ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AN INTELLIGENT USE OF STEMMER AND MORPHOLOGY ANALYSIS FOR ARABIC INFORMATION RETRIEVAL

Ph.D. Thesis by

Ali Abrahem Ali ALNAIED

Department of Electrical and Computer Engineering

April, 2020

ANKARA

AN INTELLIGENT USE OF STEMMER AND MORPHOLOGY ANALYSIS FOR ARABIC INFORMATION RETRIEVAL

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Electrical and Electronics Engineering, Department of Electrical and Computer Engineering

by

Ali Abrahem Ali ALNAIED

April, 2020

ANKARA

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"AN INTELLIGENT USE OF STEMMER AND MORPHOLOGY ANALYSIS FOR ARABIC INFORMATION RETRIEVAL"** completed by **ALI ABRAHEM ALI ALNAIED** under the supervision of **ASST. PROF. DR. M.ABDULLAH BÜLBÜL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Ph.D.

Asst. Prof. Dr. M.Abdullah BÜLBÜL

Supervisor

Assoc. Prof. Dr. Osman DÜZGÜN

Jury Member

Asst. Prof. Dr. Fahreddin Şükrü TORUN

Jury Member

Asst. Prof. Dr. Ali Osman ÇIBIKDİKEN

Jury Member

Asst. Prof. Dr. Cevat RAHEBİ

Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: **Signature:**

Name & Surname: Ali Abrahem Ali ALNAIED

ACKNOWLEDGMENTS

Praise is to Allah, the Almighty for having led me at every point of my biography. And best pray and peace on Prophet **Mohammed**.

I would also like to take this chance to express my sincere gratitude to my supervisor, Asst. Prof. Dr. **M. Abdullah BÜLBÜL**, for his constant support, sharing the knowledge, mentoring, and his precious recommendations that assisted me all the time of my research and while writing this thesis. Thanks to you.

I should also thank you for Asst. Prof. Dr. **Mucahid KUTLU** for his support and suggestions. This support means a lot to me.

As well I would like to take this opportunity to thank the discussion committee, Assoc. Prof. Dr. **Osman DÜZGÜN,** Asst. Prof. Dr. **Özkan KILIÇ,** Asst. Prof. Dr. **Şükrü TORUN**, and Prof. Dr. **Mehmet Hakkı** SUÇİN for their advice and your brilliant comments and suggestions, during my thesis.

Special acknowledgment is given to the examiners of the thesis, Asst. Prof. Dr. **Cevat RAHEBİ** and Asst. Prof. Dr. **Ali Osman ÇIBIKDİKEN** for agreeing to be the examiners of my PhD viva.

A special thanks to my family. Words cannot express how grateful I am to my mother, and father for all of the sacrifices that you've made on my behalf. Your prayers for me were what sustained me thus far. I would also like to very special thanks to my loyal wife for her supporting me, especially I can't thank you enough. To my children for their encouragement and the understanding and patience, they have shown during my period of study, giving me the strength to keep going.

Last, but not least, I thank my personal friends for their unconditional support and encouragement.

2020, March **Ali Abrahem Ali ALNAIED**

AN INTELLIGENT USE OF STEMMER AND MORPHOLOGY ANALYSIS IN ARABIC INFORMATION RETRIEVAL

ABSTRACT

In the past several years, the Arabic information retrieval has garnered significant attention due to increasing the Arabic text on the web. A considerable number of researchers share similar opinions on the benefits of morphology and stemming in Arabic information retrieval systems, especially for internet search engines; a problem exacerbated by the enormous amounts of datasets on the internet. The Arabic language is ranked as the seventh top language on the web. It is the highest growth of the ten top online languages. Therefore, the number of Arabic documents increases rapidly. Also, the Arabic language has a serious challenge due to the complexity of its alphabet morphological. In NLP tasks it becomes hard to select an effective index term of information retrieval systems. Thus, indexing terms is a complex and difficult process, especially when it concerns the indexing of Arabic documents. Year after year, many methods are being published to overcome the Arabic stem problem for successful retrieval of documents. Therefore, this research present a novel method to extracting an Arabic stem called Arabic Morphology Information Retrieval (AMIR. The main goal and advantage of our method are to generate/extract stem by applying a set of rules and matches the relationship between some Arabic letters to find the root/stem of the respective words to use as indexing terms for the text searching in Arabic retrieval systems. Furthermore, we highlight the use of these rules and their benefits for different Arabic information retrieval systems. Consequently. AMIR can be considered to operate around minimum morphological complexity. Finally, AMIR has been tested using the EveTAR (2016) dataset on Arabic tweets and the obtained results show that the AMIR results outperform the state-of-the-art results. Therefore, our approach has been able to improve the performance of Arabic stem and increases retrieval as well as being active against any type of stem and we believe that it's difficult to develop a new Arabic system retrieval method without uses a good morphology analysis support it.

Keywords: Natural language processing, Arabic information retrieval systems, Arabic morphology, light stemming algorithm, Arabic language, text analysis, Arabic stemming algorithms, Arabic speech search, rule-based stemming, indexing.

ARAPÇA BİLGİ ERİŞİMİNDE KÖK VE BİÇİM ANALİZİNİN AKILLI BİR KULLANIMI

ÖZ

Son yıllarda, internet üzerindeki Arapça metinlerin çoğalmasıyla, Arapça bilgi erişimi önemli derecede ilgi görmektedir. Çok sayıda araştırmacı Arapça bilgi erişim sistemlerinde, özellikle de internetteki çok büyük miktarda veri yığını oluşmasıyla daha da karmaşık bir probleme dönüşen internet arama motorlarında, biçimbilim ve köke indirgemenin yararlarına dair benzer görüşleri paylaşmaktadırlar. Arapça dili, internetteki diller arasında yedinci sıradadır ve ilk on çevrimiçi dil arasında en hızlı yükselişi göstermektedir. Arapça belgelerin sayısı da hızla artmaktadır. Arapça dilinin, alfabe biçiminin kompleks olması nedeniyle de ayrı bir zorluğu vardır. NLP çalışmalarında, bilgi erişim sistemleri için etkin bir dizin terimi bulmak oldukça zordur. Dolayısıyla terimlerin dizinlenmesi, özellikle mesele Arapça belgelerin dizinlenmesi olduğunda daha da karmaşık ve zor bir süreç haline gelmektedir. Bu araştırma, Arapça köke indirgeme üzerine "Arapça Biçimbilim Bilgi Erişimi (AMIR)" adlı yeni bir yöntem sunmaktadır. Yöntemimizin temel amacı ve avantajı, Arapça bilgi erişiminde metin aramada, dizinleme terimleri olarak kullanmak amacıyla, bir dizi kural uygulayarak ve bazı Arapça harfler arasındaki ilişkiyi eşleştirerek kök oluşturmak/bulmaktır. Ayrıca, bu kuralların kullanımına ve farklı Arapça bilgi erişim sistemleri için faydalarına da dikkat çekiyoruz. AMIR'in minimum biçimsel karmaşıklık düzeylerinde etkili olarak çalıştığı belirtilebilir. Sonuç olarak, AMIR, Arapça tweetler üzerinde, EveTAR (2016) veriseti kullanılarak test edilmiştir ve elde edilen sonuçlar, AMIR'in aynı fonksiyonu gören literatürdeki en güncel araçlardan daha iyi performans sergilediğini göstermiştir. Dolayısıyla bizim yaklaşımımız, her türden köke karşı duyarlı olma kapasitesine sahip olduğu gibi, Arapça kök indirgeme ve bilgi erişiminde performansı daha da iyi bir noktaya taşıyabilecektir ve biz inanıyoruz ki iyi bir destekleyici biçimsel analiz kullanılmadan, yeni bir Arapça bilgi erişim sistemi geliştirmek zor olacaktır.

Anahtar Kelimeler: Doğal dil İşleme, Arapça bilgi erişim sistemleri, Arapça biçimbilim, hafif kök indirgeme, Arapça dili, metin analizi, Arapça kök bulma algoritmaları, Arapça konuşma araması, kural tabanlı köke indirgeme, dizinleme.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

TERMINOLOGY

Natural Language Processing NLP uses the rules of native languages to examine the content and meaning of text.

Morphology is a branch of linguistic studies that deals with the internal structure of words.

Morpheme is the smallest meaning-bearing unit in language.

Inflection is the alteration of a word to express different grammatical categories.

Derivation is the process of forming a new word on the basis of an existing word.

Stem is the part of the word that is common to all of its inflected variants.

- **Prefixes** coming before the stem.
- **Suffixes** coming after the stem.
- **Infixes** inserted into the stem.

Lexical meaning of a base or root word without considering any prefix or suffix.

Dual: Arabic morphological designating two subjects.

Feminine: Arabic morphological designating female subject.

Masculine: Arabic morphological designating male subject.

Root non affix lexical content morpheme, which cannot be analyzed into smaller parts.

Corpus is a structured collection of text covering a large number of words from different domains of a given language.

Stop-words are words that have little semantic meaning that are removed from the index and the query.

Precision the number of correct documents retrieved as the result of q query or search divided by the number of documents retrieved.

Ranking referring to the position of a document on the search results for a particular query.

Recall the number of documents retrieved containing correct content as the result of a corpus divided by the possible number of documents in the corpus containing correct

Part Of Speech POS: which is a feature of the word that indicating whether it is a verb, a noun, or a particle.

Lemmatizer: This algorithm collects all inflected forms of a word to break them down to their root dictionary form or lemma.

Linguistics is the scientific study of language. It involves analysing language form, language meaning, and language in context.

Information Retrieval (IR) is the task of ranking a list of documents or search results in response to a query.

Keywords are search terms that that define what your content is about.

Indexing term is used to convert queries into representation or data structure to improve the efficiency of retrieval.

Lemma is a word that stands at the head of a definition in a dictionary.

Inverted index is a database index storing a mapping from content, such as words to its locations in a document or a set of documents.

Rule-based is a system that applies human-made rules to store, sort, and manipulate knowledge to interpret information in a useful way.

Word is a unit of language, consisting of one or more spoken sounds or their written representation.

CHAPTER 1

INTRODUCTION

Chapter Overview

Every day, the internet offers a huge volume of data to the users of the internet needs, and many users on the internet need to retrieve documents by using only a few words, or a query to fetch all information or documents that relevant to their search query. However, the quality of the search depends on the query's words; if query words were not precise enough, it may influence the search ability to retrieve the correct documents. Therefore, to ensure a satisfying level of precision for Arabic information retrieval systems it requires decomposed words in the query into meaningful components before submitted to the retrieval system. Stemming defines as transferring the modified words to their origin instead of their current status. So, this process depends on the stemmer and morphology. Thus, the advantage of using stemmer and morphology in information retrieval systems is to conflate query terms into indexing terms. Therefore, the indexing terms are one of the most challenging morphologies in natural language processing, especially in the Arabic language. It can retrieve successfully user information needs and increasing precision. Consequently, in this theses, good progress has been made in retrieving Arabic information based on morphology analysis which is an efficient way to improve retrieval systems. Arabic stemmer processing still needs more research to offer any contribution that a larger framework such as information retrieval.

This chapter provide a comprehensive introduction to research and the research objectives which will be set as follows: In section 1.1 the introduction, section 1.2 then provides a brief explanation of the research objectives, and then section 1.3 goes to features of the proposed approach, finally, in section 1.4 provide a brief overview of the related work.

1.1 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 the user's specifications. This return satisfies the user's needs. The Arabic language has different semantic and phonetic structures when compared to other languages [1]. This difference has also posed a significant issue as to how Arabic users benefit from search engine optimization.

Recently, the 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]. 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 are composed of three letters, orthographic variations, sophisticated and very rich in morphology. Indeed, the 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]. 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, Arabic language verbs and nouns are derived from a set of 10,000 roots [5].

Many researchers have investigated the impact of stem and morphology on the information retrieval process over many years [6, 7, 8]. However, the Arabic language is rich in morphology which is considered the most important factor to improve retrieval effectiveness. The stem is a technique used for reducing the grammatical form of a word based on inflection and derivation. The work suggested 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. Al-Stem which was later on modified by the University of Massachusetts. The Al-Stem Stemmer was further modified by David Graff whereby (ل, وي, لل, ال, فم, كم, وم, بم, نت, ست, ت, مت, ات, بت, بال, فال, وال, يا, فا, وا) can be removed from the word's prefixes and suffixes [9]. The study [10], classified as a light stemmer. This stemmer was developed to improve retrieve query searches. The

author factored the length of the words to be used for removing affixes and suffices. Additionally, he normalized some specific Arabic characters [11]. Also, the stemmer's words were blind, however; a robust and efficient Arabic Stemmer Algorithm can decrease data storage and computational time [12].

Basically, the essential component of a word is its stem; for instance, Arabic stems are different as compared to other languages like English. Arabic nouns can take the form of being plural, singular, dual, gender; feminine or masculine, and verbs can be present, past, future, and command verb. In contrast, stemming refers to a computational technique used to reduce words to their respective stems or roots. One disadvantage of existing Arabic stemmers is that they exhibit and are prone to immense stemming error-rates [13].

The stem should not be mistaken for the word lemma. Stem not mandatory for the stem to be characterized by actual words. However, the stem can allude to the morphological variant’s least common denominator. The lemmatizer is the process of using morphological and vocabulary to analyze words to execute things properly (De Roeck & Al-Fares, 2000). Also, lemmatizer is aimed at only removing inflectional endings (Alpkocak, 2012). Treatment of complex operations is known as segmentation of the word-formations into morphemes. Both lemmatizer and stemmer are essential when developing an efficient and good performing of the Arabic information retrieval system. The main reason behind this is that the stemmer can leverage on ad-hoc prefix and suffix stripping rules together with exception lists (Goldsmith, 2000). However, the lemmatizer must carry out a complete and detailed morphological analysis, which inherently based on a dictionary and grammatical rules. One of the most effective approaches in finding Arabic stems is by correctly performing morphological analysis, under a similar stem if the stemmer deducts many words in its bid to stem results, then semantically or morphologically unrelated words complement each other in a similar stem.

The Arabic language is characterized by complex and rich morphologies, which critically study and analyze the internal structures of words dealing with two morphemes: affixes and stems. Consequently, it is a requirement that

comprehensive and elaborate morphological analyzer ought to be used for the Arabic retrieval system to be effective (El-Beltagy & Rafea, 2009). One of the most important problems in a search system is to find a similar article for a newly submitted query. So, there are different techniques are proposed before in the literature that tried to solve this problem to enable the user to quickly obtain an answer to his request.

As were mentioned early, Arabic poses many challenges for retrieval performance and most of these challenges due to orthography and morphology. Therefore, several early works suggested to improving the retrieval performance for Arabic texts depends on stemmer, morphology, and lemmatizer. However, most of these works are extracted root or stem without distinguishing whether the removed stemmer is core letters of the root or not. Thus, the simplest way to develop Arabic information retrieval is to analyze Arabic morphology. Hence, Arabic morphology is considered an essential stage in natural language processing applications such as the search engine. That is because Arabic morphology can produce different word formation, including affixation behavior, roots, and pattern. Therefore. In this thesis, we attempt to perform a better retrieval system for Arabic text according to Arabic morphology structure (whether the word is noun, verb, adverb, and objectives), i.e., it analyzes to extract the best word formula uses as indexing term that can answer user's needs.

In this thesis, our goal is to evaluate the performance of a new method proposed uses to enhance the performance of Arabic information retrieval. A good analysis of Arabic morphology and stemmer has been made to improve search ability to retrieve the correct documents. So, one of the main advantages of this work is to produce a high-performance level stemmer and morphology by removal via implementing morphological analysis. As well as over the last few years, so many researches have been carried out for Arabic information retrieval problems, but still many problems such as plurals in infix and conjunction words unsolved, which can always be challenging and confusing, especially for Arabic language retrieval, in this work, we presented AMIR rules trying to touch upon the major components of Arabic lexicon its: roots, patterns, and affixes including prefixes, and suffixes

in addition to infixes that based on morphology analysis, to improve the retrieval systems of the Arabic documents. So, the complexity of the Arabic morphology means that many inflectional and derivational operations that need more effective works if we compared with other language rules.

Thesis contribution may be briefly described as follows:

- The comprehensive processing of Arabic texts to improve root extraction. Existing schemes extract the roots by removing affixes from a word without distinguishing whether the removed letters are core letters of the root or not, like study [14, 15]. This is because, in the Arabic language it is not easy to determine the conjunctions of pronouns, gender, plural, prepositions, stop-words, 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 thesis aims to address by proposing a method to validate whether the removed letters are core letters of the root. Therefore, in this thesis, we will attempt to extract the Arabic root/stem based on a validation of the letters before removing affixes by building an AMIR dictionary that generates over 1,400 words from each root. Therefore, the method proposes a root extraction based on morphology features by matching the word with all possible affixes and patterns attached to it. To the best of our knowledge, a single root can generate 1000 words using previous studies. Thus, our method increases the efficacy of extracting a word effectively while minimizing the ambiguity since it depends on validation 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 (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 the noun; prefix ل (l) is an inflectional morpheme that refers to prepositions; prefix و (w) is an inflectional morpheme that refers to stop-word; infix ا (a) is an inflectional morpheme that refers to the plural form, and suffix كن (kun) is an inflectional

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

- The second contribution relates to improved precision in Arabic information retrieval using an infix stemmer. In English, affixes can generally be divided into two groups: (prefixes and suffixes). While, in the Arabic language, affixes can be divided into three groups: (prefixes, infixes, and suffixes). Therefore, existing schemes are unable to extract the 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 morphological structure of the language. Also, extracting the root of a word in its plural form can always be challenging and confusing, 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). The result is a change in the meaning of the word. According to AMIR rules No 5, this case is not permitted. Using the work [16] stemming can give better precision in information retrieval. 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 English. As mentioned earlier, the English language uses suffixes and prefixes to determine the plurality of a word. Consequently, an infix is a very important factor that can improve Arabic retrieval systems. Therefore, we proposed is capable to solve problems of the plural form while still allowing the extraction of stem/root of Arabic words thus resulting in increased.

1.2 Research objectives

The main goal and advantage of this work are to improve the stemming issue in Arabic information retrieval systems by drawn the relationship between the root and the patterns in the Arabic language and how they are used to produce new words using as indexing term for the text searching in Arabic retrieval systems. Therefore, in this work, we have described Arabic stemmer and morphology analysis, which considered the most important factors uses to develop the fields of Arabic information systems, as detailed in chapter 3 and chapter 4, which talks about the AMIR system, in this chapter has drawn the relationship between the roots and the patterns to generate Arabic stem by inserting the root's radicals into their respective slots on the pattern template. This can be done by applying a set of Arabic morphological rules that use to match the relationship between some Arabic letters to produce a new word using as indexing term. Then, we highlight the use of these rules and their benefits for different Arabic information retrieval systems. Therefore, one of the advantages of the AMIR system is a validation of the letters before removing affixes by applying a set of rules regarding the relationship among Arabic letters to find the root or stem of the word.

The Objectives being addressed in this thesis can be defined as follows:

- To investigate the effectiveness of the morphological analysis on Arabic information retrieval performance.
- To analyze the morphology, which made different formulations to the word based on determining the correct word according to morphology analysis.
- Design a new approach to the analysis of Arabic morphological to build different possible word forms that used as a keyword in application search systems to support Arabic information retrieval.
- Handle infix stemmer, where the Arabic stemmer can be divided into three main types: prefixes, infixes, and suffixes. This can helps to solve problems of the plural that attached in infix.
- Compare the retrieval performance of this thesis approach with Arabic morphology algorithms that are available in the literature.

1.3 Features of the Proposed approach

In this thesis, our method aims to build a new approach of the Arabic language uses to solve problems that Arabic information retrieval face by study deeper for the most important factors that use in related work that have an impact on Arabic information retrieval systems such as stemmer, light stemmer, morphology, and lemmatizer[1] with minimum sufficient resources for information retrieval purposes. Our proposed method transforms inflected word form to its lemma (This means that produce different word forms with the same meaning). For nouns, plural, verbs, adverb, and adjectives; and for an adverb, it is the perfective is the singular indefinite (masculine if possible) form.

To implement our system, the following features are considered:

- The proposed method gives the benefits of power to Arabic stem and gives an increase in precision by using an infix stemmer.
- Enhance Arabic root extraction algorithms by Constructor rule-based dictionary (AMIR) that generate over 1,400 words from each root, this dictionary is used to enhance the performance of the retrieval systems in the Arabic based on morphological transformations related to query word affixes.
- The proposed method is capable to improve the extraction root in the Arabic language base on several specific linguistic rules to generate/extract Arabic root.
- Relying on only a stemmer may lead to a much ambiguous word. Therefore, the proposed method removed affixes from a word based on morphology features. For example, the proposed method uses Arabic morphology to extract patterns to identify the current word.
- The proposed method increases the efficacy of extracting a word effectively while minimizing the ambiguity since it depends on validation before removing affixes.

[1] **Lemmatizer** usually refers to doing things properly with the use of vocabulary and morphological analysis of words,

- Since Arabic morphology is more complicated than English, the proposed method able to remove the plural attached in infix to get singular (infix stemmer is not used in the English language).
- Proposed method constructing Arabic POS tagger that's using to find the stems of Arabic words.
- In the proposed method, morphology rules are adopted to reduce the Arabic word into its abstract lemma form. This means that our method captures all semantic features of the word to extract candidate word uses as an indexing term in information retrieval systems.
- The proposed method identify affixes including prefix, infix, and suffix to find out the corresponding word suitable for Arabic information retrieval at the lemma level.

1.4 Related work

In recent decades, many works have been done to developed Arabic information retrieval systems such as [17, 18, 19]. But still, some weaknesses and problems facing retrieval systems to the most sufficient and reliable information. Arabic retrieval systems rely immensely on morphological and stemming operates but only very few of the studies have used lemmatizer. In this section, focuses more on what has been achieved on stemming and morphology in previous works.

Khoja's stemmer previously showed the first attempt to find the Arabic root by the removal of prefixes and suffixes. According to the studies [20, 21]. The Khoja stemmer produced an effective root-extracting stemmer. The Khoja stemmer subjective to the follows steps:

- Remove diacritics.
- Remove conjunction (punctuation, numbers, stopwords)
- Remove definite ال (the) and Remove conjunction و (and).
- Remove suffixes and Remove prefixes.
- Check results against a list of patterns. If a match found, then extract the characters in the pattern representing the root.

- Extracted root against a list of known “valid” roots.
- Replace letters ا (a), letters وي (ya) by letters و (y) and letters of Hamza ئ (y) and ؤ by أ
- Checked roots to see if they should contain a double character, if yes, then we added the character to the root.

However, the Khoja stemmer still weaknesses the root like the word مباريات (the Matches) which Khoja produce the root برا (a word did not have meaning). The researcher (porter, 1980) developed the Porter tailored for the English language. This stemmer leverages on two-step rewriting rules. This is achieved by removing approximately 60 different suffixes (Al-Fedaghi & Al-Anzi, 1989). 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 stemmer, and it was later used by several researchers such as [22].

(Larkey, Ballesteros, & Connell, 2007) Show better retrieval efficiency, among described in light stemming; it merely removes prefixes and suffixes depends on a listed in a predefined. However, it does not guarantee the production of better results when evaluating experiments, as proposed by (Aitao & Fredric, 2002) and (Ahmed, Emam, Essam, Nabil, and Hassan. 2019) it is common to use word segmentation for highly agglutinative languages like Arabic, or highly compounding.

In existence are numerous root extraction techniques for Arabic known as heavy stemming or stemming based root words, works by removing all prefix, infix, and suffix to improved Arabic information retrieval performance, shown by (Al-Shalabi, 1996).

(Arfath, Mohamed, Diab; Ahmed, Eskander, 2013) are one of the best and most techniques for Arabic and Arabic dialect processing tool designed for morphological analysis in a context that combines different aspects used systems for Arabic processing, they apply language, support vector machine (SVM) and models in the predictions of word tags based on feature modelling component.

(Kareem, Abdelali, 2016) proposed Farasa which is a new method using to extract Arabic word segmentations. This proposed method is more efficient in terms of the query time used to produces word segmentations; however, this technique cannot

handle any infixes segmentations. However, in this thesis, our system leverages some FARASA components but has its own rules that allow handling problems of infixes, also to improve generates prefixes and suffixes.

(BAKEEL, A, 2019) which proposed a new model for identifying the verb root produced in a tool "RootIT" by root extraction without disambiguation out of traditional methods. Numerous Arabic morphology systems have been devoted toward morphed requirements of words; Thus, these proposed method removes the prefixes and suffixes without using any linguistic rules. Therefore, in this thesis, our system has developed a root extraction technique that gives support to natural language processing to include morphology features by matching the word with all possible affixes and patterns attached to it; for example, the word المسلمون (the Muslims), where we can see the concatenation of morphemes to form a word. To distinguish between these morphemes, we say that سلم (lesson) is the root morpheme; prefix م (m) is a derivational morpheme where it refers to the noun; prefix ال (the) is an inflectional morpheme that refers to definition; suffix ون (wn) is an inflectional morpheme indicating the plural.

According to (Al-Saqqa, S., A. Awajan, and S. Ghoul. 2019) the most commonly used stemmers in the Arabic language are light stemmer and (Khoja, 1999). In the work (Mustafa, Mustafa, Aldeen, E. Ahmed, 2019) the authors of these papers proposed different stemming techniques based on light stemming by utilizing extra suffixes in the total number of prefixes and suffixes to be removed; thus, these studies were unable to extract stem/root of words that contain Infixes. In Arabic morphology, many words that have infixes in it and removing an affix depends on the morphological structure of the language. The truncation of infix from Arabic words is very important for an effective Arabic stemmer.

According to work (Kreaa, Ahmad, and Kabalan. 2014), several researchers have been developed the lights stemming over the past decade, particularly light-10, for example (Jaffar, Mohd, and Kanaan, 2003). This is because the lights stemming is the most commonly used stemmers in the Arabic language. Stemmers can generally be divided into two groups: first, light stemmers which provided by (Larkey, L.S., L. Ballesteros, and M.E, 2007). The second stem proposed in (Khoja, S., 2001). On the other hand,

there are several comparisons have been made between the Khoja stemmer and Light stemmer in different research fields such as topic classification [23], the similarity between words [24], and information retrieval. In these related works, the performance of the Light stemmer outperforms the Khoja stemmer.

Recently, many studies focused on extraction lemma from a word which is a new topic for the Arabic language, for example, the work [25] proposed a hybrid method for the extraction of lemmas. Similarly, the study [26] presented a system based on linguistic rules, where the system begins by searching for the stem of the word. It uses the pattern to identify the POS tag, then, exploits this information to extract the lemma from the word.

Finally. Lexical recognition tests are established for English (Lemh¨ofer and Broersma. 2012), and other languages such as French (Brysbaert, 2013), and Spanish (Izura, Cuetos, and Brysbaert, 2014), but there is still very little work on the Arabic language. The study by (Ricks. 2015) neglects lexical diacritics, a very important feature of the Arabic this is because of many challenges for automatic processing (Farghaly and Shaalan, 2009).

1.5 Thesis Outline

The main goal of this thesis is to demonstrate that based-rule stem based on morphological features can be used to successfully process retrieval systems. A suitable method called AMIR has been proposed and then developed using several morphological rules to compare its performance with the state-of-the-art stemming methods. Most of the original work appears in Chapters 3, 4, and 6.

- In CHAPTER 2, we go into the details of Arabic information retrieval and Arabic language, which considered more complex than the other language such as English.
- In CHAPTER 3, we discussed different aspects included Arabic stemmer and morphology and also focuses mainly on how existing methods are extracting stem or root from a word, and how AMIR, LUCENE, and the FARASA are extracted stem from the word by decreasing the complexity of morphology to

find the optimum model of the stem and the morphology for the Arabic language.

- In CHATER 4, we introduced with detail the AMIR dictionary and AMIR rules that can be implemented by discusses the effects of using different rules (rule-based stem), thus this chapter described different operations that are concerned with word formations to find the optimum stemming method. Also, three options of feature extraction (prefix, infix, and suffix) are proposed to reduce the complexity of morphological features to be used instead of light stemming in the Arabic retrieval systems.
- In CHAPTER 5, we have presented a brief survey of the most popular information retrieval models which we will adopt in this work.
- In CHAPTER 6, this chapter mainly focuses on the system evaluation system and the results of the experiments which we have been analyzed in this chapter.
- Finally, in CHAPTER 7, we give some concluding remarks of the research with a summary of achieved results, and also some notes around the future work.

CHAPTER 2

Information Retrieval and Arabic Language

Chapter Overview

Over the last decade, Arabic information retrieval has garnered significant attention due to increasing the Arabic text on the web. A considerable number of researchers share similar opinions on the benefits of morphology and stemming in Arabic information retrieval systems, especially for internet search engines; a problem exacerbated by the enormous amounts of data on the internet. Therefore, in this chapter, we will emphasize more on important aspects of information retrieval systems like broken plurals, derivation, affixation, morphology, and language. The Arabic language is a Semitic language as reported in [27], such as Aramaic and Hebrew, over 330 million people use it as their first language. Arabic has spoken in a large area including the Middle East, North Africa, most of the Arabian Peninsula, and other parts as reported in [28]. Therefore, a linguistic word composed are different between Arabic and English, hence Arabic morphology is very different from English; for example, Arabic offers more inflection word than English which are comes with the gender, numbers, person, noun, and adverb. While in English, inflection word can come with numbers within the sentence: (Office à Offices) and person within the sentence: (e.g., I play à he plays). In Arabic, there is no distinction is made between upper and lower case, and the rules for punctuation are much looser than in English. In addition to that, one big difference between the Arabic and English languages is that Arabic doesn't use abbreviations or capitalization, Arabic letters are only written in cursive, and Numbers are written from left-to-right. This chapter mainly describes information retrieval and the Arabic language and will be organized as follows: section 1.1 Arabic information retrieval, section 1.2 then provides a brief explanation of the Arabic language including Arabic nouns and verbs, and then section 1.3 goes to Arabic orthography, lastly in section 1.4 provide a brief overview of the Arabic grammar.

2.1 Arabic Information Retrieval

An information retrieval system is an information system using to store items of information that need to be processed or searched retrieved as reported in [29]. Information retrieval has started in the 1950s as reported in [30], using computers to search collections of documents with search engines web browsers, like Mozilla Firefox, Internet Explorer, Bing, Yahoo!, etc. These applications bring together a network of self-declared “experts” to answer requests posted by users. Several studies have been done to develop the best and easy way to retrieve data from different data sources [31, 32]. This task becomes one of the most important challenges in computer science after the World Wide Web became popular. In 1995, it was just less than 1% of the world population who access the internet but now there are more than 3 billion people, which accounts for around half of the world’s population, have an internet connection. The number of web sites in 1995 it was around 6 million pages and now of web sites over than1 trillion of web sites. The advances in smartphone technology and mobile internet connection speeds also boosted the number of internet users. Search engines are the most effective tools to search/locate information answers to a variety of queries on diverse subjects on the web. They try to process and index a huge volume of constantly changing mostly textual data. The inverted index is the internal data structure that stores information about indexing terms and in which documents these terms appear. The information retrieval searching process starts with sending the user request/query to the search engine. In a nutshell, query terms and indexing terms are matched and a ranking function is used to score each document. The search engine displays the documents by decreasing scores hopefully relevant to the user’s query.

Earlier information retrieval systems mostly work on English documents and web pages. However, they have rapidly grown and developed for many languages. Arabic is one of the most important languages for web search engines because more than 389 million people speaking the language in 25 different countries and most of them live in North Africa, Middle East, and West Asia. The Arabic information retrieval systems face several problems such as orthographic variations, very rich and complex morphology, ambiguous words because most of the root words consist of only three

letters, and inflectional and derivational morphology produce various word formations, where most Arabic morphology extracted from the Holy Quran [33].

Concerning information retrieval is fundamentally linguistic. This means that information retrieval based on the concept of linguistic. In Arabic, the meaningful word is divided into three categories namely noun, verb, letter, where, Arabic nouns can accept the definition ال (the) which are always attached to it at the begins whereas verbs don't accept the definition ال (the). Verbs often denoted to an event where it happens in the past, current or future. While the letter is every word that doesn't have meaning like the word أن (in) see more examples in the [34]. Furthermore, Arabic has a very complex morphology because plural forms are formed by adding meddle or ends to the word [35]. While English language plural forms are formed by adding suffix only to the word. However, existing schemes for Arabic are unable to extract the stem or root of words having an infix. Thus, in this thesis, one of the most challenges is to extract the stem or root of a word in its plural form, especially when a plural form is in the infix.

2.2 Arabic language

Arabic is a Semitic language family, where Arabic is the most widely used in the Semitic language. The history of Semitic language is described in [36]. The basic Arabic alphabet contains 28 alphabets (29 if hamza is counted as a separate letter as reported in [37]) with three short vowels, namely Alef (ا)، Waw (و)، and Ya' (ي). Words are written style in horizontal lines from right to left. The shape of each letter depends on its position. For example, the letter س (s) has a different shape depends on its position in the word as shown in Table 2.1, and all letters as reported in [38].

Table 2.1 Example of Arabic letters shape.

