نموذج رقم (1)

إقـــــرار

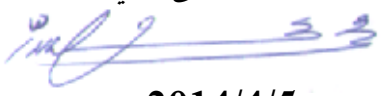
أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

تحسين التجذير العربي الخفيف في أنظمة استرجاع البيانات

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name	**محمد يحيى علي المصدَر**	اسم الطالب:
Signature		التوقيع:
Date:	**2014/4/5**	التاريخ:

Islamic University, Gaza, Palestine

Research and Postgraduate Affairs

Faculty of Engineering

Computer Engineering Department

Improving Arabic Light Stemming in Information Retrieval Systems

Mohammed Yahya Almusaddar

Supervisor

Prof. Ibrahim S. I. Abuhaiba

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1435H (2014)

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza
مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي 1150

الرقم... ج س ع/35/ ..... Ref
التاريخ ..... 2014/03/10 Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **محمد يحيى علي المصدر** لنيل درجة الماجستير في **كلية *الهندسة*** قسم **هندسة الحاسوب** وموضوعها:

تحسين التجذير العربي الخفيف في أنظمة إسترجاع البيانات

Improving Arabic Light Stemming in Information Retrieval Systems

وبعد المناقشة التي تمت اليوم الاثنين 09 جمادى الأولى 1435هـ، الموافق 2014/03/10م الساعة الثانية عشرة والنصف ظهراً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

أ.د. إبراهيم سليمان أبو هيبة	مشرفاً ورئيساً	.....
د. محمد أحمد الحنجوري	مناقشاً داخلياً	.....
د. إيهاب صلاح زقّوت	مناقشاً خارجيًا	 2-4-2014

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في **كلية *الهندسة*** / قسم **هندسة الحاسوب**.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله وإلى التوفيق،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 ص.ب 108 الرمال غزة فلسطين هاتف
public@iugaza.edu.ps www.iugaza.edu.ps

Dedication

To My Father, Mother,

To my wife,

To My Daughters,

To my Sister,

To My Friends,

To all who helped me, I devote this work.

Acknowledgements

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 thank my research supervisor, Prof. ***Ibrahim S. I. Abuhaiba****, for guiding and helping me throughout this research and other courses.*

As well I would like to thank the discussion committee, Dr. ***Mohammed Alhangouri and*** *Dr.* ***Ihab Zakout,*** *for their guides and commentaries.*

Table of Contents

List of Abbreviations

AI	Artificial intelligence
API	Application Programming Interface
CLEF	Conference and Labs of the Evaluation Forum
DFR	Divergence from Randomness
ETS	Educated Text Stemmer
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICT	Information and Communications Technology
IDE	Integrated Development Environment
IDF	Inverse Document Frequency
IR	Information Retrieval
LDC	Linguistic Data Consortium
NIST	National Institute of Standards and Technology
OSAC	Open Source Arabic Corpora
OOV	Out of Vocabulary
TC	Text Classification
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency

TREC	Text Retrieval Conference
URL	Uniform Resource Locator

List of Figures

List of Tables

تحسين التجذير العربي الخفيف في أنظمة استرجاع البيانات

محمد يحيى المصدَر

الملخص

استرجاع البيانات يقصد بها استخراج معلومات معينة من مستندات نصية من الصحف أو المجلات أو مواقع الويب، بهدف الوصول الى المعلومة المحددة. نتيجة للأبحاث المكثفة في مجال استخراج البيانات، يوجد العديد من الطرق التي تم تطويرها لتحسين استرجاع اللغة العربية في هذا المجال.

هذه الرسالة تتناول تحسين استخراج البيانات باللغة العربية من خلال التحسين على التجذير الخفيف وعمليات ما قبل المعالجة، والمساهمة في تطوير البرمجيات المفتوحة المصدر، أيضاً اقتراح نموذج لتنقية وتوحيد النص وحذف الكلمات المستبعدة.

قمنا بتقديم واجهة رسومية تقوم بعمليات ما قبل استرجاع البيانات، من خلال التحسين على عملية تنقية النص بتقديم مجموعة من القواعد لكي يتم استخدامها كمعيار في عملية تنقية النص، ثم بعد ذلك التحسين على حذف الكلمات المستبعدة عن طريق تقديم قائمتين منفصلتين للكلمات المستبعدة، الأولى قائمة تحتوي على عدد كبير جدا من الكلمات المستبعدة، والثانية قائمة خفيفة لكي نمنح الباحثين خيارات أوسع كمساهمة جديدة في مجال الكلمات المستبعدة وتأثيراتها على عمليات استرجاع البيانات.

قدم هذه البحث مساهمة جديدة وهي استخدام قائمة للكلمات المعربة حيث أن هذه الكلمات لا يجوز تطبيق قواعد التجذير الخفيف عليها لأنها لا تنتمي للغة العربية، تم إدراج هذا التحسين في الخوارزمية المقترحة للتجذير الحفيف ومقارنتها مع خوارزميات اخرى، ويبلغ عددها 100 كلمة وتم مقارنة الخوارزمية المقترحة مع خوارزميات موجدة في مجال التجذير الخفيف واستخراج نتائج أفضل مقارنة مع خوارزميات كل من Larkey و Khoja

في هذا البحث تم استخدام تطبيق مشهور من تطبيقات استرجاع البيانات والمخصص للأبحاث وهو منصة Terrier IR، قمنا بإضافة دعم اللغة العربية على هذا التطبيق واضافة الواجهة الرسومية التي قدمناها مع الخصائص الخاصة بها مع هذا التطبيق لتكوين إطار تطويري وبحثي لمجال استرجاع البيانات باللغة العربية.

تم استخدام نظام TF-IDFوالخاص بترتيب النتائج التي تم استرجاعها على هذا الإطار والحصول على نتائج باللغة العربية باستخدام إطار مفتوحة المصدر يمكن الباحثين في المستقبل من استخدامه والتطوير عليه بكل حرية.

تم استخدام مجموعات البيانات OSAC في الحصول على النتائج، واعتمدنا في البيئة التطويرية للإطار على كل من لغة البرمجة الجافا وعلى منصة Terrier IR مع جهاز ذو معالج CORE I7بتردد يبلغ 3.4 جيجا هرتز، وذاكرة عشوائية 8 جيجا بايت.

Improving Arabic Light Stemming in Information Retrieval Systems

Mohammed Yahya Almusaddar

ABSTRACT

Information retrieval refers to the retrieval of textual documents such as newsprint and magazine articles or Web documents. Due to extensive research in the IR field, there are many retrieval techniques that have been developed for Arabic language.

The main objective of this research to improve Arabic information retrieval by enhancing light stemming and preprocessing stage and to contribute to the open source community, also establish a guideline for Arabic normalization and stop-word removal.

To achieve these objectives, we create a GUI toolkit that implements preprocessing stage that is necessary for information retrieval. One of these steps is normalizing, which we improved and introduced a set of rules to be standardized and improved by other researchers. The next preprocessing step we improved is stop-word removal, we introduced two different stop-word lists, the first one is intensive stop-word list for reducing the size of the index and ambiguous words, and the other is light stop-word list for better results with recall in information retrieval applications.

We improved light stemming by update a suffix rule, and introduce the use of Arabized words, 100 words manually collected, these words should not follow the stemming rules since they came to Arabic language from other languages, and show how this improve results compared to two popular stemming algorithms like Khoja and Larkey stemmers.

The proposed toolkit was integrated into a popular IR platform known as Terrier IR platform. We implemented Arabic language support into the Terrier IR platform. We used TF-IDF scoring model from Terrier IR platform.

We tested our results using OSAC datasets. We used java programming language and Terrier IR platform for the proposed systems. The infrastructure we used consisted of CORE I7 CPU ran speed at 3.4 GHZ and 8 GB RAM.

Keywords: *light Stemming, preprocessing, Information Retrieval, Arabic Language, stop-word removal, open source*

Chapter 1 Introduction

This chapter introduces Information Retrieval (IR) and its importance and introduce the stemming process, also describes Arabic language and summarize its grammar.

IR systems along with natural language processing, speech recognition, text mining and automated document classification are very important these days, especially with the huge increase in the number of text databases and the amount of digital content available on-line. The demand for better techniques to access data faster and more efficiently is called for. Researchers who are enhancing these fields focus mainly on English or Latin languages, other languages such as Arabic language still needs more research to improve these fields.

1.1 Information Retrieval

In general, IR refers to the retrieval of unstructured data. Most often, it is related to text retrieval, i.e. the retrieval of textual documents such as newspaper and magazine articles or web documents

Due to extensive research in the IR field, there are many retrieval techniques that have been developed and which are being researched over the last three decades, two important techniques that have been used in this thesis are described below.

One of the major techniques that has been used in preprocessing stage in IR, natural language processing, speech recognition, and text mining is Stemming. **Christopher *et al.*** [1] define stemming as a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. In other words, it is the process of removing any affixes (prefixes that added to the beginning of the word, infixes that added to the middle of the word, or/and suffixes that added to the ending of the word) from words to reduce these words to their stems or roots under the assumption that words with the same stem are semantically related.

There are two major approaches that are followed for Arabic stemming. One approach is called light stemming (also called stem-based stemming) by which a word prefix and suffixes are removed, the other one called Root-based stemming (also called aggressive stemming) which reduces a word to its root. Another two approaches that have been researched are Statistical stemming and Manual constructing of dictionaries, the last one is not efficient and therefore not so popular. Studies show that light stemming outperforms aggressive stemming and other stemming types **Al-Maimani *et al*.** [2].

Another technique that is important as the stemming is Stop-word removal which is a procedure of eliminating language that do not apply any significance to a textbook. These common words are called stop words. Categories of stop-words cover adverbs, conditional pronouns, prepositions, and pronouns, transformers such as verbs, and letters, referral names and affixes such as prefixes, infixes, and postfixes.

1.2 Arabic Language

Arabic is the official language of millions of Arabs and Muslims around the globe. Although spoken Arabic can be easily understood by all Arabs, each Arab country has its own way of pronouncing Arabic words and it has its own variations of used vocabularies. Arabic has a complex morphology compared to English and European languages. Arabic has around 5 million words that are derived from around 11,300 roots compared to 400,000 key words in English which has a total of 1.3 million words (this exclude scientific names and nomenclature). Tens to hundreds of derivatives can be derived from a single root. This is the secret of why Arabic has almost three times more words than English.

1.2.1 Importance

The amount of Arabic digital information is growing rapidly on the web, statistics show that the number of Internet users in the Middle East has increased enormously due to the increasing in Information and Communications Technology (ICT) awareness and its importance within the Arab countries. According to **Internet World Stats** [3] the number of Internet users in the Middle East has increased from 3.3 Million in 2000 to 90 Million in 2012, with internet penetration of 40.2 % of the

population. This clearly shows the need to find effective tools and techniques for handling Arabic language.

1.2.2 Grammar

In this subsection we present a quick review about Arabic grammar. An Affix is a Morpheme added to a word to change its function or meaning. There are three basic ways of doing this:

- Prefix - by adding a morpheme to the beginning of a word.
- Infix - some languages add morphemes to the middle of the word, and Arabic is one such language.
- Suffix - by adding a morpheme to the end of a word.

