Turkey*

b *Department of Computer and Information Sciences, Northumbria University, UK*

c *Department of Computer Engineering, Ankara Yildirim Beyazit University, Turkey*

# a r t i c l e i n f o

*Article history:*

Received 13 November 2019

Revised 23 January 2020

Accepted 18 February 2020

Available online 7 March 2020

*Keywords:*

Natural language processing Arabic morphological analysis Information retrieval systems Arabic stemmer

# a b s t r a c t

Arabic Information Retrieval has gained significant attention due to an increasing usage of Arabic text on the web and social media networks. This paper discusses a new approach for Arabic stem, called Arabic Morphology Information Retrieval (AMIR), to generate/extract stems by applying a set of rules regarding the relationship among Arabic letters to find the root/stem of the respective words used as indexing terms for the text search in Arabic retrieval systems. To demonstrate the usefulness of the proposed algo- rithm, we highlight the benefits of the proposed rules for different Arabic information retrieval systems. Finally, we have evaluated AMIR system by comparing its performance with LUCENE, FARASA, and no- stemmer counterpart system in terms of mean average precisions. The results obtained demonstrate that AMIR has achieved a mean average precision of 0.34% while LUCENE, FARASA and no stemmer giving 0.27%, 0.28% and 0.21, respectively. This demonstrates that AMIR is able to improve Arabic stemmer and increases retrieval as well as being strong against any type of stem.

© 2020 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Artificial Intelli-

gence, Cairo University. This is an open access article under the CC BY-NC-ND license ([http://creative-](http://creativecommons.org/licenses/by-nc-nd/4.0/)

[commons.org/licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

1. Introduction

A key objective of search engines is to leverage online massive information available from the internet or social media to return query results as per user’s specifications. This return satisfies the user’s needs. The Arabic language has different semantic and pho- netic structures when compared to other languages [[1]](#_bookmark13). This differ- ence has also posed a significant issue as to how Arabic users benefit from search engine optimization. Recently, Arabic language has attracted significant interest from researchers to optimize users’ searches. The main challenge is that there are few webpages authored in the Arabic language [[2]](#_bookmark14). The other daunting challenge of the Arabic information retrieval systems has been the inability to solve problems such as the ambiguity of words as most roots

\* Corresponding author.

*E-mail address:* [a\_alnaied@yahoo.com](mailto:a_alnaied@yahoo.com) (A. Alnaied).

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

**Production and hosting by Elsevier**

are composed by three letters, orthographic variations, sophisti- cated and very rich morphology.

Construction of Arabic words is based on abstract forms known as roots. A root, in phonetics, is the most basic word that serves as a base to generate other derivatives obtained by blending suffixes or affixes on the root to produce verbs, adjectives and nouns [[3,4]](#_bookmark15). It is worth noting that the Arabic language is very inflectional as it has trilateral roots used to derive over 85% of its words. Typically, Ara- bic language verbs and nouns are derived from a set of 10,000 roots [[5]](#_bookmark18).

The stem, which is a technique for reducing the grammatical form of a word based on inflection and derivation. Brent [[6]](#_bookmark19) is a crucial step, especially for Arabic information retrieval because the same word may have many different forms. Also, the Arabic language has a significant number of stemming techniques and a notable one is Kareem Darwish’s [[7]](#_bookmark20) Al-Stem which was later on modified by University of Massachusset’s. [[8]](#_bookmark21) The Al-Stem Stem- mer was further modified by David Graff whereby (,ﻓﻢ ,ﺍﻝ ,ﻟﻞ ,ﻭﻱ ,ﻝ ﻭﺍ ,ﻓﺎ ,ﻳﺎ ,ﻭﺍﻝ ,ﻓﺎﻝ ,ﺑﺎﻝ ,ﺑﺖ ,ﺍﺕ ,ﻣﺖ ,ﺕ ,ﺳﺖ ,ﻧﺖ ,ﺑﻢ ,ﻭﻡ ,ﻛﻢ) can be removed from the word’s prefixes and suffixes [[9]](#_bookmark22). The Aljlayl Stemmer [[10]](#_bookmark23), classified as a light stemmer. This stemmer was developed by Mohammed Aljlayl for use to retrieve query searches. The author factored the length of the words to be used for removing

<https://doi.org/10.1016/j.eij.2020.02.004>

1110-8665/© 2020 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

affixes and suffices, additionally, he normalized some specific Ara- bic characters [[10,11]](#_bookmark23).Also, the stemmers stem words were blind, however, a robust and efficient Arabic Stemmer’s Algorithm can decrease data storage and computational time [[12]](#_bookmark24).

Arabic stems are different when compared to other languages such as English, French etc. In addition, Arabic nouns can take the form according to several factors such as plurality (plural, sin- gular, or dual), gender (feminine or masculine), and grammatical tense (present, past, future, and command). In contrast, stemming refers to a computational technique used to reduce words to their respective stems or roots. One disadvantage of the existing Arabic stemmers is that they are prone to immense stemming error-rates [[13]](#_bookmark25).

This paper proposes a new approach to produce a high- performance tool to generate/extract Arabic stems by implement- ing a morphological analysis using specific linguistic rules. The tool is compared against LUCENE and FARASA methods and the results show that our approach outperforms the other methods.

This paper describes the two contributions as follows:

* The comprehensive processing for Arabic texts to improve the root extraction. Existing schemes extract the roots by removing

affixes from a word without distinguishing whether the removed letters are actually core letters of the root or not, like study [[14,15]](#_bookmark26). This is because, in Arabic language it is not easy to determine the conjunctions of pronouns, gender, plural, prepositions, stopwords, and etc. that are connected directly to the word. This means that the existing schemes cannot authenticate whether the removed letters are the roots or not. This is the gap that the proposed scheme aims to address by proposing a method to validate whether the removed letters are actually core letters of the root. Therefore, our proposed technique attempts to extract the Arabic root/stem based on a validation of the letters before removing affixes by building AMIR dictionary that generate over 1400 words from each root. Therefore, the method proposes a root extraction based on mor- phology features by matching the word with all possible affixes and patterns attached to it. To the best of our knowledge, a sin- gle root can generate 1000 words using previous studies. Thus, our method increases the efficacy of extracting a word effec- tively while minimizing the ambiguity since it depends on val- idation before removing affixes where each input term is compared against with all the words in the dictionary until a match is found; if no root is found, the original word is returned unchanged. For example, the word ﻭﻟﻤﺪﺍﺭﺳﻜﻦ ‘‘walimadarisikin” (and for our schools) shows the concatenation of morphemes to form the word. To distinguish between these morphemes, we say that ﺩﺭﺱ (lesson) is the root morpheme; prefix ﻡ (m) is a derivational morpheme where it refers to noun; prefix ﻝ (l) is an inflectional morpheme that refers to prepositions; prefix ﻭ (w) is an inflectional morpheme that refers to stopword; infix ﺍ (a) is an inflectional morpheme that refers to plural

fusing, especially when a word in plural form is in the infixes. Therefore, the proposed method aims to produce a high- performance tool to extract Arabic root/stems by adding infixes to prefixes and suffixes. For example: the word ﻣﻜﺎﺗﺐ (offices) by removing infix ﺍ (a), will result in the word ﻣﻜﺘﺐ (office); thus, the word is changed from plural form to get its singular one by applying AMIR rule No 3. Using the word ﻛﺎﺗﺐ (author) and by removing infix ﺍ (a) will result in the word ﻛﺘﺐ (wrote). Thus, the word would result in a change of the meaning. According to AMIR rules No 5, this case is not permitted. Using the work

[[16]](#_bookmark26) stemming can give better precision in information retrie- val. Therefore, we believe that our proposed method will improve the precision in Arabic information retrieval through the use of infix extraction unlike other languages such as Eng- lish. As mentioned earlier, English language uses suffixes and prefixes to determine the plurality of a word. Consequently, an infix is very important factor that can improve Arabic retrie- val systems. Therefore, we proposed is capable to solve a prob- lems of the plural form while still allowing the extraction of stem/root of Arabic words thus resulting in an increased.

1. Related works

Over the last few decades, several works have been carried out for Arabic information retrieval problems. However, many weak- nesses and problems still face the Arabic language retrieval since they mainly rely on morphological and stemming analysis with lit- tle attention or emphasis on lemmatization. This section discusses recent advances in stemming, and morphological analysis and how they have an impact on the retrieval of documents in Arabic.

Khoja’s stemmer [[17]](#_bookmark26) previously showed the first attempt to find the Arabic root by the removal of prefixes and suffixes. The author [[18]](#_bookmark26) developed the Porter stemmer tailored for the English language. This stemmer leverages two-step rewriting rules and is achieved by removing approximately 60 different suffixes by [[19]](#_bookmark26). Up to now, the Porter Stemmer has been documented to have an exemplary performance, especially in its precision and recall of evaluations. However, this stemmer has the drawback of being very aggressive in its creation of stems and ends up over stemming. Therefore, the proposed method intends to solve the Khoja’s prob- lem of over stemming and aggressiveness as our method provides specific patterns of a word. This will reduce the major drawback of Khoja’s stemmer.