Arabic Letter	Begins	Middles	Ends
س (s)	سـ	ـسـ	ـس

Also, Arabic numerals are written from left to right, and most root contains three letters, though some roots contain more than three letters. And also, the prepositions

and stop-words may be writing separately or linked directly with the same letters, like و، ل، ك، ف، ب which means (inside, in, as, for, and). Notes that grammatical in Arabıc often deal wıth the ends of the word [39] more detail can be seen in the [40]. Whereas morphology focus on how the word is constructed [41]. Therefore, Arabic morphology can use to produce different word forms for nouns and verbs, as described as follows:

2.2.1Arabic Nouns

In Arabic, nouns can be divided into two main types, first one is rigid which is not taken from the other (words that are not derived) such as the word "شمس" (Sun), and the second type is derivative which is taken from the other (derived words) such as the word "مدرس" (Teacher) where the root is "درس" (study), which we will adopt in this thesis. Also, nouns are prefixed with the definite article (al-) as reported in [42], for example, [43, 44]. Furthermore, nouns can be singular, dual, or plural; and also some letters writing at begins and ends can indicate whether a noun is feminine or masculine; for instance: the word يكتب (he write) which indicates to masculine.

2.2.2 Arabic Verbs

In Arabic, most verbs contain three letters, though there are also a few verbs that may contain more than three letters. More detail about the Arabic verbs as described as following:

2.2.2.1 Command verb

Command verb is produced with a system of the prefix by adding letter ا (A) at begins of the root (note that, command verb is not used in other languages like English). Table 2.2 shows different word forms for command verb. However, in Arabic, it's easy to transfer root to command verb, these can be done by adding prefix ا (A) at begins of the root. For example, the word اذهب (you must go) which composed a letters ا (a) at begins of the word, which means that letters ا (a) indicates to command verb, thus our system will extract the same word اذهب (you must go) without removing prefixes ا (a). This means that our system does not remove prefix ا (a) that refers to the command verb.

Table 2.2 Example of commend verb in Arabic.

Command verb conjugations for the word شَرِبَ	
اِشْرِبْ: أَنْتَ (You must drink)	اِشْرَبَا: أنتما (you must drink, which indicates to dual)
اِشْرَبِي : أَنْتِ (you must drink)	اِشْرَبْنَ : أَنْتُنَّ (you must drink, which indicates to feminine)
اِشْرَبُوا: أَنْتُمْ (you must drink)	وَاشْرَبُوا: أَنْتُمْ (and you must drink, which indicates to plural)

2.2.2.2 Past verb

In Arabic, the past verb can be generated by adding diacritical marks[2] to the root, for example, كَتَبَ (he wrote). This means that diacritical marks are used to differentiate words either a verb or a noun. So, the past verb always produced by the remove all affixes that attached to the word. Thus, the past verb could be referring to gender (feminine/masculine) by the suffix during adding the suffixes like "ت" (ta). These refer to feminine; for instance: the word دَرَسَتْ (she studied). The past verb also could be also referring to plural by adding the suffix "وا"; for instance: the word كَتَبُوا (They wrote). Table 2.3 shows different word-forms of the past verb. Hence, our strategy for the past verb is removing all affixes from the word.

Table 2.3 Example of past verb in Arabic.

.Past Tense conjugations for the word شَرِبَ (Drink)	
شَرِبَتْ : هِيَ (she drank)	شَرِبُوا : هُمْ (They drank)
شَرِبَ : هُوَ (he drank)	شَرِبْنَ : هُنَّ (They drank, which indicates to feminine)
شَرِبْنَا : نَحْنُ (we drank)	شَرِبَتَا : هُمَا (both of they drank, which indicates to feminine)

2.2.2.3 Present verb

In Arabic, the present verb is very easy, it can produce by adding the: ي (y)، ت (t)، or ن (n) at the beginning of the verb as described in [45]. The letter ي (y) indicates to the masculine, the letter ت (t) is indicated feminine, and the letter ن (n) which indicates to

[2] See section 2.4 which present Arabic diacritical marks.

plural form to both feminine or masculine. However, there is the relation between letter ت (t) and the vowel ي (ya) to generate new words, This means that if the word contains letter ت (t) and the vowel ي (ya) in the same word; this will generate a noun, for example, the word تَدْرِيس (teaching), so in this case, the letter ت (t) can be used individually to produce present verb or related with others to generate noun, in this thesis we use this rule when we extract stem from the word as seen in the AMIR rules chapter 4. Table 2.4 shows examples for the present verb.

Table 2.4 Example of present verb in Arabic.

Present Tense conjugations for the word شَرِبَ (Drink)	
شَرِبُوا : هُمْ (They drank)	شَرِبَتْ : هِيَ (she drank)
شَرِبْنَ : هُنَّ (They drank, which indicates to feminine)	شَرِبَ : هُوَ (he drank)
شَرِبَتَا : هُمَا (both of they drank, which indicates to feminine)	شَرِبْنَا : نَحْنُ (we drank)

2.2.2.4 Future verb

The future verb is produced with a system of the prefix that makes up the future verb by adding the letters: سا "sa" [46]. Table 2.5 shows different word forms for the future verb. In fact, in the Arabic language, it's easy to transfer word to the future verb by adding the prefixes سا "sa." at begins of the word. Note that conjunctions in Arabic such as prepositions may be linked directly with the future verb.

Table 2.5 Example of future verb in Arabic.

Future Tense conjugations for the word شَرِبَ (Drink)	
سَيَشْرَبُونَ : هُمْ (they will drink, which indicates to plural)	سَأَشْرَبُ : أَنَا (I will drink)
سَيَشْرَبُ : هُوَ (he will drink)	سَتَشْرَبُ : هِيَ (she will drink)
سَتَشْرَبُونَ: أَنْتُمْ (they will drink, which indicates to plural)	سَنَشْرَبُ : نَحْن (we will drink)

2.3 Arabic Orthography

The development effective way to retrieval systems for the Arabic language is relatively urgent. Since the Arabic language was introduced information retrieval, a number of problems arisen; for instance orthography variations. Therefore, many of

these problems have been solved but still, there are some remains unsolved. In this section, we present a quick review of Arabic orthography variations.

Arabic can be written with or without the diacritics such as كَتَبَ (he wrote) and كتب (to write), but the computer considers these two words as same. Orthography with diacritics is a less ambiguous word, and more phonetic, but diacritics are used only in specialized contexts, such as children's reading books, dictionaries, and the Qur'an. In addition, letters change forms according to their position. Both orthography and morphology increase the amount of lexical variation which means that a word can be found in a huge number of different forms [47].

In Arabic, one of the most important problems to find a similar article in huge documents that depend completely on the diacritics of the word. The Modern Standard Arabic Language MSAL is characterized by the absence of diacritics. For example, two sentences 1) "مُحَمّدٌ طَالِبُ ذَكِيّ ", “Mohamed clever student, 2) "طالَبَ مُحَمَّدٌ بِحَقِّهِ” which means “Mohamed demanded his right, these two sentences contain the same word "طالب" in different diacritics. Therefore give us different meaning that means there is Ambiguity in the word if the word without diacritics. For these reasons, we try to find or develop a new method to disambiguate such differences due to diacritics and enhance the performance of finding a similar document.

We can use Arabic WorldNet to find semantic similarity of the article in search but it requires disambiguating word senses in sentences. Word sense disambiguation (WSD) tries to find the correct sense of a word in a sentence. We can search for a library if possible to apply WSD on Arabic text, or try to implement a WSD method for Arabic. In English WorldNet, the first and the most popular sense of a word is correct generally for around 60% of the time. If such information is available, we can always select the most popular sense of each Arabic word. We need disambiguation for identifying the word senses. For example, the following two sentences contain the same word "دَرسَ" and same diacritics but the meaning for this word is different”دَرسَ العِلم على شَيخِ الأزْهر”, He studied science at Al-Azhar Sheikh" and “دَرسَ القَمحَ وَالشَّعِير”, Threshing wheat and barley". In the first sentence, the meaning of the word is "studied" but in the second sentence its meaning is "threshing".

2.4 Arabic Grammar

Arabic grammar is similar to other Semitic languages. Therefore, an Affix is a morpheme added to a root to change its function/meaning.

Affixes can generally be divided into three groups:

- Prefixes - add morphemes at the beginnings of the root.
- Infixes - insert morphemes at the middles of the word, it’s not used in other languages like English.
- Suffixes - add morphemes at the ends of the root.

In Arabic, there are three types of roots as follows:

- Tri-literal roots.
- Quad-literal roots.
- Penta-literal roots.

As were mention early, Arabic language consists of 28 letters, only 3 are vowels [48] as shown following:

- ا (Alif).
- و (Wāw).
- ي (Ya').

Indeed, the basic form of the word is a root, thus any further analysis or changes on the root will impact results, this is because in some cases it loses the meaning of the word. In addition, the Arabic language is written from right to left and it consists of 28 characters as mentioned earlier. These characters can have diacritical marks on them as reported on [49] such as follows:

- Fathah
- Shaddah
- Damma
- Kasra

These decide how a word should be pronounced. If these diacritical marks are improperly used, this may cause errors in the pronunciation and hence change

meanings of words. Moreover, in Arabic pronoun linked always at the ends of the word [50]. It indicates to two kinds of gender:

- Masculine
- Feminine

In addition, the numbers in Arabic always come with pattern فاعل "FAal" to indicate the order. For example, الثالث (the thırd). Arabic has three main cardinalities cases:

- Singular
- Dual
- Plural

As far as the noun case is concerned, the dual noun cases are formed by adding two suffixes ان (-an) in the nominative cases and تان (-tan) in the genitive cases [51]. For example in the nominative case like ولد - ولدان (boy - two boys) and in the genitive case like سنة - سنتان (year - two years).

CHAPTER 3

Stemmer and Morphology

Chapter Overview

The Arabic language has a serious challenge due to the complexity of morphology. However, the best way to develop information retrieval in Arabic is to analyze Arabic morphology and stemmer. In the NLP tasks, it becomes hard to select an effective candidate word in retrieval systems. This is because it needs a good morphology to support it. Many methods are published to overcome this problem and successfully retrieved documents. But still, there is some weakness that should be solved like plural in infix, which a complex and difficult process, especially when it concerns the indexing of Arabic documents. In linguistic, morphology operations are concerned with word formation that most linguists agree that morphology is the study of the internal structure of words that deal with two morphemes: stemmer and affixes. So, it becomes more difficult to develop an efficient stemming algorithm due it is required that a good morphological analysis that should be used for effectiveness Arabic text retrieval to fetch all documents relevant to user needs. Arabic morphology could be divided into two main types: derivational morphology and inflectional morphology. Consequently, there is a strong relationship between inflectional and derivational morphology that inflectional morphology often comes after the derivational process. Furthermore, the best way to extract stem in the Arabic word is to conduct morphological analysis correctly. If the stem removes too much from the word, then the result becomes weak and does not perform very effectively. The stem is a pre-processing task that uses to reduce different grammatical word forms such as verb, noun, adjective, adverb, etc. Hence, in this work, we present AMIR that dealing with morphology to produce stem among all possible formations in the derived word. This means that AMIR is used to produce the stem or root from the word, this stem could be used as indexing term in Arabic retrieval systems, so that it is able to solve problems that Arabic information retrieval facing like plural in infix, conjunctions in Arabic such as preposition. This chapter is organized as the follows: section 3.1 presents Arabic

stem, then in section 3.2 and section 3.3, we will describe two types of popular stem's techniques uses for English and Arabic, we will present Arabic morphology in section 3.4, then root-pattern morphology is exposed in section 3.5, lemmatizer, stemmer and morphology analysis is exposed in section 3.6, in the rest of this chapter, we briefly comparison between lemmatizer, stem, and morphology analysis in section 3.7.

3.1 Arabic Stemmer

Basically, the stem is a technique for reducing the grammatical form of the word based on inflectional and derivational morphology. It a crucial step especially for Arabic information retrieval because the same word may have many different forms. However, a word's stem is its most basic form (the basic component of the language is a word.). Stemming has a strong impact not only on the precision and recall of searching. Therefore, the essential component of a word is its stem and Arabic stems are different as compared to other languages. For example: In English, affixes stemmer can generally be divided into two categories: prefixes and suffixes stemmer. While, in the Arabic language, the affixes stemmer can be divided into three categories: prefixes, infixes, and suffixes stem. In addition, Arabic nouns can take the form of being singular, dual, and plural; gender; feminine/masculine; verbs such as; present, past, future, and command which not used in other languages like English. And also, one of the other differences between Arabic and English stem is the plural form, where Arabic plural form can be indicated by ends or meddles of the word, whereas plural form in English could be indicated only by the ends of the word. Thus, the plural form is one of the most challenges in this thesis, especially when a word in plural form is in the infix. Consequently, the stem is a process that uses to reduce words to their respective stems or roots. One disadvantage of existing methods of Arabic stem is don't dealing with infix stem, and also, the stem does not conflate irregular forms like (men, man).

As shown in Figure 3.1, Arabic stem is classified into two categories: (i) a statistical stemmer which employing statistical information from a large corpus of a given language and (ii) a Rule-based stemmer which employing dictionary targeting to remove inflected affixes from a word based on morphology rules, which we will adopt in this work.

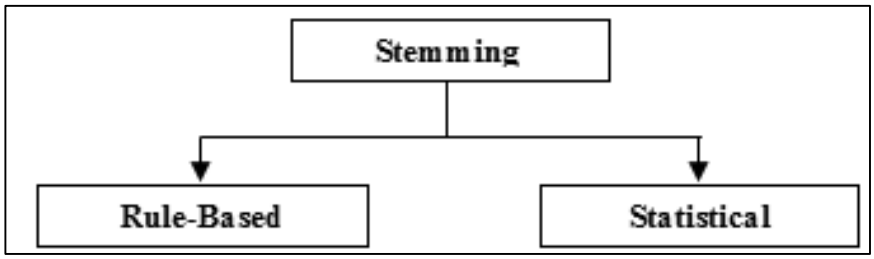

Figure 3.1 Types of Arabic Stemmer.

3.2 Porter Stem

Globally, the most widely used for English stem is Porter Stem, which is proposed in [52]. It proposed to remove inflectional endings only, such as -es, -ies, or –s, for example; (schools, school) and (studies, study). Up to now, the Porter stem performs well, especially in precision or recall evaluations. Therefore, in this work, we used lemmatizer to remove or replace inflectional morphology at the beginnings and the endings based on the morphology rules. These rules removed or replaced affixes based on the relationship among Arabic letters to find the stem or root of the respective words used as indexing terms in Arabic retrieval systems. For example, lemmatizer like (Porter Stem) tries to solve the problems of the plural in suffix. Therefore, proposed method try to replacing the suffixes such as ـتهن ، ـتهم ، ـتكن ، ـتكم ، تان- ، ات- that indicates to plural by replacing it with the suffix ة-

For example, the following suffixes highlighted in red which indicates plural should be replaced with another suffix to become singular, like in English (studies study).

مكتبية → مكتبات ، مكتبتان ، مكتبتكم ، مكتبتكن ، مكتبتهم ، مكتبتهن.

Thus, this could be much improvement to solve problems that Arabic information retrieval face. Table 3.1 shows examples of how the Arabic word construct and what morpheme should be removed to extract the stem or root of a word. Then, this word should be capable and suitable to use as an indexing term to get better retrieval. So, more examples can be seen in the next Table for the root wrote (he wrote).

Table 3.1 Example of AMIR stemmer.

Actual Word	Number	Affixes composed			AMIR Stemmer	Word Translate
		Prefix	Infix	Suffix		
المكتب	Singular	ال	-	-	مكتب	The office
مكاتب	Plural	-	ا	-	مكتب	The offices
مكتبكم	Plural	-	-	كم	مكتب	Your offices
الكاتب	Singular	ال	ا	-	كاتب	The author
المكتبات	Plural	ال	-	ات	مكتبة	The Libraries
كتابك	Singular	-	ا	كم	كتاب	Both of your book
كتبوا	Plural	-	-	وا	كتب	They wrote

3.3 Light Stemming

Light Stemming[3] is an algorithm uses to extract Arabic stem by removing a set of prefixes and suffixes from a word. Consequently, this algorithm was first proposed by Larkey (2007), and become one of the widely used in Arabic information retrieval system in previous works. Therefore, This work addressing stem by remove prefixes and suffixes if it's found in the listed of the light 10 as shown in Table 3.2. There are two disadvantages of light stemming: first light stemming is remove prefixes and suffixes without dealing with infix stem, and the second is extracting stemmer by removing prefixes and suffixes from a word without distinguishing whether the removed letters are actually core letters of the root or not.

Table 3.2 Removing prefixes and suffixes by light stemming 10.

	Remove prefixes	Remove Suffixes
Light10	ال، وال، بال، كال، فال، لل، و	ها، ان، ات، ون، ين، يه، ية، ه، ة، ي

[3] As far as have been developed the light stemming in Arabic, it has still removal prefixes and suffixes without distinguishing whether the removed letters are actually core letters of the root or not.

Due this algorithm has a good impact on Arabic text, so many works have been made to develop this algorithm to improve Arabic stem, thus the following is some proposed methods that have been made to develop light stemming as follows:

Larkey, 2007 developed several light stemmers algorithms for Arabic, and they have compared light stemming with several stemmers based on morphological analysis**.** They also evaluate their effectiveness for information retrieval using standard TREC data. The test they performed shows that light stemming allows remarkably well for information retrieval without providing correct morphology analyses.

Mustafa, 2019 which is the most modern stemming algorithm for developing Arabic stemming based on light stemming. This work proposed two different stemming techniques by utilizing extra suffixes in the total number of prefixes and suffixes to be removed as shown in Table 3.3 which extended-Light stemmer is greater than their peers in light 10. They found that the proposed stemmers yield 5.13% and 13.1% improvement over light stemming 10 in retrieval performance with 0.369 average precision and 0.397, respectively.

Table 3.3 Light stemming example for removing prefixes and suffixes developed by Mustafa.

Prefixes Stemmer	Suffixes Stemmer
ال، وال، بال، كال، فال، لل، وبال، ولل، فل	ها، ان، ات، ون، ين، يه، ية، ة، ي، وا، تي، هما، نا، هم، ت

Jaffar, 2019 this work has developed an algorithm based on the light stemming to restore Arabic data by adding extra prefixes and suffixes to the list of light 10 as shown in Table 3.4.

Table 3.4: Light stemming example for removing prefixes and suffixes developed by Jaffar.

Prefixes Stemmer	Suffixes Stemmer
ال، وال، بال، كال، فال، لل، و	ها، ان، ات، ون، ين، هن، هم، ته، تي، ني، يه، ية، ه، ة، ي

In this study, the main goal is to improve Arabic information retrieval through produce the best stem of the word that can be used in retrieval systems such as search engines.

Hence, we developed the light stemming by adding extra prefixes, suffixes, and infixes, which not used in other light stemming proposed as shown in Table 3.5.

Table 3.5 Light stemming example for removing prefixes and suffixes developed by AMIR[4].

Prefixes Stemmer	Infixes Stemmer	Suffixes Stemmer
است، ست، فست، وست، ال، وال، فال، وكال، فكال، وبال، فبال، وت، فت، ون، فن، كت، وكت، وم، فم، بم، لم، ولم، فلم، وكم، وكالم، فكالم، وللم، ولم، فللم، وللم، وبالم، فبالم، وا، فا، وب، و، ف، ب، ل، ول، فل	ا ، و	ات، وا، ون، وه، ان، تي، ته، تم، كم، هن، هم، ها، ي، تك، نا، ين، يه، ية، ي، ا، تكما، تكنا، تهما، تهم، تهن، تهم، تكم، تكن، ن

Furthermore, in this work, we trying to increase the efficacy of extracting a word effectively while minimizing the ambiguity since it depends on validation before removing affixes, thus for each input term, AMIR starts by check if the length of the term is greater than three-letters, if yes, AMIR starts by check if the length of the term is greater than three-letters, if yes, AMIR will search the query term in AMIR dictionary, if the term is found, then AMIR will apply set of rules to extract stem or root from the word if it accepted on AMIR rules, otherwise, the original term is returned unchanged. This process will describe in detail in chapters 4 and 6.

3.4 Arabic Morphology

In this section, we will introduce the morphological structure of the Arabic word, which is an important tool in the field of information retrieval. Therefore. Morphology analysis is an important feature in indexing and search systems because if affixes removed too much from the word, the result can be building unrelated word or it may lose the meaning. However, a morphology analyzer can be helpful to avoid the complicated processing of the indexing term. Since Arabic is morphologically more complicated than English, Figure 3.2 shown Arabic word structure, which explains how morphemes put together to construct Arabic words (each content of word called morpheme). So, words are the basic units of analysis [53]. There are two important

[4] See Figures 4.1, 4.3, which presents how AMIR add all possible prefixes and suffixes to generate word.

notions of the word: open class, which means it's possible to produce a new word to this class; for example noun, verb, adverb, Etc. The close class, which not possible to add new words; for example, prepositions, stop-word. On the other hand, the open class can be divided into groups: free class, which means it can stand alone, such as شمس "sun", and bound class, which means, it cannot stand alone like the derived word. Therefore, the bound class is a morphological morpheme, which can be divided into two groups: inflectional morpheme and derivational morpheme. In Arabic, inflectional morpheme usually does not change the word meaning, it indicates to number, gender, and verb tense; for instance: يلعب - لعب (playing - play), while derivational morpheme usually changes the word meaning, where derivational morphology used to generate words as described in the [54]; for example the word كاتب (Author). The root is كتب (he wrote), through inserting the infix ا (a) word will change the word type from verb to noun

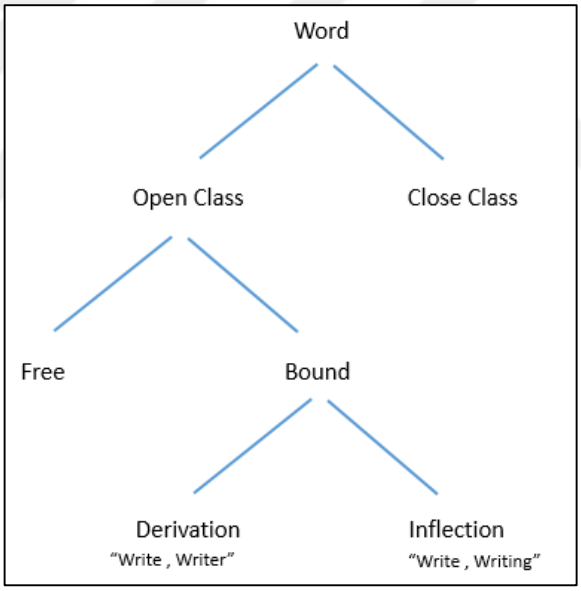

Figure 3.2 Arabic word structure.

As were a mention early, Arabic morphology is the most important branch of Arabic language dealing with how the word is constructed. It is only applied with nouns, verbs, adjectives, adverbs because they can be conjugations into different forms (e.x., of conjugations is prepositions). However, Arabic verbs can generally be divided into two groups: rigid and morphological, where rigid means in Arabic "جامد" and it can come only in one formula, for example, the word ليس (not), and morphological, which

means "متصرف " in Arabic, and it can come in different formulas, for example, the word يقول (he said). Note that the nouns are also divided into rigid and morphological as described in [55, 56]. On the other hand, Arabic verbs can generally one of the following forms: past, present, imperative (command verb), and future; such as فرح، يفرح، افرح، سافرح (be happy, he is happy, he was happy, we will be happy). While nouns conjugating can be subjected as number (singular, dual, and plural), such as نهر، نهران، أنهار (rivers, two rivers, a river), gender (masculine, feminine). Whereas Arabic prepositions and stop-words can be separately or linked directly with the word. This means that prepositions and stop-words can link with the Arabic word. Therefore, in this thesis, we attempt to build a word that can give a positive effect on indexing terms among all possible word formations from the derived word. This will be done by applying a set of morphological rules used to generate the stem or root from the word. Hence, the proposed method trying to improve the effectiveness of results retrieval by using Arabic morphology features that allows a query term to focus more on the meaning of a term as shown in Table 3.6.

Table 3.6 Indexing term with and without AMIR.

Semantic	بصيام Fasting	كالصائم As fasting	صائم Fasting	صيام Fasting	الصوم Fasting	الصائمون The Fasters
Without AMIR						
Indexing term	بصيام Fasting	كالصائم As fasting	صائم Fasting	صيام Fasting	الصوم Fasting	الصائمون The fasters
With AMIR						
Indexing term	صيام Fasting	صائم He fasting	صائم He fasting	صوم Fasting	صوم Fasting	صائم Fasting

3.5 Root-Pattern Morphology

Consider Figure 3.3, which illustrate the example of Arabic words generation process, the following terms are defined as:

- Surface Word: in short an inflected stem, where stem with or without prefixes, infix, and suffixes attached in it.
- Stem: a word generated by all possible inflected forms. This operation is usually performed after the derivational.

Note that: The following steps are followed in the construct of stem:

- The first step is the extract of all patterns.
- The second step is the extract of all noun prefixes.
- The third step is the extract of all noun suffixes.
- The fourth step is the remove conjunctions
- The fifth step is the extract of root.
- The sixth step is the sum of the five previous steps to construct the stem.

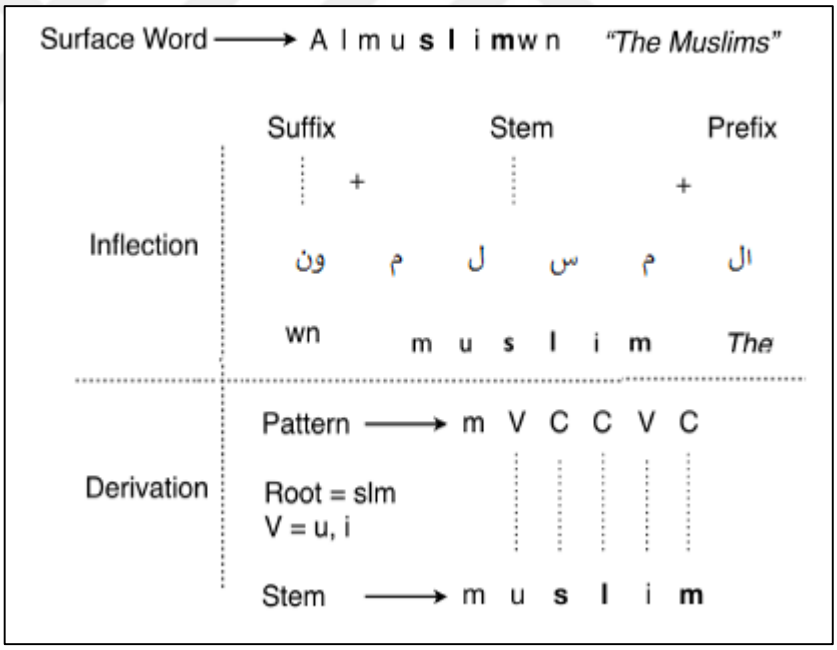

Figure 3.3 Example of Arabic word generation process

In this example, there are two operations performed to generate Arabic word: first is derivational, which always results produced with a new meaning of the word by adding the patterns radicals into their respective root's as shown Table in 4.7, the second inflectional morphology, which is used merely to marker the gender, number that gives

it its surface form without altering its meaning. Note that, there are many different patterns and more stems that can be generated using the same root and these different patterns we will attempt to remove by using the AMIR system.

As shown in Table 4.7 there are different types of Arabic stem can be generated with the same root such as مكتب، يكتب (office, he write). Since Arabic has rich in morphology, there are three cases indicating to number, single, dual, and plural. For example, the word, "مكتبهم" "Makatabham" which means in English their office. The root is "كتب" Kataeb (he wrote), and the pattern is "MaCaCaCkam" or "Makatab" (office), the suffix "ham" denotes to a masculine plural.

Through suffixes are used to mark persons; such as the suffix "ي" (y) for the singular; the suffix "ان" for the masculine dual; the suffix "تان" for the feminine dual; the suffixes "ون، هم، كم" for the masculine plural; and the suffixes "هن، كن، ات" for the feminine plurals as shown in the previous table.

Table 3.7 Example of root-pattern Arabic morphology

Arabic Word	Class	Pattern	English translate
سلام	salAm	CaCyC	Peace
مسلم	Msalam	muCCiC	Muslim
سليم	Salym	CaCyC	flawless, sound
الاسلام	Al<islAm	<muCCiC	Islam
سَالم[5]	sAlim	CACiC	Safe
تَّسَليمَ	TasalًOaim	taCaCiaC	he received
السلم	Al<silm	<CiCC	Peace

3.6 Lemmatizer, Stemmer and Morphology Analysis

Stem and morphology are a very rich area of research in the field of information retrieval. Therefore, existing schemes use to dealing with morphology analysis and stemming to improve search efficiency but only a few offered with Lemmatizer.

[5] See rule number 3 on page 50, which is applied to extract infix stems.

Therefore, NLP applications such as the search engine. Lemmatizer, stemmer, and morphology are considered as the essential stage to determine word formations. The goal of both stemmer and Lemmatizer are used to reduce inflectional forms and sometimes derivational forms of a word into a common base form. While morphology is used to reduce both inflectional forms and derivational forms of a word into a common base form. Indeed, Lemmatizer usually refers to doing things properly with the use of vocabulary and morphological analysis of the words, normally aiming to remove inflectional endings, for instance: the word كتبوا (they wrote) by removing inflectional ending وا (WA) then they produce the word كتب (write). On the other hand, the treat of word-forms usually known as "segmentation".

Both stemming and lemmatization used to develop a performing of Arabic information retrieval. The reason for this is that the stem can use suffix and prefix stripping off from a word without morphological analysis support it while the Lemmatizer must do a complete morphological analysis (based on actual grammatical rules and a dictionary). In addition to that, the best way to find an Arabic stem is to conduct morphological analysis correctly, if the stemmer removes too much from the stemmers the result is that, produced unrelated words under the same stemmer. Thus, Arabic has rich and complex morphology that study the internal structure of words that deal with the two morphemes such as stems and affixes. Therefore it is required that a good morphological analyzer should be used for the effectiveness of Arabic retrieval.

The Arabic information retrieval systems mostly work on stemmer and morphology analysis rather than Lemmatizer which is a relatively new topic for Arabic language processing, hence only a few studies focused directly on Lemmatizer to extract stemmer from Arabic texts. The goal of both stemmer and Lemmatizer is to produce stemmer from different word formations. Therefore, Lemmatizer consists of assigning to the surface of each word in a text its corresponding lemma. Therefore, Lemmatizer is a process based on the extension of the syntagmatic unit, which is an important component in any natural language like Arabic. In this work, we will be using the Lemmatizer technique in our approach by applying a set of proposed rules of the Arabic language for each word requested by users to extract the exact word, and then selecting the analysis by matches the best current word that uses in operation searches. For examples:

Exp1: The output of the Lemmatizer of suffixes by using the AMIR:

بيت. → بيتهم، بيتكما، بيتهن, بيتكم

مكتبة[6] → مكتبات، مكتبتان، مكتبة، مكتبتكم

Exp2: The output of the Lemmatizer of prefixes by using the AMIR:

بيت. → البيت ، فالبيت، بالبيت، وللبيت

مكتبة → فالمكتبة، وكالمكتبة، فمكتبة، يالمكتبة

3.7 Comparison between Stemmer and Lemmatizer

Information retrieval systems normally divided words into three groups namely: root, stem, or lemma. So, many schemes in the information retrieval field considered the root which leads to high recall but low precision. This is because of the complexity of the language. However, searching for the words by root, yields getting the other words that may not relevant to the user's request. Other schemes show the importance of using stem for improving retrieval precision and recall based on inflected words. While lemmatizer used in information retrieval to enhance the performance, especially in Arabic. In Arabic, lemmatizer often captures similar words. This is because lemmatizer can be broken in different plural forms to their singular form. So, lemmatizer use lemmas, not surface forms[7] in the Arabic lexical, in this thesis, we need to perform lemmatizer. This step is very necessary as Arabic is a morphology-rich language as reported in [57]. On the other hand, both stemming and lemmatizer generate the root form of the inflected words that stem might not be an actual word whereas, the lemma is an actual language word. So the stemmer and lemmatizer could be related but the lemmatizer used to produce lemma. Therefore, the best way to improve Arabic retrieval performance to get a power search is by using the lemmatizer. This is because lemmatizer is more accurate and it takes the context of the word in mind. Table3.8 shown some examples of the Arabic words to extract lemma and stem.

[6] See rule number 7, on page 51, which is applied to deal with lemmatizer by using replacements.

[7] The surface form of a word is the form of a word as it appears in the text. while the lexical form of a surface form is the lemma.

Therefore, in this work, we have applied both stemmer and lemmatizer where stemmer applied in the AMIR rules R1 and lemmatizer applied in the AMIR rules No 7, see AMIR rules section 4.2.