There are four types of roots in Arabic:

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

Trilateral are the majority of roots. The root is the basic form of an Arabic word and any further analysis / changes on the root will result in losing the meaning of the word. The Arabic root of whatever word is morphologically mapped to:

- A standard form of three characters فعل
- A standard form of four characters فعلل
- A standard form of five characters فعللَ

Arabic consists of 28 characters which are written from right to left. These characters can have diacritical marks on them which are:

- Damma
- Fathah
- Kasra
- Shaddah

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, Arabic has two genders:

- Feminine
- Masculine

Also Arabic has three cardinalities:

- Singular
- Dual
- Plural

And Arabic has three grammatical cases

- Nominative
- Genitive
- Accusative

Also Arabic has two tenses

- Perfect
- Imperfect

Also Arabic nouns are made differently depending on the noun gender, cardinality, and grammatical case. The special character HAMZA can be composed in different glyphs:

- أ
- إ
- آ

These glyphs are dropped in many cases and replaced with a common shape of Alif (ا). The characters (ى and ي) are used based on different pronunciation. Moreover, characters ة and ه are used interchangeably. Most noun are change with the definite article (ال) which are always attached to its following word. There is no space between a word and its prefix, postfix, and pronoun. Arabic conjunction letters like (و) which means ‘and’ are written next to the word. Plurals are formed by adding suffixes or infixes to the noun or by completely changing its pattern like in case of broken plurals which do not follow a general rule. [2]

1.2.3 Challenges in Arabic IR systems and Arabic Language

Arabic IR has a long road ahead. Researchers need to formulate more efficient results for Arabic IR challenges such as

- Lack of resources for test-beds
- Regional variations, spelling variations and machine translations.
- Preprocessing techniques need to be united. Tokenization still has its challenges due to clitics and ambiguity.
- Linguistic stemming is a very rich area of research. A straightforward algorithm is needed for broken plurals.
- Automatic diacritization

Arabic is a challenging language for a number of reasons:

1. Orthographic with diacritics is less ambiguous and more phonetic in Arabic, certain combinations of characters can be written in different ways. For instance, sometimes in glyphs combining HAMZA with ALEF (أ) the HAMZA is dropped (ا). This causes the glyph ambiguous as to whether the HAMZA is present.
2. Arabic has a very complex morphology recording as compared to English language. For instance, to convey the possessive, a word shall have the letter (ي) attached to it as a suffix. There is no disjoint Arabic-equivalent of “my”.
3. Arabic words are derived: Arabic words are usually derived from a root (a simple bare verb form) that usually contains three letters. In some derivations, one or more of the root letters may be dropped. In such cases tracing the root of the derived word would be a much more difficult problem.
4. Broken plurals are common. Broken plurals are somewhat like irregular English plurals except that they often do not resemble the singular form as closely as irregular plurals resemble the singular in English. Because broken plurals do not obey normal morphological rules, they are not covered by existing stemmers.
5. In Arabic we have short vowels which give different pronunciation. Grammatically they are required, but omitted in writing Arabic texts.

6. Arabic synonyms are widespread. Arabic is considered one of the richest languages in the world. This makes the exact keyword match is inadequate for Arabic retrieval and classification

1.3 Topic Area

Stemming is used in IR systems to reduce variant word forms to common roots in order to improve retrieval effectiveness. As in other languages, there is a need for an effective stemming algorithm for the indexing and retrieval of Arabic documents.

As an important factor in Arabic IR, preprocessing step which contains stemming and stop-word removal need to be enhanced and add to different open source IR applications to allow researchers interested in Arabic IR use the advantage of these applications in improving Arabic IR

1.4 Thesis Questions

IR systems depend, among other things, on a preprocessing step to extract correct stems or roots especially on indexing stage. According to **Al-Nashashibi *et al*.** [4] there is no available standard stemmer for Arabic language.

Through the developing of stemming algorithm, important issues must be handled such as under-stemming, over-stemming and miss-stemming. Other issues that require handling for stemmers are compound words, proper nouns, foreign Arabized words, and irregular forms of words, these types of issues affect the process of stemming and needs more research. A general stemmer algorithm should handle most of these issues.

Before Stemming step **Al-Maimani *et al.*** [2], suggests that Stop-word elimination could be enhanced by looking into various stop-words used in different Arab countries. Moreover, there are many frequently used Arabic words which are not categorized as stop-words, although these words are essential to the meaning of a sentence, they can be ignored, more research is needed to look into how to avoid these types of words. Stop-word removal may affect the performance of stemming algorithms as some researchers claimed, this will be investigated through this thesis.

An existing available stop-word list will be used and updated to include new stop-words and other special cases.

One of the most important techniques in Arabic IR that will be used in this thesis is normalizing, until now no one show specifications to one region or personal preferences which should be considered to get the more standardized Arabic text.

Many normalizing procedures introduced by Arabic IR researchers have no justifications to its normalization techniques, for that more standardized normalization step is needed. Two normalizing techniques were reviewed to implement more general normalization procedures [5], [6].

There are many tools that include different stemming techniques for both monolingual and multilingual languages like **WEKA** [7]**, RapidMiner** [8] and other software packages. These packages may or may not contain Arabic language support, **Saad and Ashour** [9] added Arabic language support in which they including Arabic stemming algorithms to two of the most popular data mining, clustering and pattern recognition software which are WEKA and RapidMiner.

Indeed, Arabic language support needs more contribution especially for public packages and software that are reliable and popular among other researchers. The two software packages WEKA and RapidMiner are not specialized in IR and researchers prefers more specialized tools in the area they improve or enhance. There are other IR tools and packages that need Arabic language contribution, one of the most popular is **Terrier IR Platform** [10]**.**

The terrier is a highly flexible, efficient, and effective open source search engine, readily deployable on large-scale collections of documents, although Terrier uses UTF internally, and can support datasets written in languages other than English, Terrier IR do not contain Arabic Stemming algorithms, other areas that need to take care of is Terrier tokenization and normalizing processes which need to be modified and tested to handle Arabic words well. Terrier includes all stemmers from the **Snowball stemmer project** [11] which lack of Arabic stemming algorithm.

1.5 Thesis Significance

One of the major issues of IR systems is the choice of word that concerns the difference between the terms used for describing documents and the words used by the researchers or users to describe their information need, stemming is a key technique that solves this issue.

With the massive growth of the Arabic content as mentioned earlier, the need for effective stemming techniques for Arabic language has raised up, although many linguistic techniques and light stemmers have been developed for Arabic language there still many weaknesses and problems that need to be solved and enhancing the retrieval process still has a long way to go.

There are still many challenges and open outlets to be analyzed deeply. So the need for general or standard techniques is significant especially for stemming algorithms.

The main objective of the stemming process is to remove all possible affixes and thus reduce the word to its stem or root. Using stemming, many IR applications split words from prefixes and suffixes to their word stem, to make the search broader in the meaning that it can ensure that the greatest number of relevant matches is included in search results.

As shown in Figure 1-1 when the IR system deals with the word without stemming the same semantic words produce many variants which consume the size and the process time, in contrast as shown in Figure 1-2 stemming can improve the effectiveness of retrieval related document and to reduce the size of the index using less term that have the general form. Stemming allows a query term to focus more on the meaning of a term and closely related terms and less on specific character matches.

Figure 1-1: IR Indexing without stemming

Figure 1-2: IR Indexing with stemming

This thesis focus on improving light stemming techniques, because of extensive investigation done by Larkey et al. [12] which concluded that for IR systems the using of light stemming provided the highest performance followed by using root extraction, both improved the performance of IR systems but light stemming outperformed root extraction stemming. Many of the proposed stemmers, available in literature, do not handle some special cases in Arabic language.

Light stemming can be further improved by combining hybrid stemming techniques like statistical stemming techniques combined with light stemming techniques and root-based techniques.

A question that has been raised by many researchers, especially in the IR field, as to whether the root extraction is really the most effective stemming method for Arabic. These researchers point out a major drawback using root extraction is that it creates ambiguity in IR settings by creating too few conceptual category results in a loss of distinguishing semantic information present in words that could be important for more accurate IR. Despite these drawbacks, this technique can solve particular cases that light stemming cannot handle.

There are many Arabic stemming algorithms, but most of them are not available publicly and the need for the general Arabic stemmer algorithm is rising. There are many issues that challenge the researchers in stemming techniques that need researching to improve the effectiveness of IR systems.

Other important techniques in Arabic IR that must be taken care of is normalizing, until now no one show specifications to one region or personal preferences which should be considered to get the more standardized Arabic text.

Stemming improve IR systems by minimizing the storage requirements and increasing matching between documents terms and query terms. Stemming can cause errors in the forms of over-stemming, miss-stemming and under-stemming. These errors decrease the effectiveness of stemming algorithms, however reducing one type of errors can lead to an increase of the other.

Over-stemming occurs when two words with different stems are stemmed to the same root. An over-stemming example is when the word "probe" and "probable" are merged together after stemming.

Under-stemming occurs when two words that should be stemmed to the same root are not, for example, when the stemmer fails to conflate the words "adhere" and the word "adhesion" to the same root.

Miss-stemming is defined as "taking off what looks like an ending, but is really part of the stem for example, stemming the word "red to "r" or the word "reply to "rep" **Al-Shammari and Lin** [13]**.** These types of error need more research, especially in Arabic stemming techniques.

Other types that are common source of error in IR systems are vowel omission due to Arabic Orthography and Out of Vocabulary (OOV) due to the influence of foreign languages. Vowels distinguish often one word from another. While human Arabic readers can decipher the meaning of a term by context, the standard practice of omitting vowels in Arabic text causes ambiguity that is difficult for computers to handle. The one major exception to the practice of vowel omission is the text of the Qur'an, in which all diacritics must be included to avoid this inherent ambiguity. OOV adopted the use of many foreign words. Transliteration of these words into the Arabic alphabet is not standardized. Additionally, OOV words do not follow standard Arabic transformation rules. Thus, OOV words complicate the process of term stemming **Morris** [14]**.**

Stop-word elimination is very important step and can improve the effectiveness of IR systems. Stop-word removal can be enhanced by looking in more words that not

helping the Retrieval System to get better results, these types of words need to be eliminated from documents and queries. Stop-words can enhance and add significant improvements to process Arabic documents. This thesis will improve and enhance a general stop-words list and handle special cases that could improve the effectiveness of the stemming algorithm

Terrier IR Platform is open source, and is a comprehensive, flexible and transparent platform for research and experimentation in text retrieval. Research can easily be carried out on the standard Text Retrieval Conference (TREC) and Conference and Labs of the Evaluation Forum (CLEF) test collections.

One limitation with Terrier IR is the Arabic language support in tokenization and stemming steps, the need for implementing and integrating Arabic support is very obvious.

1.6 Thesis Contribution

This thesis will contribute to the following:

- Implement a light stemmer algorithm with the use of other techniques features like root-based, dictionary based then taking care of existing issues in other stemming algorithms and open challenges that stemming field are facing with taking care of normalization process to get the more standardized text.
- Cover the issue of foreign words by different languages and countries due the difference in spoken Arabic and other Arabic words that could be neglected and added to Arabic stop-words list or create own list of these types of words.
- Add the suggested Arabic stemmer algorithm to one of the most popular IR research platform “Terrier IR platform” along with tokenization and normalization modification to handle Arabic language.
- Collect and manage intensive stop-word list to reduce the size of vocabulary as much as possible, also provide light stop-word list which is much smaller than intensive stop-words list and slightly affects the recall.
- Contribute an open source Graphical User Interface (GUI) which explains the preprocessing steps in IR systems and including different stemming types to be the basis of Arabic stemming algorithm.

- Improve the normalization process of existing Arabic light stemming algorithms.
- Contribute to open source communities by providing code to be publicly accessible and improve other open source projects that lack of Arabic support.