Larkey [[20,21]](#_bookmark26) shows better retrieval efficiency as described in Light stemming which merely removes prefixes and suffixes depending on a predefined list. However, it does not guarantee the production of better results when evaluating experiments. Therefore, the proposed method intends to increase the production of better results by incorporating the use of infixes to suffixes and prefixes. There are a number of root extraction techniques for Ara- bic language known as heavy stemming or stemming based root words which work by removing all affixes as described by Khoja

form; and suffix ﻛﻦ

(kunn) is an inflectional morpheme

[[22]](#_bookmark27).

indicating the gender. Lastly, the proposed method is capable to improve the extraction root in Arabic language, and this is a major improvement in previous methods.

* The second contribution relates to an improved precision in

Arabic information retrieval using infix stemmer. In English,

affixes can generally be divided into two groups: (prefixes and suffixes). However, in Arabic language, affixes can be divided into three groups: (prefixes, infixes, and suffixes). Therefore, existing schemes are unable to extract stem or root of words having an infix. In Arabic morphology there exist many words that have infixes and removing an affix depends on the morpho- logical structure of the language. In addition, extracting the root of a word in its plural form can always be challenging and con-

Darwish [[23]](#_bookmark27) proposed FARASA system which also segments Arabic text into words. However, this stemmer technique handles prefixes and suffixes. Our proposed, AMIR leverages on some FAR- ASA components but has its own rules that allow for handling problems of infixes in addition to improving the generation of pre- fixes and suffixes.

Numerous Arabic morphology systems have been devoted towards morphed requirements of words, like the study [[14]](#_bookmark26) which proposed a new model for identifying the verb root produced in a tool (RootIT) by a root extraction without disambiguation out of traditional methods. Therefore, this paper removes the prefixes and suffixes without using any linguistic rules. Our proposed method proposes a novel root extraction technique that gives

support to natural language processing to include morphology features by matching the word with all possible affixes and

original root; and *suf f ixm*is a number of suffixes adding to the root after ends letters of original root.

patterns attached to it.

According to [[24,25]](#_bookmark27) the most commonly used stemmers in the

*pref ix*1

+ + *pref ixn*

+ T1 + T2 + *inf ix*2

+ *T*3 + *suf f ix*1

+ ....

Arabic language are light stemmer and Khoja stemmer. Indeed, stemmers can generally be divided into two groups: first, light stemmers such as the stemmer provided by [[8]](#_bookmark21). The second one represents the root derivation stemmer proposed in [[22]](#_bookmark27). In addi- tion, the work [[25]](#_bookmark27) proposed two different stemming techniques based on light stemming by utilizing extra suffixes in the total number of prefixes and suffixes to be removed. Thus, the new added prefixes and suffixes are extended prefixes (ﻓﻞ ,ﻭﻟﻞ ,ﻭﺑﺎﻝ ,ﻟﻞ ,ﻓﺎﻝ ,ﻛﺎﻝ ,ﺑﺎﻝ ,ﻭﺍﻝ ,ﺍﻝ) and extended suffixes

+ *suf f ixm*

where *infix*2 is inserted after T2 of original root.

For sequence 3.1, 3.2 affixes is added to the root to generate new words as follows:

fol- as words a generate to root of begins to adding prefixes 58 ● ﺍ, ﻭﺍ, ﻭﻥ, ﻭﻩ, ﺍﻥ, ﺍﺕ, ﺍﻱ , ﻛﻢ, ﻛﻢ, ﻭﻥ, ﻓﻦ, ﺑﻦ, ﻱ, ﻥ, ﺏ, ﻳﻦ, ﻳﻪ, ﻩ, ﻱ, ﻭﻝ, ﻓﻞ, ﻓﺎ‘‘, lows:

ﻓﻠﻞ, ﻭﻟﻞ, ﺍﺳﺖ, ﺳﺖ, ﻭﺍﻝ, ﻭﻛﺎﻝ, ﻭﻓﺎﻝ, ﻭﺑﺎﻝ, ﻓﺒﺎﻝ, ﻭﺕ, ﻓﺖ, ﻭﻥ, ﻓﻦ, ﻛﺖ, ﻡ, ﻭﻡ, ﻓﻢ, ﺑﻢ, ﻟﻢ, ﻭﻟﻢ,

(ﺕ ,ﻫﻢ ,ﻧﺎ ,ﻫﻤﺎ ,ﺗﻲ ,ﻭﺍ ,ﻱ ,ﺓ ,ﻳﺔ ,ﻳﻪ ,ﻳﻦ ,ﻭﻥ ,ﺍﺕ ,ﺍﻥ ,ﻫﺎ) which extended-Light stemmer is greater than their peers in light 10 [[20]](#_bookmark26). Therefore, these studies were unable to extract stem or root of words that contain infixes. While our proposed method is able to extract stem or root of Arabic words by segmenting the word to remove its infixes using its prefixes and suffixes thus allowing the generation of the corresponding root (if it exists).

In addition, there exist various methods to show the perfor- mance of the light stemmer such as Jaffar [[26]](#_bookmark27) to restore Arabic data by adding extra prefixes and suffixes to the list of light 10 [[20]](#_bookmark26). The new added prefixes and suffixes are: prefixes (ﻭ ,ﻟﻞ ,ﻓﺎﻝ ,ﻛﺎﻝ ,ﺑﺎﻝ ,ﻭﺍﻝ ,ﺍﻝ) and suffixes (ﻱ ,ﺓ ,ﻩ ,ﻳﺔ ,ﻳﻪ ,ﻧﻲ ,ﺗﻲ ,ﺗﻪ ,ﻫﻢ ,ﻫﻦ ,ﻳﻦ ,ﻭﻥ ,ﺍﺕ ,ﺍﻥ ,ﻫﺎ). Although, the technique removes the affixes it does not handle infix stemmer problems. Therefore, in this work we have designed a technique to solve a problems of Arabic information retrieval systems. This is achieved by using specific linguistic rules to remove infixes in addition to suffixes and prefixes.

1. AMIR dictionary

ﻓﻠﻢ, ﻭﻛﻢ, ﻭﻛﺎﻟﻢ, ﻓﻜﺎﻟﻢ, ﻭﻟﻠﻢ, ﻭﻟﻢ, ﻓﻠﻠﻢ, ﻭﻟﻠﻢ, ﻭﺑﺎﻟﻢ, ﻓﺒﺎﻟﻢ, ﻭﺍ, ﻓﺎ, ﻭﺏ, ﻭ, ﻑ, ﺏ ,ﻝ

* 3 infixes inserting to the root to generate a words as follows: ﺍ

(alif), ﻭ (waaw), and ﻱ (yaa).

fol- as words a generate to root of ends the to adding suffixes 25 ● ﺍﺕ, ﻭﺍ, ﻭﻥ, ﻭﻩ, ﺍﻥ, ﺗﻲ, ﺗﻪ, ﺗﻢ, ﻛﻢ, ﻫﻦ, ﻫﻢ ﻫﺎ, ﻱ, ﺗﻚ, ﻧﺎ, ﻳﻦ, ﻳﻪ, ﻩ, ﻱ, ﺍ, ﺗﻜﻤﺎ, ﺗﻜﻨﺎ, lows:

,ﺗﻬﻤﺎ, ﺗﻬﻢ, ﻛﻲ

* Affixes can be associated with each other to generator words as follows:
  1. Prefixes with Infixes (58 × 3 = 174)
  2. Prefixes with Suffixes (58 × 25 = 1450)
  3. Infixes with suffixes (3 × 25 = 75)
  4. All affixes together (58 × 3 × 25 = 1450)

Note that not all combinations of above affixes can be joined

together. In case (a), there are 7 prefixes cannot join with infixes. In case (b) there is no exceptions, all prefixes can join with all suf- fixes. In case (c) there are 24 exceptions which are not permitted. In case (d) affixes can connected to each other if they do not form the exceptions above. These exceptions motivate the following definition.

Definition 1. The morphological structure of derivational word

AMIR dictionary is constructed from several Arabic grammatical rule-based such as syntactic rules and morphological rules. There- fore, AMIR dictionary uses morphological features that enable it to specify all inflected forms for each stem templates, which are a combination of the affix with the root. AMIR dictionary is com- posed of two main phases. The first phase is to add patterns to the root (in Arabic, patterns known as ﺍﻭﺯﺍﻥ ‘‘awzan”). This

is:

is:

Derivational = (adverb + root) | (particle + root) | (particle + root + possessive\_pronouns) |

(root + possessive\_pronouns) |

(noun + root) | (particle + noun + root)

Definition 2. The morphological structure of inflectional word

process is done by adding some specific letters to the root such as the letterﻡ (m) like the word ﻣﺪﺭﺱ (teacher). Eight patterns can be added to each root, see AMIR rules. These specific letters refer to a derivation morpheme that contains the basic Arabic patterns forms. The second phase is to add affixes. These affixes indicate the inflectional morpheme, where linked directly to patterns or core root like pronouns, gender, prepositions, and stopwords. Therefore, we extract stem by seeking input words in AMIR dic- tionary, and then segmentation of word and returning the corre- sponding root if it exists based on AMIR rules by removing all inflection morpheme and keep all derivation morpheme as shown in [Fig. 1](#_bookmark4).