Table 3.8 Example of the Arabic stemmer and lemma

Term	Stemmer	Lemma	Root
مدرسات	مدرس	مدرس	درس
طالبتان	طاالب	طاالبة	طلب
زرعات	زرع	زرع	زرع
مصروفات	مصروف	مصروف	صرف
دراستان	دراس	دراسة	درس
معلمون	معلم	معلم	علم
الاعاب	اعاب	لعب	لعب
عمليات	عملي	عملية	عمل
ممرضتان	ممرض	ممرضة	ممرض
طائرات	طائر	طائرة	طائر
سيارات	سيار	سيارة	سير
تدريسهم	تدريس	تدريس	درس
مطارات	مطار	مطار	طار
مشروبات	مشروب	مشروب	شرب
مسافرون	مسافر	مسافر	سفر
العالمين	عالم	عالم	علم
ورثكم	ورث	ورث	ورث
الصلاوات	ميراث	ميراث	ميراث
مقالات	مقال	مقالة	قول
مسلمون	مسلم	مسلم	سلم
معاملات	معاملة	معامل	عمل
الجامعات	حامعة	جامع	جمع
مستشفيات	مستشف	مستشفي	شفي
زراعات	زراع	زراعة	زرع
طابعات	طابع	طابعة	طبع
حسابات	حساب	حاسبة	حسب
مصورون	مصور	مصور	صور

3.8 Comparison between Arabic stem algorithms

In this section, we compared AMIR stem with two counterpart systems: LUCENE and FARASA. Since the Arabic language is a very inflectional language, there are two cases that denoted to plural in the Arabic language, which at the infixes and suffixes. This means that many inflectional forms that ought to be performed to mark numbers at the infixes and suffixes. A plural at the meddling of a word with infixes “و , ا” (a, y). An example of this case, the word مساجد “Masajed" (Mosques), which contains the infix ا (a) which indicates to plural. Thus, the AMIR system is able to remove plural

in infix to produce its singular form. This can be done by applying AMIR rule No 3, as shown in the next chapter. So, AMIR system extracting the word مسجد (Mosque) instead of مساجد (Mosques) by removing infix ا (a), which denotes the plural. While the system of FARASA and the LUCENE extract the same word مساجد (Mosques) as it is. The reason for that is the system of FARASA and the LUCENE do not handle plural in infix. Therefore, in this case, they failing to produce their singular form. On the other hand, when the plural is in the suffix; in this case, the plural is denoted by the following:

“ون، ات، تان، هم، كم، هن، كن، تهم، تهن، تكم، تكن، تكما، تكنا، تهما، تهنا”.

Consequently, the word مكتبات “MkatabaT" (libraries). To distinguish between these morphemes, we say that “Kataeb" كتب (he wrote) is the root; the prefix م (m) is a derivational morphology where it refers to the noun; the suffix ات (at) is an inflectional morphology where it refers to feminine plural. So, there are two patterns marker in this word, the first is the prefix “م" (M), which denotes the noun as were mentioned early and the second is the suffix ات (at) which refers to plural. However, to change the plural form to singular, the following process is followed; the suffix “ات" (at) should replace by the suffix “ة" (at) to generate another common word مكتبة “MkatabT" (library). See AMIR rules in Section 4.3. This process is called replacement “الإبدال” in the Arabic language, to the best of our knowledge, the replacement was not used in previous works.

This technique is an important factor that can improve Arabic retrieval systems. Therefore, this work is trying to solve problems of the plural form especially that attached in the infix thus resulting in an increase in precision. So, we compared AMIR against FARASA and LUCENE for the same example: the word مكتبات (libraries), where AMIR extractor the word مكتبة (library) by replacing the suffix ات (at) by suffix ة (taa). While FARASA and LUCENE both extractor the word مكتب (office) by removed suffix ات (at); thus, they produce a word that has a different meaning. Therefore, the main advantage of this work is to provide highly accurate results in linguistic knowledge by use morphology features. The fact that this new scheme can dissect a plural word and then get its singular form. Table 3.8 shows the comparison

of how LUCENE, FARASA, and AMIR dealing with inflectional and derivational morphology to extract stem or root from the word. In the rest of this chapter, we described LUCENE, FARASA, and AMIR in more detail, where LUCENE and FARASA trying to find the root or stem from the word based on remove possible affixes that attached to word whereas AMIR is rule-based stemmer.

Table 3.9 AMIR and LUCENE dealing with inflection and derivation to extract stemmer

	LUCENE	**FARASA**	**AMIR**
Inflectional	LUCENE uses the light stemming technique to exact inflectional morphology including prefixes and suffixes without dealing with infixes.	FARASA uses the linear kernels technique to exact inflection morpheme including prefixes and suffixes without dealing with infixes.	AMIR uses the rules-based dictionary to exact inflectional morphology including prefixes and suffixes in addition to infix.
Derivational	LUCENE requires less knowledge than AMIR such as LUCENE removed the inflectional suffixes without dealing with replacement.	FARASA requires less knowledge than AMIR such as FARASA removed the inflectional suffixes without dealing with replacement.	AMIR removes morphemes that are inflectional in suffixes by replacing suffixes with other suffixes to produce the best stem.

3.8.1 LUCENE Stemmer algorithm

LUCENE is a free open source in the field of information retrieval, it was writing in 1997 by Doug Calting. It was originally written completely in Java, which has been ported and can be used with many programming languages like Python, C++, Delphi, Perl, and PHP, Java, and C #. Also, LUCENE is the most important tool for research

because it has more software libraries, and developed by Apache Foundation[8]. So LUCENE is a full-text search library of documents, which makes it easy to add search functionality to an application. In recent years, LUCENE has become an important tool in the information retrieval library and the most common search framework as reported in [58]. Therefore, LUCENE became popular and the best search available on the internet that used for preparing text for indexing. Also, LUCENE uses as a search engine for searching and browsing for documents. This is because LUCENE is a simple, powerful, and high-performance search engine. Figure 3.4 shows the main steps done by LUCENE performs to index the documents. During indexing, the text is extracted from the original content, then a document is created that contains fields holding the contents. After creating the fields holding the contents, the contents of the Fields are passed through the Analyzer to tokens, then after tokens, the contents are passed through the Indexer to generate an inverted index, then, the index is written to an existing index file, Finally, this index file is stored in a user-specified directory. In this work, we used LUCENE to index the EveTAR 2016 corpus to create a search index, also we used both LM and the BM25 by use with LUCENE ranking functions to compute the textual similarity feature.

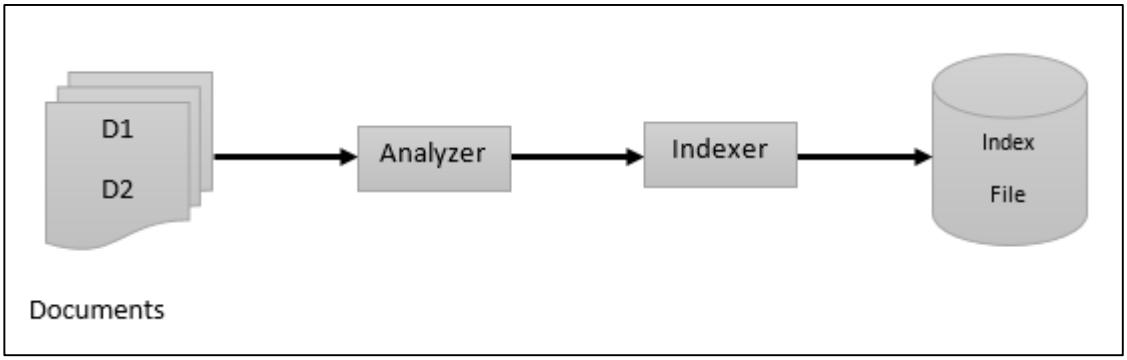

Figure 3.4 steps in the LUCENE

[8] The Apache Software Foundation (ASF) is an American non-profit corporation to support Apache software projects, including the Apache HTTP Server. The ASF was formed from the Apache Group and incorporated on March 25, 1999.

3.8.2 FARASA Stemmer algorithm

FARASA is a fast and accurate Arabic segmented (Arabic translate means insight). It's a segmentation[9]scheme which is one of the most important tools in Arabic, FARASA was writing by [59]. It has been made available for use on http://qatsdemo.cloudapp.net/farasa/. FARASA uses features and lexicons include stems, affixes, and lemmatizer to their segmentation which is based on the vector space model to rank possible a word that using linear kernels by breaking words to their constituent clinics. The following some Arabic words that segmentation by using FARASA:

Example 1: **من السهل أن ترى الناس على حقيقتهم ومن الصعب أن ترى نفسك على حقيقتها**

FARASA: **من ال+سهل أن ترى ال+ناس على حقيق+ت+هم و+من ال+صعب أن ترى نفس+ك على حقيق+ت+ها**

Example 1: **ليست الحياة تعيسة لكن قلوبنا بالبعد عن الله كئيبه**

FARASA: **ليس+ت ال+حيا+ة تعيس+ة لكن قلوب+نا ب+ال+بعد عن الله كئيب+ه**

Recently, FARASA is accurate text processing for Arabic text such as segmentation, POS tagging, and lemmatizer. It offers faster performance than [60]. The FARASA system relies on probabilistic models of stems, prefixes, and suffixes, instead of using context information to produce high tokenization and performance which have been used in this work to compare the retrieval performance of the proposed method.

3.8.3 AMIR Stemmer algorithm

This subsection discusses the AMIR algorithm to find the stem or root of the word that use as an indexing term in the field of Arabic information retrieval systems. AMIR algorithm works as follows:

[9] See http://qatsdemo.cloudapp.net/farasa/demo.html

3.8.3.1 Tokenization & Normalization.

Arabic tokenization has been implemented in several solutions to resolve ambiguous words. For instance, characters can be written in different ways, such as character (ء) Hamza can be composed in different ways (أ، إ، آ). This causes 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 أ ، آ ، إ by ا
- Replacing final ى ، ئ by ي.
- Replacing final ه by ة.

3.8.3.2 Keyword Extraction

We represent AMIR steps to extract Keywords as follows:

1. Convert the user request text into words and put them into a list.
2. Check the lists whether prepositions or stop-word are found. If found, remove any matched from the list.
3. Search AMIR Dictionary to find given terms in the list; if a match found, then extract root/stem if accepted on AMIR rules.
4. 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.

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 the AMIR dictionary, if any match found in the given list, then extract root/stem based on AMIR rules. 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 refers to preposition, and prefix و (and) which refers to stop-word. So, we will get مدرس (teacher), which used as indexing term.

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

More explanation around how AMIR works as follows: AMIR starts by receiving a request from user's query; then check if the length of the three-letter word or more, and then seeks the query words in AMIR dictionary, if the word is found, thus then extract root or stem if accepted on AMIR rules. An example of this process: The word requested is مدرسة (school); when this word gives to other Arabic retrieval systems, it will return the stem مدرس (Teacher) by removing the suffix ة (taa), in this case, the word meaning has changed. through the use, AMIR rules No 2 as shown in the next chapter, which says that if the word composed prefixes م (M) and suffixes ة (taa) joined together in the same word; thus, this case will a produce noun (always refer to places)., Hence, AMIR system suggested to keep the word formations as they are. So, AMIR system will extract the word مدرسة (school). This result is increases precision, which we aim to have.

CHAPTER 4

AMIR Rule-Based Stemmer for Arabic

Chapter Overview

There are only a few methods proposed for Arabic that using to extract stem from a word according to Rule-Based stemmer that leverages Arabic stemmer in segmenting the word into valid morphological features, which can uses as indexing terms in web searching to gives internet users access to millions of articles, news, and services. However, Arabic morphology has challenges that fall into two categories; the ambiguity problem, which can be always faced by good analysis for Arabic morphology, and the second is conjunction words such as stop-words and prepositions. Note that stop-words and prepositions can be linked directly with the word in the Arabic language, which adds another challenge to the stemmer operation. Therefore, in this chapter, we will present the state-of-the-art Arabic stem algorithm, which uses a rule-based dictionary to generator words based on morphology features by matching the word with all possible affixes and patterns attached to it. So, the proposed method uses morphology features to building an AMIR dictionary that generates over 1,400 words from each root. Furthermore, the method proposes increases the efficacy of extracting a word effectively while minimizing the ambiguity since it depends on validation 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. So, we attempt to produce rules capable of producing the root better than other existing methods like Khoja.

On the other hand, in this thesis, we explain through the morphology features of the Arabic language why the Arabic morphology is much more complicated than the other language like English. Hence, this thesis poses the big challenge faced by researchers to extract the stem or root of the word to produce a high-performance technique uses to extract Arabic root/stems by adding infixes to prefixes and suffixes. Note that, in Arabic morphology there exist many words that have infix, and removing an infix

depends on the morphological structure of the language. So, there are no rules proposed to extract the stem or root of words having an infix. Therefore, the proposed method aims to produce a high-performance tool to extract Arabic root/stems by adding infixes to prefixes and suffixes.

Consequently, this thesis investigated in detail the impact of stem and morphology on Arabic retrieval systems, and how to bridge the gap in Natural Language Processing (NLP) task. So, we described in more detail, how morphological units are grouped to produce new words based on the derivational and inflectional process. Thus, we used morphology features to generation words to building a rule-based stemmer dictionary for Arabic that can help to extract the stem or root of the word to uses as indexing term in Arabic information retrieval. Therefore, we can get the intelligent use of stem and morphology in the Arabic domain that we aim to have.

This chapter will be organized as follows: Section 4.1 provides the AMIR dictionary, which is a rule-based dictionary constructed from several Arabic morphology features. The AMIR rules are exposed in section 4.2, and section 4.3 presents conjunctions in Arabic, and we will focus on two types of conjunctions; namely stop-word and prepositions.

4.1 AMIR Dictionary

The process of information retrieval consists of finding all relevant documents, which are ordered by relevant for the user query. Thus, the highest-ranked document is considered to be the most likely relevant document. Also, the aims of natural language processing like information retrieval to transform the potentially ambiguous words of queries into unambiguous words. Furthermore, indexing terms is a complex and difficult process, especially when it concerns the indexing of Arabic documents. Year by year, many methods are published to overcome this problem and successfully retrieved documents. Therefore, in this thesis, we present the AMIR dictionary, which constructed from several Arabic morphology rules to get the intelligent use of morphological analysis of the Arabic information retrieval system. Therefore, the utility of the AMIR dictionary is to use morphological features and specify all inflected forms for each stem templates by a combination of the affix with the root.

In general, inflectional morphology often does not change the basic meaning of a word, while derivational morpheme is often changing the basic meaning of a word. Table 4.1 shows an example of derivational and inflectional morphology.

The Arabic language has two main phases to generate or produce the stem which is inflectional and derivational morpheme as described on the next page.

Table 4.1 Derivational and inflectional morpheme example

	Word	**Example**
Derivational	Write "كتب"	Writer "كاتب"
Inflectional	Write "كتب"	He wrote "كتبَ" He is writing "يكتب"

4.1.1 Derivational Morpheme

The derivational morpheme is a process of generating a word with a new meaning (called in some other works by derivational morphology). This process is done by adding patterns into root based on the set of morphological rules (see AMIR rules in the next section). Note that, in Arabic, patterns are known as "" as described in اوزان[10] [61, 62]. However, in Arabic, several patterns can be adding to each root to generate or produce an Arabic word. These patterns can be inserted to root within the prefix, infix, and suffix, whereas in other languages can be inserting at the prefix and suffix only. Therefore, in this thesis, we will be adding all possible patterns that attached to the root to generator all derivational word formations. Thus, these patterns must not remove when stem extract from a word. This is because, as mentioned earlier, derivational morpheme often generate a word with a new meaning; Thus the word type probably will change from noun to adverb, adjective, or noun to another noun like the word "مكتبة" (library), if where remove the prefix "م" (m), it became "كتبة" (Kutiba), thus, the word change their meaning.

Table 4.2 as showing on the next page, which presents different derivational morpheme operations that ought to be performed to mark nouns to adverbs and

[10] Note that the Arabic language uses derivational to generate both verbs and nouns

adjectives. Therefore, the derivational morpheme can be inserting to root within begins, middles, and ends. Noted that derivational morpheme is adding to root directly. This means that derivational morpheme operation always comes before inflectional morpheme, also inflectional morpheme could be adding to root directly without any derivational morpheme attached.

Table 4.2 Arabic language derivational process

Arabic Word	Pattern	Derivational	Root	English translate
كاتب	KaCaCyC	KAtab	كتب	Author
مسلم	MuCCiC	Msalam	سلم	Muslim
مدارس	MaCACaC	MadAras	درس	Schools
الجامعة	Al<CACCta	Al<gAmaTaa	جمع	The university
دروس	CACiC	sAlimwn	درس	Lessons
الرجلات	Al<CaCaCTaa	Al<rahalat	رحل	The flights
منزلية	TaCaCraC	ManazalYeT	نزل	Things for house

4.1.2 Inflectional Morpheme

The inflectional morpheme[11]is a process of generating a word with different forms within the same meaning. This process is done usually after the derivational morpheme. However, in Arabic, different inflectional morpheme can be adding to roots or patterns to produce Arabic words within the same meaning. So, inflectional morpheme operations are the process that indicates inflectional morphology which generally refers to gender, numbers, preposition, stop-word, verb tense, nouns (singular, dual, plural). Therefore, inflectional morpheme often not affect the basic meaning of the word. So, we can remove inflection from the word when we extracting the stem. There are three main types of the inflectional morpheme in Arabic language namely prefixes, infixes, and suffixes, which are described as following:

[11] Inflectional affixes are always suffixes in English, whereas in Arabic, inflectional affixes can be prefixes, infixes, and suffixes.

4.1.2.1 Prefixes

Prefixes are the part of the word that adding to the beginning of a root, which very important in the word structure. Three types of the prefix can be concatenated to the Arabic patterns or root namely: definitions, interrogation, and conjunction. Figure 4.1 shows all possible prefixes that can be adding to the root. It's clear from figure 4.1 there are one to four prefix can be adding to each root. Figure 4.1 presents one of the most challenging inflectional morphemes in natural language processing, which we will consider when we generator a word.

The conjunctions in Arabic often placed at the beginning of the word like prepositions, which is not easy to distinguish if it was linked directly with the word. Thus, by using AMIR it's easy to remove when we extract stem from the word. For example: In the Arabic language, there are so many original root's letters begins with letter و (and), or ل (for), etc. for instance, if we remove و (and) from the word `ورثكم` (they inherit), we will get رثكم (the word does not have meaning). So, we can remove these prefixes if it did not find in root's contents. Thus, AMIR removes only necessary affixes that not changing the meaning of the word based on removing inflectional prefixes that accepted in the AMIR rules.

The Arabic language is considered a highly inflectional language. So, the fundamental is different between Arabic and English language; for example, morphological features for English are usually marked by suffixes for nouns, whereas in Arabic are usually marked by both prefixes and suffixes for nouns. And also, the Arabic language offers more inflectional operations than English. This means that many inflectional morphemes that ought to be performed to mark number, gender, verbs, and person. For example, the word المسلمون "Almuslimwn" (the Muslim) shows the concatenation of morphemes to form the word. To distinguish between these morphemes, we say that مسلم (slim) is the root morpheme; the definiteness marker ال is an inflectional morpheme; prefix م (mu) is a derivational morpheme, which denotes to the noun; and the suffix ون (wn), which is an inflectional morpheme.

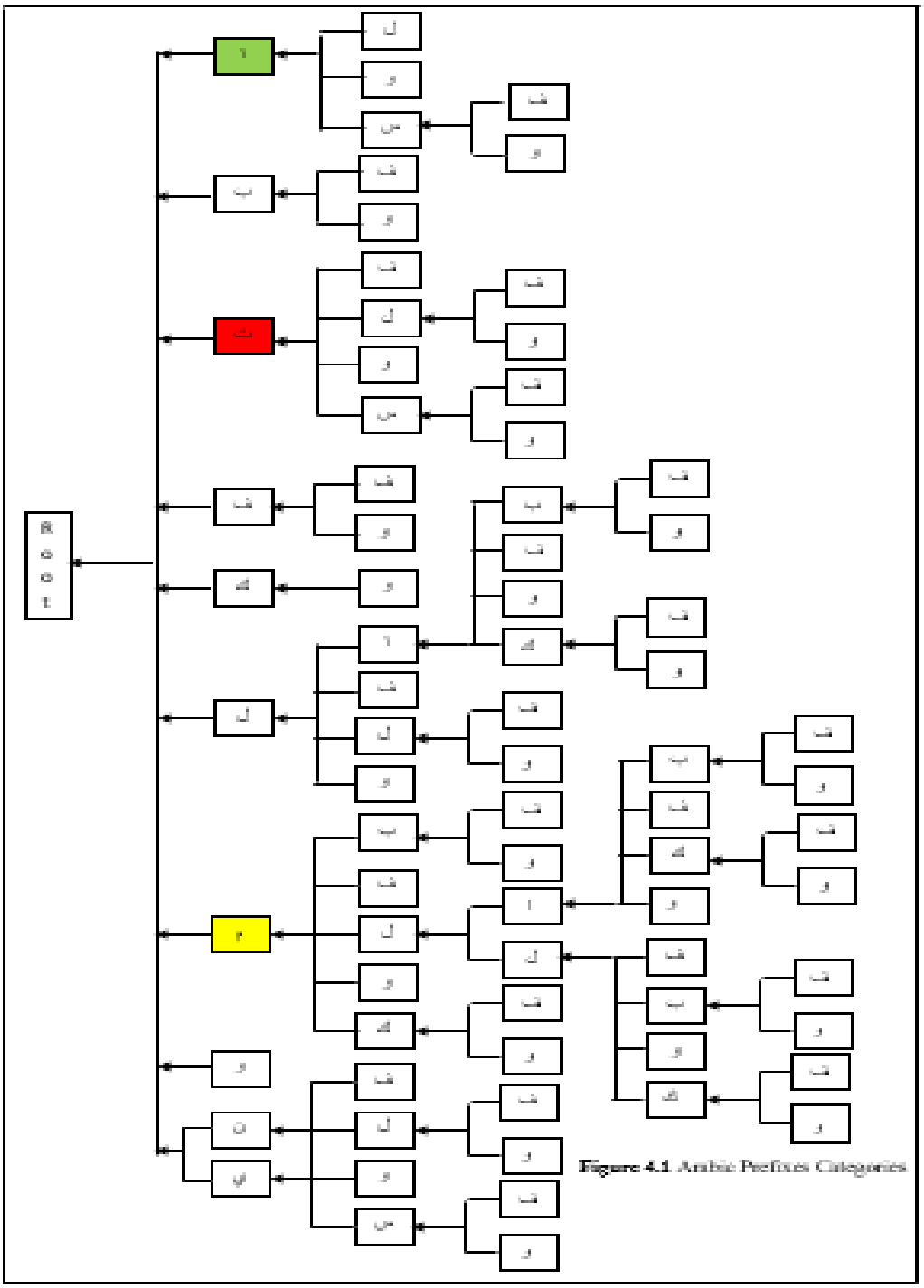

Figure 4.1 Arabic Prefixes Categories

The perfective form is used suffixes to marker persons, whereas the imperfective form used suffixes to mark the persons, and used prefixes to mark the number and gender. Table 4.3. Shows an example of perfective and imperfective verbs in Arabic.

Table 4.3: Perfective and imperfective verbs in Arabic Example.

Type	Word	Person	Number	Gender	Stem	Translate
Perfective	كتبوا	1st	Singular	M	katabtu	I wrote
	كتبً	2nd	Singular	F	Katabti	you wrote
	كتبن	3rd	Plural	F	Katabna	They wrote
	يكتبوا	1st	Singular	M	yaktubtu	he writes
	تكتبوا	1st	Singular	F	taktubtu	he writes
Imperfective	نكتبوا	1st	Plural	M	naktubw	we write
	تكتبي	2nd	Dual	F	taktubA	you write
	تكتبون	3rd	Plural	M	taktubyna	they write
	يكتبون	3rd	Plural	M	yaktubw	they write

4.1.2.2 Infixes

In Arabic, affixes are divided into three groups: prefix, infix, and suffix. While infix does not exist in many other languages like English. The English language has no infixes. Infix is an affix that is inserted within a root. Although infix does exist in Arabic, it has not yet been used in most of the Arabic studies in the literature. However, he infix can be placed in the root after the first or second letters of the original root, where sequence 4.1 is applied when the infixes inserted after the first letter of the root, and sequence 4.2 is applied when the infixes inserted after the second letter of the root.

$$prefix_1 + \ldots + prefix_n + \text{T1} + infix_1 + \text{T2} + \text{T3} + suffix_1 + \ldots + suffix_m \quad \textbf{(4.1)}$$

Where N is the number of prefixes, which adding to begins of the root; T1 is the first letter of the original root; is inserting after the T1, which adds to the meddling of the

root; T2 is the second letter of the original root; T3 is the third letter of the original root; and m is the number of suffixes, which adding to ends of the root. An example of this case: the word لاعب "lAeb" (player).

$$prefix_1 + \ldots + prefix_n + T1 + T2 + infix_2 + T3 + suffix_1 + \ldots + suffix_m \quad \textbf{(4.2)}$$

Where $infix_2$ is inserted after T2, An exemple of this case: the word دروس (lessons). It is to be noted the infix that is inserted after the first letter of the root often denoted to derivational morphology, where it indicates to the adverb or noun. Whereas the infix that inserted after the second letter of the root often referred to inflectional morphology, where it indicates to number (plurals).

Arabic vowels always placed in the infix, where vowel ا (a) is placed after the first letter of root, which it indicates to the adverb or noun in this case; it also indicates to the plural if the word begins with the letter م (m) like the word مصانع (Factories) by removing infix ا (a), will result in the word مصنع (Factory); thus the word is changed from plural form to get its singular one. The vowel ا (a) if it a placed after the second letter of root, is indicated to the noun. While vowels ي (ya) and و (w) often placed after the second letter of root, which it indicates to the adverb or the plural. Therefore, in this work, we used the infix to solve problems of plural form attached in the infix. As were mentioned early, stemming affixes can give better precision in information retrieval. Therefore, we believe that by using the proposed method we will be capable to solve problems of the plural in infix in addition to plural in suffix. Figure 4.2: shows an example of Arabic vowels inserting to infix.

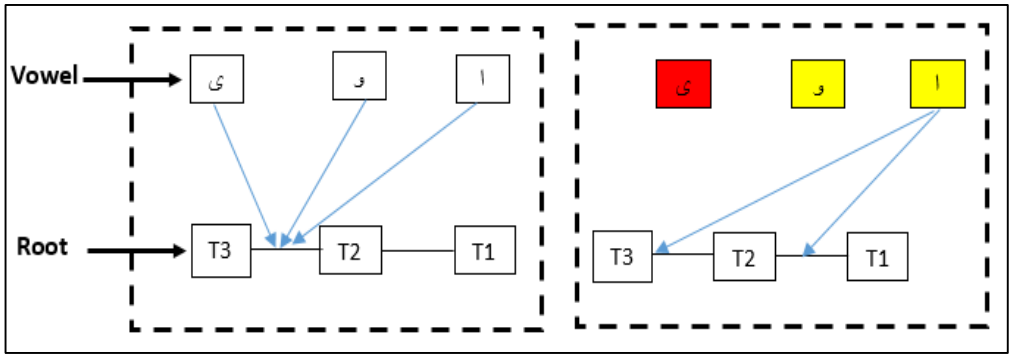

Figure 4.2 Arabic infixes Categories

4.1.2.3 Suffixes

A suffix is an affix that adding to the back of the root. The Arabic language uses suffixes just like in English. Three types of suffix can be concatenated to the Arabic root namely: gender, possessive, and number (plural in the suffix). In Arabic, the problem of the plural is difficult to deal, because no obvious rules exist, also no proposed methods stemming that can process them. Therefore, we discussed Arabic plural form, through an application of their patterns on stems (singular, dual, plural, masculine and feminine) by altering to their correct singular forms. In the AMIR rules as present in the next section, if the word ends with the suffix[12]ات (ta), it means that the suffix ات (ta) is the plural form. For instance; if we remove the suffix ات (ta) from the word `مكتبات` (Libraries), we will get مكتب (office), then change the word meaning. So, AMIR proposed altering to their plural to singular by replacing the suffix ة (taa) instead of the suffix ات (ta). Thus, if we replace the suffix ة (taa) instead of the suffix ات (ta) from the word `مكتبات` (Libraries), we will get مكتبة (Library) which singular form. Also, we will apply the same with other suffixes such as تكم (takam), تهم (taham) which refer to masculine plural; تكن (takan), تهن (tahan) which refer to feminine plural; تهما (tahama) which refer to masculine dual; تهنا (tahana) which refer to feminine dual. Thus, we replace these suffixes (ات , تكم , تهم , تكن , تهنا , تهما , تهن) from a word by the suffix ة (taa). Examples of these cases:

Singular		Plural
دراسة	➔	دراسات , دراستان
مبارة	➔	مبارات , مبارتان
مقالة	➔	مقالات , مقالتان
سيد	➔	سيدات , سيدتان
معلم	➔	معلمات , معلمتان
معلم	➔	رضات , ممرضتان

[12] Note that when replacing the suffix ات by the suffix ة, the word must contain the vowel ا at the meddle, otherwise, we should remove instead of replacing.

Figure 4.3 shows all possible suffixes that can be adding to the root. It's clear from figure 4.3 that there are one to four suffixes that can be adding to each root. And also shows clearly, how suffixes put together to construct word (each word may contain more than one suffix). In Arabic, a derivational morpheme is often not placed at ends of the word, while inflectional morpheme is often placed at ends of the word. Therefore, Existing Arabic stems produce some error-rates due to stem may remove the original letter of the word, which definitely will lose the meaning of the word. Therefore, AMIR addresses this problem by generating different Arabic stem during this affixes that presented in Figure 4.1 and Figure 4.3.

The construction of Arabic words is based on abstract forms known as roots. A root, in Arabic morphology, is the most basic word that serves as a base to generate other derivatives or inflective obtained by blending prefixes, infixes, and suffixes on the root to produce adverbs, verbs, nouns, and adjectives. Figure 4.4 shows the AMIR process to generate or extract a word.

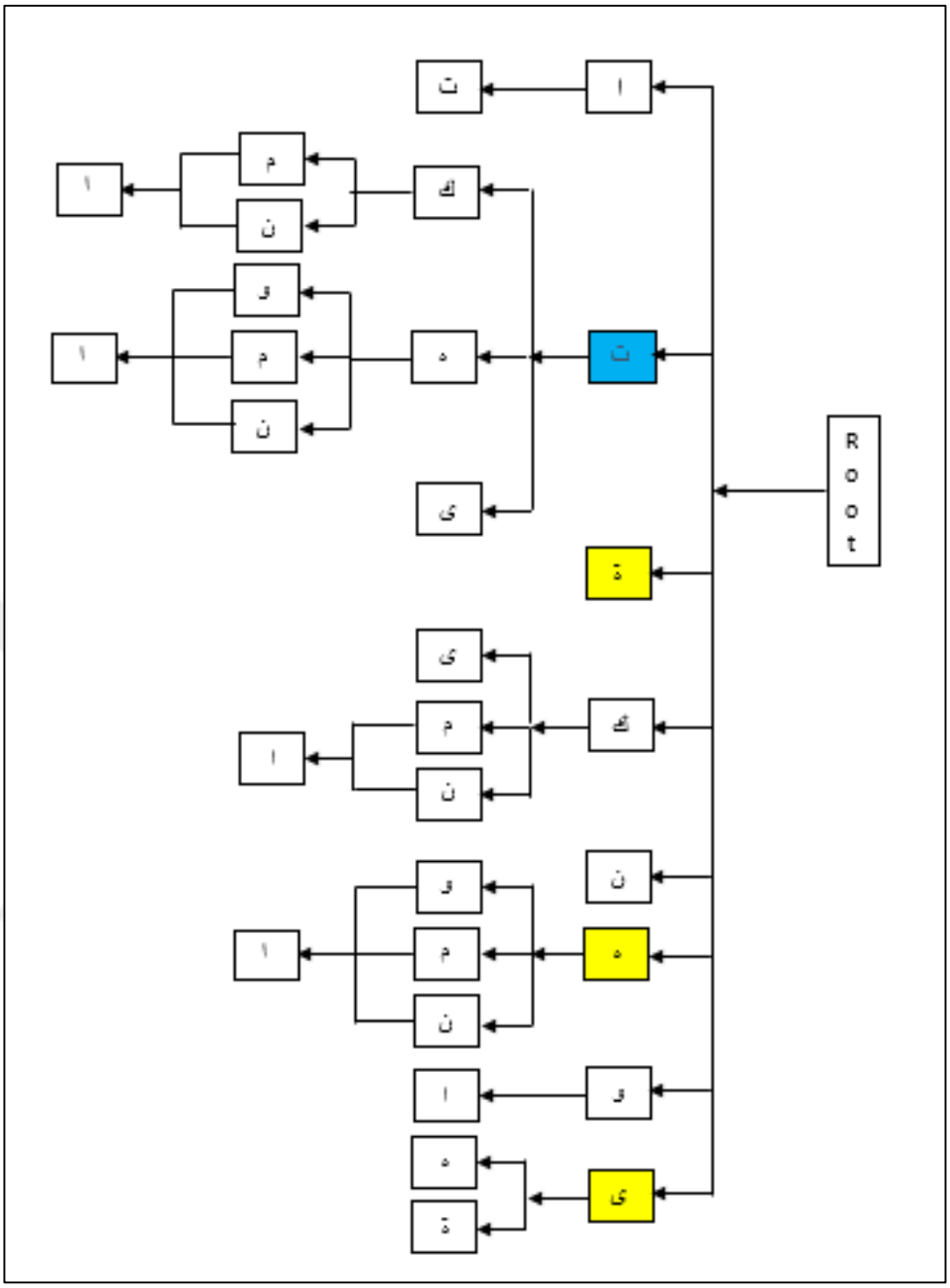

Figure 4.3: Arabic Suffixes Categories.