1.7 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 introduces the related work, that are relevant to our work, next in Chapter 3 we will overview some background, then in Chapter 4 we will describe the methodology, where in chapter 5 we will show the results of our work, and finally the conclusion of the research in Chapter 6, which will summarize the research, and some notes around the future work.

Chapter 2 Related Work

2.1 Arabic Stemming Algorithms

The research for the stemming of Arabic text has been fairly popular. This popularity has resulted in the development of a large number of stemming approaches over the past ten years, most of which can be divided into two main groups light stemming and root extraction, distinguishable by the lexical character of their resultant stems.

According to the desired level of analysis, Arabic stemming algorithms can be classified, into four classes:

- Manually constructed dictionaries.
- Root-based stemming (morphological analyzers).
- Statistical stemmers.
- Light stemmers.

Most modern stemming algorithms exhibit a combination of these approaches known as hybrid methods which this thesis will consider as base research methodology.

Sembok and Ata [15] have developed algorithms for Arabic stemming and incorporated it in their experimental system in order to measure retrieval effectiveness. The results have shown that the retrieval effectiveness has increased when stemming is used. Their analysis suggests that most of the errors are due to the order in which the stemming rules are applied.

Otair [16] evaluates many Arabic stemming algorithms and he shows that Arabic stemming algorithm is still one of the most information retrieval challenges. He compared the most of the commonly used light stemmers in terms of affixes lists, algorithms, main ideas, and information retrieval performance. The results show that the light10 which is a Larkey algorithm stemmer outperformed the other stemmers. Finally, recommendations for future research regarding the development of a standard Arabic stemmer were presented by him, these recommendation meet with our conclusion for a standard Arabic stemming approach.

2.1.1 Light Stemmers

The light stemming approach is also called affix which is a process of stripping off a small set of prefixes and/or suffixes, without trying to deal with infixes, or recognize patterns and find roots.

In their work **Al Hajjar *et al***. [17] presented a new application that allows the evaluation of Arabic roots extraction methods in the same conditions on a corpus of two thousand words. The implemented methods in their system are:

- Light stemmer.
- Arabic stemming without a root dictionary.
- MT-based Arabic stemmer.
- N-gram based on similarity coefficient.
- N-gram based on dissimilarity coefficient.

They found that the method "Stemming Arabic without a Root Dictionary" achieves the best results.

Al-Shalabi *et al.* [18] proposed a new rule-based light stemming to solve ambiguity problem related to light stemming by finding if a specific sequence in the word is an affix or is part of the original work to determine to split it or not. The method they proposed depends on a set of possible affixes in which contains prefix and suffix. In prefix set they combined all possible antefixes and prefixes, on the other hand, the suffix set combines all possible suffixes and postfixes. Before they stem any word, they compared it against an Arabic word patterns list collected from different resources if there is a match, then this prefix is part of a word and will not be truncated. If the word does not match any Arabic pattern, then the relationship between the suffix and prefix will be examined, if there is no relationship between the suffix and prefix then at least one of them is part of the word. If the relationship is valid then the count of the letter of the word will be considered, after few rules that based on the letter, number, they consider solving the issue of the plural form of irregular nouns. To compare their work they use the sample terms list and compare it to Larky light stemmer and to Khoja root-extraction stemmer.

Larkey *et al.* [12] developed several light stemmers for Arabic, and they evaluate their effectiveness for IR using standard TREC data. They also have compared light stemming with several stemmers based on morphological analysis**.** They found that light stemming allows remarkably well IR without providing correct morphological analyses.

Al Ameed *et al.* [19] carefully analyzed the Text Retrieval Conference 2002 (TREC-2002) light stemmer algorithm then they design new algorithms to get better stemmed results. The enhancement attempts move toward into two main approaches, the first covers the TREC pre-defined removable suffixes and prefixes groups, including the contents of the terms in each set and the number of terms. The second approach focuses on modifying the sequence of algorithm components execution. The proposed light stemmer algorithms were assessed by using more than 1450 Arabic words, including different sets of affixation, patterns, as well as hollow verbs and various types of strong verbs. The proposed approaches provide better accepted (meaningful) outcomes of Arabic words with up to 30-50% more than TREC stemmer outcomes.

For Affix removal algorithms [5], [6].and [20] introduce a very close set of rules with certain conditions to remove prefixes and suffixes from words. Each one of them define a set of rules which apply to words in order to remove prefixes or suffixes like removing definite articles for prefixes or removing pronouns for suffixes.

2.1.2 Root-based Stemmers

Root-based stemming uses morphological analysis to extract the root of a given Arabic word. One of the most popular root-based stemmers are the **Khoja's stemmer** [21]. The Khoja algorithm removes suffixes, infixes and prefixes and uses pattern matching to extract the roots. Khoja's stemmer includes lists of diacritic characters, punctuation characters, definite articles, and 168 stop-words that will be used to normalize search terms. The algorithm suffered from problems, especially with broken plurals. The same apply for **Buckwalter** [22] who manually constructed dictionaries generate dictionaries (tables) listing the Arabic prefixes, suffixes, and stems/roots

Goweder *et al.* [23] adapted the existing root stemmer developed by Khoja and added different methods for identifying broken plurals. Broken plurals are nouns that do not follow a specific pattern for pluralizing and, as a result, do not resemble their root closely in the plural form. They found that practicing light stemming and adding broken plurals to a search term improved performance for IR systems.

Taghva *et al.* [24] tried to improve the Khoja stemmer to make it more competitive, they eliminated the need for a dictionary to support their root stemmer, thus eliminating the intensive maintenance and system requirements associated with a dictionary of the entire Arabic language. The normalization implemented by them is more conservative that previously practiced by Khoja.

Al-Omari *et al.* [25] introduced the design and implementation of a new Arabic light/heavy stemmer called Arabic Rule-Based Light Stemmer, which is not based on Arabic root patterns. Instead, it depends on mathematical rules and some relations between letters. A series of tests were conducted on Arabic Rule-Based Light Stemmer to compare the effectiveness of this new Arabic stemmer with the effectiveness of Khoja stemmer. The test they performed shows clearly Arabic Rule-Based Light Stemmer is more effective than the other tested Arabic stemmer.

Al-Kabi [26] shows that many Arabic stemming algorithms are presented during the last fifteen years. Khoja Arabic stemmer is considered by a number of researchers as a standard stemmer for Arabic language. Therefore his study attempts to improve the effectiveness of this standard stemmer, through the adoption of new additional (Patterns, Forms). Adding those patterns help to improve its accuracy by around 5%.

2.1.3 Statistical Stemmers

Darwish and Ali [27] presented a method for generating morphologically related tokens from Wikipedia hypertext to page title pairs. They indicated that the method overcomes some of the problems of statistical stemming to yield statistically significant improvements in Arabic retrieval over using statistical stemming. The technique can also be applied to words to yield results that statistically indistinguishable from statistical stemming. The technique had the added advantage of detecting variable spellings of transliterated named entities. For future work, they would like to try the proposed technique on other languages, because it would

likely be effective in automatically learning character-level morphological transformations as well as overcoming some of the problems associated with stemming. It is worthwhile to devise models that concurrently generate morphological and phonologically related tokens.

Boudlal *et al.* [28] presented a morphological analysis system for unvoweled Arabic sentences. The morphological analysis process often gives multiple solutions. They showed that by introducing the context, an approach based on hidden Markov models can have very good results in choosing the correct root of the word.
The results of tests carried out on both parts of the system are very encouraging and can be improved by further analysis of hamzated words in the analysis out of context, by using a larger corpus in the Markovian approach. Currently, they also extend the work of the first module in order to generate other tags of the words (noun, verb, particle, adjective, adverb, and possible vowelization,) then, they use an adaptation of the Markovian approach to identify the best vowelization of the word in context

2.1.4 Hybrid Stemmers

Morris [14] claims that the most promising direction of research lies in stemming methods that combine hybrid stemming techniques with statistical string matching techniques, especially *n*-gram based approaches. Although some researchers consider the use of stem-based techniques and root-based techniques as hybrid methods, the use of hybrid techniques have better results than the use of one technique alone.

Dilekh and Behloul [29] apply five different methods of stemming, and compare their results. They reason out which method yields better performance in IR. Among these five methods of stemming, they propose a new hybrid method of stemming from the integration of three different techniques (deletion of affix, the use of dictionaries, and morphological analysis) to improve the overall performance of the process of stemming. They also showed the effectiveness of removing suffixes before prefixes during the stemming process of Arabic texts.

Boubos *et al.* [30] used genetic algorithms and pattern matching machine learning to generate a morphological analyzer for Arabic verbs to derive morphological rules from word-pattern pairs. The input to the learning module consists of learning patterns. First, they compiled a file containing the 3,089 patterns .On each other line,

the pattern is written with its letters separated by spaces. On the lines following each pattern, the root is written. The file is then fed into the learning module. To perform a morphological translation for a new input word, the word is a string matched to the source part of the rules.

Alhanini and Aziz [31] proposed enhanced Arabic stemming algorithm that use both the light stemming and dictionary-based stemming and designed to overcome the dis-advantages of them both. For example light stemming cannot solve the problem of the broken (irregular) plurals for nouns and irregular verbs these types of issues have been handled by the dictionary-based stemmer. In contrast, the words that cannot be stemmed in the dictionary-based stemmer because they are not found in the lexicon of Arabic stems have been handled by the light stemmer. They evaluate their work and use an in-house collected corpus from Arabic newspaper archives.

Hadni *et al.* [32] proposed a hybrid method which incorporates three different techniques for Arabic stemming to overcome some of the existing stemming algorithms weaknesses. The three techniques are: Khoja Stemmer, Light Stemmer and N-Gram. Each one of these techniques needs individually some adaptation to be appropriate for Arabic language usage. Then they prepare a valid root file derived from a dictionary and a stop-word file. After that they removed prefix / suffix, by checking if the word matches of the patterns, extract the relevant word, otherwise to remove the suffix and prefix respectively, with verification of the length after each removal of affixes. Finally the valid root is found by using the bi-gram and the dice measure similarity.

Al-Nashashibi *et al.* [4] proposed linguistic approach for root extraction as a preprocessing step for Arabic text mining. The linguistic approach is composed of a rule-based light stemmer and a pattern-based infix remover. The rule-based light stemmer removes prefixes and suffixes from the word according to specific rules. The pattern-based infix remover removes infixes from the word according to specific patterns. They proposed a correction algorithm to handle some of the issues in Arabic words like: weak, hamzated and irregular words.

The accuracy of the extracted roots was determined by comparing them with a predefined list of 5400 trilateral and quadrilateral roots. The performance was tested

on collected in-house consisting of eight categories with and without the correction algorithm which improve the linguistic approach about 14 % as they claimed. They aim to improve Arabic stemming techniques and to use stems instead of words in the Arabic Text Mining field. In another paper **Al-Nashashibi *et al.*** [33] compared the performance of different techniques for stemming Arabic words with the previous approach and prove that their approach is more efficient.

Al-Shammari and Lin [13] presented Educated Text Stemmer (ETS). ETS is a dictionary free, simple, and highly effective Arabic stemming algorithm that can reduce stemming errors in addition to decreasing the computational time and data storage as they claimed. They propose an alternative technique that defines a rule to stem words instead of chopping off the letters. This rule is set by the syntactical structure of the word. For example, verbs require aggressive stemming and need to be represented by their roots. Nouns on the contrary, only require light suffixes and prefixes elimination. They use an evaluation method proposed by Paice, applied to compare various English stemmers to compare their work. This evaluation approach proves to be not viable, according to **AlSerhan *et al.*** [34] experiments, negative through exposes, such negative results of uses such approach for Arabic language. The Paice method is considered with grouping the stems from the perspective of IR system rather than the correctness of the stem itself. For this reason and for the results they obtained they claimed that the Paice evaluation method is not viable for Arabic language in order to compare the accuracy performance of two different stemming algorithms.