Affixes divide into three groups: prefixes, infixes, and suffixes. Arabic word contain one-to-four prefixes, one or two infixes, and one-to-three suffixes. Sequence (3.1) is uses to generate stem when the infixes placed after the first letter of the original root, and sequence (3.2) used to generate stem when the infixes placed after the second letter of the original root.

*pref ix*1 + .... + *pref ixn* + *T*1 + *inf ix*1 + *T*2 + *T*3 + *suf f ix*1 +· · · .

+ *suf f ixm*

where *pref ixn* is a number of prefixes adding to the root; T1 is the

first letter of original root; *infix*1 is inserting after the T1 of original root; T2 is the second letter of original root; T3 is the third letter of

Inflectional = (particle + root) |

(particle + root + possessive \_pronouns) | (root + possessive\_pronouns)

Derivational structures of Arabic often change word meaning and it consist of prefixes, infixes, and suffixes in derived word. While inflectional structures not change word meaning and it con- sist of prefixes, infixes, and suffixes. In this paper, we removed the inflectional and kept the derivational.

* 1. *Arabic stemmer*

Stemmer is a pre-processing tool used to reduce different gram- matical forms/word forms, such as: nouns, adjectives, verbs, adverbs, etc. Therefore, in this paper, we proposed a new approach to improve Arabic stemmer by adding all possible affixes to the root in order to use them as indexing terms in Arabic search oper- ation or information retrieval systems.

As shown in [Fig. 2](#_bookmark5), Arabic stem is classified into two categories:

(i) a statistical stemmer employing statistical information from a large corpus of a given language for morphologically complex texts and (ii) a Rule-based stemmer-employing dictionary targeting to remove inflected affixes from the words based on language specific rules, which we will adopt in this work. Globally the most widely used English stemmer is the Porter Stemmer [[18]](#_bookmark26) (called lemma- tizer). It proposed to remove inflectional endings only such

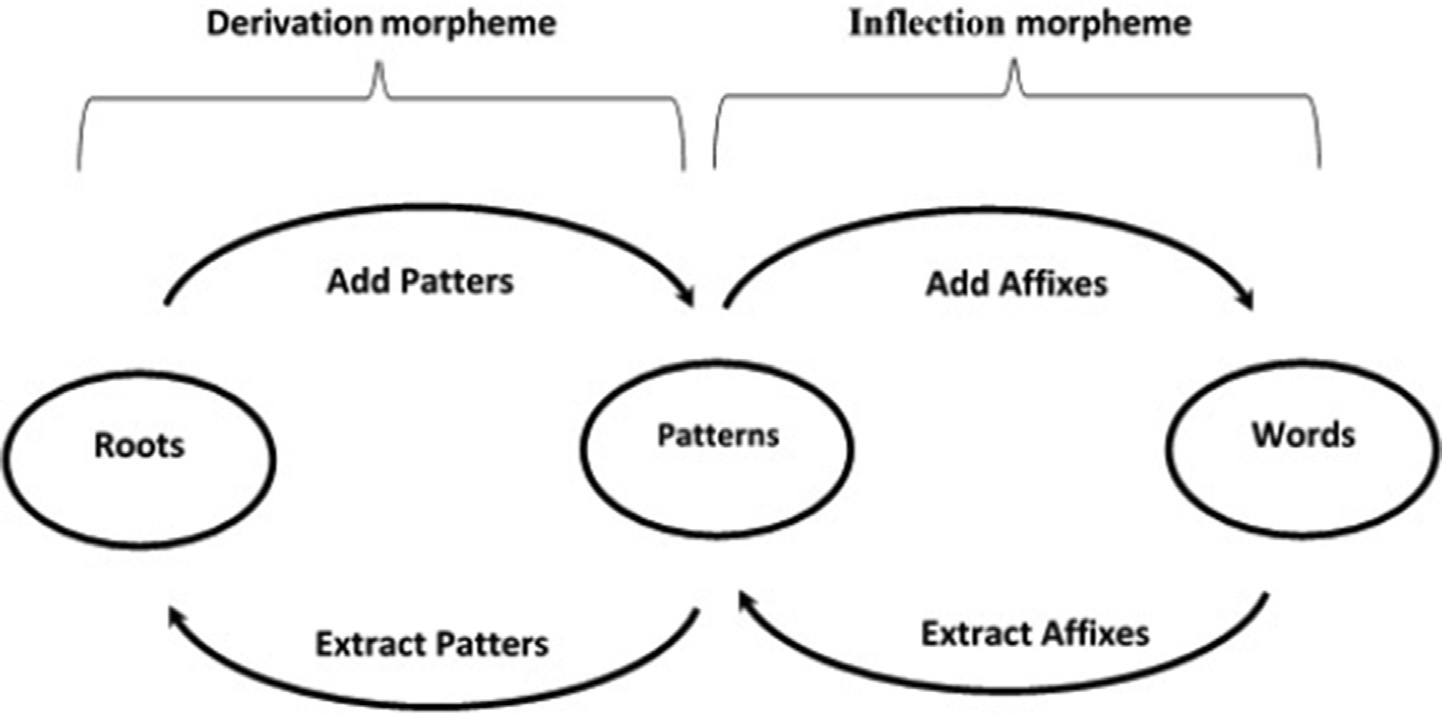


Fig. 1. AMIR Model for generate/extract stem.

as -es, -ies, or -s. In this paper, we proposed similar approach in Arabic, aims to remove all inflectional morpheme (beginnings, middles, and endings), see [Table 1](#_bookmark6).

Light stemming [[20]](#_bookmark26) is one of the most renowned Arabic stem- mers, aims to strip off a set of prefixes and suffixes. However, exist- ing light stemmers [[20,25,26]](#_bookmark26) are extracting root/stem by removing prefixes and suffixes from a word without deal with infix stemmer. In this paper, we developed a light stemming by adding extra prefixes and suffixes in addition to infixes in order to improve the effectiveness of information retrieval systems in Arabic. There- fore, the proposed method is capable to produce a high- performance tool to extract Arabic root/stems by removing stickers from a word included infixes to suffixes and prefixes.

* 1. *AMIR rules*

Our proposed approach is bottom-up and rule-based. First, it attempts to find substrings of words, which are mostly stems, or in other cases morphemes that can be derived from stems. The next process is to join each core with word elements, thus gener- ating words according to the governing rules. Finally, the rules check to ensure that each core allows for a correct generation thus resulting in the correct stem of the given word. AMIR rules are composed of three phases: substring tagging, rule matching and anti-rule matching, as shown in [Fig. 3](#_bookmark7) below.

Substring tagging: the morphological information that charac- terizes possible substrings of respective words is extracted. Based on the results, we can accurately determine which word substrings are morphemes. This phase is also instrumental in ensuring that clusters of each morpheme are extracted. The clusters are used in the rule matching phase. Rule matching: each core that has been extracted from the substring tagging phase is used to determine the rules employed in the extraction. Anti-rule matching: this is



**Stemming**

**Rule-Based Stemmer**

**Statistical Stemmer**

Fig. 2. Overview of the types of Arabic stemmer.

an essential phase to extract the required anti-rules from the anti-rules-based repository for every core in the given list. This ensures that every core with any anti matching rules with the word’s morphemes gets removed from the given core list. This last anti matching rule phase ensures that every core’s stem in the core’s list is indeed the correct word’s stem. AMIR Rules is con- structed from different Arabic grammatical rule-based according to morphological analysis. Therefore, these rules depend on a mod- ification of a word into an appropriate stem. So, selecting the rules depend on special letters adding/inserting to the root. [Table 2](#_bookmark8) shows intelligent use of morphological analysis and stem in Arabic Information Retrieval System using AMIR rules R, where T1 is the first letter of the original root, T2 is the second letter of the original root, and T3 is the third letter of the original root.

* 1. *AMIR algorithm*

This section discusses AMIR algorithm to find the root/stem that uses as index term in the field of Arabic information retrieval sys- tems. AMIR algorithm works as follows:

* + 1. *Tokenization & normalization*

Arabic tokenization has been implemented in several solutions to resolve ambiguous words. For instance, characters can be writ- ten in different ways, such as character (ﺀ) Hamza can be composed in different ways (ﺁ ,ﺇ ,ﺃ). This cause more ambiguous as to whether the Hamza is present. Therefore, at most one token is assigned to each letter at any one time as follows:

* + - * Replacing initial ﺇ ,ﺁ ,ﺃ by ﺍ
      * Replacing final ﻯ ,ﺉ by ﻱ.
      * Replacing final ﻩ by ﺓ.
    1. *Keyword extraction*

We represent AMIR steps to extract Keywords as follows:

Convert the user request text into words and put it into a list. Check the lists whether prepositions or stop-word are found. If found, remove any matched from the list

Search AMIR Dictionary to find given terms in the list; if a match found, then extract root/stem if accepted on AMIR rules. Else, if a match not found, do nothing.

Step1: Convert the user request text into words to create a word list by selecting the words that contain more than three letters.