To describe the next Figure 4.4, we say that there are two phases' uses to generate or extract stem: the first phase is to add patterns. This process is done by adding patterns to the root. These patterns are highlighted in green, yellow, red, and blue in Figure 4.1, Figure 4.2, and figure 4.3. Note that patterns that highlighted in yellow in figure 4.1 can be conjunctions with patterns in figures 4.2 and figures 4.3. This means that patterns highlighted in each color may conjunctions to other patterns to the generator word. For

example prefix, م (m), which highlighted in yellow in Figure 4.1 may be linked with infix ا (a) highlighted in yellow in Figure 4.2, or the suffix "ة" (taa), which highlighted in yellow in Figure 4.3.

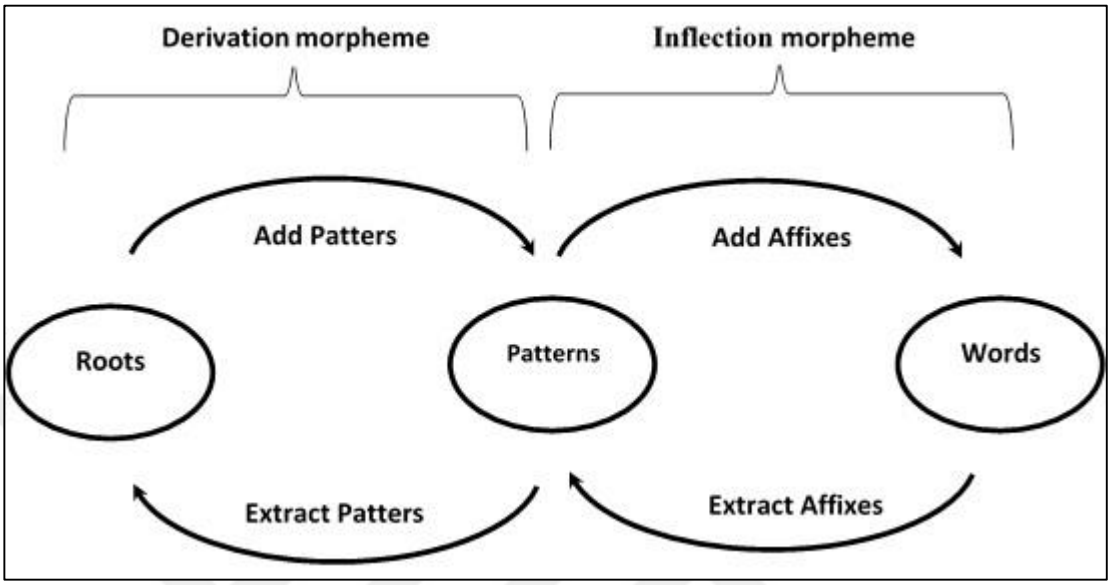

Figure 4.4 AMIR process to generate/extract stem or root from the word[13]

This means that AMIR can generate or extract stems by applying a set of rules regarding the relationship among Arabic letters to find the root or stem of the respective words.

Figure 4.1, 4.2, and 4.3 shows the list of affixes that add to root, to generator words. Therefore, we will use sequence 4.1 and sequence 4.2 to adding or inserting the following affixes:

- ✓ 64 Morphemes list are added to begin of the root to generator a word, which shows in Table 4.4.
- ✓ There are three morphemes are add to meddles of the root, which is: ا (alif), و (waaw), and ي (yaa).
- ✓ 30 Morphemes lists are added to the ends of the root to generator words, which shows in Table 4.5.

[13] AMIR is capable to generate 1400 word from each root by adding all possible affixes that presented in figures 4.1, 4.2, and 4.3.

Table 4.4 shows a list of prefixes that can be added to the root to generate words.

Table 4.4 Arabic stemmer prefixes list

كم	اي	ات	ان	وت	ون	وا	ا
يه	فب	وب	ب	وي	ي	بن	فن
فلل	لل	فل	ول	ل	ة	ه	ية
ال	است	وسي	سي	وست	ست	فا	ولل
فم	وم	م	وكال	كال	بال	فال	وال
وبالم	فللم	وللم	للم	فلم	و لم	لم	بم
وسن	سن	فت	وت	ف	وب	ب	فبالم
فكال	فبال	وبال	بال	فا	وك	ك	فسن

Table 4.5 shows a list of suffixes that can be added to the beginning of the root to generate Arabic words. Note that plural and numbers often placed at the ends of the Arabic word.

Table 4.5 Arabic stemmer suffixes list.

ه	و	ون	وا	ات
تها	تك	تي	ان	ة
كما	كن	كم	هن	هم
تهن	تكي	تكم	هما	كنا
يه	ا	كي	تهما	تكما
تان	ن	ك	ت	ية

Affixes[14] can associate with each other to generator more words as follows:

Prefixes with Infixes → (a)
Prefixes with Suffixes → (b)
Infixes with Suffixes → (c)
Prefixes with Infixes, and Suffixes → (d)

Note that, not all combinations of affixes can be joined together, there are some combinations of affixes that are not permitted. In case (a), 7 prefixes that cannot join with infixes. In case (b) there are no exceptions, all prefixes can join with all suffixes. In case (c) 24 exceptions that are not permitted. In case (d) affixes can be linked together if not found in the exceptions above. This exception motivates the following definitions.

Definition 1. The morphological structure of derivational word is:

Derivational = (adverb + root) | (particle + adverb + root) |

(particle + noun + root + possessive pronouns) |

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

(particle + subjective +root)

Definition 2. The morphological structure of inflectional word is:

Inflectional = (particle + root) |

(particle + root + possessive _pronouns) |

(root + possessive pronouns)

Derivational structures of Arabic consist of prefixes and infixes in derived words. While inflectional structures consist of affixes including prefixes, infixes, and suffixes.

[14] Note that affixes in Figures 4.1, 4.2, 4.3, can be linked together to generate more words

AMIR is Rule-Based stemming, which uses morphological rules to produce a high-performance technique to generate Arabic stems by implementing a morphological analysis using specific linguistic, which remove only affix that not change the meaning of the word, whereas essential affix that may change the meaning of the word should not be removed if any are exist. Therefore, the AMIR method is capable to improve the extraction root in the Arabic language, and this is a major improvement in previous methods.

4.2 AMIR Rules

AMIR rules are bottom-up and rule-based dictionary based on the Arabic Morphology analysis. However, AMIR rules can be classified into two categories: the first thing it attempts to do is finding substrings of words which are mostly stems or in other cases morphemes that get derived from stems. The next process is to join each core with word elements, therefore generate words according to the governing rules. Finally, the Rules check to ensure that each core exhibiting at least a correct generation is passed as a correct, and consequently, the individual stem is also the correct stem of the given word. AMIR rules are composed of three phases, which include: Rule matching, Substring tagging, and Anti rule matching, as shown in figure 4.5 below:

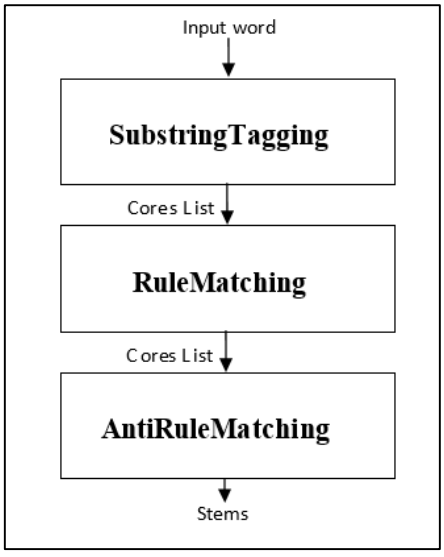

Figure 4.5 AMIR rules steps

Substring tagging, in this phase, morphological information that characterizes possible substrings of respective words is extracted. Based on the results, we can accurately know which word substrings are morphemes and substrings, which are not morphemes. This phase is also instrumental in ensuring that we know the clusters of each morpheme. The clusters are used in the rule matching phase.

Rule matching, in this phase, each core that has been extracted from the substring tagging phase, we can know the rules used in extracting. Finally, the last phase is instrumental in extracting anti rules from the anti-rules-based repository for every core in the given list. What then follows is that it is ensured 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.

As were mentioned early, AMIR Rules are constructed from different Arabic grammatical rule-based according to morphological analysis. Therefore, these rules depend on modifying the word into an appropriate stem. So, selecting rule depends on special letters adding/inserting to the root. Table 4.6 shows an 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.

Table 4.6 An intelligent use of morphological analysis and stem in Arabic information retrieval systems using AMIR Rules

Rule	Syntax	Description
R1	Prefix م (m) + Root ⟶ Noun	In Arabic, some prefixes indicate nouns, such as, If we adding prefix م (m) to the root, then will change word type to the noun (which refers to a person). For instance, if we adding prefix م (M) to the root `درس` (study), we will get 'مدرس' (Teacher). Thus, these prefix م (M) must not remove in derived words whereas we remove other extra affixes if any found.
R2	Prefix م (m) + Root + Suffix ة (taa) ⟶ Noun	As indicated above in rule (R1). prefix (m) denoted to noun, but there are some conjunctions in Arabic like prefix with suffix can be joined together to indicates to the noun; such as: if we adding prefix م (M) and suffix ة (Taa) to the root, this we will produce noun (This noun often indicates to a place in the most cases).

Table 4.6 (continued**) A**n intelligent use of morphological analysis and stem in Arabic information retrieval systems using AMIR Rules

Rule	Syntax	Description
R3	Prefix م (m) + T1 + Infix ا (a) + T2 + T3 ⟶ Noun	In Arabic, plurals can be identified by the middle or ends of the word. However, it's easy to remove plural in the suffix because removing plural in the suffix will not change the meaning of the word. While removing plural in the infix in some cases to change the meaning of the word. Therefore, this rule says that the infix ا (a) denoted to noun if the word does not contain prefix م (m), thus, these prefix ا (a) should not remove from the word. Whereas if the word contains prefix (m), in that case, the infix ا (a) refers to a plural noun. For example, if we remove infix ا (a) from the word `مكاتب` (Offices), we will get مكتب (office). Thus, we reduce them to their singular form.
R4	T1 + T2 + Infix و (w) + T3 ⟶ Noun	As indicated above in the rule (R3), plurals can be identified by the middle of the word by adding infix that indicates plural after the second letter of the root. Therefore, this rule tells us that if we insert Infix و (w) after the second letter of the root. This will produce a plural form of the noun. For example, if we remove infix و (W) from the word `دروس` (lessons), we will get درس (lesson). Thus, we reduce them to their singular form by remove infix و (W). So, if a word contains the infix و (W) in derived word, this means that it should be removed from a word.
R5	T1 + Infix ا (a) + T2 + T3 ⟶ Noun	This rule says If we insert infix ا (a) after the first letter of the root. This will produce a noun. For example, if we remove infix ا (a) from the word `كاتب` (author), we will get كتب (he wrote). However, word type and meaning have changed. So, the infix ا (a) is indicates the noun. Thus, infix ا (a) should not be removed from the word because it will change the word meaning.
R6	Prefix ت (taa) + T1 + T2 + Infix ي (y) + T3 ⟶ Noun	In this rule, derivational produce a noun based on the relation between letters. So, if we adding prefix ت (ta) and infix ي (ya) after the second letter of the root, this will produce a noun. For example, if we adding prefix ت (ta) and insert infix ي (ya) to the root `درس` (study), this will produce the word `تدريس` (Teaching), Therefore, we should not remove prefix ت (ta) and infix ي (ya) if they found in the same word.
R7	T1 + T2 + Infix ا (a) + T3 + suffix ة (taa) | suffix ات (at) | suffix تان (tan) ⟶ Plural	This rule of derivation is called replacement الإبدال. So, this operation is often done at the ends of the word. Thus, the advantage of the replacement is that it can be addressed the problem of the plural form in suffix. Hence, If a word contains the infix ا (a) and ends by suffix ات (at) OR suffix تان (tan), which indicates to plural. In this case, the suffix ات (at) and suffix تان (tan), it should replace by suffix ة (taa). For example, the word دراسات (studies), and the word دراستان (two studies), which indicates to plural by replacing the suffix ات (at) and suffix تان (tan) with suffix ة (taa), So, we will get the word دراسة.

Table 4.6 (continued**) A**n intelligent use of morphological analysis and stem in Arabic information retrieval systems using AMIR Rules

Rule	Syntax	Description
R8	T1 + T2 + Infix ا (a) + T3 + suffix ية (ya) ⟶ Plural Noun.	As indicated in the above rule (R7), replacement operation can lead to the right formula of the stem. So, in this rule, we will replace suffix ية (yat) or suffix ية (yah) by suffix ي (ya). For example, if we remove suffix ة (taa) from the word `دراسية` (things for study), we will get دراسي.

According to the rules in table 4.6, we can easily extract stem or root from the word, for example, AMIR system removes any prefixes that placed before the prefix م (m) that means it's not necessary to check whether the word contains the prefixes و , لل , ال, .. because it always indicates to definitions, prepositions, and stopwords that not change the word meaning. While the light stemmer system extract stem or root from the word by a checklist of prefixes if the word begins with to remove. So, the advantage of the AMIR system is that Arabic word can contain more there prefixes in the same word, thus by applied AMIR rule which suggest removing any prefixes that comes before the prefix م (m) whereas the light stemmer often removes only one prefixes from the word.

4.3 Conjunctions in Arabic

Conjunctions[15]in Arabic is called "حروف العطف", which uses to link between words in a sentence such as prepositions. In Arabic, the prepositions can be writing separately or linked directly with the same word. While in English words prepositions must writing separately. The following table 4.7 shows a list of all conjunctions in Arabic.

Table 4.7 List of conjunctions in Arabic.

ثُمَّ	ف	و
لا	أم	بل
حتي	لكن	او

[15] A conjunction in Arabic is making a word follow another one by using a preposition.or stopword or definitions.

On the other hand, in the Arabic language, the most common conjunction uses are prefixed ف (so, then), and prefixed و (and), which can be writing separately or linked directly with the word, and also prefixed ف and prefixed و gives the most general way to join between words in a sentence.

The following two examples demonstrate how the conjunctions use to join between words in a sentence:

Example 1: جاء احمد ومحمد Ahmed and Muhammad came

The prefix و (and) is used to combine two words with each other.

. Example 2: اجتهد محمد فنجح Muhammad works hard, so he succeeded

The prefix ف (so) is used to combine two words with each other.

Example 3: ليأخذ الطالب العلم من المعلم The student takes knowledge from the teacher

The prefix ل (for) is used to combine two words with each other.

Figure 4.6 on the next page, we have shown an example of the most oft-used conjunctions in the holyQuran[16], which highlighted in yellow from 5:91-95.

[16] Note that conjunctions in the holy**Quran** are a particle that connects two words together. The most common conjunctions in the Quran is the prefixed particle wa, this means in English as "and" as highlighted in yellow in figure 4.6.


إِنَّمَا يُرِيدُ ٱلشَّيْطَٰنُ أَن يُوقِعَ بَيْنَكُمُ ٱلْعَدَٰوَةَ وَٱلْبَغْضَآءَ
فِى ٱلْخَمْرِ وَٱلْمَيْسِرِ وَيَصُدَّكُمْ عَن ذِكْرِ ٱللَّهِ وَعَنِ
ٱلصَّلَوٰةِ ۖ فَهَلْ أَنتُم مُّنتَهُونَ ﴿٩١﴾ وَأَطِيعُوا۟ ٱللَّهَ وَأَطِيعُوا۟
ٱلرَّسُولَ وَٱحْذَرُوا۟ ۚ فَإِن تَوَلَّيْتُمْ فَٱعْلَمُوٓا۟ أَنَّمَا عَلَىٰ رَسُولِنَا
ٱلْبَلَٰغُ ٱلْمُبِينُ ﴿٩٢﴾ لَيْسَ عَلَى ٱلَّذِينَ ءَامَنُوا۟ وَعَمِلُوا۟ ٱلصَّٰلِحَٰتِ
جُنَاحٌ فِيمَا طَعِمُوٓا۟ إِذَا مَا ٱتَّقَوا۟ وَّءَامَنُوا۟ وَعَمِلُوا۟ ٱلصَّٰلِحَٰتِ
ثُمَّ ٱتَّقَوا۟ وَّءَامَنُوا۟ ثُمَّ ٱتَّقَوا۟ وَّأَحْسَنُوا۟ ۗ وَٱللَّهُ يُحِبُّ
ٱلْمُحْسِنِينَ ﴿٩٣﴾ يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا۟ لَيَبْلُوَنَّكُمُ ٱللَّهُ بِشَىْءٍ
مِّنَ ٱلصَّيْدِ تَنَالُهُۥٓ أَيْدِيكُمْ وَرِمَاحُكُمْ لِيَعْلَمَ ٱللَّهُ مَن يَخَافُهُۥ
بِٱلْغَيْبِ ۚ فَمَنِ ٱعْتَدَىٰ بَعْدَ ذَٰلِكَ فَلَهُۥ عَذَابٌ أَلِيمٌ ﴿٩٤﴾ يَٰٓأَيُّهَا
ٱلَّذِينَ ءَامَنُوا۟ لَا تَقْتُلُوا۟ ٱلصَّيْدَ وَأَنتُمْ حُرُمٌ ۗ وَمَن قَتَلَهُۥ
مِنكُم مُّتَعَمِّدًا فَجَزَآءٌ مِّثْلُ مَا قَتَلَ مِنَ ٱلنَّعَمِ يَحْكُمُ بِهِۦ ذَوَا
عَدْلٍ مِّنكُمْ هَدْيًۢا بَٰلِغَ ٱلْكَعْبَةِ أَوْ كَفَّٰرَةٌ طَعَامُ مَسَٰكِينَ
أَوْ عَدْلُ ذَٰلِكَ صِيَامًا لِّيَذُوقَ وَبَالَ أَمْرِهِۦ ۗ عَفَا ٱللَّهُ عَمَّا
سَلَفَ ۚ وَمَنْ عَادَ فَيَنتَقِمُ ٱللَّهُ مِنْهُ ۗ وَٱللَّهُ عَزِيزٌ ذُو ٱنتِقَامٍ ﴿٩٥﴾

Figure 4.6 Conjunctions in the holy Quran[17]

As later subsections will explain, there are two types of conjunctions in the Arabic language, which are the prepositions and the stop-words that may be linked directly

[17] There are different conjunctions in the holy Quran whose syntactic diversity requires that a lot of attention be paid to Arabic syntax.

with the word. This is why most of the existing schemes cannot authenticate whether the removed letters are the roots or not. This is because it's hard to extract stem from the word that may contain four prefixes in the same word such as the word ففالمدرسة (so the school). Therefore, AMIR addressed this problem by generator all conjunctions words in Arabic to add these conjunctions to the AMIR dictionary. Therefore, the AMIR technique attempts to extract the Arabic root/stem based on a validation of the letters before removing affixes. The next two subsections present the most conjunctions in Arabic, which as follows:

4.3.1 Stop-word

Stop-words are words that have little semantic meaning that is removed from the index and the query because they have a very high frequency that can affect the retrieval effectiveness. Arabic language allows writing stop-words such as "و" (and) separately or linked with the word. Wherever the root letters are called "صحيح" as reported in [63] which means the root letters do not contain any hamzah or vowel, For instance; the stop-word و which means "and" in the English language. It may be writing separately like the word "و كتب" (and he wrote), or it may be linked directly to the word like "وكتب".

As were mentioned early, it is not easy to extract the stop-words "و" (and) if linked directly with the word. This is because there are so many Arabic roots are begin with letters و (and). So this is the challenge that we trying to addressing in this work since a stop-word is may be linked directly with the word. Hence, in this thesis, the AMIR system removes stop-words based on a validation of the letters before removing affixes by utilizing the AMIR dictionary.

In Arabic, many root's contents begins with letter و (and), or ل (for), etc. for instance, if we remove و (and) from the word `ورثكم` (inherited from someone), we will get رثكم (the word does not have meaning because the word has to remove an essential letter from the root). So, AMIR removes unnecessary affixes only that not changing the meaning of the word. Also, the verbs of which the root starts with و (and) this و (and) is dropped in the present tense as reported in [64].

The most popular web search on the internet is Google. So, we used Google to search for the derived words that have a root begins with و (and). Hence, we chose the word `ورثكم` (inherited from someone), which contents the root ورث (he inherited) and suffix كم (kam), which indicates to plural. So, when we search the `ورثكم`; by using Google, we got 7,310 results. Whereas, when we search the root `ورث`. we got about 2,990,000,000 results, and they're relevant results. This means that Google removing affixes from a word without distinguishing whether the removed letters are core letters of the root or not. Therefore, AMIR method increases the efficacy of extracting a word effectively while minimizing the ambiguity since it depends on validation before removing affixes where each input term is compared against with all the words in the AMIR dictionary until a match is found; if no root is found, the original word is returned unchanged.

4.3.2 Prepositions

A preposition is a word used to link nouns, pronouns, or phrases to other words such as the word `فاجْتَنِبُوا` (to avoid). This word linked directly to the word. Therefore, these problems cannot be solved by previous studies, because the word contains two prefixes which are the prefix ف (in) and prefix ا (a). Thus, the AMIR scheme generated all possible prepositions that linked directly with the word. Therefore, we will use these words when we extract stem from the word easily by using AMIR rules. Now, it's easy to deal with prefixes that indicate to prepositions or stop-words if AMIR rules applied. This is because AMIR seeks the word in the AMIR dictionary if it found or not. For example, AMIR rule No R1 says that it must remove all prefixes that placed before م (M), for example:

The prepositions and stop-words always placed at begins of the word in the Arabic language., so, it's easy to remove it by drop all prefixes that refer to stop-words and prepositions according to use AMIR rules[18] more detail in the following examples:

Ex.1 Remove preposition:

فمصنع → مصنع

Ex.2 Remove stop-word:

ومنزل → منزل

Ex.3 Remove preposition:

فالمصنع → مصنع

Ex.4 Remove stop-word:

والمنزل → منزل

Ex.5 Remove preposition:

فللمصنع → مصنع

Ex.6 Remove stop-word:

وللمنزل → منزل

[18] These two examples are using AMIR rule No 1 to extract stem, see AMIR rules in section 4.2.

CHAPTER 5

A Brief Survey of Information Retrieval Models

Chapter Overview

Retrieval models are one of the essential concepts in the information retrieval system, and also retrieval models are used to find the top-k answers to a given user query. It is not a trivial task to select the most important/relevant results from a huge amount of documents. One of the most important models used in information retrieval systems is a statistical model that includes the vector space model. This model measures the similarity between the query and the documents in the collection, it considering the distinct query terms and the distinct terms in each document to occupy n-dimensional vectors, where n is the number of unique terms in the collection, and then documents are ranked according to this similarity score.

Another popular model used to represent documents and queries is the language model. This model is a statistical language model such that each document is viewed as a language model. Indeed, the statistical language model is a probability distribution over all possible words in a language. Another important tool in retrieval models is learning to a rank strategy which becomes very popular in recent years. This strategy learns a ranking function using implicit or explicit relevance data. In the following subsections, we give the details of each different approach.

In this chapter we will present some basic retrieval models which are organized as follows: In Section 5.1, we describe Vector Space Model; which used to determine whether the document is relevant/matched or not relevant/not matched to the user's query. In Section 5.2, we will describe the Language Model that assigns probabilities to sequences of words. Then in Section 5.3, we will present the N-grams Language model, which is a set of N-items, and also in this section, we will explore Unigram, Bigram, and Trigram. After that In Section 5.4, we will do a comparison between the

Vector Space Model and Language Model. And in Section 5.5, we will describe the Query likelihood Language Model, The logic behind this model is to compute the probability of producing/generating the terms. Lastly in Section 5.6, we will present in the short Translation Model.

5.1 Information Retrieval Models

Information retrieval models have many approaches and techniques have been developed, and also have a long history. In traditional information retrieval models, matching between each document and query. Thus, one issue is to select an appropriate model for the representation of user queries and documents. So, it should be capable to determine relevant documents in terms of similarity between queries and documents. Basically, the information retrieval model describes the document representation, the user query representation and the retrieval process. However, one of the most important problems in Arabic information retrieval systems is to find similar articles/documents for a newly submitted query. So that the user can quickly obtain the answer. There are different methods such as vector space models, language models, translation models, and word kernels are proposed before. Therefore, in the next sections, we will briefly describe these models. Indeed, information retrieval models can be divided into three groups: probabilistic model, vector space model, and Boolean model as shown in Figure 5.1.

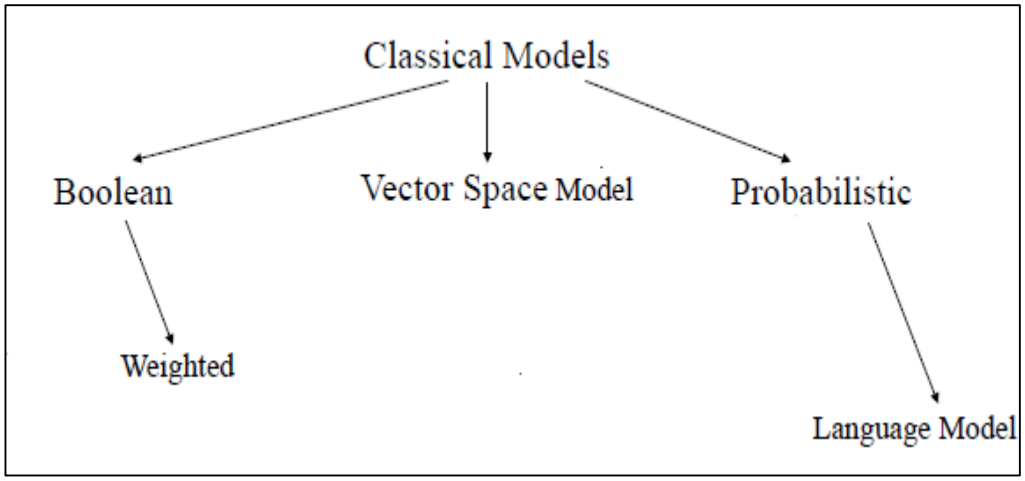

Figure 5.1 Information retrieval models

Boolean model[19] is an information retrieval model that uses set theory to request any Boolean combination of descriptors by using the operators: AND, OR, and NOT for query formulation. Whereas, Weighted Boolean model is an extension of the Boolean Model with weights, where Weight denoting the representatively of a term for a document. While the vector space model requests a set of descriptors each of which has a positive number associated with it which will describe in detail in the next section.

5.2 Vector Space Model

Vector Space Model VSM is one of the most important models in the information retrieval model to representing text documents, which start from document indexing the terms that are extracted from the document text, and weighting of the indexed terms to enhance retrieval of document relevant to the user query. After that, rank the document according to a similarity measurement. In addition to that, the vector space model represents documents and user queries by vectors in a multidimensional space. For instance, $d_j = (w_{1,j}, w_{2,j}, .., w_{t,j})$, and $q = (w_{1,q}, w_{2,q}, \dots, w_{n,q})$. Each dimension corresponds to a term vocabulary that is used to build a list (index) of terms. Note that the morphological analysis reduces different word forms to be indexed as described in the previous chapter. In the VSM, a collection of documents can be represented by a term-document matrix, where each cell in the matrix corresponds to the weight of a term in the document if a term (word) appears in the document. This tells us that the value in the vector is non-zero, which means similarity does exist and zero otherwise. Furthermore, the performance of the VSM depends on the term weights scheme, that is, the functions that determine the components of the vectors is the TF-IDF weighting scheme, it's one of the most widely used term weighting schemes as reported in [65]. TF-IDF weighting scheme product of two statistics, term frequency (TF) and inverse document frequency (IDF) to compute the weight of a term for a document, where (TF) is the number of times a term occurs in a document. Traditional TF-IDF uses term frequencies and document frequencies to generate a

[19] Note that weighted boolean has a logical formula, which similar to the boolean model. The probabilistic model function is to rank documents by their probability of relevance based on the user's query.

weighted term that is used for document representation [66]. Therefore, the (TF) weights for the terms can be computed as the following formula:

$$TF_{ij} = \frac{f_{ij}}{max_{ij}} \quad \textbf{(5.1)}$$

Where f_{ij}a is a number of times term i appears in document j, and max_{ij}denotes the maximum frequency of the term in document j.

IDF is a measure of what portion of the document collection contains the term. Therefore, the IDF values for the terms can be defined as follows:

$$IDF_i = \log_2\left(\frac{N}{df_i}\right) \quad \textbf{(5.2)}$$

Where N is a total number of documents and i is a term in the collection, df_i is a number of documents that contain the term i.

The VSM is used to determine whether the document is relevant/matched or not relevant/matched to the user's query. So, the standard way of quantifying the similarity (also called cosine similarity) between each document vector and the original query vector where the query is represented as the same kind of vector as the documents $\vec{V}(d)\ and\ \vec{V}(q)$. This task can be done by computing their cosine of the angle between two vectors (document and query vectors). Therefore, if these vectors are related, the cosine of their angle will be one, and if the vectors are orthogonal (unrelated), the cosine of their angle will be zero, which is defined as follows:

$$\cos(\theta) = \frac{d \,.\, q}{||d||\ ||q||} \quad \textbf{(5.3)}$$

Where d is a document vector, q is a query vector, θ is the angle between two vectors, $||d||$ is the norm of vector d, $||q||$ is the norm of vector q. The norm of the vector is defined as the following:

$$||q|| = \sqrt{\sum_{i=1}^{n} q_i^2} \quad \textbf{(5.4)}$$

The resulting scores can then be used to select the top-scoring documents for a query.

The steps of the VSM can be represented as follows:

- Preprocessing which converts documents and query to a vector of all distinct terms.
- Construct the document term matrix.
- Computing TF-IDF weights.
- Convert each document to a weighted vector and also construct the query vector, then compute similarity scores between document and query vectors.
- Ranking the documents based on the similarity score, in descending order and return the top-k documents as the query results.

An example of the vector space model, suppose that we have a collection C that consists of three documents and we needs to answer query Q as shown below:

D1: ارض قمر سماء "land moon sky"

D2: سماء نجوم فضاء "space stars sky"

D3: سحاب ارض شمس "sun land cloud"

Q: ارض ارض شمس "sun land land"

There are three documents D1, D2, and D3, each of them contain terms, some appear only in one document and other appear in two documents, the similarity between each document and query can be computed as shown in following steps:

Step1: We counted the number of times of all distinct terms (word) appears[20] in a document by calculating the TF scores for each word, and their frequency on each of the document, by using formula 5.1 as shown in Table 5.1. We supposed the words in the vectors are ordered alphabetically.

Table 5.1: Term frequency (TF)

	ارض	سحاب	سماء	شمس	فضاء	قمر	نجوم
D1	1	0	1	0	0	1	0
D2	0	0	1	0	1	0	1
D3	1	1	0	1	0	0	0

[20] Note that 0 that means did not appear in the document and 1 otherwise.

Step 2: We computed the term appears across the index by calculating the IDF for terms occurring in all the documents, by using formula 5.2 as shown in Table 5.2. The total number of documents is N=3.

Table 5.2 Inverse document frequency (IDF)

Term	**IDF**
ارض	$\log_2 (3/2) = 0.58$
سحاب	$\log_2 (3/1) = 1.58$
سماء	$\log_2 (3/2) = 0.58$
شمس	$\log_2 (3/1) = 1.58$
فضاء	$\log_2 (3/1) = 1.58$
قمر	$\log_2 (3/1) = 1.58$
نجوم	$\log_2 (3/1) = 1.58$

Step 3: we computed the weight of a term (w_{ij}) this model is known as term frequency–inverse document frequency model, by multiply the TF scores with the IDF values of each term, which defined as: $\mathbf{w_{ij} = TF_{ij} * IDF_i}$ [21], both TF_{ij} and IDF_i were defined in steps 1, 2. Hence, Table 5.3 shows the obtained matrix of documents-by-terms.

Table 5.3: TF.IDF weights.

	ارض	سحاب	سماء	شمس	فضاء	قمر	نجوم
D1	0.58	0	0.58	0	0	1.58	0
D2	0	0	0.58	0	1.58	0	1.58
D3	0.58	1.58	0	1.58	0	0	0

[21] TF-IDF provides which words have the most important within the text documents.

Step 4: We calculated the TF-IDF vector for the query term as shown in Table 5.3. Therefore, we divided the frequency by the maximum frequency (2), then multiply with the IDF values.

Table 5.4 Q weight

	ارض	سحاب	سماء	شمس	فضاء	قمر	نجوم
Q1	(2/2)*0.58 = 0.58	0	0	(1/2)*1.58 = 0.79	0	0	0

Step 5: We calculated the length of each document and of the query by using formula 5.4 as follows:

Length of **D1** = sqrt (0.58^2+0.58^2+1.58^2)= 1.78

Length of **D2** = sqrt (0.58^2+1.58^2+1.58^2)= 2.31

Length of **D3** = sqrt (0.58^2+1.58^2+1.58^2)= 2.31

Length of **query** = sqrt (0.58^2+0.79^2)= 0.98

Step 6: Finally, we computed the cosine similarity values between the query vectors by calculating the cosine of the angle between two vectors, query by using formula 5.3 as shown on the next page.