A hybrid stemming method was proposed by **Goweder *et al.*** [35] that attempts to determine the stem of a word according to linguistic rules. The proposed method integrates three different stemming techniques to improve the overall performance of the stemming process. The three techniques are: affix removal, dictionaries, and morphological analysis.

2.2 Normalizing

In normalizing and due to the absence of Arabic normalization standard each researcher has a different idea of normalizing the Arabic text which must be addressed. **Larkey *et al.*** [20] choose to normalize the text as shown in Figure 2-1:

Algorithm 2.1: Larkey Normalizing Algorithm

Purpose: Normalize Arabic Words

Input:

- Arabic Words

Output: Normalized Arabic Words

Procedure:

1. Converting it to Windows Arabic encoding (CP1256)
2. Removing punctuation
3. Removing diacritics (short vowels)
4. Removing non-letters
5. Replacing all modified alefs إ أ with ا
6. Replacing final ى with ي
7. Replacing final ة (a letter usually used to indicate the feminine gender) with ه

Figure 2-1: Pseudo Code of Larkey Normalizing Algorithm

Aljlayl and Frieder [6] integrated their discussion of normalization with that of stemming by listing it as steps one through four in a seven-step stemming process their normalization technique as shown in Figure 2-2:

Algorithm 2.2: Aljlayl and Frieder Normalizing Algorithm

Purpose: Normalize Arabic Words

Input:

- Arabic Words

Output: Normalized Arabic Words

Procedure:

1. Removing diacritics (short vowels)
2. Replacing all modified alefs إ أ with ا
3. Replacing final ى with ي

4. Replacing the two final letters ى and ء with ئ
5. Replacing the two final letters ي and ء with ئ
6. Replacing final ه with ة
7. Remove the prefix و if the word is three characters or longer

Figure 2-2: Pseudo Code of Aljlayl and Frieder Normalizing Algorithm

Lastly, **Chen and Gey** [5] explained their normalization standards in the "Preprocessing" section of their study. Text normalization for the stemmer created by Chen and Gey as shown in Figure 2-3:

Algorithm 2.3: Chen and Gey Normalizing Algorithm

Purpose: Normalize Arabic Words

Input:

- Arabic Words

Output: Normalized Arabic Words

Procedure:

1. Converting it to Windows Arabic encoding (CP1256)
2. Considering punctuation marks to be delimiters
3. Replacing all modified alefs إ أ with ا
4. Replacing final ه with ة
5. Replacing final ى with ي
6. Removing the diacritic shadda(ّ)

Figure 2-3: Pseudo Code of Chen and Gey Normalizing Algorithm

None of the studies justify its normalization techniques or makes an argument for or against certain components. Characteristics of the Arabic language itself and Arabic in word processing environments support the most basic normalization. While with stemming it can be argued that less stemming is safe and therefore better to ensure that meaning is not subtracted from the keywords, with normalization, the more normalized a text is the better. In more standardized text, a term is more likely to find

a match when unique characters, discrepancies in type, and regional preferences are eliminated. [36]

2.3 Stop-word Removal

El-Khair [37] measure the effectiveness of three stop-words lists for Arabic IR which are general stop list, corpus-based stop list and combined stop list. He used popular weighting schemas and used an Arabic Newswire dataset from the Linguistic Data Consortium (LDC) he found that the overall performance of the general stop list was better that the other two lists.

Similar research to **El-Khair's** [37] was investigated by **Atwan et al.** [38],they measured the effectiveness of three stop-word lists with light stemming for Arabic information retrieval (General stop-words list, Khoja stop-words list, Combined stop-words list). They used vector space model as the popular weighting scheme was examined. The idea is to combine (General and Khoja) stop-words lists with light stemming to enhance the performance, and compare their effects on retrieval. The Linguistic Data Consortium (LDC) Arabic Newswire data set was used. The best performance was achieved with the combined stop-words list, with light stemming.

Al-shalabi et al. [39] designed and implemented an algorithm for removing the Arabic stop-words based on a deterministic finite machine, they created the stop-list of more than 1000 words using several sources, one is the list compiled from other researchers work and the other are translated from English stop-lists to Arabic. They tested their system using 242 Arabic Example from the Proceedings of Saudi Arabian National Computer Conferences with a total number of 47897 words and another set from the holy Quran

2.5 Arabic Stemming Toolkit

According to **Al-Nashashibi *et al*.** [4] there is no available standard stemmer for Arabic language. And yet researchers still search in the same areas as we saw in normalizing how different researcher implement the same technique for normalizing, but without any standard even for other preprocessing steps like encoding,

tokenization and stop-word removal, so the need for a basic standard approach with the support of open source tools will help to guide the research in this field.

The contribution to Arabic tools by enhancing existing open source tools is introduced by **Saad and Ashour** [9] who added Arabic language support in which they including Arabic stemming algorithms to two of the most popular data mining, clustering and pattern recognition software which are WEKA and RapidMiner. But these tools are for Text Classification (TC) and may not be optimized for IR applications, so the need for a basic toolkit for preprocessing and the support of Arabic language in an open source platform is needed for creating a standard approach for enhancing and improving preprocessing and Arabic IR.

Chapter 3 Background

3.1 Arabic Stemming Techniques

Word stemming is one of the most important factors that affect the performance of IR systems. Arabic stemming algorithms have become a popular area of research. Many computational linguists and researchers have designed and developed algorithms to solve the problem of morphology and stemming.

Each researcher proposed his own gold standard, testing methodology and accuracy measurements to test and compute the accuracy of his algorithm.

Stemming aim to find the lexical root or stem for words in natural language, by removing affixes attached to its root, because an Arabic word can have a more complicated form with those affixes.

There are four kinds of affixes: antefixes, prefixes, suffixes and postfixes that can be attached to words. An Arabic word can represent a phrase in English, for example the word ليحدثونهم which means "to speak with them" is decomposed as shown in Table 3-1 [40].

Table 3-1: Arabic Affixes Example.

Antefix	Prefix	Root	Suffix	Postfix
ل	ي	حدث	ون	هم
Preposition meaning "to"	A letter meaning the tense and the person of conjugation	speak	Termination of conjugation	A pronoun Meaning "them"

In this Chapter we will see how different stemming algorithms including light stemming technique stem Arabic words in different ways and the goal of all of these

techniques to find a better representation of the word as the root of the word or light term of the word for better accuracy and save space and time in indexing process.

Indeed, there is no clear classification of stemming types, although there is general classification, but different types and names is used interchange so we will identify every name and classify in the literature and then group them.

3.1.1 Light-Stemming Technique

Light Stemming is to find the representative indexing form of a word by removing affixes, the main goal of light stemming is to get a better reduce the small size of the word to improve the retrieval performance of an Arabic IR system.

Light stemmers are mainly used in IR. Its main idea is to truncate prefixes and suffixes from a word to produce a stem. Furthermore, light stemmers find the representative indexing form of a word by application of truncation of affixes. Light stemmer is not concerned with root extraction

In other words, light stemming refers to the process of stripping off a small set of prefixes or suffixes without trying to deal with infixes, also most Light stemming techniques classifies these affixes into four kinds of affixes as shown in Table 3-1.

Many researchers create a new enhanced groups of prefixes and suffixes based on TREC-2002 terms, in Table 3-2 and Table 3-3 researchers [18] group both prefixes and suffixes to combine all possible cases.

Table 3-2: Arabic Prefixes set 1

Prefix 1	ي ت ن ب ل
Prefix 2	لن لت لي با فا كا سن ست سي لل ال فن في فت
Prefix 3	وسا وسن وست وسي ولل كال فال بال وال ولن ولت ولي ولا
Prefix 4	وبال وكال

Table 3-3: Arabic Suffixes set 1

Suffix 1	ه ة ك و ي ن ا ت
Suffix 2	تم وا نا كن كم ها هن هم ات ون ين ان تك تا يا ما يه ته تن تي

Researchers in [19] also proposed an enhanced Prefixes and suffixes list shown in Table 3-4 and Table 3-5 respectively and based also on TREC-2002 Prefixes and Suffixes lists.

Table 3-4: Arabic Prefixes set 2

Prefix 1	ي ت ن
Prefix 2	ال لل سي سا ست سن كا فا با ب ل لي لت لن فت في فن
Prefix 3	وال بال فال كال ولل وسي وست وسن وسا ولا ولي ولت ولن
Prefix 4	وبال

Table 3-5: Arabic Suffixes set 2

Suffix 1	ه ة ك و ي ن ا ت
Suffix 2	ان ين ون ات هم هن ها كم كن نا وا تم ني تن ته يه ما يا تا تك

TREC-2002 developed as an improved version of TREC-2001 [41] and TREC-2001 stemmer was a modified version of the Larkey [20] stemmer in which two additional prefixes identified.

TREC Light Stemmers remove only prefixes and suffixes. Five pre-defined groups of removable prefixes and suffixes were offered. For prefixes, there are three groups (one, two and three characters), and two groups (two and three-characters) for suffixes, Table 3-6 and Table 3-7 shows the TREC-2002 predefined set of prefixes and suffixes respectively.

Table 3-6: TREC-2002 predefined Prefixes set

Prefix 1	و ل ب
Prefix 2	كا وس سي لا وب وت وم لل يا وا ال فا ول وي
Prefix 3	كال وال فال بال ولل مال اال سال لال

Table 3-7: TREC-2002 predefined Suffixes set

Suffix 1	ه ة ي ت
Suffix 2	ها يه هم نا ما وا يا ني هن كم كن تم تن ان ين ون ات

There are many types of Arabic stemming algorithms. We named different types, in the next few subsections.

3.1.2 Root-based Technique

Root-based stemmers extract the root from the word by means of morphological analysis. It attempts to restore original root of a word and group words accordingly.

The two basic steps of root-based stemmers are first to remove prefixes, and suffixes. Second, is to extract roots by analyzing Arabic words according to their morphological components. Most root-based stemmers matches the remaining word against the patterns of the same length to extract the root, this is accomplished by rule-based techniques, table lookup, or a mixture of two.

3.1.3 Rule-Based Technique

Most rule-based techniques use light stemming algorithm idea to then improve the removing process of affixes by adding specific rules to the removal process to improve light stemming and solve issues that original light stemming algorithm cannot handle, for example it add to the removal process Arabic root patterns to match it with the word before removing it.

There are some researchers [4] who use rule-based stemming to improve root extraction, which tell us that different stemming techniques can be used together only the application will matter and based on we choose the appropriate stemming technique.

3.1.4 Dictionary-Based Technique

The dictionary-based stemming is the morphological approach that depends on a set of lexicons of Arabic stems, prefixes, and suffixes to extract the stem of words. This stemming can find the stem of the broken (irregular) plurals of nouns and irregular verbs, because the stem of these irregular words had been entered [31].

The first step of this stemming is preprocessing which identifying sentence boundaries, to split the running text into tokens so that it can be fed into morphological analyzer and parser processing. This step is to remove redundant and misspelled space. It also to resolve the orthographic variation in Arabic writing which can be changed or unchanged the meaning.