Table 1

AMIR lemmatizer Example.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Word | Prefix | Infix | Suffix | AMIR Stemmer | Word Translate |
| ﺍﻟﻤﻜﺘﺐ | ﺍﻝ | – | – | ﻣﻜﺘﺐ | The office |
| ﻣﻜﺎﺗﺐ | – | ﺍ | – | ﻣﻜﺘﺐ | Offices |
| ﻣﻜﺘﺒﻜﻢ | – | – | ﻛﻢ | ﻣﻜﺘﺐ | Your office |

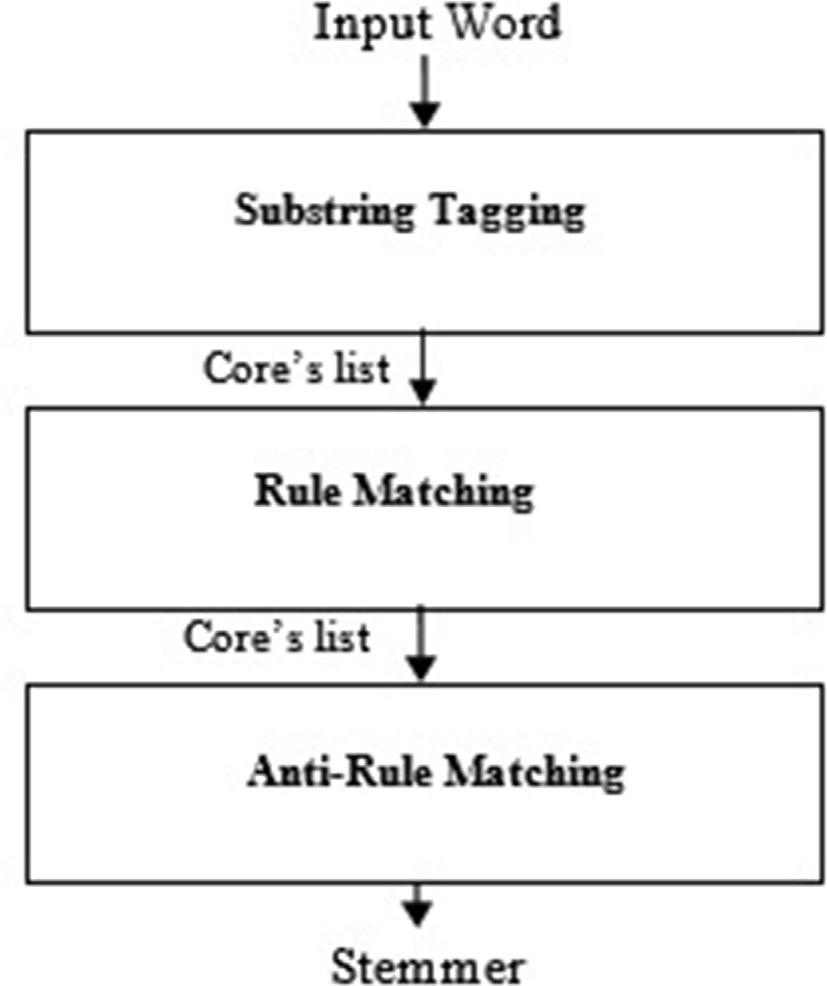


Fig. 3. AMIR rules steps.

Step 2: Check the created lists, if prepositions or stop-word found, if they found, then remove prepositions or stop-word from the list.

Step 3: Search in AMIR dictionary, if any match found in the given list, then extract root/stem based on AMIR rules, after that, use these as index term. For example; if we give the word ‘ﻭﻟﻤﺪﺭﺱ’ (And for a teacher) to AMIR dictionary which is consist of three prefixes ﻡ (m), ﻝ (for), and ﻭ (and). So based on AMIR rule 1, we will remove prefix ﻝ (for) which refer to preposition, and prefix ﻭ (and) which refer to stop-word. So, we will get ﻣﺪﺭﺱ (teacher) which using as index term.

Step 4: if a match not found in AMIR dictionary, then not do anything.

1. Experiments and results

This section aims to verify the effectiveness and the quality of AMIR performances with the relevance measures.

* 1. *Dataset*

In this paper, the experiments were carried out with EveTAR (2016) dataset on Arabic tweets, which cover different types of Arabic events detection. The EveTAR is essential evaluation tools in the field of information retrieval, which are comparable to sim- ilar Text Retrieval Evaluation Conference TREC. EveTAR dataset includes a crawl of 355 M which contained roughly 59,732 Arabic

Table 2

shows an intelligent use of morphological analysis and stem in Arabic Information Retrieval System using AMIR rules R.

Rule Syntax Description

R1 Prefix ﻡ (m) + Root ? Noun In Arabic, prefixes ﻡ (M) indicates to noun. Therefore, If we add prefix ﻡ (m) to the root, then it changes word

type to noun. For example, if adding prefix ﻡ (M) to the root ‘ﺩﺭﺱ‘ (lesson), we will get ‘ﻣﺪﺭﺱ’ (Teacher). Thus, we kept the prefix ﻡ (M) in derived words and we will removing any other extra prefixes. So, this rules tells that we replacing any inflection begins with ﻭﻛﺎﻟﻢ ,ﻛﺎﻟﻢ ,ﻓﻠﻠﻢ ,ﻭﻟﻠﻢ ,ﻟﻠﻢ ,ﻛﻢ ,ﻭﻡ ,ﻓﺎﻟﻢ ,ﻭﺍﻟﻢ ,ﻟﻢ ,ﺍﻟﻢ ,ﻭﻟﻢ by ﻡ (M), and we remove other prefixes if any found.

R2 Prefix ﻡ (m) + Root + Suffix ﺓ (taa) ? Noun Prefixes ﻡ (M) and suffixes ﺓ (taa) if they joined together, it will produce noun (always refer to places). For

example, if we adding prefix ﻡ (M) and suffix ﺓ (Taa) to the root ‘ﺩﺭﺱ‘ (lesson), we will get ‘ﻣﺪﺭﺳﺔ’ (school). Therefore, we will keep prefix ﻡ (M) as indicated in the rule (R1). Also we will keep suffix ﺓ (taa) as they are, and we remove other extra prefixes if any found.

R3 Prefix ﻡ (m) + T1 + Infix ﺍ (a) + T2 + T3 ? Plural Form

Infix ﺍ (a) refer to plural form, if it joined with prefix (m) in the same word. For example, ‘ﻣﻜﺎﺗﺐ‘ (Offices) is plural form, if we remove infix ﺍ (a), it will change to singular form. So, we will get ﻣﻜﺘﺐ (office). Therefore, we will keep the prefix ﻡ (M) and we will removing infix ﺍ (a) and any other extra prefixes if any found.

R4 T1 + T2 + Infix ﻭ (w) + T3 ? Plural Form Infix ﻭ (w) indicates to plural form when word does not including prefix ﻡ (M), such as: ‘ﺩﺭﻭﺱ‘ (lessons). Thus,

we changed to their associated singulars by removing infix ﻭ (w). For example, if we remove infix ﻭ (w) from ‘ﺩﺭﻭﺱ‘ (lessons), we will get ﺩﺭﺱ (lesson). Therefore, we will remove all affixes if any exist.

R5 T1 + Infix ﺍ (a) + T2 + T3 ? Noun This rule say If word included infix ﺍ (a) and does not including prefix ﻡ (M). it will refer to noun. For example

‘ﻛﺎﺗﺐ‘ (author), if we remove infix ﺍ (a) from ‘ﻛﺎﺗﺐ‘ (author), then, we will get ﻛﺘﺐ (Wrote). So, the word meaning have changed. Therefore, we kept infixes ﺍ (a) as they are, and we removed any other extra prefixes and suffixes.

R6 Prefix ﺕ (taa) + T1 + T2 + Infix ﻱ (y) + T3 ? Noun This type of derivational create a noun from relation between letters. Thus, if prefix ﺕ (taa) and infix ﻱ (y)

joined together in the same word, so this will indicates to noun. For example: ‘ﺗﺪﺭﻳﺲ‘ (Teaching). Therefore, we will keep prefix ﺕ (taa) and infix ﻱ (y), if they combinations in the same word and we will remove any other extra prefixes and suffixes if any found.

R7 T1 + T2 + Infix ﺍ (a) + T3 + suffix ﺓ (taa) | suffix ﺍﺕ

(at) | suffix ﺗﺎﻥ (tan)) ? Noun

R8 T1 + T2 + Infix ﺍ (a) + T3 + suffix ﻳﺔ (ya) ? Plural Form

This type of derivation called replacement (alibdaal - ﺍﻹﺑﺪﺍﻝ). Therefore, If word included infix ﺍ (a) and ends by suffix ﺍﺕ (at) or suffix ﺗﺎﻥ (tan) in the same word. Thus, we replace inflectional suffixes ﺍﺕ (at) or ﺗﺎﻥ (tan) by ﺓ (taa). So, we reduce them to their singular by replace inflectional suffixes ﺍﺕ (at) or ﺗﺎﻥ (tan) by ﺓ (taa). For example: ﺩﺭﺍﺳﺎﺕ (studies) or ﺩﺭﺍﺳﺘﺎﻥ (two studies) if we replace inflectional suffixes ﺍﺕ (at) or ﺗﺎﻥ (tan) by ﺓ (taa), then we will get ﺩﺭﺍﺳﺔ (study).