The similarity[22] value between D1 and query is:

cos(D1,q) = (0.58*0.58 + 0*0 + 0.58*0 + 0*0.79 + 0*0 + 1.58*0 + 0*0) / (1.78*0.98) = 0.19

The similarity value between D2 and query is:

cos(D2,q) = (0*0.58 + 0*0 + 0.58*0 + 0*0.79 + 1.58*0 + 0*0 + 1.58*0) / (2.31*0.98) = 0

[22] Note that, if a query term does not appear in a document, then the probability value becomes zero. So, in this case, smoothing methods should be applied to avoid zero probability problems.

The similarity value between D3 and query is:

cos(D3,q) = (0.58*0.58 + 1.58*0 + 0*0 + 1.58*0.79 + 0*0 + 0*0 + 0*0) / (2.31*0.98) = 0.69

Step 7: we ranked search results concerning the query amongst three documents according to the similarity values. The ranking of documents based on decreasing cosine similarity will be: D3, D1, and D2.

Thus, the vector space model is a simple linear algebra-based model and allows computing the lexical similarity between queries and documents. Additionally, it can compute partial matching for documents and produce a ranking according to the relevance. The disadvantage of the model is that it cannot compute semantic similarity between queries and documents because it relies on the exact word matches and does not account for semantically similar but different words such as "car" and "automobile".

5.3 Language Model

Language model LM is a statistical model in the Natural Language Process NLP task that is used to compute the probability of any context like sentence or sequence of words. Typically, models that assign probabilities to sequences of words are called language models. The language model is a function that puts a probability model measure over strings drawn from some vocabulary.

Most of the language model approach to rank the documents by the probability of the query which is defined as $P(Q|M_d)$. Each sentence/sequence of words can give different probabilities. Therefore, documents are ranked according to the highest probability of the query. On the other hand language model computes the probability of a sentence or sequence of the word of length n, it assigns a probability as follows:

$$P(t) = P(t1, t2, \ldots, t_n) \quad \textbf{(5.5)}$$

For terms t the following is the probability of these terms appearing in the order of "t1, t2 ..,t_n" based on some corpus. For example, the language model given sentence/sequence of words like “its water is so transparent” as follows:

P (its water is so transparent) = **p** (its) **p** (water |its) **p** (is |its water) **p** (so |its water is) **p** (transparent | its water is so)

The language model steps can be defined as below:

- Estimate a document LM by estimating the probability of each word.
- Computing the query likelihood.
- Rank documents based on likelihood probability which is relevant of user's query.

5.4 N-grams

N-gram models are widely used in statistical natural language task, which basically is a set of N-items such as sentence/sequence of words as shown the following formula:

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_1 | w_1, \ldots, w_{i-1}) \quad \textbf{(5.6)}$$

Where W is word and $\mathbf{W_m}$ is total number of words appear in the sequence. N-grams are used to predict the previous words (N-1) in a sequence to predict the next word (N).

An n-gram of size is defined as follows:

- N=1 → this is known as Unigrams which is essentially the individual words in a sentence.
- N=2 → this is known as Bigrams.
- N=3 → this is called Trigrams, and when N>3 this is sometimes referred to as four grams or five grams and so on. The formula for all the above models can be defined as the following next page:

Unigram as the following formula:

$$P(w_{1,n}) = P(w_1)P(w_2) \ldots . P(w_1) \quad \textbf{(5.7)}$$

Bigram as the following formula:

$$P(w_{1,n}) = P(w_1)P(w_2|w_1) \ldots . P(w_n|w_{n-1}) \quad \textbf{(5.8)}$$

Trigram: as the following formula:

$$P(w_{1,n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1,2}) \ldots . P(w_n|w_{n-2,n-1}) \quad \textbf{(5.9)}$$

An example of the **N-grams**[23] model, suppose we have the sentence "Welcome to YILDIRIM BEYAZIT University Ankara"

If **N=2** (Bigrams) Then, we have five n-grams in this case, which is described as the following:

Welcome to

to YILDIRIM

YILDIRIM BEYAZIT

BEYAZIT University

University Ankara

[23] The **n-grams** typically are collected from a text that are phrases cut out of a sentence with N sequential words.

Let me explain with an example.

Unigram - [Let] [me] [explain] [with] [an] [example.]

Bigram [let me] [me explain] [explain with] [with an] [an example]

Trigram [let me explain] [me explain with] [explain with an] [with an example]

If **N=3** (Trigrams) Then, we have four n-grams in this case, which shown as following:

Welcome to YILDIRIM

to YILDIRIM BEYAZIT

YILDIRIM BEYAZIT University

BEYAZIT University Ankara

5.4.1 Unigram

Unigram is a probability distribution over individual words included in the text (which also known as the bag of words model), which means it does not depend on any of its previous words. Unigram computes as: the probability of word i = Frequency of word (i) / total number of words. For example, the probability of the sentence: "sea east, sea west". By reference to the formula No. 5.7 we compute the probability of "sea east, sea west" as below:

P (Sea, east, Sea, west) = P (Sea) P (east |Sea) P (Sea |Sea, east) P (west |Sea, east, Sea).

Above sentence contains three words, the probability of each word independent as below:

P (sea) = 2/4

P (east) = 1/4

P (west) = 1/4

P (sea east west) = P (sea) x P (east) x P (west) = 2/4 x 1/4 x 1/4 = 0.31

P (sea east) = P (sea) x P (east) = 2/4 x 1/4 = 0.13

5.4.2 Bigram

A Bigram is an n-gram for N=2, this model used to predict the conditional probability based on the previous word, which is also called Markov assumption, assumes that we can predict the probability of the next word by only looking at the previous word, as shown follows:

$$Bigram = \mathrm{P}(W_i|W_1\ W_2\ \ldots\ ,W_{i-1}) \approx \mathrm{P}(W_i\ |W_{i-1}) \quad \textbf{(5.10)}$$

The simplest way to estimate bigram probability is by using Maximum Likelihood Estimation (MLE), based on taking counts from the corpus and normalizing them. The general equation for estimating the probability for an MLE Bigram is presented as follows:

$$\mathrm{P}(\mathrm{W_i}\ |\mathrm{W_{i-1}}) = \frac{\mathrm{Count(W_{i-1}\ ,W_i\ )}}{\mathrm{Count(W_{i-1}\ )}} \quad \textbf{(5.11)}$$

Where count (W_{i-1} , W_i) is a number of times word $\mathbf{W_{i-1}}$ followed by word $\mathbf{W_i}$ and Count(W_{i-1}) is the number of times word $\mathbf{W_{i-1}}$ appears in the corpus. For example, Let assumes that our corpus is: "I live in Turkey, I am a graduate student at YILDIRIM BEYAZIT university in Ankara and I learned the Turkish language in active language center about three days per week". We will try to compute the probability **P(I am graduate student at Ankara)** by **Bigram** language model, note that n**o**rmalizes means count each term in the corpus as shown below:

Table 5.5 Query weight

I	**Am**	**Graduate**	**Student**	**at**	**Ankara**
3	1	1	1	4	1

Bigram counts as shown in the next Table 5.6.

Table 5.6 Bigram counts

	I	Am	Graduate	Student	at	Ankara
I	0	1	0	0	0	0
Am	0	0	1	0	0	0
graduate	0	0	0	1	0	0
student	0	0	0	0	1	0
At	0	0	0	0	0	1
Ankara	0	0	0	0	0	0

Result: by refer to the formula (5.11)

ex : $P(\text{ am }|\text{I }) = \frac{Count(I\ am)}{Count(I)} = 1/3 = 0.333$

Bigram = P (am | I) x P (graduate | am)x P(student | graduate) x P(at | student) x P(Ankara|at)

= 0.333 * 1 * 1* 1 * 0.25 = 0.08325

5.4.3 Trigram

A Trigram is an n-gram where N=3 and this model is mostly used in the past two decades in literature. Trigram model computes the probability of the next word based on only (the most recent) previous two words, as the following formula:

$$\text{Trigram} = \text{P}(W_i\ |W_{i-2}, W_{i-1}\ ) = \frac{\text{Count}(\text{W}_{\text{i}-2},\text{W}_{\text{i}-1},\text{W}_{\text{i}})}{\text{Count}(\text{W}_{\text{i}-2},\text{W}_{\text{i}-1})} \quad \textbf{(5.12)}$$

The probability of word W_i depends only on two previous words, W_{i-2} and W_{i-1} as shown in the formula No.5.9. For example: Let the same example in Bigram to compute the Trigram

P **(I am graduate student in Ankara)**

By refer to the formula (**5.9)**

$$\textbf{P (graduate |I am)} = \frac{\textbf{Count(I am graduate)}}{\textbf{Count(I am)}} = 1/1 = 1$$

Trigram = P(graduate | I am) x P(student | am graduate)x P(in |graduate student) x P(Ankara | student in) . Trigram=1 * 1 * 1 * 0 = 0

There are several ways of avoiding zero probabilities and making the estimates for n-grams. This is known as smoothing methods. However, smoothing is a technique to give some probability massing (unseen) words to n-grams that have not occurred in the corpus. Therefore, the probability of unseen trigram in the corpus is zero. Thus, to avoid unseen trigram in the corpus needs a smooth estimate of the probability of unseen words. This can be done by linear interpolation of a trigram, bigram, and unigram frequencies and also a uniform distribution on the vocabulary as shown below:

$$P(Wi\,|W_{i-2}, W_{i-1}) = \lambda_1 P(Wi\,|W_{i-2}, W_{i-1}) + \lambda_2 P(Wi\,|W_{i-1}) + \\ + \lambda_3 P\,(Wi\,) \quad \textbf{(5.13)}$$

Where $\sum_i \boldsymbol{\lambda_i} = \mathbf{1}$

$$\text{P(graduate| I am)} = \boldsymbol{\lambda_1} \frac{Count(I\ am\ graduate)}{Count(I\ am)} + \boldsymbol{\lambda_2} \frac{Count(\ am\ graduate)}{Count(\ am)} + \\ \boldsymbol{\lambda_3} \frac{\boldsymbol{C(graduate)}}{\sum_w \boldsymbol{C(W)}}$$

P(graduate |I am) = 0.90 * 1 + .05 * 1+ .05* (1/29) = 0.951724137

P(student | am graduate) = 0.90 * 1 + .05 *1 +.05 * (1/29) = 0.951724137

P(at |graduate student) = 0. 90 * 1 + .05 *1 + .05* (1/29) =0.951724137

P(Ankara | student at) = 0.90 * 0 + .05* 0.25 + .05 * (1/29) = 0.014224137

= 0.951724137 * 0.951724137 * 0.951724137 * 0.014224137

= 0.01226939

5.5 BM25 Model

In information retrieval, BM25 stands for Best Matching BM (it is also known as Okapi BM25) is a ranking function model used by search engines to rank matching documents according to their relevance to a given search query to retrieve relevant information search, It is developed in the 1980s by [67]. The BM25 is a retrieval model based on the probabilistic retrieval, where BM25 performs very well in many ad-hoc retrieval tasks. However, The BM25 is considered as the state-of-the-art model. it a retrieval model based on the probabilistic retrieval. However, both BM25 and TF-IDF[24] using two main components: TF and IDF, where the BM25 model often achieves better performance as compared to TF-IDF. Thus, in this thesis, we have used both BM25 and TF-IDF to rank matching documents according to their relevance in addition to Language model LM. The implementation of BM25 by using the LUCENE open-source search library, a widely-deployed variant. The result obtained of this study proves that the BM25 performance better than LM.

[24] TF*IDF is a way of approximating how the word is important in the relevance of the text.

5.6 Comparison between Vector Space Model and Language Model

In this section, we have compared two information retrieval models which are the Vector Space Model VSM and the Language Model LM. Therefore, VSM and LM which are the most popular information retrieval models currently use. They are closely related, a vector in the VSM and a probability distribution in the LM are similar in containing the term weights although they have very different mathematical intuitions. Also, there is only one difference that a probability distribution is normalized to sum to one, while a vector does not have such a requirement. The two models provide similar functionality (represent documents and weight terms).

In VSM, the document and the query are represented as a vector, and also the VSM ranks documents based on the vector space similarity between the query vector and document vector. The way to compute the similarity between vectors is by computing cosine similarity. So, the cosine similarity between two vectors or two documents on the vector space is a measure that calculates the cosine of the angle between them. In addition, VSM uses linear algebra tools to model the documents and terms. A document is represented as a vector and the terms are its elements. While the Language Model is a branch of probabilistic models, where a document is viewed as a language model, which is essentially a probability distribution over its terms. While the LM was first used in natural language processing to model the probability of a sequence of words. Therefore, the LM ranks the documents according to their probabilities to generate the query terms. Thus, we found that both used a similar functionality, which represents documents and weight terms. So, the difference between them in the methodologies, wherein the LM, a probability distribution is normalized to sum to one, while in the VSM, a vector does not have such a requirement. Therefore, two models provide similar functionality and closely related, a vector in the VSM and a probability distribution in the LM.

5.7 Query likelihood Language Model

Query likelihood Language model is a language model used in information retrieval, where the language model constructed query likelihood from each document in the collection. So, the logic behind the query likelihood language model is to compute the

probability of producing/generating the query terms under each document language model. Documents are then ranked based on the highest probability of the query in the document. Therefore, we need to rank documents given the query q, and also we need to compute P(d|q), by the Bayes rule as shown follows:

$$P(d\,|q) = \frac{P(q|d\,)P(d)}{P(q)} \quad \textbf{(5.14)}$$

Where P (q), probability of the query, is the same for all documents and P (d) is a probability of the document. So, we can ignore P (q) expression in the above formula because it is the same for all documents. Then the prior probability of each document P (d) could be considered as equal. Therefore, we need to compute only P (q|d). So, for each language model constructed for each document, we compute the probability of producing/generating the query q. This is interpreted as being the likelihood of a document being relevant to a given query. It is possible that some terms in a query do not occur in documents. Even though other terms of the query appear in the document, the probability of generating the query will be zero because of the zero probability of non-existing terms. Jelinek-Mercer is one of the smoothing methods. This method involves a linear interpolation of the maximum likelihood model with multiple documents. So the formula of the Jelinek-Mercer smoothing No.5.15 is suitable only when you have one document in the corpus as follows:

$$P(wi) = \lambda 1\, P_{ml}\,(wi) + (1 - \lambda 1)\frac{1}{N} \quad \textbf{(5.15)}$$

Where P_{ml} ($\boldsymbol{wi}$) is the maximum likelihood of word w_i, which computed by the ratio of the frequency of w_i to the total vocabulary size ($\frac{\boldsymbol{C(w)}}{\boldsymbol{N}}$**).** Here $\boldsymbol{\lambda 1}$ is the smoothing factor.

The following formula No.5.16 is suitable when you have many documents in the corpus.

$$P\,(w|\,d) = (1 - \lambda)\,\frac{C(w,d)}{|\,d\,|} + \lambda\, P(w\,|c) \quad \textbf{(5.16)}$$

For example: Suppose we have a collection containing more than one document and the number of total words in document d1 is 100**.** Therefore, in table 5.7 on the left

side, we give the words in d1 along with the frequency of each word in decreasing order. In table 5.8 on the right side shows the probability[25] of each word in the collection C (P(w | C)). Assume that our collection contains 10,000 words.

To compute P("text" | d1) and P("network" | d1) using the Jelinek-Mercer smoothing method. This formula No.5.16 is suitable when you have many documents in the corpus.

Table 5.7 Example of word frequency

d 1	
Item	No. of times term appears in the document
Text	10
Mining	5
Association	3
Database	3
Algorithm	2
Query	1
Efficient	1
…	
…	

Table 5.8 Probability of each word

Collection	
Item	P(w|C)
The	0.1
A	0.08
Computer	0.02
Database	0.01
Text	0.005
…	
…	
Network	0.001
Mining	0.0009
…	
….	

We computed the probabilities by using the formula No.5.16 as shown below:

P("text" | d) = (1- λ) 10 /100 + λ ∗ 0.005

P("network" | d) = (1- λ) 0 /100 + λ ∗ 0.001

If we set smoothing factor by $\lambda = 0.05$, then

P("text" | d) = 0.95*0.1 + 0.05*0.005 = 0.095+0.00025 = 0.09525.

P("network" | d) = 0.95*0 + 0.05*0.001 = 0.00005.

[25] Note that even though the term "network" does not found in d1, its probability is not zero due to smoothing.

5.8 Translation Model

Translation Model can bridge the lexical gap between the query and the document, and it can capture the lexical similarity by a translation model. And also it can generate query words not in a document by translation to alternate terms with similar meaning. (Berger and Lafferty) proposed a very important extension to the basic exact matching query likelihood function by allowing the query likelihood to be computed based on a translation model of the form p(u|v), which gives the probability that word v can be *semantically translated* to word u. The query likelihood is computed in the following way:

$$P(Q|D)= \prod_{i=1}^{m} \sum_{W \in V} p(q_i|w)p(w|D) \quad \textbf{(5.17)}$$

Where p(qi|w) is the probability of translating word w into qi. This translation model can be understood by imagining a user who likes document **D** would formulate a query in two steps: In the first step, the user would sample a word from document **D**; in the second step, the user would *translate* the word into possibly another different but semantically related word. In our case, we do not have two different languages but we need related sentences, phrases (or queries), and documents to be mapped in the same language. One way to obtain such data is to pair queries and their related documents but this requires too many relevance judgments. They first choose query terms from documents and use them as a parallel corpus to learn translation probabilities.

CHAPTER 6

System Evaluation

Chapter Overview

In this chapter, we analyzed and evaluated the effectiveness of AMIR that produces intelligent stem that is used to perform very searches in retrieving Arabic information, AMIR stem effect is very large as compared to that existing in many other stem studies. So, there are different search topics are to be queried. Therefore, two methods were used to compare evaluate the quality of AMIR namely LUCENE and FARASA to ascertain which method can produce high-performance stem than others, based on the outputs mean average precision. Consequently, the primary evaluation measure we have used in this work is the mean average precision MAP, which is one of the most commonly used metrics in information retrieval, in addition to and precision P at 10 and P at 20 to analyze the change in retrieval precision, this could be evidence that using AMIR stem yields a much-improved precision. Furthermore, the measures were computed with Trec_eval software as presents in Section 6.1.3. Trec_eval tool which is commonly used in the TREC community for evaluating an ad hoc retrieval run. So, in this work, the performance measures used Trec_eval 9.0 to evaluate rankings by relevance judgments for each query. On the other hand, in our experiments, the retrieval performance of the proposed method AMIR has been compared with the LUCENE stem, FARASA stem, and No stem using the BM25 model and language model LM with Dirichlet. In addition to that, the TFIDF weighting also was used to evaluate the quality of our scheme performances, which is a very popular weighting in information retrieval. So, TFIDF delivers results that are highly relevant score to a query, to compute the term weight for each term in the document to estimate whose frequencies are highly relevant to a query. Therefore, in this chapter, we're going to look at the experiments of the proposed approach and results, which we conducted for evaluating our method on the retrieval performance in Arabic.

This chapter is mainly system evaluation and will be organized as follows: section 6.1 provides experiments and results, which divided into five subsections, in the

subsection 6.1.1 we will present a brief overview of the dataset that we will use of measuring the effectiveness of our system, then in the subsection 6.1.2 we described various measures that use to evaluate retrieval, and in the subsection 6.1.3 we provided evaluate the performance of our approach, then in the subsection 6.1.4 We present results obtained, in the subsection 6.1.5 we will describe our result, and in Section 6.2 we briefly show comparison our approach with two proposed methods.

6.1 Experiments and results

In this section, we simply plan to verify the effectiveness and the quality of AMIR performed with relevance judgments. be note that the AMIR system was implemented using Java language, the main reason for choosing Java was huge flexibility in the software creation and the experiments were performed on a PC with windows 10 Pro operation system, 1.70GHz Intel(R) Core(TM) i5-3317U processor, 4 GB of RAM, and 500 GB HDD. So, we present an overview of the tests performed, as follow:

6.1.1 Dataset

The experiment was carried out with Arabic Test collections EveTAR on tweets [68] that are comparable to similar Text Retrieval Evaluation Conference TREC. Test collections EveTAR are evaluation tools that are essential for advancing the state-of-the-art in the field of Arabic information retrieval that supports multiple information retrieval EveTAR includes a crawl of 355M which contained roughly 61946 articles on an Arabic tweet represented in Unicode and encoded in UTF-8, and covers 50 significant events for which about 62K tweets were judged with the substantial average inter-annotator agreement. Then, we will compute a certain number of measures the performance of information retrieval using Trec_eval[26]tools available on the internet to measure MAP P.5, 10, 20, 50. Finally, we compare the results obtained by our method with FARASA (Darwish, 2016) and LUCENE (Core, SolrTM, PyLucene). The result of the experiments (for each document collection) can be seen in the results section.

[26] See like: Trec_eval tools (https://trec.nist.gov/trec_eval/)

6.1.2 Measures

In the literature, we have seen various methods to test the effectiveness of the Arab information retrieval system from relevant and irrelevant documents. Typically evaluation measures are computed across multiple queries and averaged to produce a final score. Therefore, the primary evaluation measure used in this work is the mean average precision (MAP), in addition to the precision at 10 (P@10) and precision at 20 (P@20) to analyze the change in retrieval precision. And also we used TF.IDF weight to evaluate the quality of our scheme performances. So, how the measurements were calculated that used in our experiments will be present as follows:

Precision P is the total number of predicted positives values (TP) as the ratio of a true positive. They are also, defined as the following formula 6.1:

$$P = \frac{TP}{TP+FP} \quad \textbf{(6.1)}$$

Where TP is True Positive, FP is False Positive, which will be using to shows quantifies how good our model is effectiveness performing the query.

Mean Average Precision MAP which is defined as the following formula 6.2:

$$MAP = \frac{\sum n_{k,j}\, AveP(q)}{Q} \quad \textbf{(6.2)}$$

Where **Q** is a collection of a set of queries, and AveP(q) is the average precision for a given query.

Term frequency (TF), which can be calculated by the following formula 6.3:

$$TF_{i,j} = \frac{n_{i,j}}{\sum n_{k,j}} \quad \textbf{(6.3)}$$

Where $\mathbf{n_{i,j}}$ **is** a number is that term t_i occurs in the document d_j. and $\sum \mathbf{n_{k,j}}$ is the sum of the number of the term t_i occurs in all terms in document d_j

Inverse Document Frequency (IDF) which can be calculated by the following formula on the next page:

$$\mathrm{IDF_t} = \log(\frac{N}{n}) \quad \textbf{(6.4)}$$

Where N is a number of the documents and n is a number of the documents with term t.

Term Frequency and Inverse Document Frequency (TF-IDF), which can be calculated by the following formula:

$$\text{TF-IDF} = \mathrm{TF_{i,j}} *. \mathrm{IDF_t} \quad \textbf{(6.5)}$$

Where $\mathrm{TF_{i,j}}$ and $\mathrm{IDF_t}$ is computed by reference to the formula **4.3** and the formula **4.4**.

The best ways to measure the performance of the retrieval system is to capture statistics and metrics, thus we have been used two statistical methods as follows:

6.1.2.1 Using statistical metrics

We have employed TREC_EVAL software, which is a tool used to evaluate rankings information, to measure precision @ 10, precision @ 20, and Mean Average Precision MAP as evaluation metrics. There are two different files that TREC_EVAL uses, first is the quels file that is a human-generated file that tells whether a retrieved document is relevant or not for each query, according to the following format delimited by spaces. The format for the quels file[27] as follows:

query-id 0 document-id relevance

Where query-id is referring to the query number, document-id is to identify the document, and relevance is to identify the judged document (0 means non-relevant and 1 for relevant).

Figure 6.1: Shows an example of a snippet from the qrels file.

.

[27] Note that the quels file is a set of relevance judgments of which documents are relevant, which include around 61946 relevant judgment for each query.

E01	Q0	5501800098735104 00	0
E01	Q0	550187841100070913	0
E01	Q0	550200917908148224	0
E01	Q0	550202458442461184	0
E01	Q0	550212751239680000	0
E01	Q0	550212775021379584	0
E01	Q0	550214252146229248	1
E01	Q0	550214533952724993	1
E01	Q0	550214847833862146	1
E01	Q0	550214867937153024	1
E01	Q0	550215016591683586	1
E01	Q0	550215753849647104	1
E01	Q0	550216115058524160	1
E01	Q0	550216499416555520	1
E01	Q0	550216953915129858	1
E01	Q0	550217788699062273	1
E01	Q0	550218073664258049	1
E01	Q0	550219444610293760	1
E01	Q0	550219846848233472	1
E01	Q0	550220253284667392	1
E01	Q0	550220483942449152	1
E01	Q0	550220568763453440	0
E01	Q0	550220921852936192	1

Figure 6.1 An example of a snippet of the qrels file

The second file is the results file, which contains a ranking of documents according to higher scores for each query. We have created the results file by using Java language according to the following format delimited by tab spaces. The format for the results file as follows:

<query_id>, <document No>, <rank>, <score>, <system >

Where "query-id" refers to the query number; document-id is to identify the document number; rank is showing the document position in the ranking; the score is to indicate the similarity degree between query and document, which is sorted by relevance, this

in means the most relevant document will have higher scores; lastly system is referred to system name like AMIR. The next Figure 6.2 shows an example of a snippet from the result files generated by using the AMIR system.

E06	Q1	5517279472842342 40	1	23.271646 AMIR
E06	Q1	551735429813534721	2	23.271646 AMIR
E06	Q1	551712909454503936	3	23.271646 AMIR
E06	Q1	551669742034227201	4	23.271646 AMIR
E06	Q1	551662046774034432	5	23.271646 AMIR
E06	Q1	551449343400890368	6	23.271646 AMIR
E06	Q1	551415760791535618	7	23.271646 AMIR
E06	Q1	551419636357791745	8	23.271646 AMIR
E06	Q1	551385327286226944	9	23.271646 AMIR
E06	Q1	551383359293648896	10	20.4983 AMIR
E06	Q1	551465102486929409	11	20.17767 AMIR
E06	Q1	552024206993805312	12	19.456533 AMIR
E06	Q1	551838164412026881	13	19.456533 AMIR
E06	Q1	551838164768555009	14	19.456533 AMIR
E06	Q1	551728583933427713	15	19.456533 AMIR
E06	Q1	551388125562281984	16	19.456533 AMIR
E06	Q1	551392418935750657	17	19.456533 AMIR
E06	Q1	552116745780363265	18	18.028454 AMIR
E06	Q1	552116745780363265	19	18.028454 AMIR
E06	Q1	551450942361837568	20	18.028454 AMIR
E06	Q1	551697731417538560	21	16.225042 AMIR
E06	Q1	551683353918070784	22	16.225042 AMIR
E06	Q1	551685445810401281	23	15.150319 AMIR

Figure 6.2 An example of a snippet of the result files generated by using AMIR system

Figure 6.2 shown the result file that generated by using the AMIR system, where the results were ranked of documents according to the highest score similarity degree between query and document, where the highest score was 23.271646. Now, we

generated the result file for AMIR and we will do the same with FARASA and LUCENE as shown next.

E06	Q1	551465102486929409	1	20.282215 FARASA
E06	Q1	552116745780363265	2	18.132034 FARASA
E06	Q1	552116745780363265	3	18.132034 FARASA
E06	Q1	551727947284234240	4	18.132034 FARASA
E06	Q1	551735429813534721	5	18.132034 FARASA
E06	Q1	551712909454503936	6	18.132034 FARASA
E06	Q1	551669742034227201	7	18.132034 FARASA
E06	Q1	551662046774034432	8	18.132034 FARASA
E06	Q1	551449343400890368	9	18.132034 FARASA
E06	Q1	551450942361837568	10	18.132034 FARASA
E06	Q1	551415760791535618	11	18.132034 FARASA
E06	Q1	551385327286226944	12	18.132034 FARASA
E06	Q1	552024206993805312	13	15.171213 FARASA
E06	Q1	551838164412026881	14	15.171213 FARASA
E06	Q1	551838164768555009	15	15.171213 FARASA
E06	Q1	551728583933427713	16	15.171213 FARASA
E06	Q1	551388125562281984	17	15.171213 FARASA
E06	Q1	551392418935750657	18	15.171213 FARASA
E06	Q1	552099024795217920	19	15.000885 FARASA
E06	Q1	552099024795217920	20	15.000885 FARASA
E06	Q1	551707831041150976	21	15.000885 FARASA
E06	Q1	551383359293648896	22	14.967562 FARASA
E06	Q1	551659151827275776	23	12.551356 FARASA

Figure 6.3: An example of a snippet of the result files generated by using FARASA system.

As shown in Figure 6.3 which represents the result file generated by using FARASA system, the highest score was 20.282215, which is less than was achieved by the system of AMIR. Now we also generated the result file for FARASA. Therefore, we generated the result files for the LUCENE system as showing as next.

E06	Q1	551727947284234240	1	19.487757 LUCENE
E06	Q1	551735429813534721	2	19.487757 LUCENE
E06	Q1	551712909454503936	3	19.487757 LUCENE
E06	Q1	551669742034227201	4	19.487757 LUCENE
E06	Q1	551662046774034432	5	19.487757 LUCENE
E06	Q1	551449343400890368	6	19.487757 LUCENE
E06	Q1	551385327286226944	7	19.487757 LUCENE
E06	Q1	552024206993805312	8	17.203176 LUCENE
E06	Q1	551415760791535618	9	17.203176 LUCENE
E06	Q1	551388125562281984	10	17.203176 LUCENE
E06	Q1	551392418935750657	11	17.203176 LUCENE
E06	Q1	551383359293648896	12	14.149532 LUCENE
E06	Q1	551465102486929409	13	13.117601 LUCENE
E06	Q1	552116745780363265	14	12.067222 LUCENE
E06	Q1	552116745780363265	15	12.067222 LUCENE
E06	Q1	551450942361837568	16	12.067222 LUCENE
E06	Q1	551419636357791745	17	10.852409 LUCENE
E06	Q1	551378352255561731	18	10.852409 LUCENE
E06	Q1	551375356469706752	19	10.852409 LUCENE
E06	Q1	551838164412026881	20	10.652563 LUCENE
E06	Q1	551838164768555009	21	10.652563 LUCENE
E06	Q1	551728583933427713	22	10.652563 LUCENE
E06	Q1	551647626689409024	23	9.580165 LUCENE

Figure 6.4 An example of a snippet of the result files generated by using LUCENE system

Figure 6.4 which represents the result file generated by using LUCENE system, the highest score was 19.487757, which is less than was achieved by both the systems of AMIR and FARASA. Therefore. we have been generated three the result files, one for AMIR and another for FARASA and LUCENE, so now these three files evaluated by using trec_eval software as shown next page.

TREC_EVAL is a program used to evaluate ranking documents that are sorted by relevance according to the following format:

trec_eval [-q] [-a] qrels_file Resultd_file

Where trec_eval is the name of the executing program, -q is a parameter that shows all detail of queries, -m is shown only main measures like MAP, -a is a parameter that shows the summary output. Consequently, Table 6.5 showing a screenshot for trec_eval to evaluate the performance measure values (P@10, P@20, and MAP) for the AMIR system as shown in the follows.

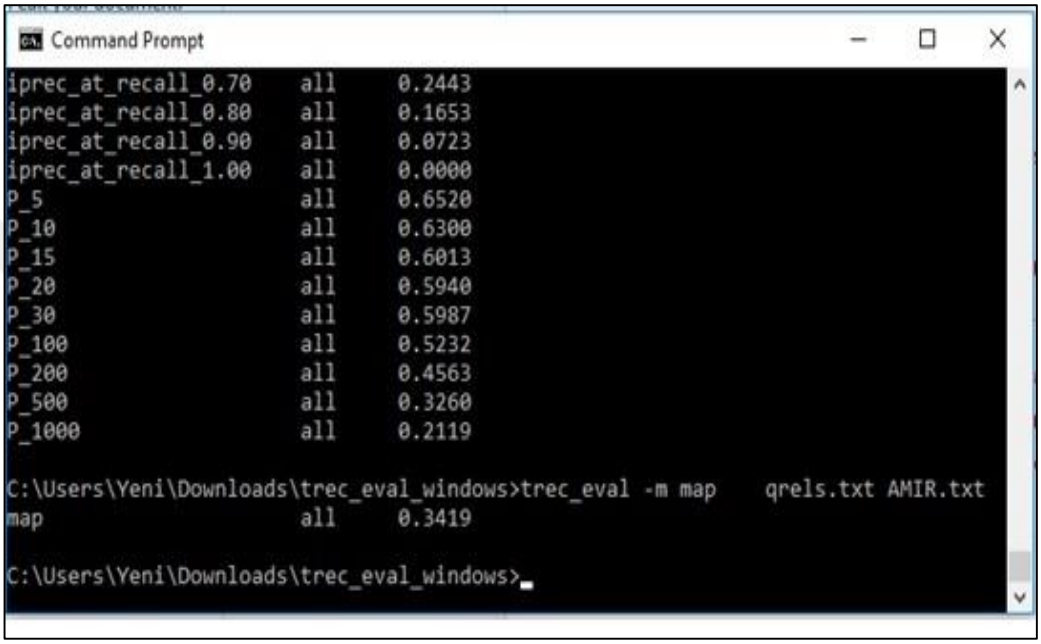

Figure 6.5 AMIR result to measure the MAP, P@10, and P@20 by using TREC_EVAL

6.1.2.2 Using frequency metrics

TF.IDF is a popular information retrieval technique, which weighs word's frequency, abbreviated as TF, which helps us to locate important terms of a document in collection for ranking purposes, and the term's inverse document frequency commonly abbreviated as (IDF). Therefore, in this work, TF.IDF has been used to evaluate the quality of our scheme performance retrieval. Thus, we compared TF.IDF score of our scheme AMIR with LUCENE and FARASA for the first ten queries to find out which method has succeeded in removing all unnecessary stem such as conjunctions and plural that may change the meaning of the words or the form of the word. Consequently, we computed the term weight for all approaches by using TF.IDF to

estimate whose frequencies are high. Table 6.1 shows the summary of produced stem approaches for AMIR, LUCENE, and FARASA.