The second step is the named entity recognition. The main purpose of this step is to identify Arabic names using some heuristic also some lists of special verbs that are identified as introducing person names and descriptions that are identified to be linked to person names. The general idea behind this process is that most of the Arabic names are real words that frequently used.

3.1.5 Statistical Stemming Technique

In statistical stemmer, related words are grouped based on various string similarity measures and often involve n-gram technique [42]. Equivalence classes can be formed from the words that share some initial letter n-gram or by refining these classes with clustering techniques. An n-gram is a set of n consecutive characters extracted from a

word. The main idea behind this approach is that, similar words will have a high proportion of n-grams in common. Typical values for n are 2 or 3, these corresponding to the use of diagrams or trigrams, respectively.

3.2 Popular Arabic Stemmers

Here we explain some of the popular Arabic stemming algorithms, each one of them cover a type of Arabic stemmers, to get a clear understanding of different types of stemming techniques.

3.2.1 Larkey Algorithm

A popular light stemming, the algorithm stages are described in Figure 3-1

Algorithm 3.1: Larkey Light Stemming Algorithm

Purpose: Stemming Arabic Words

Input:

- Dataset
- Stop-word list

Output: Stemmed Dataset

Procedure:

1. Stop-words removal.
2. Remove punctuation, diacritics, non-letters, and non-Arabic.
3. Replace initial آ, أ, إ with ا
4. Convert ى to ي and ة to ه
5. Remove the letter (و) from the beginning of the word only if the resulting word is more than 3 letters.
6. Remove definite articles "آال" ، "فال" ،"وال" ، "البال"، from the beginning of the word only if the resulting words are two or more letters.
7. Remove the suffixes "ين" ، "ون" ، "ات" ، "ان" ، "هاي" ، "ة" ، "ه" ، "ية" ، "يه"، from the end of the word (longer first) only if the resulting word length is two or more.

Figure 3-1: Pseudo Code of Larkey Algorithm

The Larkey light stemmer ignored many suffixes, which can cause a high under stemming error rate. Additionally, the Larkey list of suffixes and prefixes can be original letters.

3.2.2 Khoja Algorithm

Khoja's stemmer removes the longest suffix and the longest prefix. It then matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop-words. The Khoja's algorithm initially removes suffixes, infixes and prefixes and uses pattern matching to extract the roots, but suffered from problems especially with nouns. A summarization of the Khoja's stemming procedure is shown in Figure 3-2

Algorithm 3.2: Khoja Root-Based Stemming Algorithm

Purpose: Stemming Arabic Words

Input:

- Dataset
- Stop-word list
- Assets and patterns files

Output: Stemmed Dataset

Procedure:

1. Replace initial إ, أ, آ with ا.
2. Stop-words removal.
3. Remove punctuation, non-letters and diacritics.
4. Remove definite articles from the beginning of the word.
5. Remove the letter (و) from the beginning of the word and (ة) from the end of the word.
6. Remove prefixes and suffixes
7. Comparing the resulting word to patterns stored in the dictionary, if the resulting root is meaningless the original word is returned without changes.

Figure 3-2: Pseudo Code of Khoja Algorithm

3.3 Preprocessing

One of the most important steps in IR is document preprocessing which aims to prepare the document or text to be used in the IR systems, preprocessing step can contains many sub steps to say that the document is ready to be used in IR systems, for example a document needed to be converted to the right encoding before next step, also the text needed to be normalized to unify all the document with specific rules guided by the IR system, then the size of the document need to be reduced from unnecessary words that will not affect the retrieval process.

Preprocessing phase aims to transform the collected Arabic text documents into an easily accessible representation of texts that is suitable for the Stemming algorithm. Mostly preprocessing step include the removing of punctuation marks, diacritics, stop-words and non-letters.

3.3.1 Encoding

A lot of the texts of the documents and topics are encoded in different schemes, Windows CP-1256 encoding [43] is a code page used to write Arabic, Persian, and Urdu under Microsoft Windows, it encodes every abstract single letter of the basic Arabic alphabet, not glyphs.

Another popular encoding is UTF-8 [44] which is a variable-width encoding that can represent every character in the Unicode character set. UTF-8 has become the dominant character encoding for the World Wide Web, UTF-8 is also increasingly being used as the default character encoding in operating systems, programming languages, Application Programming Interfaces (API), and software applications.

3.3.2 Normalization

Normalization often removes punctuation, diacritics (primarily weak vowels) and non-letters also it may replace different variation of a letter with a general form of the same letter, although this may affect the retrieval efficiency as these normalization steps may change the meaning of the word which leads to ambiguous results.

3.3.3 Stop-word Removal

Stop-words are very common words that appear in the text that carry little meaning, they serve only a syntactic function but do not indicate the subject matter. These stop-words have two different impacts on the IR process. They can affect the retrieval process because they have a very high frequency and tend to diminish the impact of frequency differences among less common words, affecting the weighting process.

The removal of the stop-words also change the document length and subsequently affects the weighting process. They also affect efficiency due to their nature and the fact that they carry no meaning, which may result in a large amount of unproductive processing.

The removal of the stop-words can increase the efficiency of the indexing process as 30% to 50% of tokens in a large text collection can represent stop-words [37].

Identifying a stop-words list or a stop-list that contains such words in order to eliminate them from text processing is essential to an IR system. Stop lists can be divided into two categories, domain independent stop-lists and domain dependent stop-lists. They can be created using syntactic classes or using corpus statistics, which is a more domain dependent approach, used for well-defined fields. They can also be created using a combination of the syntactic classes and corpus statistics to obtain the benefits of both approaches.

Stop-words belong to several word groups such as conjunctions, prepositions, adverbs etc., and often referred to as function words. Most often the using of stop-word lists for English is rather small, consisting of 200-400 words, due to the morphological richness of the language, the list contains all possible morphological variants of each stop-word [39].

3.4 Terrier IR Application

Terrier [45], Terabyte Retriever, is a project that was initiated at the University of Glasgow in 2000, with the aim to provide a flexible platform for the rapid development of large-scale IR applications, as well as a state-of-the-art test-bed for research and experimentation in the wider field of IR.

The Terrier project explored novel, efficient and effective search methods for large-scale document collections, combining new and cutting-edge ideas from probabilistic

theory, statistical analysis, and data compression techniques. This led to the development of various retrieval approaches using a new and highly effective probabilistic framework for IR, with the practical aim of combining efficiently and effectively various sources of evidence to enhance the retrieval performance.

3.4.1 Indexing Process

Figure 3-3 [10] outlines the indexing process in Terrier. The terrier is designed to allow many different ways of indexing a corpus of documents. Indexing is a four stage process, and at each stage, plugins can be added to alter the indexing process. This modular architecture allows flexibility in the indexing process at several stages: in the handling of a corpus of documents, in handling and parsing each individual document, in the processing of terms from documents, and in writing the index data structures. In addition, Terrier allows the direct parsing and indexing of compressed collections.

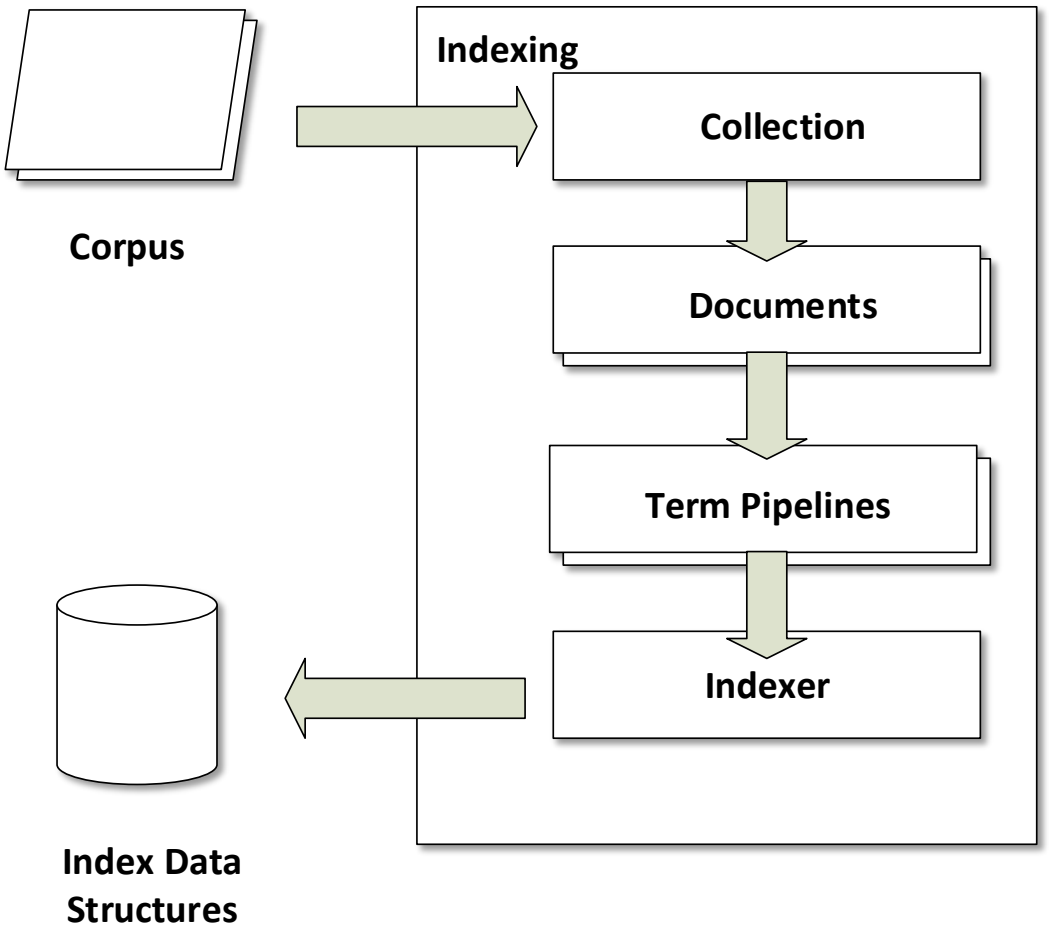


Figure 3-3: Overview of the indexing architecture of Terrier IR Platform

3.4.2 Indexing Structures

A Terrier index consists of 4 main data structures, in addition to some auxiliary files:

- **Lexicon:**

The lexicon stores the term and its term id (a unique number for each term), along with the global statistics of the term (global term frequency and document frequency of the term) and the offsets of the postings list in the Inverted Index.

- **Inverted Index:**

The inverted index stores the postings lists of a term. In particular, for each term, the inverted index stores: the document id of the matching document, and the term frequency of the term in that document. The fields in which the term occurred are encoded using a bit set. If the index has been made with positional information, the postings list will also contain the block ids in the document that the term occurs in. Positional information allows phrasal and proximity search to be performed. The postings list in Terrier is highly compressed. In particular, the document ids are encoded in the inverted index using Gamma encoding, the term frequencies are encoded using Unary encoding, and the Block ids, if present, are encoded using Gamma encoding.

- **Document Index:**

The Document Index stores the document number (an external unique identifier of the document), the document id (internal unique identifier of the document), the length of the document in terms of tokens, and the offset of the document in the Direct Index.