As indicated in rule (R7), replacement affixes can produce right formula of stem. Thus, we replace inflectional suffixes ﻳﺔ (ya) by ﻱ (y). This is because suffix ﺓ (taa) do not change the word meaning.

tweets represented in Unicode and encoded in UTF-8, and covers 50 significant events for which about 62 K tweets. We also evaluate our approach using Trec\_eval software which is available at: (<https://trec.nist.gov/trec_eval>). It uses different measures of infor- mation retrieval. In our work, we have used precision @ 10, preci- sion @ 20, and Mean Average Precision MAP as evaluation metrics see evaluation results section.

* 1. *Comparison of AMIR with LUCENE and FARASA algorithms*

In this section, we have compared AMIR stemmer with two counterpart systems: LUCENE and FARASA. [Table 3](#_bookmark9) shows the dif- ferent stemmers for AMIR, LUCENE, and FARASA, which is slightly different from each other. For Arabic Language, there exit a number of methods to extract infixes and suffixes to indicate a plural form of words. For example, the word ﻣﺴﺎﺟﺪ (Mosques) is shown in [Table 3](#_bookmark9) for query No 10. Which composed of the infix ﺍ (a) that indicates to plural, AMIR method is able to remove plural using infixes to generate singular forms by applying AMIR rule No 3. As such, AMIR system extracts the word ﻣﺴﺠﺪ (Mosque) instead of a ﻣﺴﺎﺟﺪ (Mosques) by removing the infix ﺍ (a). While both FARASA and LUCENE extract the same word ﻣﺴﺎﺟﺪ (Mosques) thus failing to generate the singular form. This is because both FARASA and

LUCENE do not handle plural using infixes. Another example, when a plural form is in the suffix; this type of derivation called replace- ment alibdaal - ﺍﻹﺑﺪﺍﻝ, which is not applied in previous studies such as LUCENE and FARASA; for example, the word ﻣﻜﺘﺒﺎﺕ (libraries), where AMIR extractor the word ﻣﻜﺘﺒﺔ (library) by replacing the suffix ﺍﺕ (at) by suffix ﺓ (taa) by applying AMIR rule No 7. While FARASA and LUCENE both extractor the word ﻣﻜﺘﺐ (office) by removed suffix ﺍﺕ (at); thus, they produce word that has different meaning. Therefore, the advantages of AMIR is that it provides highly accurate results into the linguistic knowledge by use mor- phology. The fact that this new scheme can dissect a plural word and then get the its singular form.

[Fig. 4](#_bookmark10) shows the steps of each search that requests/topics from the text collection. First, we denote sets of documents in the text collection as D1, D2... Dn. We denote sets of queries Q1, Q2.. . Qn and extract terms as T1, T2 ,Tn for each query. We also denote

the search methods as S1, S , Sn where a search method consists

of all processing stem for each query term and document term. Therefore, search method S can process a set of queries Q and pro- duce a ranked list of document D hits H for each query Q. we sum- marized as:

## (D; S; Q ) → H

Table 3

Summary of produce stemmer approaches.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Query | Actual Text | English Translation | AMIRStemmer | LUCENE stemmer | FARASA stemmer |
| 1 | ﻣﻘﺘﻞ ﺣﻮﺛﻴﻴﻦ ﻓﻲ ﺍﻧﻔﺠﺎﺭ ﻓﻲ ﺍﻟﻴﻤﻦ | Houthis killed in explosion in Yemen | ﻣﻘﺘﻞ ﺣﻮﺛﻲ ﺍﻧﻔﺠﺎﺭ ﻳﻤﻦ | ﻣﻘﺘﻞ ﺣﻮﺙ ﺍﻧﻔﺠﺎﺭ ﻳﻤﻦ | ﻣﻘﺘﻞ ﺣﻮﺛﻲ ﺍﻧﻔﺠﺎﺭ ﻳﻤﻦ |
| 2 | ﻟﻴﺘﻮﺍﻧﻴﺎ ﺗﺴﺘﺨﺪﻡ ﺍﻟﻴﻮﺭﻭ ﺑﺪﻝ ﺍﻟﻠﻴﺘﺎﺱ | Lithuania uses euro instead of litas | ﻟﻴﺘﻮﺍﻧﻴﺎ ﺗﺴﺘﺨﺪﻡ ﻳﻮﺭﻭ ﺑﺪﻝ ﻟﻴﺘﺎﺱ | ﻟﻴﺘﻮﺍﻧﻴﺎ ﺗﺴﺘﺨﺪﻡ ﻳﻮﺭﻭ ﺑﺪﻝ ﻟﻴﺘﺎﺱ | ﻟﻴﺘﻮﺍﻧﻴﺎ ﺗﺴﺘﺨﺪﻡ ﻳﻮﺭﻭ ﺑﺪﻝ ﻟﻴﺘﺎﺱ |
| 3 | ﻓﻠﺴﻄﻴﻦ ﺗﻄﻠﺐ ﺍﻻﻧﻀﻤﺎﻡ ﻟﻠﻤﺤﻜﻤﺔ ﺍﻟﺠﻨﺎﺋﻴﺔ  ﺍﻟﺪﻭﻟﻴﺔ | Palestine asks to join the International  Criminal Court | ﻓﻠﺴﻄﻴﻦ ﺗﻄﻠﺐ ﺍﻧﻀﻤﺎﻡ ﻣﺤﻜﻤﺔ ﺟﻨﺎﺋﻲ  ﺩﻭﻟﻲ | ﻓﻠﺴﻂ ﺍﻧﻀﻤﺎﻡ ﺍﻻﻧﻀﻤﺎﻡ ﻣﺤﻜﻢ ﺟﻨﺎﺉ  ﺩﻭﻝ | ﻓﻠﺴﻄﻴﻦ ﺗﻄﻠﺐ ﺍﻧﻀﻤﺎﻡ ﻣﺤﻜﻢ ﺟﻨﺎﺋﻲ  ﺩﻭﻟﻲ |
| 4 | ﺗﺤﺪﻳﺪ ﺍﻟﻤﺸﺘﺒﻪ ﺑﻬﻢ ﻓﻲ ﻫﺠﻮﻡ ﺷﺎﺭﻟﻲ ﺇﺑﺪﻭ | Identify suspects in Charlie Hebdo attack | ﺗﺤﺪﻳﺪ ﻣﺸﺘﺒﻪ ﻫﺠﻢ ﺷﺎﺭﻟﻲ ﺇﺑﺪﻭ | ﺗﺤﺪﻳﺪ ﻣﺸﺘﺐ ﻫﺠﻮﻡ ﺷﺎﺭﻝ ﺇﺑﺪﻭ | ﺗﺤﺪﻳﺪ ﻣﺸﺘﺒﻪ ﻫﺠﻮﻡ ﺷﺎﺭﻝ ﺇﺑﺪﻭ |
| 5 | ﺍﺧﺘﺮﺍﻕ ﻛﻮﺭﻳﺎ ﺍﻟﺸﻤﺎﻟﻴﺔ ﺣﺴﺎﺑﺎﺕ ﺳﻮﻧﻲ | Hacking Korean accounts | ﺍﺧﺘﺮﺍﻕ ﻛﻮﺭﻳﺎ ﺷﻤﺎﻟﻲ ﺣﺴﺎﺏ ﺳﻮﻧﻲ | ﺍﺧﺘﺮﺍﻕ ﻛﻮﺭﻳﺎ ﺷﻤﺎﻝ ﺣﺴﺎﺏ ﺳﻮﻥ | ﺍﺧﺘﺮﺍﻕ ﻛﻮﺭﻳﺎ ﺷﻤﺎﻟﻲ ﺣﺴﺎﺏ ﺳﻮﻧﻲ |
| 6 | ﺑﻨﺎﺀ ﺃﻭﻝ ﻛﻨﻴﺴﺔ ﻓﻲ ﺇﺳﻄﻨﺒﻮﻝ ﻗﺮﻥ | Construction of the first church in Istanbul | ﺑﻨﺎﺀ ﺍﻭﻝ ﻛﻨﻴﺲ ﺍﺳﻄﻨﺒﻮﻝ ﻗﺮﻥ | ﺑﻨﺎﺀ ﺃﻭﻝ ﻛﻨﻴﺲ ﺇﺳﻄﻨﺒﻮﻝ ﻗﺮﻥ | ﺑﻨﺎﺀ ﺃﻭﻝ ﻛﻨﻴﺲ ﺇﺳﻄﻨﺒﻮﻝ ﻗﺮﻥ |
| 7 | ﻫﺠﻮﻡ ﺣﺰﺏ ﺍﻟﻠﻪ ﻋﻠﻰ ﻣﺰﺍﺭﻉ ﺷﺒﻌﺎ | century  Hezbollah attack on Shebaa Farms | ﻫﺠﻢ ﺣﺰﺏ ﺍﻟﻠﻪ ﻣﺰﺭﻋﺔ ﺷﺒﻌﺎ | ﻫﺠﻮﻡ ﺣﺰﺏ ﺍﻟﻠﻪ ﻣﺰﺍﺭﻉ ﺷﺒﻌﺎ | ﻫﺠﻮﻡ ﺣﺰﺏ ﺍﻟﻠﻪ ﻣﺰﺍﺭﻉ ﺷﺒﻌﺎ |
| 8 | ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﺗﺨﻄﻒ ﺷﺒﺎﺏ ﻓﻲ ﻧﻴﺠﺮﻳﺎ | Boko haram kidnaps youths in Nigeria | ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﺗﺨﻄﻒ ﺷﺒﺎﺏ ﻧﻴﺠﺮﻳﺎ | ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﺗﺨﻄﻒ ﺷﺒﺎﺏ ﻧﻴﺠﺮﻳﺎ | ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﺗﺨﻄﻒ ﺷﺒﺎﺏ ﻧﻴﺠﺮﻳﺎ |
| 9 | ﺳﻴﻄﺮﺓ ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﻋﻠﻰ ﻗﺎﻋﺪﺓ ﻋﺴﻜﺮﻳﺔ ﻓﻲ | Bucco is banned on a military base in Nigeria | ﺳﻴﻄﺮ ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﻗﺎﻋﺪ ﻋﺴﻜﺮﻱ | ﺳﻴﻄﺮ ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﻗﺎﻋﺪ ﻋﺴﻜﺮ | ﺳﻴﻄﺮ ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﻗﺎﻋﺪ ﻋﺴﻜﺮﻱ |
|  | ﻧﻴﺠﻴﺮﻳﺎ |  | ﻧﻴﺠﻴﺮﻳﺎ | ﻧﻴﺠﻴﺮﻳﺎ | ﻧﻴﺠﻴﺮﻳﺎ |
| 10 | ﻫﺠﻤﺎﺕ ﻋﻠﻰ ﻣﺴﺎﺟﺪ ﻓﻲ ﻓﺮﻧﺴﺎ | Attacks on mosques in France | ﻫﺠﻢ ﻣﺴﺠﺪ ﻓﺮﻧﺴﺎ | ﻫﺠﻢ ﻣﺴﺎﺟﺪ ﻓﺮﻧﺴﺎ | ﻫﺠﻢ ﻣﺴﺎﺟﺪ ﻓﺮﻧﺴﺎ |