Table 6.1 Summary of produced stemmer approaches

Actual Text		AMIR stem	LUCENE stem	FARASA Stem
1	مقتل حوثيين في انفجار في اليمن	مقتل حوثي انفجار يمن	مقتل حوث انفجار يمن	مقتل حوثي انفجار يمن
2	ليتوانيا تستخدم اليورو بدل الليتاس	ليتوانيا تستخدم يورو بدل ليتاس	ليتوانيا تستخدم يورو بدل ليتاس	ليتوانيا تستخدم يورو بدل ليتاس
3	فلسطين تطلب الانضمام للمحكمة الجنائية الدولية	فلسطين تطلب انضمام محكمة جنائي دولي	فلسط تطلب انضمام محكم جناي دول	فلسطين تطلب انضمام محكم جنائي دولي
4	تحديد المشتبه بهم في هجوم شارلي إبدو	تحديد مشتب هجم شارلي ابدو	تحديد مشتب هجوم شارلي ابدو	تحديد مشتب هجوم شارلي ابدو
5	اختراق كوريا الشمالية حسابات سوني	اختراق كوريا شمالي حساب سوني	اختراق كوريا شمال حساب سون	اختراق كوريا شمالي حساب سوني
6	بناء أول كنيسة في إسطنبول قرن	بناء اول كنيس اسطنبول قرن	بناء اول كنيس اسطنبول قرن	بناء اول كنيس اسطنبول قرن
7	هجوم حزب الله على مزارع شبعا	هجم حزب الله مزرعة شبعا	هجوم حزب الله مزارع شبعا	هجوم حزب الله مزارع شبعا
8	بوكو حرام تخطف شباب في نيجريا	بوكو حرام تخطف شباب نيجريا	بوكو حرام تخطف شباب نيجريا	بوكو حرام تخطف شباب نيجريا
9	سيطرة بوكو حرام على قاعدة عسكرية في نيجيريا	سيطر بوكو حرام قاعدة عسكري نيجيريا	سيطر بوكو حرام قاعد عسكر نيجيريا	سيطر بوكو حرام قاعد عسكري نيجيريا
10	هجمات على مساجد في فرنسا	هجم مسجد فرنسا	هجم مساجد فرنسا	هجم مساجد فرنسا

6.1.3 Evaluate

Assessing system effectiveness is highly important for developing search systems (allows the measurement of how successfully meets user needs) or information retrieval systems. Therefore, we conducted a data pre-processing step for every document and query before the stem by eliminating stop-words, preposition, punctuation marks and digits. For each method, we extracted stem from a word in all

documents to construct a new document collection stem. This means that we used the AMIR Approach to extract stem from the word in all documents. So, we produced a new collection by using the AMIR and we did the same with FARASA and LUCENE. Now, we have three collections of the stem for each method. Let say that AMIR produced collection (A); and FARASA was produced collection (F); LUCENE produced collection (L). Note that we did the same steps with queries, this means that we have three sets of the queries stem, one for AMIR AQ and also FARASA FQ and LUCENE LQ. After that, we built an inverted index using LUCENE tools for each collection that has been produced (A, F, L). And then, we searched by load the queries that were constructed form each method (AQ, FQ, LQ), and then we ranking function using two different ranked retrieval models, which are BM25 and LM model to extract the highly frequents terms-weighting and ranked a list of document-scoring function that are retrieved. The results obtained from BM25 for each collection (A, F, L) we put in new files to use measurement evaluation. This means that we have now one file for AMIR and two other files one for FARASA and one for LUCENE. We also did the same with the LM model. This means that now we have six files, where three created by using BM25 for AMIR, FARASA, and LUCENE, and three by used the LM for AMIR, FARASA, and LUCENE. Finally, we used trec_eval software to calculated MAP measure and precision at 10 and at 20, to verify the effectiveness and the quality of AMIR performed. More detail, Figure 6.6 shows the steps of each search that requests/topics from the text collection: First, we denote sets of documents in the text collection: $\boldsymbol{D_1, D_2, .., D_n}$. We denote sets of queries: $\boldsymbol{Q_1, Q_2, .., Q_n}$, and extract terms as $\boldsymbol{T_1, T_2, .., T_n}$ for each query**,** then, we also denote search methods as $\boldsymbol{S_1, S_2, .., S_n}$ where a search method consists of all processing stem for each query term and documents term. Therefore, search method S can process a set of queries Q and produce a ranked list of document D hits H for each query Q. we summarized as:

$$\mathbf{(D, S, Q) \rightarrow H.}$$

Where hits H all the k documents in D appear in the k top ranks documents D. Therefore, we evaluate the system by measure the mean average precision MAP of AMIR. (See result section).

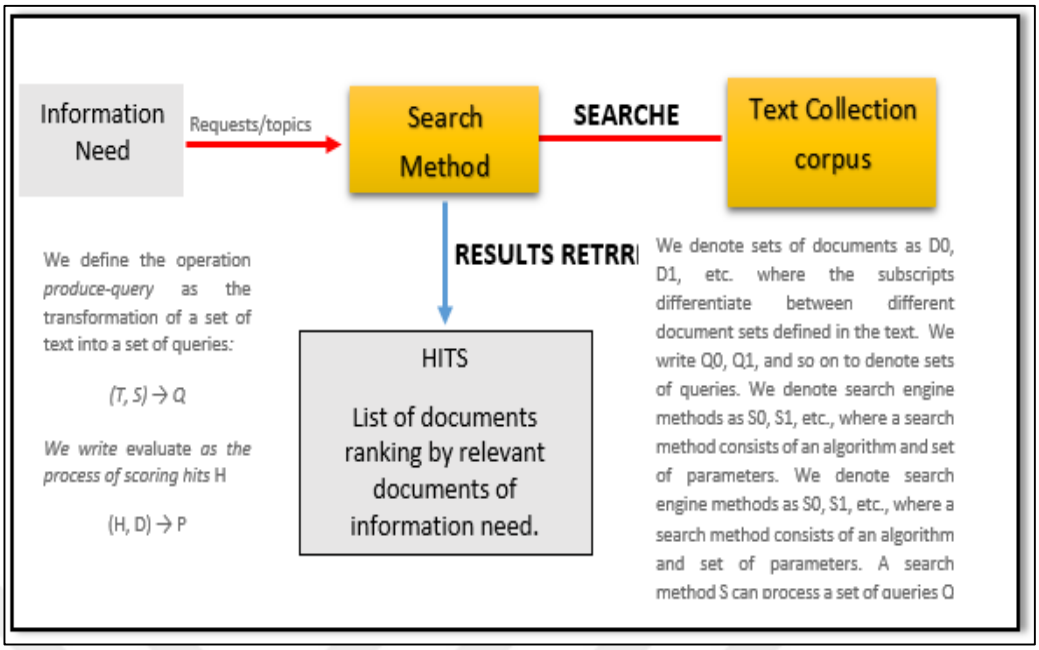

Figure 6.6 Overview of the AMIR to produce information requests/topics

6.1.4 Results

In this section, retrieval performance of the proposed method AMIR has been compared with the LUCENE, FARASA, and No stem using BM25 which is arguably one of the most widely used information retrieval functions, and language model with Dirichlet smoothing Model LM, which tries to capture the notion that some text is more likely than others [69]. Furthermore, the retrieved effectiveness was evaluated using mean average precision MAP In addition to precision at 10 (P@10) and precision at 20 (P@20) to analyze the change in retrieval precision.

Table 6.2: Summary of the results obtained for MAP by using BM25 model.

	BM25 Model		
	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 6.2 and Table 6.3 presented our experimental results, where the bold values denote the best results in each category. Thus, both Table 6.2 and Table 6.3 shows the results obtained for each system runs, 50 queries were used. These results have been analyzed in the next section.

Table 6.3 Summary of the results obtained by using LM with Dirichlet smoothing model

	LM with Dirichlet smoothing Model		
	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

Table 6.4 Query terms via TF.IDF for AMIR, LUCENE, and FARASA

Query Terms	**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 مسجداً للشيعة في باكستان	750	71	164
إعادة تشكيل مجلس الوزراء السعودي	655	34	175

The proposed method also has been used the statistical metrics to evaluate the performance retrieval. There are several term weighting schemes uses to evaluate performance retrieval proposes, but the most widely used term weighting schemes is TF-IDF. Thus, we computed the term weight for three approaches (AMIR, FARASA, and LUCENE) by using TF-IDF to estimate whose frequencies are high. So, Table 6.4 shows the achieved results of the TF.IDF.

6.1.5 Analysis Results

As were mentioned in the previous section, retrieval performance of the proposed method AMIR has been compared with the LUCENE, the FARASA, and no stem, and the retrieved effectiveness was evaluated using MAP in addition to the P@10, P@20, and TF-IDF weighs. Consequently, the results obtained of each method presents as follows: AMIR achieved a MAP value by 0.34%, while LUCENE, FARASA and no stem are 0.27%, 0.28% and by 0.21, respectively by using MB25 model. It is be 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. Also, the AMIR Accuracy of the BM25 model achieved a result of 51%, which seems a lot better than the LUCENE and FARASA algorithms where a LUCENE achieved a result of 39 % while FARASA achieved a result of 44 %. On the other hand, the AMIR achieved a MAP value by 0.32%, while LUCENE, FARASA, and no stem, are achieved a MAP value by 0.25%, 0.26%, and 0.18% respectively by using LM with Dirichlet smoothing model as shown in Table 6.3. Consequently, we found that for long queries, the BM25 model performs better than the language model LM with Dirichlet smoothing. Whereas in the short queries, the LM with Dirichlet smoothing performs better than the BM25 model. In addition to that, our evaluation results show that our approach provides high precision as compared with other methods, and also the run time of our method is slightly faster than LUCENE and FARASA methods by 172447 milliseconds, 177296, and 174585 respectively. On the other hand, the student t-test significance measure was used with p-values at or below 0.05 to claim significance to determine if the difference between the results was statistically significant or not. So, when the calculated p-value is below 0.05, it indicates that the difference between the two experimental runs is statistically significant. Therefore, the results obtained of the statistical tests show that the differences between the AMIR approach and LUCENE approach, where the p-value is 0.005508, which produced results that are statistically significant according to p-value < 0.05; and the difference between AMIR approach and FARASA approach was not statistically significant by getting P-value as 0.094249 which is greater than P>0.05. Lastly, the difference between the AMIR approach and No stem, where the p-value is 0.006334, which produced results, that are statistically significant. Thus, the

results of the statistical tests show that AMIR gives statistically significant improvements. After examining all the methods, the outcomes of the search methods are represented in Table 6.2 and Table 6.3. Therefore, the results presented in Table 6.2 and Table 6.3 clearly indicate that the proposed method is capable to solve successfully the research problems in high-performance level, so the best retrieval performance for Arabic information retrieval systems was AMIR method. After running three algorithms, we found that AMIR obtained the highest performance and LUCENE obtained lowest, whilst FARASA results fall somewhere in between. Hence, we obtained the best results and retrieval performance as compared to other systems is used.

The TF.IDF weighs of each method (AMIR, LUCENE, and FARASA) is represented all data in Tables 6.7. The results of each scheme are clear as seen in figure 6.10 which explains whether concerning terms being retrieved were relevant or irrelevant. So, Tables 6.10 clearly shows that the AMIR approach outperforms both LUCENE and FARASA counterparts, where the x-axis represents the query number, and the y-axis represents the TF.IDF score scheme that is related to that query; thus, we proposed a novel scheme that produces better approximations of the stem and the results obtained strongly indicate that the best TF.IDF values achieved when our scheme is used. Also, experimental results obtained show quantifies how good our approach is effective in performing the query.

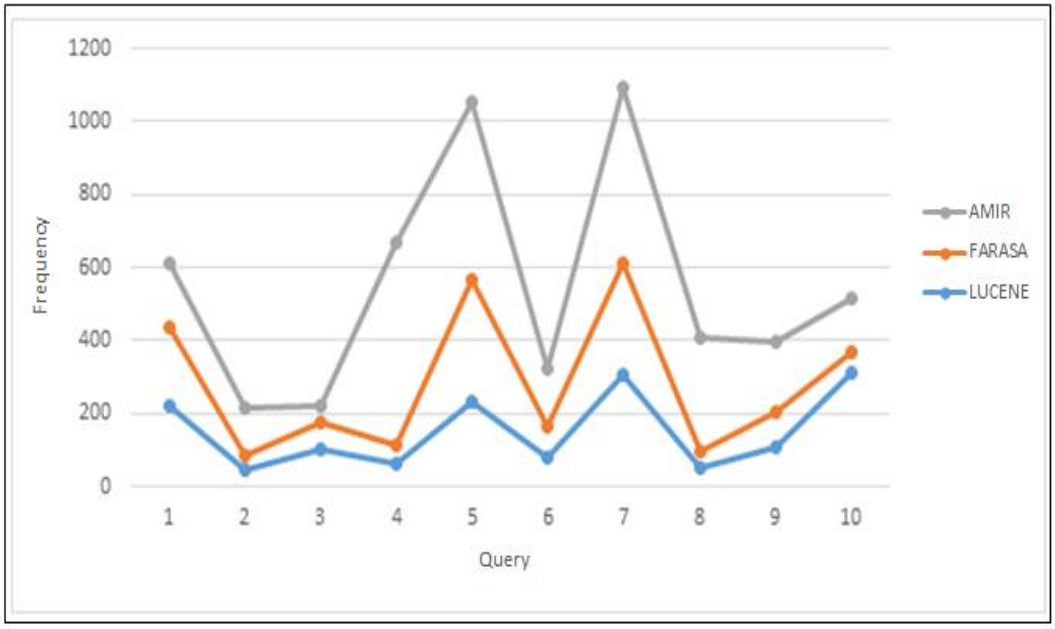

Figure 6.7 TF.IDF retrieval performance of each method.

6.2 Comparison of AMIR with LUCENE and FARASA Algorithms

In this section, we have compared the AMIR system with two counterpart systems: LUCENE and FARASA. Table 6.1 shows three methods produced stem which AMIR, LUCENE, and FARASA. So, stem can produce a big difference in weight scores. This means that if stem extract from a word correctly then the results of weight will be increased. So, as shown in Table 6.1 for query No 10, the word هجوم (attacks), which contains the infix و (ya) that indicates to plural, AMIR method is capable to remove plural in infix, then generated singular form by applying the AMIR rule No 4. As such, AMIR system extracts the word هجم (attack) instead of a هجوم (attacks) by removing the infix و (ya). While both FARASA and LUCENE extract the same word هجوم (attacks) as it is, thus failing to generate the singular form. This is because both FARASA and LUCENE do not handle plural in infix, thus resulting will be an increase in weight score.

Furthermore, there is another case of the plural in infix. This case is regarding the relationship among Arabic letters, which means if the word begins with prefix م (m) and meddles with suffix ا (a); this type of derivation is called replacement, as were mentioned early, which is not applied in LUCENE and FARASA; for example, as

shown in Table 6.1 for query No 7, the word مزارع (farms), which contains the infix ا (a) that indicates to plural, and also it contains the prefix م (m). Thus, AMIR method is capable to remove plural to get singular by removing the infix ا (a) and adding the suffix ة (taa), which will produce the word مزرعة (farm); thus, the word is changed to their singular form by applying AMIR rule No 3. Consequently, stem can give better precision in information retrieval, so AMIR system improved the precision through the use of infix extraction unlike other methods such as FARASA and LUCENE, which both returns the word مزارع (farms) as it is. This is due they not capable to solve problems of the plural in infix.

The advantage of AMIR is that it provides highly accurate results into linguistic knowledge by use morphology. The fact that this new scheme can dissect a plural word and then get its singular form. For ease of extraction gives it the precision is increased. Therefore, the best retrieval performance for Arabic information retrieval systems shows with AMIR stem.

6.3 Comparison between LUCENE, root-extraction, and AMIR Stemmer

Since Arabic is a highly inflected language, the most important research algorithms improved Arabic retrieval systems based on a morphology analyzer and light stemming. We have been investigated the effect of the morphological analysis (derivational and inflectional) on information retrieval performance. Therefore, to compare our method that uses a Rule-Based stemmer, we took a sample terms list and tested it against a root-extraction stemmer and here we used the Khoja stemmer, LUCENE stemmer method, and the result was shown in Table 6.5.

From the above results, we see that both the LUCENE stemmer and Khoja stemmer fails many times in getting the correct stem or root of the word and in many words it produced a completely new word and sometimes a wrong word that doesn't exist in the Arabic language. In addition to that, it didn't handle the broken plural forms especially in the infix like the word انهار. LUCENE stemmer method as we see it produces very general words (roots) that are far in their meaning from the original word, For example, the word الاوزان (Weights) using LUCENE stemmer will be

produded the stem which "اوز" (word have no meaning), this word does not have wrong meaning. This is because LUCENE stemmer didn't handle the broken plural forms that attached in the infix, it dealing with prefixes and suffixes only, and the other problem of LUCENE it not dealing with the technique of replacement, for example, the word دراسات (studies) where LUCNE attempts to find the stem form the word based on the stripped form, thus produced the word دراس which completely a wrong word that doesn't exist in the Arabic language, thus LUCENE failing to extract the correct stem whereas AMIR stemmer capable to dealing with replacement, so AMIR produce the stem دراسة (study) rather than دراس (word have no meaning).

On the other hand root-extraction, stemmer attempts to find roots for the words which are far more abstract than stems. It begins to remove suffixes and prefixes, then attempts to find the root form which similar to the work [70]. So, the problem in this stemming technique is that many different word formations are derived word, on the other hand, Khoja stemmer creates invalid conflation classes that result in an ambiguous query word, which leads to poor performance, for example, the word انهار (Rivers) using Khoja stemmer will be stemmed to its root which is هور (word have no meaning) where this root is not found in the Arabic language, this is because (Khoja, 1999) didn't cover all morphological rules of the Arabic language, many words with very different meanings can be formed from the same root. This is because many words are different in meaning but they originate from one identical root. Whereas our suggested stemmer AMIR produced the stems and removed all affixes effectively, thus it didn't remove them where they are part of the original word. In addition to that, thesis method handles all the broken plural forms and generates the correct singular forms, for example, the word انهار (Rivers) using AMIR produce the word نهر (River). Lastly, we found that the AMIR method able to improve and develop extract stem or root from the Arabic word, which is represented in different words forms. And also word processing that contained the infixes using the AMIR method is better than Khoja stemmer and LUCENE stemmer as well as being strong against any type of stem.

Table 6.5 Comparison between LUCENE Stemmer, root-extraction stemmer, and AMIR stemmer

Term	Stemmer (LUCENE)	Root-Extraction (Khoja)	Our System Stemmer
انهار	انهار	هور	نهر
مكاتب	مكاتب	كتب	مكتب
التدريس	تدريس	درس	تدريس
مساجد	مساجد	سجد	مسجد
الاعاب	اعاب	عيب	لعب
المسلمين	مسلم	سلم	مسلم
تعاون	تعا	عون	تعاون
المكتوب	مكتوب	كوب	مكتوب
الباب	باب	لبب	باب
التعاون	تعاون	عون	تعاون
نجوم	نجوم	نجم	نجم
الدراسات	دراس	درس	دراسة
مسافرون	مسافر	سفر	مسافر
العالمين	عالم	علم	عالم
مستشفيات	مستشف	شفي	مستشفي
مباريات	مبار	برا	مبارة
دروس	دروس	درس	درس
وتعاون	تعا	عون	تعاون
الصيف	صيف	صيف	صيف
النجوم	نجوم	نجم	نجم
الارض	ارض	ارض	ارض
البحار	بحار	بحر	بحر
فالنهر	نهر	نهر	نهر
السيارات	سيار	سير	سير
ومساجد	مساجد	سجد	مسجد
الاسواق	اسواق	سوق	سوق
يرث	يرث	ورث	ورث
المتين	متين	متين	متين
ووعد	وعد	وعد	وعد

CHAPTER 7

Conclusion and Future Work

Chapter Overview

This chapter discusses the outcome of the AMIR evaluation. It starts with evaluating the retrieval performance of the AMIR method (i.e. stem, root, word, and morphological formations). There are also different analysis methods based on the rule-based method and light stemming method, which will be discussed in this chapter. Although morphology is complex in a language, it is particularly important. For retrieval systems. This means that it has a large effect on the information retrieval system. Also, despite the infix its importance in Arabic word constructs. Infix is not classified as one of the main affixes in almost Arabic algorithms. Indeed, traditional Arabic includes plural nouns in infix and suffix. Many information retrieval systems require to extract candidate words that aid summarization and ambiguity extraction. In the proposed system, the set of morphological formation rules are used to re-produce noun plurals as singular.

As far as relevant documents are concerned, the rule-based method retrieved more documents than the other methods. This performance of morphological formation on the word and stem methods was expected, but what was not expected was that in the case of prefix stem the morphological formation retrieved more relevant documents than the light stemming method did, which was developed in this work. So, in theory, the light stemming method is expected to retrieve more relevant documents than the other methods. However, the output of the rule-based method and light stemming method were examined to find out the reasons for such performance. It was found that the light stemming method failed to retrieve any relevant documents that contained conjunctions in Arabic such as prepositions and noun plurals in infix. Therefore, in this chapter, we explain some important Arabic morphology analyses that the AMIR technique uses to build the index term uses for search, which may help to avoid the complicated processing of the indexing term. Also more, the ability of AMIR to avoid

problems such as plurals nouns, conjunctions. We also were given an overview of the conclusion of what has been done in this thesis and suggests future work.

7.1 Discussion

After the success of AMIR in the previous chapter by using the 50 queries from test collections EveTAR on tweets, where AMIR outperformed other methods according to using intelligent uses of Arabic morphology. As were mention early, morphological tasks are very hard and complex natural language processing (NLP) and require an understanding of the meaning of a text and the ability to reason over relevant facts. Thus, AMIR could solve most of the morphological problems to achieve the best results using a set of rules. (These rules have been described earlier in Section 4.3. Hence, all the successful rules are based on the concept of Lexical knowledge which is an essential component of language knowledge as reported in [71]. A lexical entry contains a gender feature and the event features, in addition to that each lexical entry has a word stem, which automatically constructed from Wiktionary, with part-of-speech and morphological attributes. The AMIR rules extract the stem using two phases. The first one depends on the morphology analysis (word formation), while the second rules depend to remove affixes (inflectional). Furthermore, all the successful rules are linguistic morphological rules. So, the complexity of morphology need different analysis to achieve high precision, i.e., to improve precision, so AMIR solved the problem of the plural in the infix based on morphology features, i.e., reducing the morphological complexity unwanted, which made the mean precision of AMIR is increased.

AMIR system uses morphological analysis to extract the best candidate word for information retrieval without losing the meaning of the word. This may help to avoid the complicated processing of the indexing term. The stem is an important feature because if too much stem removed from the word, the result can be building unrelated words or it may lose the meaning. However, AMIR has a positive effect on the indexing term for Arabic. This is because AMIR uses morphological formations rules that deal with the removal of different to extract the best candidate word used in the search for information retrieval. Therefore, the output of the AMIR system was examined and we found that AMIR performance is better than any other technique

stem for Arabic, so we have evaluated AMIR by using TF.IDF, which a popular information retrieval technique weighs. The main objective of the TF.IDF weighs is to weigh keywords in content, where the higher the TF.IDF weight, it is an indication that the word is important and vice versa. Additionally, the technique checks the relevance of the keyword throughout the dataset. Figure 6.4 shows compared with the top ten TF.IDF values which calculated to extract terms to represent the category for each method, where the AMIR approach gives a result of more effective terms than LUCENE and FARASA. Consequently, the ability of AMIR to avoid problems that face Arabic retrieval systems like plurals noun in infix as explained as follows:

- Light stemming has a positive effect on stemmed Arabic, but not for words contains vowels. This is probably because light stemming always removes affixes without considering the following: Arabic words that contain vowels, has three cases depends on the position of the vowels (if vowel placed was at beginnings, middles, or ends). In some cases, the vowels ا (A) are replaced by other vowels و (Y). For example, the word دعا (invite), in the present tense, it became يدعو (he is inviting), by replacing the vowels ا (A) with by vowels و (Y). Other case is removed vowel, such as the word ورث (inherit), at the present tense it became يرث (he inherits) see more examples in [72]. Thus, vowel و (Y) is removed in this case. Note that these vowels are original letters of the root. Therefore, we found that word processing containing vowels using the AMIR method is better than light stemming. As were mention early, the light stemming not handle infix due it deal with prefix and suffix only, so we developed light stemming by adding infix in addition to prefix and suffix. Figure 7.1 shows all affix can be adding to the light stemming list to develop it. On the other hand, in Arabic, the plural form often difficult especially when attached in the infix, the reason for that, no clear rules proposed in the previous methods that can process them to get singular, this means that no obvious rules exist to deal with plurals. This research discussed Arabic plural form, through an application of their patterns on stems (singular, dual, plural, masculine and feminine) by altering into their correct singular forms. e.g., plural in the suffix, where the AMIR proposed three rules to solve this problem. such as the word `حافلات` (buses), which contains two morphemes; To distinguish between these

morphemes, we say that infix ا (a) is an inflectional morpheme that refers to the noun; and suffix ات (ta) is an inflectional morpheme indicating the plural. So, AMIR proposed altering their singular forms by replacing ة (taa) instead of ات (ta). Thus, by applying AMIR rule No 7, then we got the word حافلة (bus) instead of the word `حافلات` (buses), so the word has changed to their correct singular form. Whereas light stemming proposed to remove suffix ات (ta). Therefore, if we remove suffix ت (ta) from the word `حافلات`, we got the word `حافل` (no meaning of the word), so, the light stemming method failed to get their singular form.

- In Arabic, the plural forms are difficult to deal with, as were mentioned early, in fact, the Arabic language seem like English in the plural in the suffix but the difference is the Arabic language can be indicated to gender with plural such as the suffix تكما (takama), which refer to the gender of masculine, whereas the suffix تهم (taham), also refer to masculine pluralتكن ; (takan) and the suffix تهن (tahan) which refer to feminine plural; the suffix تهما (tahama), which refer to masculine dual; the suffixتهنا (tahana) refer to feminine dual. Thus, we do replace these suffixes (ات, تكم, تهم, تكن, تهن, تهما, تهنا) from a word by the suffix ة (taa) to get their singular. As shown in the following Figure:

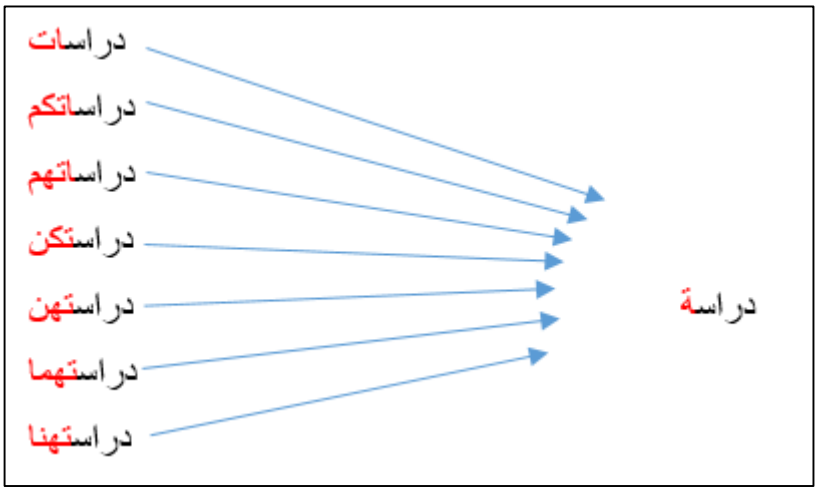

Figure7.1 Replacement Example

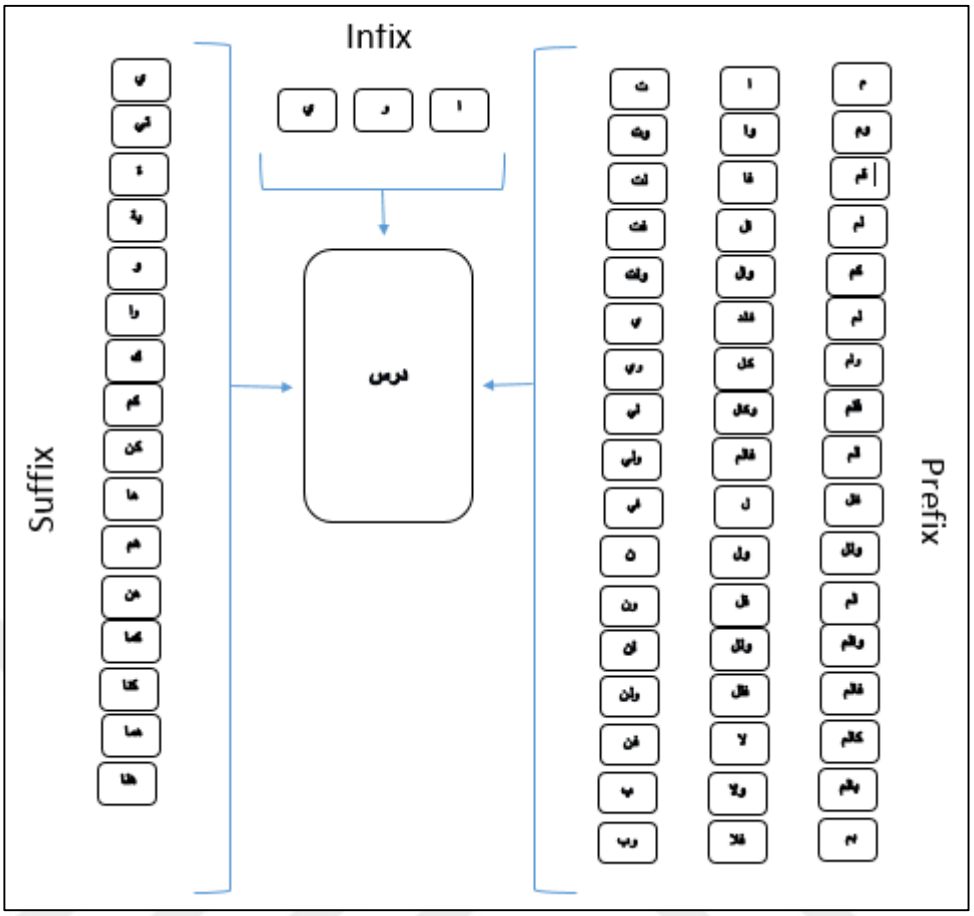

Figure7.2 AMIR generation word

- The conjunctions in Arabic are one of the problems stem that Arabic information retrieval face, where Arabic word may contain three connections linked directly in the same word. This is why the Arabic language is very inflectional language when we compared it with any other language like English. So, it is not easy to determine the conjunctions in Arabic such as stop words, prepositions, due in some cases, they may come separately or linked directly with the word. e.g., the word "فالمدرسة" (and, at school), which contains three prefixes: the prefix و (and); prefix ف (in); prefix ال (the). AMIR rules are capable to remove these conjunctions. Thus, if we applying AMIR rules No 1 for the word "فالمدرسة" (and, at school), then we got the word "مدرسة" (and, at school). Note that this rule says that if the word begins with prefix م (m), remove any extra prefixes if any found. So, it's easy to remove conjunctions in Arabic in this way. Table 7.2 shows some examples of conjunctions in Arabic.

Note that the Arabic word probably contains one or more conjunction, this means that its possible Arabic word can contain different types of conjunctions in the same word.

Table 7.1 Types of conjunctions in Arabic

Conjunctions	Example
Prepositions	So he wrote "فكتبَ"
Stop-words	and he wrote "وكتب"
Definitions	The book "الكتابَ"
Prepositions + Definitions	So the book "فالكتابَ"
Stop-words + Prepositions + Definitions	And in the book "وفالكتابَ"
Stop-words+ Definitions	and the book "والكتابَ"
Stop-words+ Definitions	So the book "فالكتابَ"
Prepositions + Prepositions + Definitions	So in the book "ففالكتابَ"
Stop-words + Prepositions + Definitions	and in the book "وفالكتابَ"

As were mention early, the AMIR system uses a set of rules of word formation (i.e. derivational morphology and inflectional morphology). The morphological analyzer is the heart of the AMIR system as reported in [73]. The main aim of using Arabic morphological operations is to remove prefixes, infixes, and suffixes attached to words to support the methods of search, such as stem and root. It was pointed out early that this study aims to cover all aspects of Arabic morphology analyzer.