- **Direct Index:**

The Direct Index stores the terms and term frequencies of the terms present in each document. The aim of the Direct Index is to facilitate easy and efficient query expansion. However, the direct index is also extremely useful for applying clustering to a collection of documents. The direct index contents are compressed in an orthogonal way to the inverted index. Term ids are written using Gamma encoding, term frequencies use Unary encoding and the fields in which the terms occur are encoded using a bit set. Block ids, if present, are encoded using Gamma encoding

3.4.3 Retrieval Process

One of the main aims of Terrier is to facilitate research in the field of IR. Figure 3-4 [10] provides an outline of the retrieval process in Terrier. Terrier's retrieval features have been selected in order to be useful for a wide range of IR research. Indeed, Terrier offers great flexibility in choosing a weighting model, as well as altering the score of the retrieved documents. Moreover, Terrier offers an automatic query expansion.

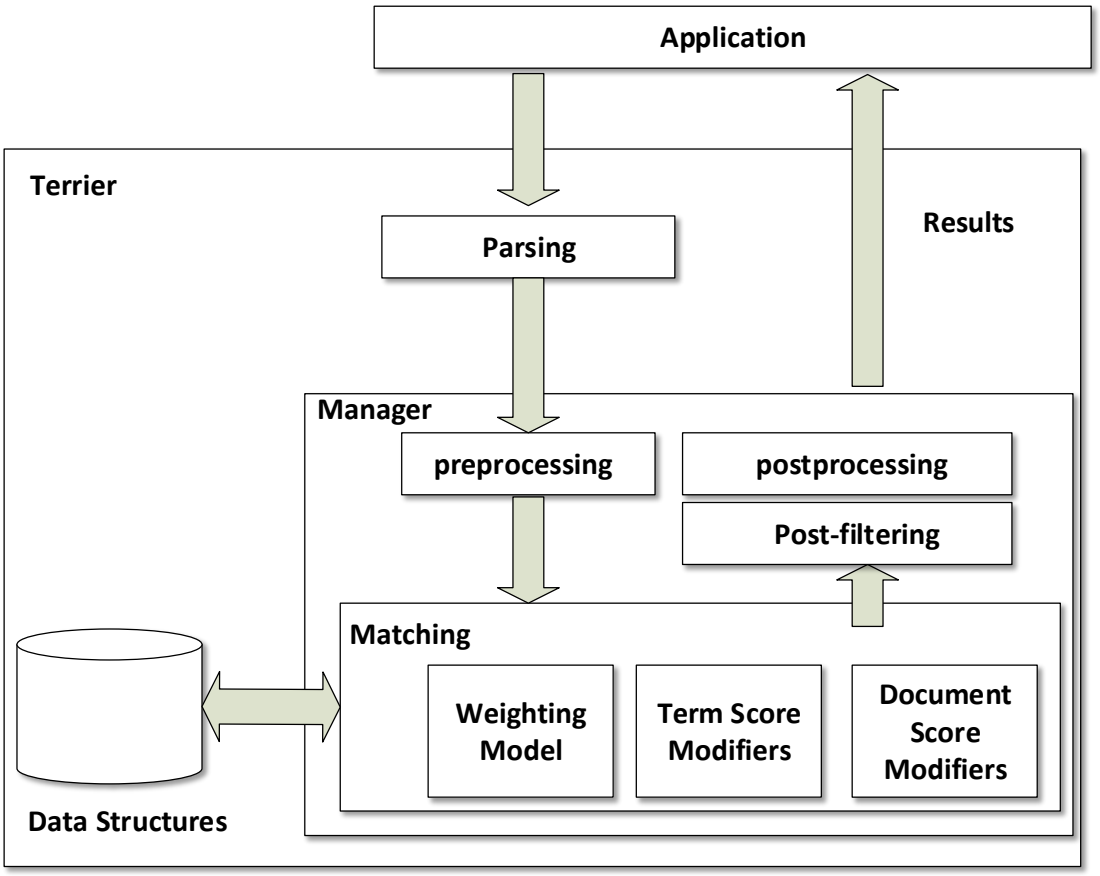


Figure 3-4: Overview of the retrieval architecture of Terrier IR Platform

3.4.4 Terrier Features

Below, you can find a succinct list of features offered by the Terrier IR platform [10].

General

- Indexing support for common desktop file formats, and for commonly used TREC research collections (e.g. TREC CDs 1-5, WT2G, WT10G, GOV, GOV2, Blogs06, Blog08, and ClueWeb09).
- Many documents weighting models, such as many parameters-free Divergence from Randomness weighting models, Okapi BM25 and language modelling.
- Conventional query language supported, including phrases, and terms occurring in tags.
- Handling full-text indexing of large-scale document collections, in a centralized architecture to at least 50 million documents, and using the Hadoop MapReduce distributed indexing scheme for even larger collections.
- Modular and open indexing and querying APIs, to allow easy extension for your own applications and research.
- Active IR research fed into the Open Source platform.
- Open Source (Mozilla Public License).
- Written in cross-platform Java - works on Windows, Mac OS X, Linux and UNIX.
- Large user-base over 7 years of public release.

Indexing

- Out-of-the box indexing of tagged document collections, such as the TREC test collections.
- Out-of-the box indexing for documents of various formats, such as Hypertext Markup Language (HTML), PDF, or Microsoft Word, Excel and PowerPoint files.
- Out-of-the box support for distributed indexing in a Hadoop MapReduce setting.
- Indexing of field information, such as the frequency of a term in a title or H1 HTML tag.
- Indexing of position information on a word, or a block (e.g. a window of terms within a distance) level.
- Support for various encodings of documents (UTF), to facilitate multi-lingual retrieval.
- Support for changing the tokenization being used.

- Indexing support for query-biased summarization.
- Support for fetching files to index by Hypertext Transfer Protocol (HTTP), allowing intranets to be easily searched.
- Highly compressed index disk data structures.
- Highly compressed direct file for efficient query expansion.
- Alternative faster single-pass and MapReduce based indexing.
- Various stemming techniques supported, including the Snowball stemmer for European languages.

Retrieval

- Provides desktop, command-line and Web based querying interfaces.
- Provides standard querying facilities, as well as Query Expansion (pseudo-relevance feedback).
- Can be applied in interactive applications, such as the included Desktop Search, or in a batch setting for research and experimentation.
- Provides many standard document weighting models, including up to 126 Divergence from Randomness (DFR) document ranking models, and other models such as Okapi BM25, language modelling and TF-IDF. Two new 2nd generation DFR weighting model, JsKLs and XSqrA_M, are also included, which provide robust performance on a range of test collections without the need for any parameter tuning or training.
- Advanced query language that supports synonyms, +/- operators, phrase and proximity search, and fields.
- Provides a number of parameter-free DFR term weighting models for automatic query expansion, in addition to Rocchio's query expansion.
- Flexible processing of terms through a pipeline of components, such as stop-word removers and stemmers.

Experimentation

- Handles all currently available TREC test collections
- Easily scriptable to evaluate many parameter settings, or many weighting models in batch form.

- Built-in evaluation tools for use with TREC ad-hoc and known-item search retrieval results, to produce various Precision and Recall measures

3.4.5 Component Description

In this section we introduce a brief description of Terrier's components.

Indexing

In Table 3-8 we introduce a brief description of the indexing process.

Table 3-8: Indexing Component Description

Name	Description
Collection	This component encapsulates the most fundamental concept to indexing with Terrier - a Collection i.e. a set of documents.
Document	This component encapsulates the concept of a document. It is essentially an Iterator over terms in a document.
Tokeniser	Used by Document objects to break sequences of text (e.g. sentences) into a stream of words to index.
TermPipeline	Models the concept of a component in a pipeline of term processors. Classes that implement this interface could be stemming algorithms, stopwords removers, or acronym expansion just to mention few examples.
Indexer	The component responsible for managing the indexing process. It instantiates TermPipelines and Builders.
Builders	Builders are responsible for writing an index to disk.

Data Structures

In Table 3-9 [10] we introduce a brief description of the Data structures used in Terrier IR platform.

Table 3-9: Data structures Component Description

Name	Description
BitFile	A highly compressed I/O layer using gamma and unary encodings.
Direct Index	The direct index stores the identifiers of terms that appear in each document and the corresponding frequencies. It is used for automatic query expansion, but can also be used for user profiling activities.
Document Index	The document index stores information about each document for example the document length and identifier, and a pointer to the corresponding entry in the direct index.
Inverted Index	The inverted index stores the posting lists, i.e. the identifiers of the documents and their corresponding term frequencies. Moreover it is capable of storing the position of terms within a document.
Lexicon	The lexicon stores the collection vocabulary and the corresponding document and term frequencies.
Meta Index	The Meta Index stores additional (meta) information about each document, for example its unique textual identifier (docno) or URL.

Retrieval

In Table 3-10 [10] we introduce a brief description of the retrieval process.

Table 3-10: Retrieval Component Description

Name	Description
Manager	This component is responsible for handling/coordinating the main high-level operations of a query. These are: • Pre Processing (Term Pipeline, Control finding, term aggregation) • Matching

	• Post-processing • Post-filtering
Matching	The matching component is responsible for determining which documents match a specific query and for scoring documents with respect to a query.
Query	The query component models a query that consists of sub-queries and query terms.
Weighting Model	The Weighting model represents the retrieval model that is used to weight the terms of a document.
Document Score Modifiers	Responsible for query dependent modification document scores.

Applications

In Table 3-11 [10] we introduce a brief description of the indexing process.

Table 3-11: Applications Component Description

Name	Description
Trec Terrier	An application that enables indexing and querying of TREC collections.
Desktop Terrier	An application that allows for indexing and retrieval of local user content.
HTTP Terrier	An application that allows for retrieval of documents from a browser.

Chapter 4 Methodology

4.1 Arabic Light Stemming Enhancement Process

The proposed system will be implemented as a standalone application, many steps of preprocessing will be modified and enhanced to construct a fundamental toolkit to help other interested researchers modify and improve Arabic stemming in the future.

This standalone application contains normalizing, tokenization, and stop-words removal sub processes so it can be introduced as a basic toolkit for Arabic stemming.

The proposed system will be integrated into the Terrier IR platform with the addition of existing processes, for example tokenization process that is available in Terrier IR will be modified to match Arabic language needs. Figure 4-1 summarize the approach that this thesis follows to implement a system for improving Arabic stemming.