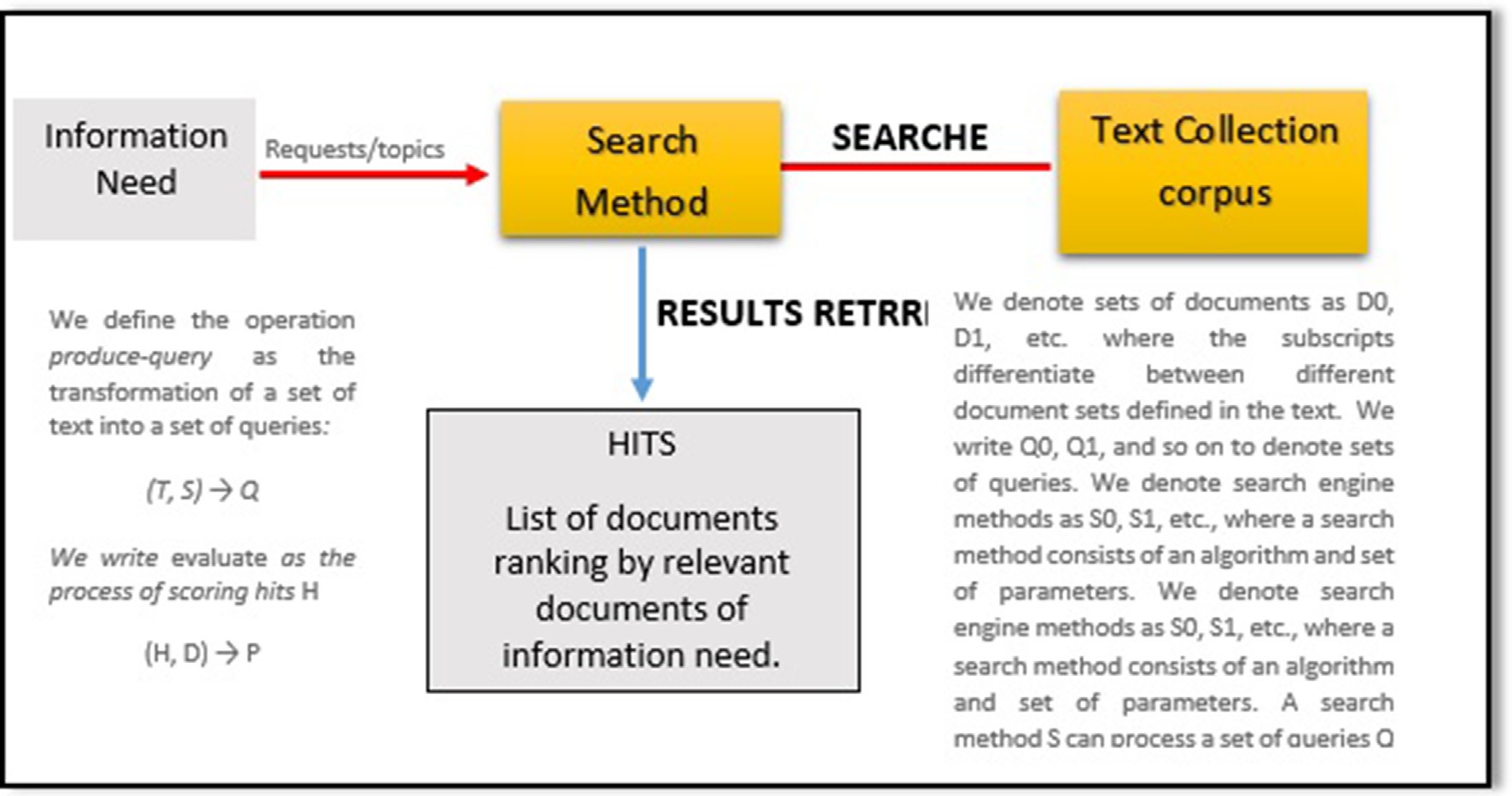


Fig. 4. Overview of the AMIR to produce information requests/topics.

Where the hits H of all the k documents in D appear in the k top ranks documents D.

[Fig. 4](#_bookmark10): Overview of the AMIR to produce information requests/topics.

* + 1. *Using statistical metrics*

We have employed TREC\_EVAL tool to measure precision @ 10, precision @ 20, and Mean Average Precision MAP as evaluation metrics. TREC\_EVAL uses two different files, quels file that is human-generated file that tells whether a retrieved document is relevant or not for each query, according to following format delimited by spaces:

## query — id 0 document — id relevance

where query-id is to identify the query, document-id is to iden-

tify the document, and relevance is to identify the judged docu- ment (0 for non-relevant and 1 for relevant). Second file is results file, which is contains a ranking of documents according to higher scores for each query. We have created results file by using Java language according to following format delimited by tab spaces:

## < query id >; < document No >; < rank >; < score >;

< system >

Where query-id is used to identify the query; document-id is used to identify the retrieved document; rank is used to identify the most relevant document; score is used to indicate the similar- ity value between document and query; system is used to identify system name. TREC\_EVAL is the executable program used to eval- uate rankings according to following format:

## trec eval [—q] [—a]*qrels* file Resultd file

where trec\_eval is the execute name, -q is a parameter for all detail

of queries, -a is a parameter for the summary output. [Fig. 5](#_bookmark12) shows the screenshot of results obtained for proposed method using tre- c\_eval to measure P@10, P@20, and MAP of retrieval experiments.

* + 1. *Using frequency metrics*

TF.IDF is a popular information retrieval technique, which weighs word’s frequency, abbreviated as TF and the term’s inverse document frequency commonly abbreviated as (IDF). In this paper,

we used TF.IDF to evaluate the quality of our scheme performances retrieval. Thus, we compared TF.IDF values of our scheme with LUCENE and FARASA for the first ten queries. It is clear from [Table 4](#_bookmark11) that the AMIR system has a great improvement as compared to LUCENE Stemmer and FARASA stemmer; this is a major improve- ment in previous methods.

[Fig. 6](#_bookmark16) shows the TF.IDF values of the first 10 queries for AMIR, LUCENE, and FARASA methods, where x-axis represents the query ID, and y-axis represents the tf:idf score scheme that is related to that query; thus, we develop a novel scheme that gives best tech- nique of affixes stemmer and the results obtained strongly indicate that the best TF.IDF values achieved when our scheme is used.