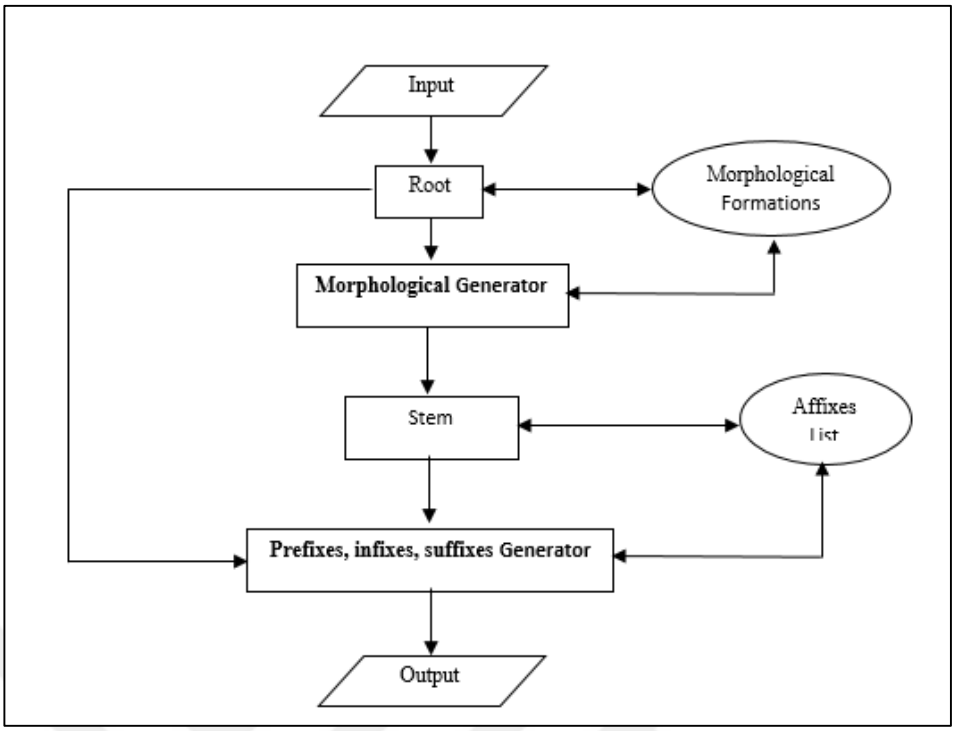

Figure 7.3 AMIR steps to generate word

Figure 4.5 shows several steps that AMIR system uses to generate words, which can be described like the steps as follows:

Morphological Generator: this step is based on the derivational morphology. The main function of this step is to add patterns attached to the root, such as:

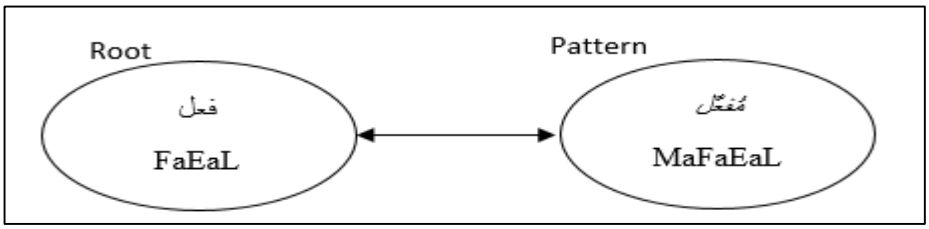

Figure 7.4 Add patterns to root

Prefixes, infixes, suffixes generator: in the previous step the derivational operation and its implementation were discussed, and in this step, we will deal with the implementation of inflectional operations. Inflectional morphology needs deeper analysis than derivational morphology. This is because many conjunctions in Arabic may be linked directly to the word. The main function of the inflectional operation is

to extract a given stem to its base form (root) from which it was derived. To do this, three lists of prefixes, infixes, and suffixes are developed, such as follows:

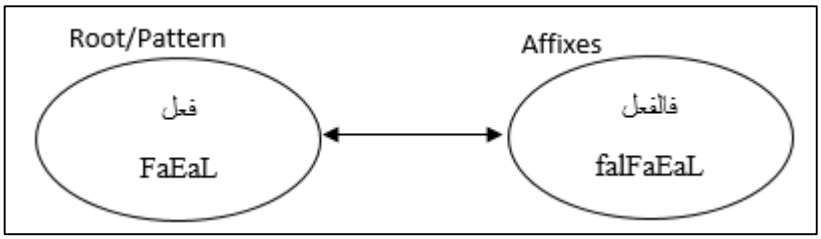

Figure 7.5 Add affixes to root/patterns

When a given word is analyzed by the AMIR to extract stem of a word, the following six decompositions are applying:

1- Prefixes + Stem ↓ (Morphological formations + Root)

The attached inflectional morphology is removed if it found, to extract stem only. Indeed, stem contains two elements, the base of the word (root) and the morphological formations. Note that not separated from each other; in other words, the two elements are linked together in the same word. For example, the word: المدرس (The teacher), if the word is passed to AMIR, then the AMIR will be passed to the morphological analyzer to extract the prefixes م (m), which refer to derivational morphology. Since there is no infix and suffix in this word, so there is no further analysis is needed. This means that keep the prefixes م (m) as it is, and remove all other prefixes attracted to the word if any found. Therefore, one contribution in this work is that there is no needs to check if a word begins with كال , وال , ال , ل and so on the other word, this technique easily capable to remove all prefixes that included in the list of light stemming. So by using the AMIR system, it's easy to remove prefixes that attached to the word and then extract the correct stem. Thus, if the word is passed to AMIR, the attached prefix ال (the), will be removed and the attached prefix م (m), will stay as it is. After the prefixes are removed, the output will be the word stem مدرس (Teacher).

2- Prefixes + Stem + Infixes ↓ (Root + morphological formations)

The derivational Morphology needs to consult the morphological forms knowledge-based to find a match for the stem in the relations between letters like the word تدريس (teaching), where prefix ت (t) and infix ي (y) refers to derivational Morphology, Another example as can be seen from Table 4.

Table 7.2 Prefixes and infixes linked together to produce word based on inflectional and derivational

Word	Stem
المكاتب	مكتب
الكاتب	كاتب
ثلوج	ثلج

3- Prefixes + Stem + Suffixes ↓ (Root + morphological formations)

Generating words based on establishing links between prefixes and suffixes forms. This inflectional and derivational morphology is capable of generating several words from a single root. This needs the morphological forms knowledge base to be consulted, as can be seen from Table 4.

Table 7.3 Prefixes and suffix linked together to produce word based on inflectional and derivational

Word	Stem
العملون	عمل
مسلمات	مسلم
ومطارات	مطار

4- Infixes + Stem ↓ (Root + morphological formations)

Indeed, in Arabic information retrieval, many word formations may not be retrieved due to the inflectional and derivational morphology maybe appear in the infix within Arabic words. Therefore, extracting the derivational and inflectional morphology in

infix was supported by AMIR by adding four rules to change the word from plural form to their singular form based on a complete knowledge base of Arabic morphological forms and rules, as can be seen from Table 4.

Table 7.4 Arabic infix inserted to root, example

Word	Stem
كاتب	كاتب
كتاب	كتاب
هجوم	هجم

5- Infixes + Stem + Suffixes ↓ (Root + morphological formations)

This step extracted all possible infixes and suffixes that may be attached to the root which also based on derivational and inflectional morphology, Table 4 shows some examples of Arabic infixes and suffixes linked together to produce a word.

Table 7.5 Arabic infixes and suffixes example

Word	Stem
كاتبة	كاتب
كتابكم	كتاب
دراسات	دراسة

6- Stem + Suffixes ↓ (Root + morphological formations)

Note that the Arabic language like the English language when dealing with the suffix. And also in English and Arabic often suffix is denoted by numbers such as school schools. Therefore, this step often based on inflectional morphology only. Table 7.6 shows an example of the suffixes attached to the Arabic word.

Table 7.6 Arabic suffixes example

Word	Stem
كتبكم	كتب
قولي	قول
درسان	درس

7- Prefixes + Infixes + Stem + Suffixes

↓

(Root + morphological formations)

This step linked[28] together all prefixes and suffixes in addition to the infixes that may be attached to the root to generate different words. This is the final step of the processor, which is also based on both the derivational and inflectional morphology, the output of this step is passed to the prefixes, infixes, and suffixes generator, as can be seen from Table 4.

Table 7.7 Arabic prefixes, infixes, and suffixes example

Word	Stem
فكتابكم	كتب
المشروبات	مشروب
الكاتبة	كاتب

.Figure 7.6 shows AMIR way to extract stem from a word. It's very simple, only by removed the inflectional morphology that added in previous steps and keeps the derivational structure as it is.

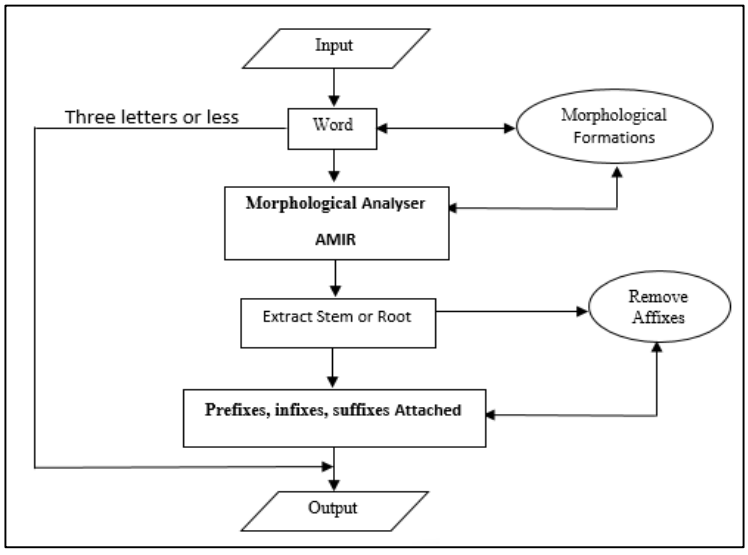

Figure 7.6 shows how AMIR system extract stem or root from the word

[28] Linked all prefixes and suffixes together is very hard and complex because the attachments are not associated with each other in all cases.

The AMIR stemmer applies the follows steps, which depend on what the word contains stem at begins, meddle, or ends.

- Only prefixes: check AMIR rules No 1 only
- Only prefixes and infixes: check AMIR rules No 1, 3, 4, 5, and 6
- Only prefixes and suffixes: check AMIR rules No 1, 2, 7, and 8
- Only infixes: check AMIR rules No 4,
- Only infixes and suffixes: check AMIR rules No 4, 7, and 8
- Only suffixes: check AMIR rules No 4, 7, and 8
- All prefixes, infixes, and suffixes: check AMIR rules No 1, 2, 3, 4, 5, 6, 7, and 8

7.2 Recommendations

Through this study, it was noted that morphological analysis affects the effectiveness of retrieval performance. This led to suggest some recommendations that related to this issue as follows:

- An Arabic test collection of documents for information retrieval is needed. So, the test collection being used as a base for developing an adequate Arabic test collection for information retrieval experiments.
- More is needed for Arabic retrieval systems for development. Unfortunately, there are few studies of the Arabic language retrieval system available for Arab researchers, especially related to Arabic morphology analysis.
- Current Arabic stem algorithms do not handle the second conjunctions in Arabic like prepositions, e.g., "ففالمدرسة" (so for the school). As a result of that, more studies are needed to improve the Arabic language retrieval system.
- Through the literature review, it was found that there is a weakness of cooperation between three groups of the scientist (i.e. computer scientists, librarians, and linguists). As a result of that, it might help, for improving the

work in this area, if the three groups could work together to enhance information retrieval performance in Arabic.

- Year after year, Arabic sites on the Internet are increasing, so the researcher hopes that advanced search engines are necessary to support the Arabic user to meet his needs.
- Current search engines like Google, Bing, and traditional information retrieval systems do not handle mixed-language queries adequately, regardless of its constituent languages, so the researcher is recommended that the mixed-language uses in the Arabic language retrieval system.
- The current Arabic stem algorithms do not get yet much focus on lemma extraction from Arabic texts, so the researcher is recommended that the lemma system can give benefits of the power of rule-based stem, and more suitable for information retrieval systems because lemma can capture all semantic features of the word.

7.3 Conclusion

The rationale behind this researcher is to improve the extraction of Arabic root/stem form a word to build effective Arabic information retrieval systems. Many researchers have investigated the impact of stem and morphology on the information retrieval process over many years. Consequently, in our experiment, comparative study of retrieval performance for AMIR, FARASA, and LUCENE method was carried out, and the results of the comparison show that the retrieval performance of the MAP for the morphological formations method (AMIR) was at a level of 0.34 %, while LUCENE, FARASA, and no stem are 0.27%, 0.28% and by 0.21, respectively. The results show the AMIR system did improve the retrieval performance of the stem methods. It is be 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 stem yields a much-improved retrieval performance of the root method in terms of precision. In other words, the AMIR system succeeds in retrieve more relevant documents that would be retrieved when compared with other stem methods. On the other hand, AMIR achieved

a MAP value by 0.32%, while LUCENE, FARASA, and no stem, are achieved a MAP value by 0.25%, 0.26%, and 0.18%, respectively, by using LM model. Furthermore, the performance of the new AMIR tested on the given Arabic words has been shown to increase results search engine output. Where AMIR outperforms Google Internet Explorer, Bing, and Yahoo in terms of the total number of result searched pages.

Concerning the morphological complexity, building good rules of morphological and stem analysis could solve hard morphological complexity that has long term dependencies to extract a word that succeeds to satisfy the user's query of data. Consequently, AMIR has been developed to solve different types of morphological complexity. However, several rules have been proposed in AMIR to generate/extract the morphology features from the Arabic word. In most performance, AMIR results outperform the state-of-the-art results. Thus, AMIR can be considered to operate around solve the complexity of morphology problems in the Arabic language.

As summarise, this thesis proposed a new approach for Arabic stem, called AMIR uses to generate/extract stems by applying a set of morphological 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 based on investigating the effectiveness of the stem and morphology analysis that are authored and/or annotated in Arabic on information retrieval performance, and how to bridge the gap in Natural Language Processing (NLP) to gain the best the intelligent use of stem and morphology in the Arabic information retrieval domain.

Finally, we believe that it's difficult to develop new Arabic system retrieval without good morphological formations support it. This thesis has shown to improve Arabic stem and increases retrieval performances through presented AMIR system for extracting stem or root form of Arabic words based on morphological formations. these morphological formations resulting from the addition of different prefixes, infixes, or suffixes to the root of the word according to the grammar., so AMIR system capable to enhance the ability to extract the stem that is used to finding the best quality solutions that facing the information retrieval by avoiding getting trapped in similarity roots such as the problem of the broken plurals for nouns, verbs, adjectives, and gender that cannot be solved by previous methods.

7.4 Future work

In this section, we will summarise some of the research points that have not yet been fully addressed by this thesis. Therefore, there are many improvements and ideas can be done to develop thesis method, so in this research, we can list a few of them as:

- Develop this algorithm by adding more rules of morphological formations to prove that it is an intelligent algorithm and can work with any kind of information retrieval.
- Add more features to improving the precision and trying to keeping the running time minimal.
- A Lemmatizer technique can be combined with the research method to get more relevant retrieval for the information to ensure the precision and the relevancy.
- Develop AMIR by adding more patterns to meet the requirements of the topic of interest to uses for identifying the current word, which makes AMIR more efficient over any type of stem.
- In Arabic, plural can be denoted by prefix, infix, or suffix. Often plural in infix used vowel letters, thus the future work, it may possible to break plural in infix by removing vowels, for example, suppose that a word contains vowel attached in infix and weight is four letters, so if we removed the vowel, the remained will be the root. Like the word هجوم (Attacks) if we remove vowel in infix then we will get هجم (Attack).

REFERENCES

[1] Bomhard, A.R., Toward Proto-Nostratic: a new approach to the comparison of Proto-Indo-European and Proto-Afroasiatic. Vol. 27, 1984.

[2] Beesley, K.R. Arabic finite-state morphological analysis and generation. in Proceedings of the 16th conference on Computational linguistics. Association for Computational Linguistics, 1996.

[3] Al Ameed, Shaikha O. Al Ketbi, Amna Ahmed Al Kaabi, Khadija S. Al Shebli. Arabic light stemmer: A new enhanced approach. in The Second International Conference on Innovations in Information Technology (IIT'05). 2005.

[4] Kanaan, Ababneh, Al-Shalabi, Al-Nobani, Building an effective rule-based light stemmer for Arabic language to inprove search effectiveness. in 2008 International Conference on Innovations in Information Technology. 2008.

[5] Al-Kharashi, I.A. and M.W. Evens, Comparing words, stems, and roots as index terms in an Arabic information retrieval system. Journal of the American Society for Information Science, 45(8): p. 548-560, 1994.

[6] Brent, M.R., Speech segmentation and word discovery: A computational perspective. Trends in Cognitive Sciences, 3(8): p. 294-301, 1999.

[7] Kareem Darwish, H. Hassan, and O. Emam. Examining the effect of improved context sensitive morphology on Arabic information retrieval. in Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Association for Computational Linguistics, 2005.

[8] Larkey, L.S., L. Ballesteros, and M.E. Connell, Light stemming for Arabic information retrieval, in Arabic computational morphology, Springer. p. 221-243, 2007.

[9] Eldesouki, M.I., W.M. Arafa, and K. Darwish, Stemming techniques of Arabic language: Comparative study from the information retrieval perspective. The Egyptian Computer Journal, 36(1): p. 30-49, 2009.

[10] Aljlayl, M. and O. Frieder. On Arabic search: improving the retrieval effectiveness via a light stemming approach. in Proceedings of the eleventh international conference on Information and knowledge management, 2002.

[11] El-Beltagy, S. and A. Rafea. A framework for the rapid development of list based domain specific Arabic stemmers. in Proceedings of the Second International Conference on Arabic Language Resources and Tools. 2009.

[12] Al-Shalabi, Ghwanmeh, Kanan, Stemmer algorithm for Arabic words based on excessive letter locations. in 2007 Innovations in Information Technologies (IIT). 2007.

[13] Paice, C.D. An evaluation method for stemming algorithms. in SIGIR'94, Springer, 1994.

[14] BAKEEL, Azman, B., Root Identification Tool for Arabic Verbs. IEEE Access. 7: p. 45866-45871, 2019.

[15] Naili, M., A.H. Chaibi, and H.H.B. Ghezala, Comparative Study of Arabic Stemming Algorithms for Topic Identification. Procedia Computer Science,. 159: p. 794-802, 2019.

[16] Carlberger, Dalianis , Hassel , Knutsson, Improving precision in information retrieval for Swedish using stemming. in Proceedings of the 13th Nordic Conference of Computational Linguistics, 2001.

[17] Abu-Salem, H., Al-Omari, M., and Evens, M. Stemming methodologies over individual query words for Arabic in¿formation retrieval. JASIS, 50 (6), 1999.

[18] Al-Fedaghi, S. S. and Al-Anzi, F. S. A new algorithm to generate Arabic root-pattern forms. In Proceedings of the 11th national computer conference. 1989.

[19] Larkey, L.S. and M.E. Connell, Structured queries, language modeling, and relevance modeling in cross-language information retrieval. Information processing management, 41(3): p. 457-473, 2005.

[20] Kazem T., Rania E., and JeÌrey C., Arabic Stemming Without A Root Dictionary, Information Science Research Institute, USA, 2005.

[21] Mohamad Ababneh, Riyad Al-Shalabi, Ghassan Kanaan, Alaa Al-Nobani, Building an Effective Rule-Based Light Stemmer, The International Arab Journal of Information Technology, Vol. 9, No. 4, July 2012.

[22] P. Willett, "The Porter stemming algorithm: then and now," Program, vol. 40, no. 3, pp. 219-223, 2006.

[23] R. Mamoun and M. Ahmed, Arabic text stemming: Comparative analysis. In Basic Sciences and Engineering Studies (SGCAC), Confer-ence of IEEE, 2016

[24] H. Froud, A. Lachkar, S. A. Ouatik, Stemming Versus Light Stemming for Measuring the Similarity between Arabic Words With Latent Semantic Analysis Model, Information Science and Technology Conference, 2012.

[25] A. Mohammed, Z. Ayah, D. Mona, The Power of Language Music: rabicLemmatization through Patterns, in: M. Zock, A. Lenci, S. Evert (Eds.), Proc. 5th Work. Cogn. Asp. Lexicon, CogALex@COLING, Osaka, Japan, December 12,2016.

[26] T. El-shishtawy, F. El-Ghannam, An accurate arabic root-based Lemmatizer for information retrieval purposes, IJCSI, Int. J. Comput. Sci. Issues. 2012.

[27] S. Ben Ismail, H. Maraoui, K. Haddar and L. Romary, "ALIF editor for generating Arabic normalized lexicons", The International Conference on Information and Communication Systems (ICICS 2017), 2017.

[28] Abd El Salam, A. L., HAJJAR, M., & ZREIK, Classification of Arabic Information Extraction methods, 2nd International Conference on Arabic Language, 2009.

[29] Salton, G., & McGill, M. J. Introduction to modern information retrieval.McGraw-Hill Book Company, 1983.

[30] Sanderson, M. and Croft, B. “The History of Information Retrieval Research.”Proceedings of the IEEE 100. (Special Centennial Issue), 2012.

[31] Hammad BALLAOUI, El Habib BEN LAHMAR, Nasser LABANI International Review on Computers and Software (I.RE.CO.S), volume 11(6), 2016.

[32] Shen, X., Tan, B., and Zhai, C., Context-sensitive information retrieval using implicit feedback, in Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval,SIGIR, 2005.

[33] Ahmed bin Muhammad bin Ahmed Al-Hamlawi, Shadal Al-Arf in the Art of Drainage, Dar Al-Kyan Printing House, Riyadh, 2006.

[34] Muhammad Khair Al-Halawani, Clear in Grammar, House of Heritage for Heritage, Sixth Edition, Damascus, 2000.

[35] Saeed Al-Afghani, Abstract in Arabic Grammar, Dar Al-Fikr for printing and publishing, Beirut, Lebanon, 2003.

[36] Gouda Mahmoud al-Tahlawi, History of semitic language, Dar Al-Qalam, 1980.

[37] Haywood, J. A., & Nahmad, H. M. A New Arabic Grammar of the Written Language. London: Lund Humphries, page 2, 1965.

[38] Abu-Chacra, Faruk, 2007, Arabic. An essential grammar, London:Routedge, pages 2-4, 2007.

[39] Ayman Amin Abdel-Ghani, Adequate Morphology, Al-Tawfikia Heritage House, Fifth Edition, Cairo, 2010.

[40] Imad Ali Juma, Arabic Grammar grammatical and morphology the Facilitator , Islamic Science Series, King Fahd Library Indexing, First Edition, 2006.

[41] Abdel Shakour Muallem Abdel Farah, The Facilitating morphology of a

Literacy Approach to Ibn Malik in a Modern Style with Examples, Tables and Training, Dar Al-Elm for Publishing, Distribution and Translation, First Edition, Cairo, 2019.

[42] AlQurashi, I.S, An analysis of simple and construct-state noun phrases in modern standard Arabic. In: Muller, S. (eds.) 22nd International Conference on Head-Driven Phrase Structure Grammar Proceedings, pp. 6–26. CSLI Publications, Nanyang Technological University (NTU), Singapore Stanford, 2015.

[43] Ryding, K. A reference grammar of modern standard Arabic. Cambridge: Cambridge University Press, pgae 54, 2005.

[44] Benmamoun, E. Construct state. In K. Versteegh, M. Eid, A. Elgibali, M. Woidich & A. Zaborski (Eds.), Encyclopedia of Arabic language and linguistics (Vol. I, pp. 477–482). Leiden: Brill, 2006.

[45] Ibrahim Shams El-Din, The easiest way to teach the morphology of verbs, Language of Arab Library, First Edition, Beirut, Lebanon, 2009.

[46] Ahmad Ibrahim Al-Hashimi, Basic Grammar of the Arabic Language, Dar Al-Kutub Al-Alami, Beirut, Lebanon, 1989.

[47] Larkey, L.S., L. Ballesteros, and M.E. Connell. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002.

[48] Harun Abdul-Razzaq, the title of the Adverb in the science of morphology, Al-Raskhoon Center, first edition, 2018.

[49] Antoine El-Dahdah, A Dictionary of Arabic grammar in charts and tables, Librairie du Liban, page 5, 1985.

[50] Nadim Hussein Daccour, Applied Grammar in the Arabic Language, Bahsoun Foundation for Publishing and Distribution, Beirut lebanon, 1998.

[51] Ibrahim Al-Huri, Arabic Language: Historic and Sociolinguistic Characteristics, English Literature and Language Review, Academic Research Publishing Group, vol. 1(4), pages 28-36, 2015.

[52] Porter, M., An algorithm for suffix stripping. Program: electronic library & information systems. 1980.

[53] Kane, Carolyn, Grammar and Composition, USA, Mark TwainMedia, Inc, 2011.

[54] Abdel Hamid Antar, The Verbs of morphology, Dar Al-Dhahria Publishing and Distribution, First Edition, 2017.

[55] Mohammed Al-Tantawi, The Names of morphology, Dhahiriya Publishing and Distribution House, First Edition, Kuwait, 2017.

[56] Fouad Nehmeh, Summary of Arabic Grammar, Dar Nahdat Misr, 9th Edition, 2015.

[57] Osama Hamed and Torsten Zesch, The Role of Diacritics in Adapting the Difficulty of Arabic Lexical , In 3rd, ELSEVIER, 2017.

[58] Stefan Langer, Joeran Beel, Apache Lucene as Content-Based-Filtering Recommender System: 3 Lessons Learned, Workshop on Bibliometric-enhanced Information Retrieval, 2017

[59] Kareem Darwish, Abdelali, Farasa: A fast and furious segmenter for arabic. in Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations. 2016.

[60] Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth.. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In In Proceedings of LREC, Iceland, 2014.

[61] Youssef Al-Hammadi, Muhammad Muhammad Al-Shinawi, Muhammad

Shafiq Ata, Basic Principles of Grammar and Morphology, general authority for affairs printing, Cairo, 1995.

[62] YShaaban Salah, Conjugation of Verbs in the Arabic Language, revised special edition, 2016.

[63] Moulana Abdus Sattar Khan, ARABIC TUTOR, Madrassah Inaamiyyah Camperdown, Volume 3, page 22, 2004.

[64] Laila Al-Sawi, Iman Saad, Al-Murshid: A Guide to Modern Standard Arabic Grammar for the Intermediate Level, page 91, 2012.

[65] Lianzhi Tan, Junjie Lin, Shengping Zhou, Attention based TermWeighting for App Retrieval, Association for the Advancement of Artificial Intelligence, 2019.

[66] Rajendra Kumar Roul, Jajati Keshari Sahoo, Kushagr Arora, oulModified TF-IDF Term Weighting Strategies for Text Categorization, Conference: 14th IEEE India Council International Conference (INDICON), 2018.

[67] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4):333–389, 2009.

[68] Almerekhi, H., Hasanain, M., & Elsayed, T. EveTAR: A New Test Collection for Event Detection in Arabic Tweets. In Proceedings of the 39th International ACM SIGIR conference, 2016.

[69] Jaffar Atwan, J., M. Wedyan, and H. Al-Zoubi, Arabic Text Light Stemmer. International Journal of Computing, 8(2): p. 1.7-23, 2019.

[70] Kadri, Y. and Nie, J.Y. (2006) Effective Stemming for Arabic Information Retrieval. Proceedings of the Challenge of Arabic for NLP/MT Conference, London, 23 October 2006.

[71] Robert Ricks, The Development of Frequency- Based Assessments of

Vocabulary Breadth and Depth for L2 Arabic, 2015.

[72] Saleh Saleem Al-Fakhiri, Morphology of Verbs with Sources and Derivatives, Asami Publishing and Distribution, Cairo, 1996.

[73] Musaid Saleh Al Tayyar, Arabic Information Retrieval System based on Morphological Analysis (AIRSMA), PhD thesis, Computer Science in the Department of the Computer, DeMontfort University, 2000.

[74] Ahmed Y. Tawfik, Mahitab Emam, Khaled Essam, Robert Nabil and Hany Hassan, Morphology-Aware Word-Segmentation in Dialectal Arabic Adaptation of Neural Machine Translation, Proceedings of the Fourth Arabic Natural Language, 2019.

[75] Ali Farghaly and Khaled Shaalan, Arabic natural language processing: Challenges and solutions, 2009.

[76] Al-Saqqa, S., A. Awajan, and S. Ghoul. Stemming Effects on Sentiment Analysis using Large Arabic Multi-Domain Resources. in Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2019.

[77] Chen, Aitao & Gey, Fredric. Building an Arabic Stemmer for Information Retrieval. TREC, 631-639. 2012.

[78] Cristina Izura, Fernando Cuetos, and Marc Brysbaert. Lextale-Esp: A test to rapidly and efficiently assess the Spanish vocabulary size. Psicol´ogica, 35(1):49–66. 2014.

[79] de Roeck, A. N., and Al-Fares, W. A morphologically sensitive clustering algorithm for identifying Arabic roots. 2000.

[80] Habash, Roth, Rambow, Eskander, Tomeh, Morphological analysis and disambiguation for dialectal Arabic. in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013.

[81] Khoja, S. and R. Garside, Stemming arabic text. Lancaster, UK, Computing Department, Lancaster University, 1999.

[82] Khoja, S. APT: Arabic part-of-speech tagger. in Proceedings of the Student Workshop at NAACL. 2001.

[83] Kreaa, A. H., Ahmad, A. S., & Kabalan, K. Arabic words stemming approach using Arabic WordNet, International Journal of Data Mining & Knowledge Management Procss, 4(6), 2014.

[84] Kristin Lemh¨ofer and Mirjam Broersma. Introducing LexTALE: A quick and valid lexical test for advanced learners of English. Behavior Research Methods, 44(2):325–343. 2102.

[85] Marc Brysbaert. LEXTALE FR: A fast, free, and efficient test to measure language proficiency in French. Psychologica Belgica, 53(1):23–37. 2013.

[86] Mustafa, Afag Salah Aldeen, Rihab E. Ahmed, Developing Two Different Novel Techniques for Arabic Text Stemming. Intelligent Information Management, 2019.

[87] Okan Ozturkmenoglu, Adil Alpkocak, Comparison of different lemmatization approaches for information retrieval on Turkish text collection, Conference: Innovations in Intelligent Systems and Applications (INISTA), 2012.