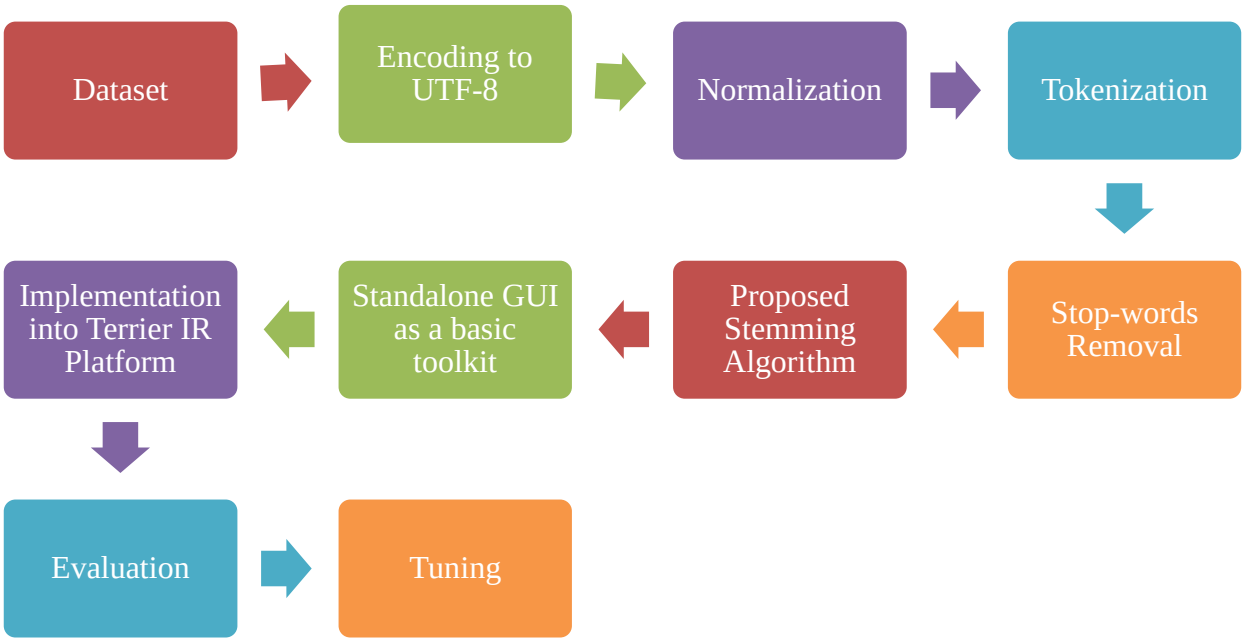


Figure 4-1: Proposed system overview

4.2 Preprocessing

The preprocessing of the Arabic text is a challenging and crucial stage. It may impact positively or negatively on the accuracy of any IR system, and therefore the improvement of the preprocessing step will lead by necessity to the improvement of any IR system very greatly. Preprocessing contains many sub processes and each one has a specific job to prepare the data to be in optimal shape so the result can be improved, the proposed system will focus on the following Preprocessing steps:

- Encoding
- Normalization
- Tokenization
- Stop-word removal

4.2.1 Encoding

This step deals with unifying the encoding to avoid character appearance issue and make the datasets compatible with the suggested algorithm and to standardize any dataset for future use.

The converting from various encoding system like Windows Arabic encoding (CP1256) to the Unicode UTF-8 system is implemented, any dataset will first be analyzed to detect the type of the encoding and then convert it to UTF-8 encoding. In the Unicode standard, version 6.3, Arabic range is from 0600 to 06FF in decimal format, Figure 4-2 [44] show this range in the proposed system.

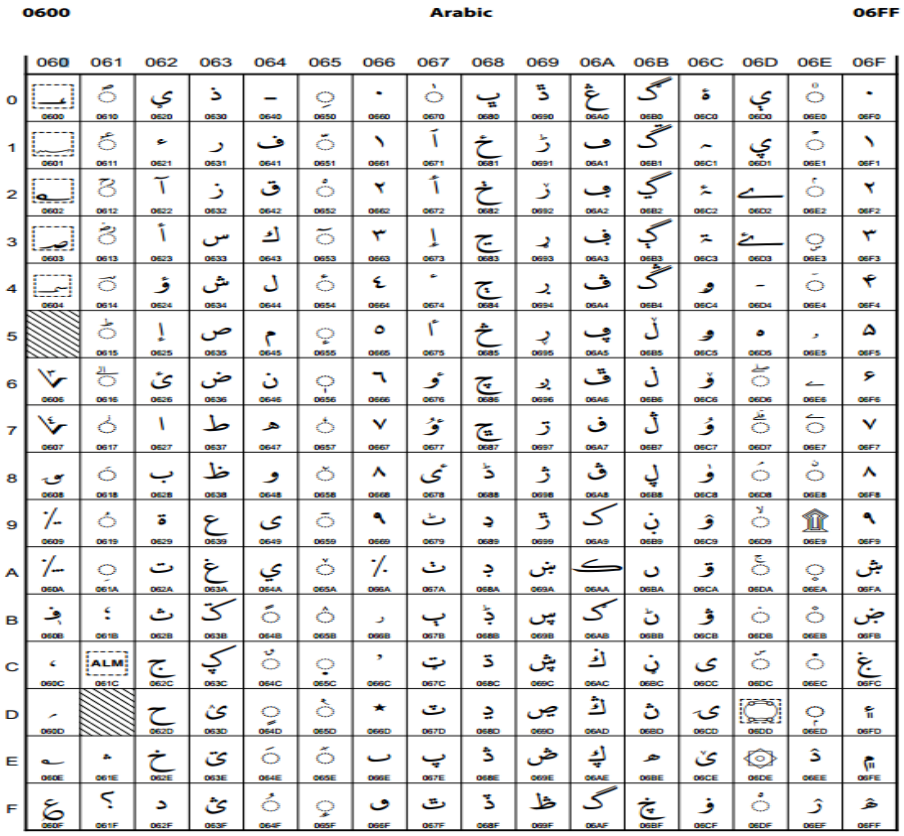


Figure 4-2: Arabic Unicode range from 0600 to 06FF

4.2.2 Tokenization

Tokenization is a necessary step in natural language processing, data mining and IR systems. The function of a tokenizer is to break down a stream of text into segments or words so that they can be used into by the algorithm.

The tokenizer is responsible in defining boundaries of a word, it is based mainly on the white spaces and punctuation marks as delimiters between words or major segments as Figure 4-3.

Figure 4-3: Tokenization Process

On the Proposed system, the tokenization process is based on dividing a string of words into token using white spaces that match the regular expression “\\s”.

4.2.3 Normalization

The normalizing process depends on the Unicode number for every character to be used as the unique identifier for this character, normalize dataset and removing of extra character is very important, normalization process that the proposed system follows is shown in Algorithm 4.1.

Algorithm 4.1: Proposed Normalizing Algorithm

Purpose: Normalize Arabic Words

Input:

- Dataset

Output: Normalized Dataset

Procedure:

1. Remove diacritics along with the short vowels, shadda and sikkun.
2. Remove all punctuation
3. Use of regular expression "\\p {Punct}"
4. Remove numbers
5. Remove non letters like special characters
6. Replace إ ,أ, and آ with ا
7. Replace final ى with ي
8. Replace final ه with ة
9. Replacing the two final letters ى and ء with ئ
10. Replacing the two final letters ي and ء with ئ
11. More normalizing steps with preference to Arabic regions to get better standardized Arabic text.

Figure 4-4: Pseudo Code of Proposed Normalizing Algorithm

The proposed system also adds advance removing features based on regular expression, this should be considered as a trend or a standard to be used as a general

normalization step for future research as regular expression has a powerful capability. For example, one of the normalizing feature that the system adds to the literature is the use of regular expression "\\p {Punct}" which used for advance removing of punctuation marks.

4.2.4 Stop-word Removal

The proposed system improves monolingual Arabic searching which may be affected by the use of stop-word removal, some of the stop-words will be removed during normalizing step, and others will give meaning to the terms.

The list of Arabic Stop-words [46] will be used and updated with the suggested enhancements like the different choice of words in Arabic World, some types of stop-words are adverbs, conditional pronouns, prepositions, pronouns, transformers (verbs, letters) and referral names.

As far as have been researched in Arabic Stop-word removal techniques no one takes care of different choices of stop-words in different Arab countries due to the differences in spoken Arabic in these countries.

In this system the Stop-word removal process combined and merged three stop-words lists to introduce a huge list of stop-words which affect the retrieval process.

- Jacques Savoy Dataset [47] contains 162 Arabic words in UTF-8
- Motaz Saad list [48] contains 940 Arabic words.
- Arabic Stop-words [46] website list contains 13015 Arabic words in UTF-8, and offer a feature that allow developer to generate stop-words based on specific options that can be added to the system.

This improves the result by reducing the size of the dataset and remove ambiguous words to be processed, the combination of the three data set cover wide range of ambiguous words.

As stop-word removal affects the recall of the IR systems, we introduce two stop-word list.

The first list is the merged one and called intensive list, which contains as much as possible of stop-words to help researchers investigate the intensive removal of stop-words and contains 13975 words, a sample of this list is shown in Table 4-2.

Table 4-2 Intensive Stop-word list sample

عنه	أما	ومنذ	عليها	وكان	من	ان
حول	اما	الذي	إن	تلك	ومن	أن
دون	إما	والذي	وعلى	وتلك	حتى	إن
مع	حين	اللذي	لكن	كذلك	وحتى	بعد
لكنه	ومن	اللذان	عن	وكذلك	وهو	ضد
ولكن	لا	اللتي	وعن	التي	يكون	يلي
له	ليسب	اللتان	مساء	والتي	به	الى
هذا	وكانت	اللذين	ليس	بين	وليس	إلى
وهذا	أي	اللاواتي	وليس	وبين	أحد	في
هذه	ما	اللائي	منذ	فيها	على	وفي

The second list is called light stop-word list and contains 119 stop-words and based on Jacques Savoy dataset [47], this help researchers to compare it to the intensive list and to see how the recall will be affected by the stop-word list removal process, a sample of light stop-word list is shown in Table 4-3.

Table 4-3 Light Stop-word list sample

إلي	وأن	أنها	كما	أي	ثم	من
على	وإن	إنها	فما	أى	او	ومن

عليها	التى	انه	عن	لا	أو	منها
عليه	التي	أنه	مع	ولا	ب	منه
اما	الذى	إنه	اذا	الا	بها	في
أما	الذي	بان	إذا	ألا	به	وفي
إما	الذين	بأن	ان	إلا	ا	فيها
ايضا	الى	فان	أن	لكن	أ	فيه
أيضا	الي	فأن	إن	ما	اى	و
كل	إلى	وان	انها	وما	اي	ف

4.3 Arabized Words

Arabized words or syndication words are a foreign word that found their way into the Arabic language, they came from many different languages around the word, and most Arabic stemmers face problems when they try to process this kind of words, so in the proposed system we skip these types of words, these words were collected from different sources [49], to become 100 words as shown in Table 4-4.

Table 4-4 Arabized words list

هيراغانا	كِشْك	قَابِيْل	فَارِس	شَطْرَنْج	رَوْب	جيب	بلغارية	إيرَان	آب
هَمْبَرْكَر	كِيمْيَاء	قُلْنُسُوَة	فَبْرَايِر	شَنْطَة	زَامْبِيَا	جَاجِيْك	بَرْبَر	إيْدِيُولُوجِيَا	آكرن
هَنْدَسَة	مَارِس	كَمُّوْن	فَرَنْج	شِيش طَاوُوق	سَالِي	جَالُوت	بَطَاطَا	إِنْجِلِيْزِيَّة	آنْسُوْن
هَوْنَدَا	مَايُو	كَنَبَة	فَرَنْسَا	صَالَوْن	سَانْدوِيش	جَفَّتَ	بَغْدَاد	اوتوكاد	أفْرَنْج
هَيْتَ	مَغُول	كَنْتُور	فَهْرَسَ	صُوفَا	سَبْتَمْبَر	جُغْرَافِيَة	بَنْدَر	اِبْرَاهِيْم	أَبْرِيْل
وِيْكِيْبِيْدِيَا	مَكَّرَوْنَة	كَنْدُوْرَة	فُرْس	طَاوِلَة	سَجَنْجَل	خَان	بَنْشَرَ	اِسْتِكَان	أَطْرَقْچِي

يَنَايِر	مُهَنْدِس	كَهْرَمَان	فُسْتَان	طَرْبُوْش	سَلَطَة	خُوَان	بُسْتَان	براغماتي	أَمُوْنِيَا
يَنْسُوْن	مُوْسَى	كُشْك	فُنْدُق	عَرَّاب	سُلَاف	دِيسَمْبَر	تَمُّوْز	براغماتية	أُرْثُوْذُكْس
يَوْنِيَوْ	نُوفَمْبَر	كُورَة	فِهْرِس	غَوْغَلَ	سِرْوَال	دِيمَاغُوجِيَّة	تُوْنِس	بعلبك	أُغَسْطَس
يُولِيُو	نِسْرِين	كُورِيَّة	فِيزْيَاء	غَوْغَلَة	شَرْشَف	رَاتِنْج	تِلْفَاز	بكتيريا	أُكْتُوبَر

These words will affect stemming and lead to ambiguous results in retrieval so in the proposed system we exclude these words from applying stemming algorithm on them, many of these words may match with light stemming rules that originally deals with pure Arabic words so their meaning and structure will be changed.