* 1. *Evaluation results*

In our experiments, the retrieval performance of the proposed method has been compared with the LUCENE, FARASA stemmers, and No stemmer using BM25 model and language model LM with Dirichlet technique in order to evaluate the quality of our scheme performances. Thus, the retrieved effectiveness was evaluated using Mean Average Precision MAP as the primary evaluation met- ric in addition to precision at 10 (P@10), and precision at 20 (P@20) in order to analyze the change in retrieval precision. [Tables 5](#_bookmark16) and [6](#_bookmark17) present our experimental results, where the bold values denote the best results in each category. Thus, in [Table5](#_bookmark16) we shows the results obtained for each system runs for 50 queries Therefore, AMIR achieved a MAP values by 0.34% while LUCENE, FARASA and no

Table 4

Summary of the results obtained from AMIR, LUCENE, and FARASA via TF.IDF values.

|  |  |  |  |
| --- | --- | --- | --- |
| Query Words | AMIR TF.IDF | LUCENE TF.IDF | FARASA TF.IDF |
| ﻭﻓﺎﺓ ﺃﺑﻮ ﺃﻧﺲ ﺍﻟﻠﻴﺒﻲ ﻧﻴﻮﻳﻮﺭﻙ | 1841 | 269 | 1641 |
| ﺍﺧﺘﺮﺍﻕ ﻛﻮﺭﻳﺎ ﺍﻟﺸﻤﺎﻟﻴﺔ ﺣﺴﺎﺑﺎﺕ ﺳﻮﻧﻲ | 1644 | 33 | 278 |
| ﺑﻨﺎﺀ ﺃﻭﻝ ﻛﻨﻴﺴﺔ ﻓﻲ ﺇﺳﻄﻨﺒﻮﻝ ﻗﺮﻥ | 1680 | 7 | 393 |
| ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﺗﺨﻄﻒ ﺷﺒﺎﺏ ﻓﻲ ﻧﻴﺠﺮﻳﺎ | 1883 | 70 | 427 |
| ﺳﻴﻄﺮﺓ ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﻋﻠﻰ ﻗﺎﻋﺪﺓ ﻋﺴﻜﺮﻳﺔ ﻓﻲ ﻧﻴﺠﻴﺮﻳﺎ | 1041 | 56 | 413 |
| ﻓﺮﺽ ﻟﺒﻨﺎﻥ ﺗﺄﺷﻴﺮﺓ ﺩﺧﻮﻝ ﻟﻠﺴﻮﺭﻳﻴﻦ | 1037 | 36 | 577 |
| ﻫﺠﻤﺎﺕ ﻋﻠﻰ ﻣﺴﺎﺟﺪ ﻓﻲ ﻓﺮﻧﺴﺎ | 1333 | 289 | 196 |
| ﺣﺮﻕ ﺑﻮﻛﻮ ﺣﺮﺍﻡ ﺑﻠﺪﺓ ﺑﺎﻏﺎ ﺍﻟﻨﻴﺠﻴﺮﻳﺔ | 862 | 49 | 333 |
| ﺗﻔﺠﻴﺮ ﺩﺍﻋﺶ ﻣﺴﺠﺪًﺍ ﻟﻠﺸﻴﻌﺔ ﻓﻲ ﺑﺎﻛﺴﺘﺎﻥ | 750 | 71 | 164 |
| ﺇﻋﺎﺩﺓ ﺗﺸﻜﻴﻞ ﻣﺠﻠﺲ ﺍﻟﻮﺯﺭﺍﺀ ﺍﻟﺴﻌﻮﺩﻱ | 655 | 34 | 175 |

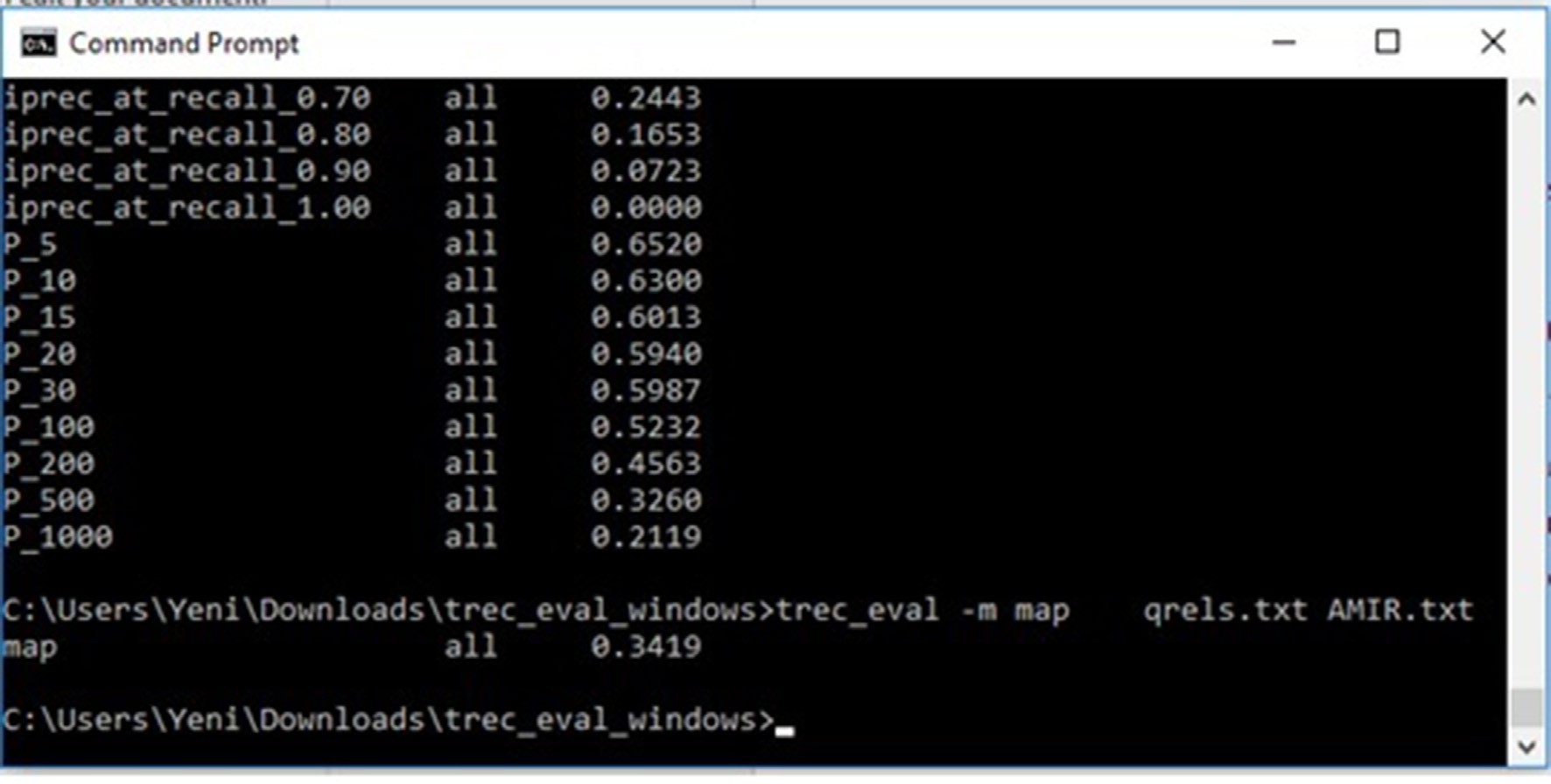


Fig. 5. Screenshot of AMIR results achieved using TREC\_EVAL to measure the MAP.

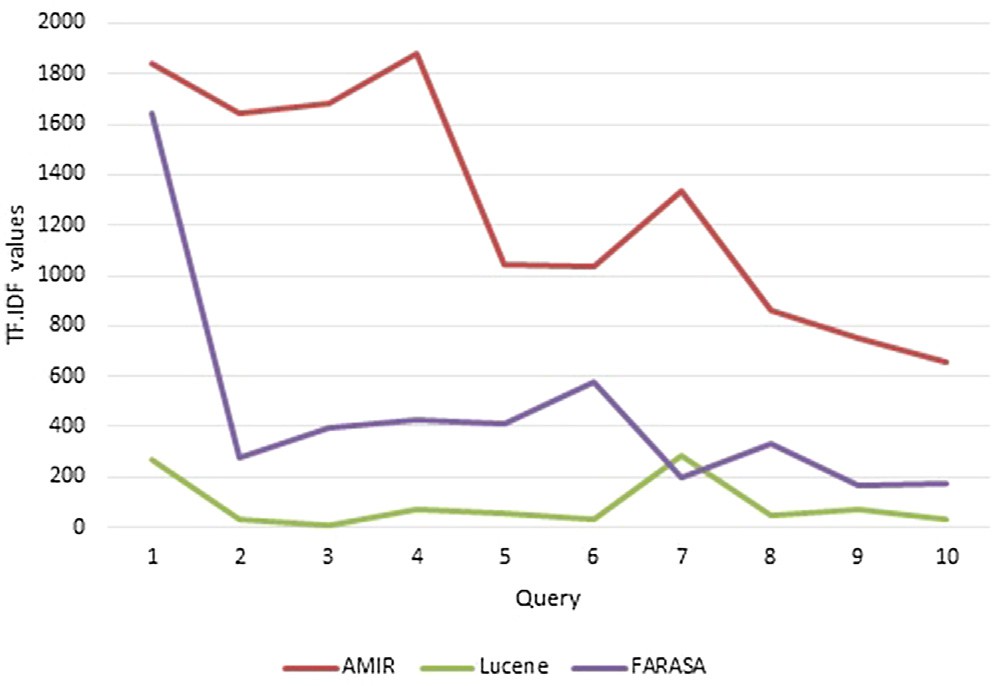


Fig. 6. The calculated TF.IDF values achieved by AMIR, LUCENE, and FARASA.