APPENDIX

Appendix A – Examples, stem of the words

Table A.1 Examples of stem of words

FARASA	LUCENE	AMIR	Word
درس	درسهم	درس	درسهم
درس	درسكم	درس	ودرسكم
ميثاق	ميثاق	ميثاق	الميثاق
جامع	جامع	جامع	الجامعات
مستشفي	مستشف	مستشفي	المستشفيات
مدارس	مدارسكم	مدرسة	ومدارسكم
انهار	انهار	نهر	انهار
مباري	مبار	مبارة	مباريات
كلم	كلم	كلم	الكلمات
يدرس	سيدرس	درس	سيدرسون
ورث	رثكم	ورث	ورثكم
قول	قولكم	قول	قولكم
صيام	صيامكم	صيام	صيامكم
صيام	صيام	صيام	الصيام
كرو	كرو	كرو	بالكروات
كرواتي	كروات	كرواتي	بالكرواتي
كرواتي	كروات	كرواتي	بالكرواتية
كروم	كروم	كروم	بالكروم
كروموزوم	كروموزوم	كروموزوم	بالكروموزوم
كري	كر	كرة	بالكريات
كريب	كريب	كريب	بالكريب
كريستال	كريستال	كريستال	بالكريستال
كريم	كريم	كريم	بالكريمة
كزاز	كزاز	كزاز	بالكزاز
كساح	كساح	كساح	بالكساح
كساد	كساد	كساد	بالكساد
كسر	كسر	كسر	بالكسر
الكسندر	كسندر	كسندر	بالكسندر
كسندروف	كسندروف	كسندروف	بالكسندروف
كشف	كشف	كشف	بالكشف
كشفت	كشفت	كشفت	بالكشفت
كعابنة	كعابن	كعابنة	بالكعابنة
كعب	كعب	كعب	بالكعب
كعبي	كعب	كعبي	بالكعبي
كفاء	كفاء	كفاء	بالكفاءات
كفاء	كفاء	كفاء	بالكفاءة

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
كفور	كفور	كفور	بالكفور
كلاب	كلاب	كلاب	بالكلاب
كلاشنكوف	كلاشنكوف	كلاشنكوف	بالكلاشنكوف
كلاشنيكوف	كلاشنيكوف	كلاشنيكوف	بالكلاشنيكوف
كلام	كلام	كلام	بالكلام
كلب	كلب	كلب	بالكلب
كلداني	كلدان	كلداني	بالكلدانية
كلس	كلس	كلس	بالكلس
كلف	كلف	كلف	بالكلفة
كلم	كلم	كلم	بالكلمات
كلم	كلم	كلم	بالكلمة
كلور	كلور	كلور	بالكلور
كلي	كل	كلي	بالكلي
كلي	كل	كلة	بالكليات
كلي	كل	كلي	بالكلية
بال	كم	بالكم	بالكم
كماش	كماش	كماش	بالكماشة
كمال	كمال	كمال	بالكمال
كمبودي	كمبود	كمبودي	بالكمبوديين
كمبيوتر	كمبيوتر	كمبيوتر	بالكمبيوتر
كمبيوتر	كمبيوتر	كمبيوتر	بالكمبيوترات
كمي	كم	كمة	بالكميات
كمي	كم	كمي	بالكمية
كنايس	كنايس	كنايس	بالكنايس
كندي	كند	كندي	بالكندي
كندي	كند	كندي	بالكندية
كنز	كنز	كنز	بالكنز
كنغورو	كنغورو	كنغورو	بالكنغورو
كنوز	كنوز	كنوز	بالكنوز
كنيس	كنيس	كنيس	بالكنيس
كنيس	كنيس	كنيس	بالكنيسة
كنيست	كنيست	كنيست	بالكنيست
كهرباء	كهرباء	كهرباء	بالكهرباء
كهن	كهن	كهن	بالكهنة
كوابل	كوابل	كوابل	بالكوابل
كوابيس	كوابيس	كوابيس	بالكوابيس
كوادر	كوادر	كوادر	بالكوادر
كوارث	كوارث	كوارث	بالكوارث
كواشف	كواشف	كواشف	بالكواشف
كواكب	كواكب	كواكب	بالكواكب
كوبي	كوب	كوبي	بالكوبية

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
كوبي	كوب	كوبي	بالكوبيين
كوري	كور	كوري	بالكوري
كوفي	كوف	كوفة	بالكوفيات
كوفي	كوف	كوفي	بالكوفية
كوكا	كوكا	كوكا	بالكوكا
كوكايين	كوكا	كوكايين	بالكوكايين
كوكب	كوكب	كوكب	بالكوكب
كولسترول	كولسترول	كولسترول	بالكولسترول
كولونيل	كولونيل	كولونيل	بالكولونيل
كوليرا	كوليرا	كوليرا	بالكوليرا
كولير	كولير	كولير	بالكوليرات
كومبيوتر	كومبيوتر	كومبيوتر	بالكومبيوتر
كومبيوتر	كومبيوتر	كومبيوتر	بالكومبيوترات
كومندان	كومند	كومندان	بالكومندان
كوميديا	كوميديا	كوميديا	بالكوميديا
كون	كون	كون	بالكون
كونا	كونا	كونا	بالكونا
كونت	كونت	كونت	بالكونت
كونترا	كونترا	كونترا	بالكونترا
كونسورسيوم	كونسورسيوم	كونسورسيوم	بالكونسورسيوم
كونكورد	كونكورد	كونكورد	بالكونكورد
كوني	كون	كوني	بالكوني
كويت	كويت	كويت	بالكويت
كويتي	كويت	كويتي	بالكويتي
كويتي	كويت	كويتي	بالكويتيين
كيان	كي	كيان	بالكيان
كيان	كيان	كيان	بالكيانين
كيبوتز	كيبوتز	كيبوتز	بالكيبوتزات
كيروز	كيروز	كيروز	بالكيروزين
كيروسين	كيروس	كيروسين	بالكيروسين
كيف	كيف	كيف	بالكيف
كيفي	كيف	كيفي	بالكيفية
كيل	كيل	كيل	بالكيل
كيلو	كيلو	كيلو	بالكيلو
كيلواط	كيلواط	كيلواط	بالكيلواط
كيلوغرام	كيلوغرام	كيلوغرام	بالكيلوغرام
كيمياء	كيمياء	كيمياء	بالكيمياء
كيمياي	كيمياي	كيمياي	بالكيمياي
كينهول	كينهول	كينهول	بالكينهول
كينون	كينون	كينون	بالكينونة
كيني	كين	كيني	بالكيني

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
كيني	كين	كيني	بالكينية
لاء	لاء	لاء	باللاءات
لايح	لايح	لايح	باللايحة
لايم	لايم	لايم	باللايمة
لانساني	لانسان	لانساني	باللانساني
لاتيني	لاتين	لاتيني	باللاتينية
لاجي	لاج	لاجي	باللاجيين
اجي	اج	اجي	اجيين
لادستوري	لادستور	لادستوري	باللادستورية
لاذقي	لاذق	لاذقي	باللاذقية
لازم	لازم	لازم	باللازم
لازم	لازم	لازم	باللازمة
لازورد	لازورد	لازورد	باللازورد
لاسلكي	لاسلك	لاسلكي	باللاسلكي
لاشرعية	لاشرع	لاشرعية	باللاشرعية
لاعب	لاعب	لاعب	باللاعب
لاعب	لاعب	لاعب	باللاعبات
لاعب	لاعب	لاعب	باللاعبة
لاعب	لاعب	لاعب	باللاعبين
لاعنف	لاعنف	لاعنف	باللاعنف
لافت	لافت	لافت	باللافتات
لافت	لافت	لافت	باللافتة
لام	لام	لام	باللام
بالا	اه	بالاة	بالاة
لامسولي	لامسول	لامسولي	باللامسولية
لانساني	لانسان	لانساني	باللانسانية
لاواعي	لاواع	لاواعي	باللاواعي
لباس	لباس	لباس	باللباس
لباق	لباق	لباق	باللباقة
لبن	لبن	لبن	باللبن
لبناني	لبنان	لبناني	باللبناني
لبنان	لبنان	لبنان	باللبنانين
لبناني	لبنان	لبناني	باللبنانيين
لجان	لج	لجان	باللجان
لجن	لجن	لجن	باللجنة
لجوء	لجوء	لجوء	باللجوء
لحاق	لحاق	لحاق	باللحاق
لحظ	لحظ	لحظ	باللحظة
لحم	لحم	لحم	باللحم
لحم	لحم	لحم	باللحمة
لحوم	لحوم	لحوم	باللحوم

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
لحي	لح	لحي	باللحية
لصحافي	لصحاف	لصحافي	باللصحافي
لصوص	لصوص	لصوص	باللصوص
ين	لص	لصين	باللصين
لضفة	لضف	لضفة	باللضفة
لعاب	لعاب	لعاب	باللعاب
لعب	لعب	لعب	باللعب
لعب	لعب	لعب	باللعبة
لعن	لعن	لعن	باللعنة
لغات	لغ	لغات	باللغات
لغم	لغم	لغم	باللغم
لفت	لفت	لفت	باللفت
لفت	لفت	لفت	باللفتة
لفتنانت	لفتنانت	لفتنانت	باللفتنانت
لقاء	لقاء	لقاء	باللقاء
لقاء	لقاء	لقاء	باللقاءات
لقاح	لقاح	لقاح	باللقاح
لقاح	لقاح	لقاح	باللقاحات
لقاح	لقاح	لقاح	باللقاحين
لقب	لقب	لقب	باللقب
لقبول	لقبول	لقبول	باللقبول
لقب	لقب	لقب	باللقبين
لكم	لكم	لكم	باللكم
لكم	لكم	لكم	باللكمات
لكم	لكم	لكم	باللكمة
للغ	للغ	للغ	باللغة
لمح	لمح	لمح	باللمحات
لمس	لمس	لمس	باللمس
لمس	لمس	لمس	باللمسات
لمس	لمس	لمس	باللمسة
لهب	لهب	لهب	باللهب
لهج	لهج	لهج	باللهجة
لهج	لهجت	لهج	باللهجتين
لهدوء	لهدوء	لهدوء	باللهدوء
لهو	لهو	لهو	باللهو
لواء	لواء	لواء	باللواء
لوايح	لوايح	لوايح	باللوايح
اللواتي	لوات	اللواتي	باللواتي
لوازم	لوازم	لوازم	باللوازم
لوجستي	لوجست	لوجستي	باللوجستية
لوح	لوح	لوح	باللوحات

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
لوح	لوح	لوح	باللوحة
لورد	لورد	لورد	باللورد
لوكيميا	لوكيميا	لوكيميا	باللوكيميا
باللوكيمياء	لوكيمياء	باللوكيمياء	باللوكيمياء
لوم	لوم	لوم	باللوم
لون	لون	لون	باللون
لون	لون	لون	باللونين
لياق	لياق	لياق	باللياقة
ليبرالي	ليبرال	ليبرالي	بالليبرالية
ليبرالي	ليبرال	ليبرالي	بالليبراليين
ليبي	ليب	ليبي	بالليبيين
ليدي	ليد	ليدي	بالليدي
لير	لير	لير	بالليز
لير	لير	لير	بالليرة
ليري	لير	ليري	بالليري
ليزر	ليزر	ليزر	بالليزر
بالليشمانيا	ليشمانيا	بالليشمانيا	بالليشمانيا
ليطان	ليط	ليطان	بالليطان
ليكوبين	ليكوب	ليكوبين	بالليكوبين
ليكود	ليكود	ليكود	بالليكود
ليمفوما	ليمفوما	ليمفوما	بالليمفوما
لين	لين	لين	باللين
ليون	ليون	ليون	بالليونة
بالم	بالم	بالم	بالم
ماخذ	ماخذ	ماخذ	بالماخذ
ماسي	ماس	ماسي	بالماسي
مازق	مازق	مازق	بالمازق
ماسا	ماسا	ماسا	بالماساة
ماكول	ماكول	ماكول	بالماكولات
مامر	مامر	مامر	بالمامرات
مامر	مامر	مامر	بالمامرة
مبد	مبد	مبد	بالمبد
متمر	متمر	متمر	بالمتمر
مثر	مثر	مثر	بالمثرات
مرخ	مرخ	مرخ	بالمرخين
مسس	مسس	مسس	بالمسسات
مسس	مسس	مسس	بالمسسة
ملف	ملف	ملف	بالملفات
ممن	ممن	ممن	بالممنين
المن	من	المن	بالمن
مهل	مهل	مهل	بالمهلات

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
ميد	ميد	ميد	بالميدين
ميذن	ميذن	ميذن	بالميذنة
ميوي	ميو	ميوي	بالميوية
بالم	ما	بالم	بالما
ماء	ماء	ماء	بالماء
ماي	ما	ماي	بالماية
ماد	ماد	ماد	بالماد
مادب	مادب	مادب	بالمادب
ماد	ماد	ماد	بالمادة
مادتين	مادت	مادتين	بالمادتين
مادي	ماد	مادي	بالمادية
ماراتون	مارات	ماراتون	بالماراتون
مار	مار	مار	بالمارة
مارشال	مارشال	مارشال	بالمارشال
مارك	مارك	مارك	بالمارك
ماريجوانا	ماريجوانا	ماريجوانا	بالماريجوانا
ماريس	ماريس	ماريس	بالماريس
ماريشال	ماريشال	ماريشال	بالماريشال
مارينز	مارينز	مارينز	بالمارينز
مازوت	مازوت	مازوت	بالمازوت
ماس	ماس	ماس	بالماس
ماسترز	ماسترز	ماسترز	بالماسترز
ماشي	ماش	ماشي	بالماشية
ماض	ماض	ماض	بالماض
ماضي	ماض	ماضي	بالماضي
ماغنسيوم	ماغنسيوم	ماغنسيوم	بالماغنسيوم
مافيا	مافيا	مافيا	بالمافيا
ماكياج	ماكياج	ماكياج	بالماكياج
مال	مال	مال	بالمال
مالاريا	مالاريا	مالاريا	بالمالاريا
مالوفا	مالوفا	مالوفا	بالمالوفا
مالي	مال	مالي	بالمالية
مان	مان	مان	بالمان
مانتيري	مانتير	مانتيري	بالمانتيري
مانح	مانح	مانح	بالمانحين
مانغا	مانغا	مانغا	بالمانغا
مانوفا	مانوفا	مانوفا	بالمانوفا
ألمانيا	مانيا	ألمانيا	بالمانيا
ماهيم	ماهيم	ماهيم	بالماهيم
ماوي	ماو	ماوي	بالماوية
مايكرو	مايكرو	مايكرو	بالمايكرو

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
مايوه	مايو	مايوه	بالمايوهات
مباحث	مباحث	مباحث	بالمباحثات
مبادي	مباد	مبادي	بالمبادي
مبادر	مبادر	مبادر	بالمبادرات
مبادر	مبادر	مبادر	بالمبادرة
مبادل	مبادل	مبادل	بالمبادلات
مبادء	مبادء	مبادء	بالمبادء
مباديء	مباديء	مباديء	بالمباديء
مبارا	مبارا	مبارا	بالمباراة
مباراتين	مبارات	مباراتين	بالمباراتين
مبارك	مبارك	مبارك	بالمباركة
مباري	مبار	مباري	بالمباريات
مباشر	مباشر	مباشر	بالمباشرة
بالغ	بالغ	بالغ	بالغ
بالغ	غه	بالغ	بالغة
مبانغ	مبانغ	مبانغ	بالمبانغ
مبان	مب	مبان	بالمبان
مبدا	مبدا	مبدا	بالمبدا
مبدع	مبدع	مبدع	بالمبدعات
مبرر	مبرر	مبرر	بالمبررات
مبضع	مبضع	مبضع	بالمبضع
مبعد	مبعد	مبعد	بالمبعدين
مبعوث	مبعوث	مبعوث	بالمبعوث
مبكر	مبكر	مبكر	بالمبكر
مبلغ	مبلغ	مبلغ	بالمبلغ
مبن	مبن	مبن	بالمبن
مبني	مبن	مبني	بالمبني
مبهم	مبهم	مبهم	بالمبهمة
مبيت	مبيت	مبيت	بالمبيت
مبيد	مبيد	مبيد	بالمبيدات
مبيراس	مبيراس	مبيراس	بالمبيراس
مبيع	مبيع	مبيع	بالمبيعات
متاخر	متاخر	متاخر	بالمتاخرات
متابع	متابع	متابع	بالمتابعة
متاجر	متاجر	متاجر	بالمتاجر
متاجر	متاجر	متاجر	بالمتاجرة
متاجر	متاجر	متاجر	بالمتاجرين
متاريس	متاريس	متاريس	بالمتاريس
متبرع	متبرع	متبرع	بالمتبرعين
متجمع	متجمع	متجمع	بالمتجمعين
متحارب	متحارب	متحارب	بالمتحاربين

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
متحدث	متحدث	متحدث	بالمتحدث
متحف	متحف	متحف	بالمتحف
متخلف	متخلف	متخلف	بالمتخلفين
متدرب	متدرب	متدرب	بالمتدربة
متدهور	متدهور	متدهور	بالمتدهورة
متر	متر	متر	بالمتر
مترشح	مترشح	مترشح	بالمترشحات
مترو	مترو	مترو	بالمترو
متسول	متسول	متسول	بالمتسولين
متصدر	متصدر	متصدر	بالمتصدر
متصدر	متصدر	متصدر	بالمتصدرين
متطرف	متطرف	متطرف	بالمتطرف
متطرف	متطرف	متطرف	بالمتطرفين
متطلب	متطلب	متطلب	بالمتطلبات
متظاهر	متظاهر	متظاهر	بالمتظاهرات
متظاهر	متظاهر	متظاهر	المتظاهرين
متعامل	متعامل	متعامل	بالمتعاملين
متعاون	متعاون	متعاون	بالمتعاونين
متع	متع	متع	بالمتعة
متعصب	متعصب	متعصب	بالمتعصبين
متغطرس	متغطرس	متغطرس	بالمتغطرسة
متغير	متغير	متغير	بالمتغيرات
متفجر	متفجر	متفجر	بالمتفجر
متفجر	متفجر	متفجر	بالمتفجرات
متفجر	متفجرت	متفجر	بالمتفجرتين
متفجر	متفجر	متفجر	بالمتفجرين
متفحر	متفحر	متفحر	بالمتفحرات
متفرج	متفرج	متفرج	بالمتفرجين
متفرق	متفرق	متفرق	بالمتفرقات
متكلم	متكلم	متكلم	بالمتكلمين
متلقي	متلق	متلقي	بالمتلقيات
متمرد	متمرد	متمرد	بالمتمردين
متن	متن	متن	بالمتن
متنازع	متنازع	متنازع	بالمتنازعين
متنبي	متنب	متنبي	بالمتنبي
متنزه	متنز	متنزه	بالمتنزهين
متهم	متهم	متهم	بالمتهم
متهم	متهمت	متهم	بالمتهمتين
متواصل	متواصل	متواصل	بالمتواصلة
متورط	متورط	متورط	متورطين
متوسط	متوسط	متوسط	بالمتوسط

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
متوفي	متوف	متوفي	بالمتوفي
مثابر	مثابر	مثابر	بالمثابرة
مثقف	مثقف	مثقف	بالمثقفين
مثل	مثل	مثل	بالمثل
مثلج	مثلج	مثلج	بالمثلجات
مثلي	مثلي	مثلي	بالمثلي
مثول	مثول	مثول	بالمثول
مجارف	مجارف	مجارف	بالمجارف
مجار	مجار	مجار	بالمجار
مجارير	مجارير	مجارير	بالمجارير
مجازر	مجازر	مجازر	بالمجازر
مجازف	مجازف	مجازف	بالمجازفة
مجاع	مجاع	مجاع	بالمجاعة
مجال	مجال	مجال	بالمجال
مجال	مجال	مجال	بالمجالات
مجالس	مجالس	مجلسة	بالمجالس
مجالي	مجال	مجالي	بالمجالي
مجال	مجال	مجال	بالمجالين
مجامل	مجامل	مجامل	بالمجاملة
مجان	مج	مجان	بالمجان
مجاهد	مجاهد	مجاهد	بالمجاهدين
مجتمع	مجتمع	مجتمع	بالمجتمع
مجتمع	مجتمع	مجتمع	بالمجتمعات
مجتمع	مجتمع	مجتمع	بالمجتمعين
مجد	مجد	مجد	بالمجد
مجذاف	مجذاف	مجذاف	بالمجذاف
مجر	مجر	مجر	بالمجر
مجر	مجر	مجر	بالمجرات
مجرز	مجرز	مجرز	بالمجرزة
مجرم	مجرم	مجرم	بالمجرم
مجرم	مجرم	مجرم	بالمجرمين
مجري	مجر	مجري	بالمجرية
مجري	مجر	مجري	بالمجريين
مجزر	مجزر	مجزر	بالمجزرة
مجل	مجل	مجل	بالمجلات
مجل	مجل	مجل	بالمجلة
مجلس	مجلس	مجلس	بالمجلس
مجمع	مجمع	مجمع	بالمجمع
مجموع	مجموع	مجموع	بالمجموعات
مجموع	مجموع	مجموع	بالمجموعة
مجموع	مجموعت	مجموع	بالمجموعتين

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
مجند	مجند	مجند	بالمجندين
مجنون	مجن	مجنون	بالمجنون
مجني	مجن	مجني	بالمجني
مجهار	مجهار	مجهار	بالمجهار
مجهود	مجهود	مجهود	بالمجهود
مجهول	مجهول	مجهول	بالمجهول
مجون	مج	مجون	بالمجون
مجوهر	مجوهر	مجوهر	بالمجوهرات
مجيء	مجيء	مجيء	بالمجيء
محابا	محابا	محابا	بالمحاباة
محادث	محادث	محادث	بالمحادثات
محاذير	محاذير	محاذير	بالمحاذير
محارب	محارب	محارب	بالمحاربة
محارب	محارب	محارب	بالمحاربين
محار	محار	محار	بالمحارة
محاسب	محاسب	محاسب	بالمحاسبة
محاصيل	محاصيل	محاصيل	بالمحاصيل
محاضر	محاضر	محاضر	بالمحاضرات
محافظ	محافظ	محافظ	بالمحافظات
محافظ	محافظ	محافظ	بالمحافظة
محافظ	محافظ	محافظ	بالمحافظين
محافل	محافل	محافل	بالمحافل
محاكم	محاكم	محكمة	بالمحاكم
محال	محال	محال	بالمحال
محاما	محاما	محاما	بالمحاماة
محامي	محام	محامي	بالمحامي
محامي	محام	محامي	بالمحامية
محام	محام	محام	بالمحامين
محاول	محاول	محاول	بالمحاولات
محاول	محاول	محاول	بالمحاولة
محب	محب	محب	بالمحبة
محبوس	محبوس	محبوس	بالمحبوسين
محتال	محتال	محتال	بالمحتالين
محترف	محترف	محترف	بالمحترف
محترف	محترف	محترف	بالمحترفين
محتفل	محتفل	محتفل	بالمحتفلين
محتم	محتم	محتم	بالمحتم
محتو	محتو	محتو	بالمحتو
محتوي	محتو	محتوي	بالمحتويات
محرر	محرر	محرر	بالمحررة
محرض	محرض	محرض	بالمحرض

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
محرق	محرق	محرق	بالمحرقة
محرك	محرك	محرك	بالمحرك
محرك	محرك	محرك	بالمحركات
محرم	محرم	محرم	بالمحرمات
محروق	محروق	محروق	بالمحروقات
محروم	محروم	محروم	بالمحرومين
محرر	محرر	محرر	بالمحررين
محسوبي	محسوب	محسوبي	بالمحسوبية
محسوس	محسوس	محسوس	بالمحسوس
محط	محط	محط	بالمحطات
محط	محط	محط	بالمحطة
محل	محل	محل	بالمحلة
محلل	محلل	محلل	بالمحللين
محلول	محلول	محلول	بالمحلول
محلي	محل	محلي	بالمحلي
محلي	محل	محلي	بالمحليين
محور	محور	محور	بالمحور
محول	محول	محول	بالمحولات
محيط	محيط	محيط	بالمحيط
محيط	محيط	محيط	بالمحيطات
محيط	محيط	محيط	بالمحيطين
مخابر	مخابر	مخابر	بالمخابرات
مخاب‌ء	مخاب‌ء	مخاب‌ء	بالمخاب‌ء
مخاتير	مخاتير	مخاتير	بالمخاتير
مخاطر	مخاطر	مخاطر	بالمخاطر
مخاطر	مخاطر	مخاطر	بالمخاطرة
مخالف	مخالف	مخالف	بالمخالفات
مخالف	مخالف	مخالف	بالمخالفة
مخالف	مخالف	مخالف	بالمخالفين
مخاوف	مخاوف	مخاوف	بالمخاوف
مخبر	مخبر	مخبر	بالمخبر
مختار	مختار	مختار	بالمختار
مختبر	مختبر	مختبر	بالمختبر
مختل	مختل	مختل	بالمختل
مخدارت	مخدارت	مخدارت	بالمخدارت
مخدر	مخدر	مخدر	بالمخدر
مخدر	مخدر	مخدر	بالمخدرات
مخدر	مخدر	مخدر	بالمخدرين
مخرج	مخرج	مخرج	بالمخرج
مخزون	مخز	مخزون	بالمخزون
مخزون	مخز	مخزون	بالمخزونات

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
مخصص	مخصص	مخصص	بالمخصصات
مخضرم	مخضرم	مخضرم	بالمخضرمين
مخطط	مخطط	مخطط	بالمخطط
مخطط	مخطط	مخطط	بالمخططات
مخطوط	مخطوط	مخطوط	بالمخطوطات
مخطوف	مخطوف	مخطوف	بالمخطوفين
مخلف	مخلف	مخلف	بالمخلفات
مخلوق	مخلوق	مخلوق	بالمخلوق
مخلوق	مخلوق	مخلوق	بالمخلوقات
مخمل	مخمل	مخمل	بالمخمل
مخيم	مخيم	مخيم	بالمخيم
مخيم	مخيم	مخيم	بالمخيمات
مداخل	مداخل	مداخل	بالمداخل
مداخيل	مداخيل	مداخيل	بالمداخيل
مدارس	مدارس	مدرسة	بالمدارس
مداعب	مداعب	مداعب	بالمداعبات
مدافع	مدافع	مدافع	بالمدافع
مدافع	مدافع	مدافع	بالمدافعة
مدافع	مدافع	مدافع	بالمدافعين
مدافن	مدافن	مدافن	بالمدافن
مدان	مدان	مدان	بالمدانين
مداهم	مداهم	مداهم	بالمداهمات
مداور	مداور	مداور	بالمداورة
مداول	مداول	مداول	بالمداولات
مداوم	مداوم	مداوم	بالمداومة
مدحل	مدحل	مدحل	بالمدحلة
مدخر	مدخر	مدخر	بالمدخرات
مدخل	مدخل	مدخل	بالمدخل
مدخن	مدخن	مدخن	بالمدخن
مدخن	مدخن	مدخن	بالمدخنين
مدد	مدد	مدد	بالمدد
مدراء	مدراء	مدراء	بالمدراء
مدراس	مدراس	مدراس	بالمدراس
مدرب	مدرب	مدرب	بالمدرب
مدرب	مدرب	مدرب	بالمدربين
مدرج	مدرج	مدرج	بالمدرج
مدرس	مدرس	مدرسة	بالمدرسة
مدرس	مدرس	مدرس	بالمدرسين
مدرع	مدرع	مدرع	بالمدرعات
مدعو	مدعو	مدعو	بالمدعو
مدعي	مدع	مدعي	بالمدعي

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
مدع	مدع	مدع	بالمدعين
مدفع	مدفع	مدفع	بالمدفع
مدفعي	مدفع	مدفعي	بالمدفعية
مدفع	مدفع	مدفع	بالمدفعين
مدفن	مدفن	مدفن	بالمدفن
مدفيع	مدفيع	مدفيع	بالمدفيعة
مدمن	مدمن	مدمن	بالمدمنين
مدن	مدن	مدن	بالمدن
مدني	مدن	مدني	بالمدني
مدني	مدن	مدني	بالمدنية
مدني	مدن	مدني	بالمدنيين
مدي	مد	مدي	بالمدي
مدي	مد	مدي	بالمدية
مديح	مديح	مديح	بالمديح
مدير	مدير	مدير	بالمدير
مديري	مدير	مديري	بالمديرية
مدين	مدين	مدين	بالمدينة
مدين	مدينت	مدين	بالمدينتين
مديوني	مديون	مديوني	بالمديونيات
مذابح	مذابح	مذبحة	بالمذابح
مذاهب	مذاهب	مذاهب	بالمذاهب
مذاييع	مذاييع	مذاييع	بالمذاييع
مذبح	مذبح	مذبح	بالمذبح
مذبح	مذبح	مذبحة	بالمذبحة
مذكر	مذكر	مذكر	بالمذكرة
مذل	مذل	مذل	بالمذلة
مذنب	مذنب	مذنب	بالمذنب
مذنب	مذنب	مذنب	بالمذنبين
مذهب	مذهب	مذهب	بالمذهب
مرا	مرا	مرا	بالمراة
مراتب	مراتب	مراتب	بالمراتب
مراجع	مراجع	مراجع	بالمراجع
مراجع	مراجع	مراجع	بالمراجعة
مراحل	مراحل	مرحلة	بالمراحل
مرار	مرار	مرار	بالمرارة
مراسل	مراسل	مراسل	بالمراسلين
مراسم	مراسم	مراسم	بالمراسم
مراسيم	مراسيم	مراسيم	بالمراسيم
مراعا	مراعا	مراعا	بالمراعاة
مرافع	مرافع	مرافع	بالمرافعة
مرافق	مرافق	مرافق	بالمرافق

Table A.1 (continued**)** Examples of stem of words

FARASA	LUCENE	AMIR	Word
مرافء	مرافء	مرافء	بالمرافء
مراقب	مراقب	مراقب	بالمراقبة
مراقب	مراقب	مراقب	بالمراقبين
مراكب	مراكب	مراكب	بالمراكب
مراكز	مراكز	مراكز	بالمراكز
مراهق	مراهق	مراهق	بالمراهقين
مراهم	مراهم	مراهم	بالمراهم
مراهن	مراهن	مراهن	بالمراهنات
مراهن	مراهن	مراهن	بالمراهنة
مراوغ	مراوغ	مراوغ	بالمراوغة
مربي	مرب	مربي	بالمربية
مرتب	مرتب	مرتب	بالمرتبة
مرتد	مرتد	مرتد	بالمرتد
مرتد	مرتد	مرتد	بالمرتدات
مرتدي	مرتد	مرتدي	بالمرتدي
مرتد	مرتد	مرتد	بالمرتدين
مرتزق	مرتزق	مرتزق	بالمرتزقة
مرتزق	مرتزق	مرتزق	بالمرتزقين
مرجان	مرج	مرجان	بالمرجان
مرجع	مرجع	مرجع	بالمرجع
مرجعي	مرجع	مرجعي	بالمرجعيات
مرجعي	مرجع	مرجعي	بالمرجعية
مرح	مرح	مرح	بالمرح
مرحل	مرحل	مرحلة	بالمرحلة
مرحلتين	مرحلت	مرحلتين	بالمرحلتين
مرحلي	مرحل	مرحلي	بالمرحلية
مرز	مرز	مرز	بالمرز
مرسوم	مرسوم	مرسوم	بالمرسوم
مرسيدس	مرسيدس	مرسيدس	بالمرسيدس
مرشح	مرشح	مرشح	بالمرشح
مرشح	مرشح	مرشح	بالمرشحين
مرشد	مرشد	مرشد	بالمرشد
مرصاد	مرصاد	مرصاد	بالمرصاد
مرض	مرض	مرض	بالمرض
مرض	مرض	مرض	بالمرضين
مرطب	مرطب	مرطب	بالمرطبات
مرغن	مرغن	مرغن	بالمرغن
مرفا	مرفا	مرفا	بالمرفا
مرفق	مرفق	مرفق	بالمرفق
مركب	مركب	مركب	بالمركب
مركب	مركب	مركب	بالمركبة

Table A.1 (continued**)** Examples of stem of words

Word	AMIR	LUCENE	FARASA
بالمركبتين	مركب	مركبت	مركب
بالمركز	مركز	مركز	مركز
بالمركزين	مركز	مركز	مركز
بالمرمر	مرمر	مرمر	مرمر
بالمرم	مرم	مرم	مرم
بالمروحيات	مروحي	مروح	مروحي
بالمروحية	مروحي	مروح	مروحي
بالمرور	مرور	مرور	مرور
بالمروض	مروض	مروض	مروض
بالمرونة	مرون	مرون	مرون
بالمريخ	مريخ	مريخ	مريخ
بالمريض	مريض	مريض	مريض
بالمريضتين	مريض	مريضت	مريض
بالمزاد	مزاد	مزاد	مزاد
بالمزادات	مزاد	مزاد	مزاد
بالمزارع	مزرعة	مزارع	مزارع
بالمزارعين	مزارع	مزارع	مزارع
بالمزاعم	مزاعم	مزاعم	مزاعم
بالمزالج	مزالج	مزالج	مزالج
بالمزايا	مزايا	مزايا	مزايا
بالمزايدات	مزايد	مزايد	مزايد
بالمزايدة	مزايد	مزايد	مزايد
بالمزج	مزج	مزج	مزج
بالمزحة	مزح	مزح	مزح
بالمزدوج	مزدوج	مزدوج	مزدوج
بالمزرعة	مزرعة	مزرع	مزرع
بالمزروعات	مزروع	مزروع	مزروع
بالمزروعي	مزروعي	مزروع	مزروعي
بالمزور	مزور	مزور	مزور
بالمزيج	مزيج	مزيج	مزيج
بالمزيد	مزيد	مزيد	مزيد
مسال	مسال	مسال	مسال

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname **:** ALI ABRAHEM ALI ALNAIED

Date of Birth **:** 19/05/1973

Phone **:** 0090 54 54 057 299

E-mail **:** a_alnaied@yahoo.com

EDUCATION

High School **:** 1989 – 1992

Baccalaureate in sciences, Tarhuna higher school, Tarhuna / Libya.

Bachelor **:** 1992 – 1996

BS in Computer Science, Computer Science Dept, higher institute of Science and Technology, Misurata / Libya.

Master Degree **:** 2002 – 2004

M.Sc. of Information & Communication Technology for Engineers, Coventry University, Coventry/ UK.

WORK EXPERIENCE

1997 – 2001

Assistant lecturer in a computer laboratory in Computer Dept. at Higher Institute of Comprehensive Vocational. Tarhuna /Libya.

2004 – 2006

Lecturer in Data Structure at the Elmergib University Libya. Al Khums/ Libya

2007 – 2008

Lecturer in Java Programming at the Higher Institute of Computer Professions in Al khamis Amseahl. Tripoli / Libya

2009 – 2010

Lecturer in Database & OOP at the Higher Institute of Electronic Professions. Alkarboly / Libya

2011 – 2013

lecturer in Database, Data structure at the College of Science and Technology-Tarhuna.

LANGUAGES

Arabic (Native), English.

TECHNICAL SKILLS

Programming Languages: Matlab, Java.

Platforms : Windows 98/2000/XP/Vista/7/8/10.

Tools : Latex, MS Office.

TOPICS OF INTEREST

Database, Data Mining, Information Retrieval.

PUBLICATIONS

- Alnaied, M., & M.Abduallah, “AN INTELLIGENT USE OF STEMMER AND MORPHOLOGY ANALYSIS FOR ARABIC INFORMATION RETRIEVAL” Egyptian Informatics Journal, Elsevier, 2020.