Table 5

Summary of the results obtained using BM25.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | BM25 |  | | |
| MAP | Prec@10 | Prec@20 |  |
| AMIR | 0.34 | 0.63 | 0.59 |  |
| LUCENE | 0.27 | 0.53 | 0.51 |  |
| FARASA | 0.28 | 0.62 | 0.57 |  |
| No stem | 0.21 | 0.45 | 0.46 |  |

Table 6

Summary of the results obtained using LM with Dirichlet smoothing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LM with Dirichlet | smoothing |  |  |
| MAP | Prec@10 | Prec@20 |
| AMIR | 0.32 | 0.60 | 0.56 |  |
| LUCENE | 0.25 | 0.47 | 0.44 |  |
| FARASA | 0.26 | 0.56 | 0.52 |  |
| No stem | 0.18 | 0.29 | 0.28 |  |

stemmer are 0.27%, 0.28% and by 0.21, respectively by using MB25 model. We also noticed that AMIR gives the best values of P@10 and P@20 by 0.63, and by 0.59, respectively. This indicates that using AMIR stemming yields a much improved precision. While AMIR achieved a MAP by 0.32% where LUCENE, FARASA, and no stemmer achieved a MAP by 0.25%, 0.26% and 0.18%, respectively, by using LM with Dirichlet smoothing model as shown in [Table 6](#_bookmark17). Therefore, we found that for long queries, the BM25 model per- forms better than the language model LM with Dirichlet smooth- ing. Nevertheless, for short queries, the LM with Dirichlet smoothing performs better than the BM25 model.

In addition, the Student *t*-test significance measure was used with p-values at or below 0.05 to claim significance in order to determine if the difference between the results was statistically significant or not. When the calculated p-value is below 0.05, it indicates that the difference between the two experimental run is statistically significant. Therefore, the results of the statistical tests show that the differences in MAP between the AMIR stemmer and LUCENE stemmer where p-value is 0.005508 produces results that are statistically significant according to p-value < 0.05. The dif- ference between AMIR and FARASA was not statistically significant by getting P-value as 0.094249 which is greater than P > 0.05. Lastly, The AMIR stemmer against No stemmer produces results that are statistically significantly by getting P-value as 0.006334. Thus, the results of the statistical tests show that AMIR gives statis- tically significant improvements. Therefore, the results presented

in [Table 5](#_bookmark16) and [Table 6](#_bookmark17) clearly indicate that the proposed method is able to solve successfully the research problems in high perfor- mance level. In addition, the best retrieval performance for Arabic information retrieval systems was AMIR method.

1. Conclusion and future work

The rationale behind this paper is to improve Arabic extraction of root/stem to build effective Arabic information retrieval sys- tems. The proposed method has shown to improve Arabic Stemmer and increases retrieval performances. In our experiment, we have compared AMIR against LUCENE, FARASA, and no-stem methods. The obtained results in terms of the mean average of precision have resulted in 0.34%, 0.27%, 0.28, and 0.21% for AMIR, LUCENE, FARASA and non-stem, respectively. This shows that our proposed AMIR stem algorithm outperforms others.

As future work, informal words such as ﺍﺗﻜﺘﺒﻮﻥ atakatabun (Do they write) needs more investigated and developed in order to apply them to information retrieval in Arabic language.

Acknowledgements

The authors gratefully acknowledge use of the services and facilities of the Ankara Yildirim Beyazit University.

Funding

This publication was made possible by the Libyan ministry of education. The statements made herein are solely the responsibil- ity of the authors.

Conflict of interest statement

There is no conflict of interest.

References

1. [Bomhard AR. Toward Proto-Nostratic: a new approach to the comparison of](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0005) [Proto-Indo-European and Proto-Afroasiatic, Vol. 27. John Benjamins](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0005) [Publishing; 1984](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0005).
2. [Beesley KR. Arabic finite-state morphological analysis and generation.](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0010) [Proceedings of the 16th conference on Computational linguistics. Association](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0010) [for Computational Linguistics; 1996](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0010).
3. [Al Ameed H et al. Arabic light stemmer: A new enhanced approach. The Second](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0015) [International Conference on Innovations in Information Technology (IIT’05),](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0015) [2005](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0015).
4. [Kanaan G et al. Building an effective rule-based light stemmer for Arabic](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0020) [language to improve search effectiveness. 2008 International Conference on](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0020) [Innovations in Information Technology. IEEE; 2008](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0020).
5. [Al-Kharashi IA, Evens MW. Comparing words, stems, and roots as index terms](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0025) [in an Arabic information retrieval system. J Am Soc Inform Sci 1994;45](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0025) [(8):548–60](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0025).
6. [Brent MR. Speech segmentation and word discovery: A computational](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0030) [perspective. Trends Cognitive Sci 1999;3(8):294–301](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0030).
7. [Darwish K, Hassan H, Emam O. Examining the effect of improved context](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0035) [sensitive morphology on Arabic information retrieval. Proceedings of the ACL](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0035) [Workshop on Computational Approaches to Semitic Languages. Association for](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0035) [Computational Linguistics; 2005](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0035).
8. [Larkey LS, Ballesteros L, Connell ME. Light stemming for Arabic information](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0040) [retrieval. In: Arabic computational morphology. Springer; 2007. p. 221–43](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0040).
9. [Eldesouki MI, Arafa WM, Darwish K. Stemming techniques of Arabic language:](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0045) [Comparative study from the information retrieval perspective. Egypt Comput J](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0045) [2009;36(1):30–49](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0045).
10. [Aljlayl M, Frieder O. On Arabic search: improving the retrieval effectiveness via](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0050) [a light stemming approach. Proceedings of the eleventh international](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0050) [conference on Information and knowledge management. ACM; 2002](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0050).
11. [El-Beltagy S, Rafea A. A framework for the rapid development of list based](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0055) [domain specific Arabic stemmers. Proceedings of the Second International](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0055) [Conference on Arabic Language Resources and Tools, 2009](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0055).
12. [Al-Shalabi R et al. Stemmer algorithm for Arabic words based on excessive](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0060) [letter locations. 2007 Innovations in Information Technologies (IIT). IEEE;](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0060) [2007](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0060).
13. [Paice CD. An evaluation method for stemming algorithms. SIGIR’94. Springer;](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0065) [1994](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0065).
14. [Bakeel,, Azman B. Root identification tool for Arabic verbs. IEEE Access](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0070) [2019;7:45866–71](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0070).
15. [Naili M, Chaibi AH, Ghezala HHB. Comparative study of arabic stemming](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0075) [algorithms for topic identification. Proc Comput Sci 2019;159:794–802](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0075).
16. [Carlberger J et al. Improving precision in information retrieval for Swedish](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0080) [using stemming. Proceedings of the 13th Nordic Conference of Computational](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0080) [Linguistics (NODALIDA 2001), 2001](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0080).
17. [Khoja S, Garside R. Stemming arabic text. Lancaster, UK: Computing](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0085) [Department, Lancaster University; 1999](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0085).
18. Porter M, An algorithm for suffix stripping. Program: electronic library & information systems; 1980.
19. Habash N, et al. Morphological analysis and disambiguation for dialectal Arabic. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2013.
20. [Larkey LS, Ballesteros L, Connell ME. Improving stemming for Arabic](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0100) [information retrieval: light stemming and co-occurrence analysis.](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0100)

[Proceedings of the 25th annual international ACM SIGIR conference on](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0100) [Research and development in information retrieval. ACM; 2002](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0100).

1. [Larkey LS, Connell ME. Structured queries, language modeling, and relevance](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0105) [modeling in cross-language information retrieval. Inform Process Manage](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0105) [2005;41(3):457–73](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0105).
2. [Khoja S. APT: Arabic part-of-speech tagger. Proceedings of the Student](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0110) [Workshop at NAACL, 2001](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0110).
3. [Darwish K, Abdelali A, et al. Farasa: A fast and furious segmenter for Arabic.](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0115) [Proceedings of the 2016 conference of the North American chapter of the](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0115) [association for computational linguistics: Demonstrations, 2016](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0115).
4. [Al-Saqqa S, Awajan A, Ghoul S. Stemming effects on sentiment analysis using](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0120) [large arabic multi-domain resources. 2019 sixth international conference on](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0120) [social networks analysis, management and security (SNAMS). IEEE; 2019](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0120).
5. [Mustafa M et al. Developing two different novel techniques for Arabic text](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0125) [stemming. Intell Inform Manage 2019](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0125).
6. [Atwan J, Wedyan M, Al-Zoubi H. Arabic Text Light Stemmer. Int J Comput](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0130) [2019;8(2):17–23](http://refhub.elsevier.com/S1110-8665(19)30346-9/h0130).