We normalized these words using the proposed system to be compared to exclude them from the dataset, since the dataset dealing with modern Arabic the change to find these words is high, in Table 4-5 we show Arabized word after normalizing.

Table 4-5 Arabized words list after normalizing

هيراغانا	كشك	قابيل	فارس	شطرنج	روب	جيب	بلغاريه	ايران	اب
همبركر	كيمياء	قلنسوه	فبراير	شنطه	زامبيا	جاجيك	بربر	ايديولوجيا	اكرن
هندسه	مارس	كمون	فرنج	شيش طاووق	سالي	جالوت	بطاطا	انجليزيه	انسون
هوندا	مايو	كنبه	فرنسا	صالون	ساندويش	جفت	بغداد	اوتوكاد	افرنج
هيت	مغول	كنتور	فهرس	صوفا	سبتمبر	جغرافيه	بندر	ابراهيم	ابريل
ويكيبيديا	مكرونه	كندوره	فرس	طاوله	سجنجل	خان	بنشر	استكان	اطرقچي
يناير	مهندس	كهرمان	فستان	طربوش	سلطه	خوان	بستان	براغماتي	امونيا
ينسون	موسي	كشك	فندق	عراب	سلاف	ديسمبر	تموز	براغماتيه	ارثوذكس
يونيو	نوفمبر	كوره	فهرس	غوغل	سروال	ديماغوجيه	تونس	بعلبك	اغسطس
يوليو	نسرين	كوريه	فيزياء	غوغله	شرشف	راتنج	تلفاز	بكتيريا	اكتوبر

We normalized these words for better matching with other tokens from dataset to exclude them from the stemming process.

4.4 Standalone Arabic Light Stemming Toolkit

In Figure 4-5 we show the structure of the proposed toolkit that consist of many components that we describe in the previous section to create a separate toolkit for Arabic stemming algorithms with enhanced GUI features.

Figure 4-5: Proposed Toolkit Overview

Figure 4-6, the toolkit allows researchers to choose different Arabic stemming algorithm or add more algorithm if they want as this toolkit will be provided as an open source project to allow other researchers to improve it

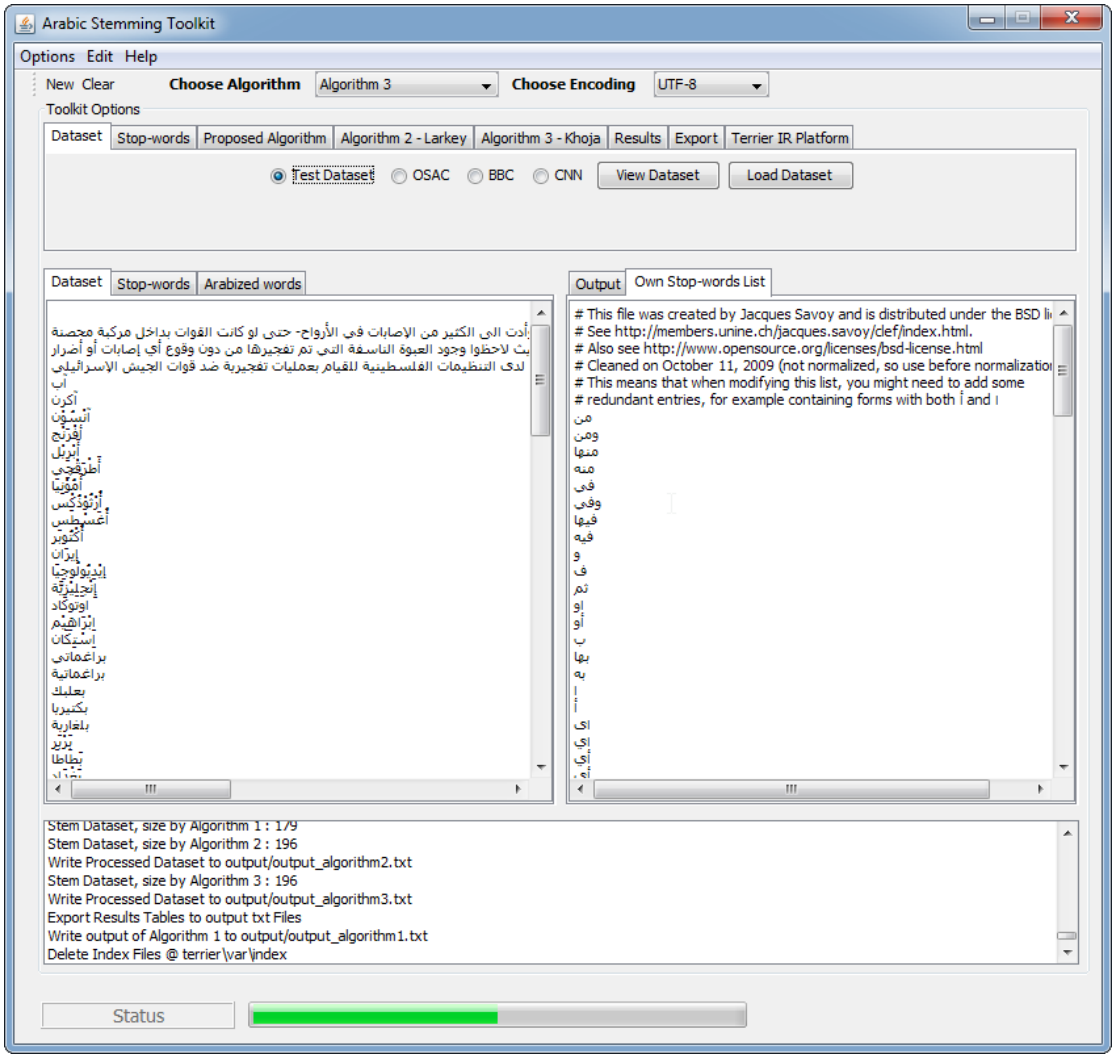


Figure 4-6: Proposed System Toolkit

Figure 4-7, the researchers can choose from different types of datasets (Test dataset, Sample of OSAC, Sample of BBC and Sample of CNN) and apply preprocessing on them alongside with stemming also they can view the dataset, the proposed toolkit also show statistics of dataset as number of words and the status of the toolkit, also the same apply for stop-word.

The toolkit allows to show either intensive stop-word list or light stop-word list as described earlier, this toolkit may be the first step towards standalone preprocessing toolkit with GUI enhancement for the Arabic language to help to create a standard approach to Arabic stemming and preprocessing.

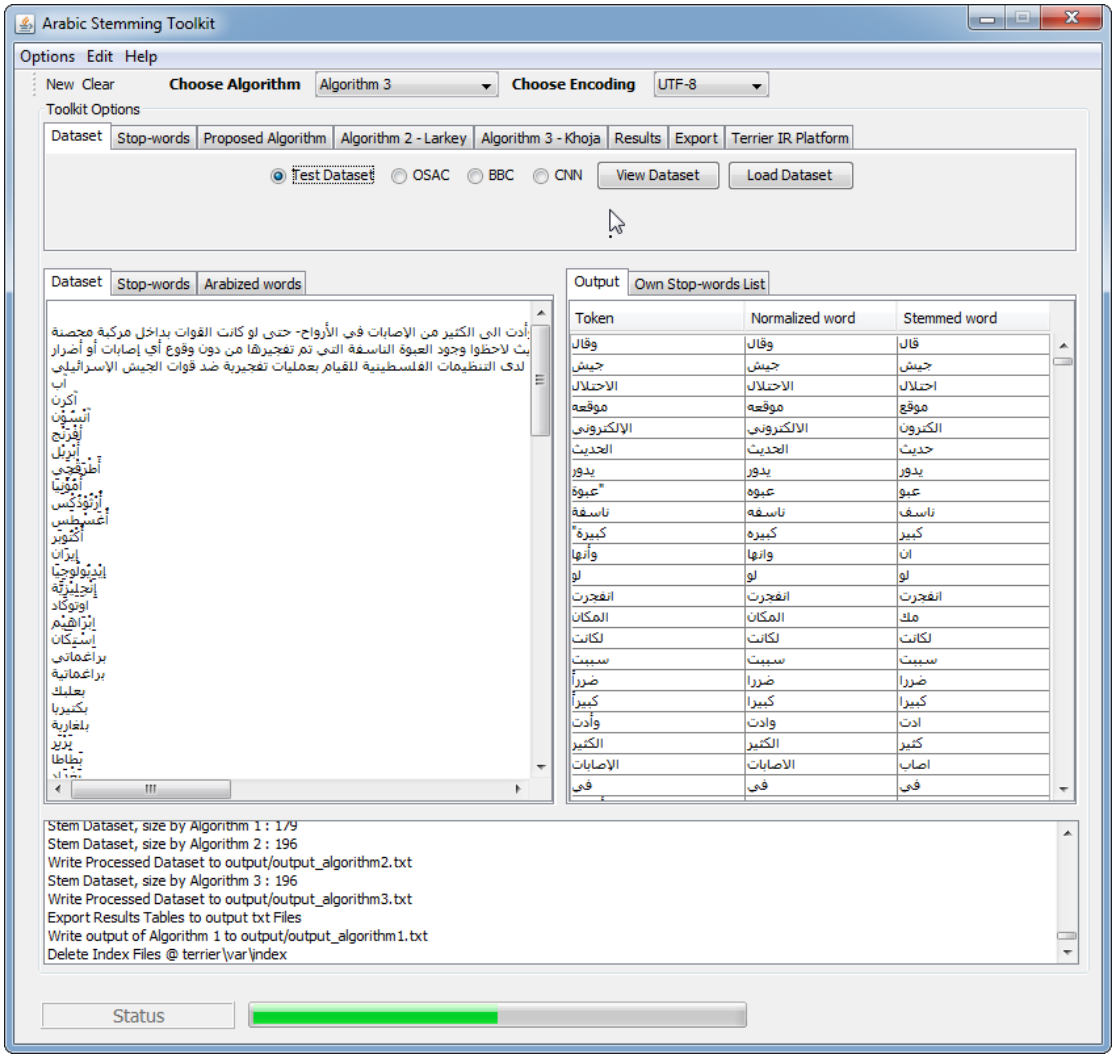


Figure 4-7: Dataset Selection and Proposed Algorithm output

After loading both dataset and stop-word list, the system preprocessing steps is used as shown in Figure 4-8.

Enhance normalizing is the first to be applied and show better normalizing output than other normalizing algorithm available online for better removal of punctuation marks that still included with tokens and words in some AI applications which may affect the accuracy of these applications.

After that stop-word removal is applied using either intensive stop-word list or light stop-word list both have different impact on retrieval results that is why the system included both to generate two different output one with light stop-word removal and the other with intensive stop-word removal then see the impact of these both output using the attached Terrier IR platform.

Each word tokenized by the tokenizer and then passed to be stemmed, the output table shows the transition that applied to each word, from normalization to stemming to investigate the differences in each process to get a better understanding of preprocessing steps.

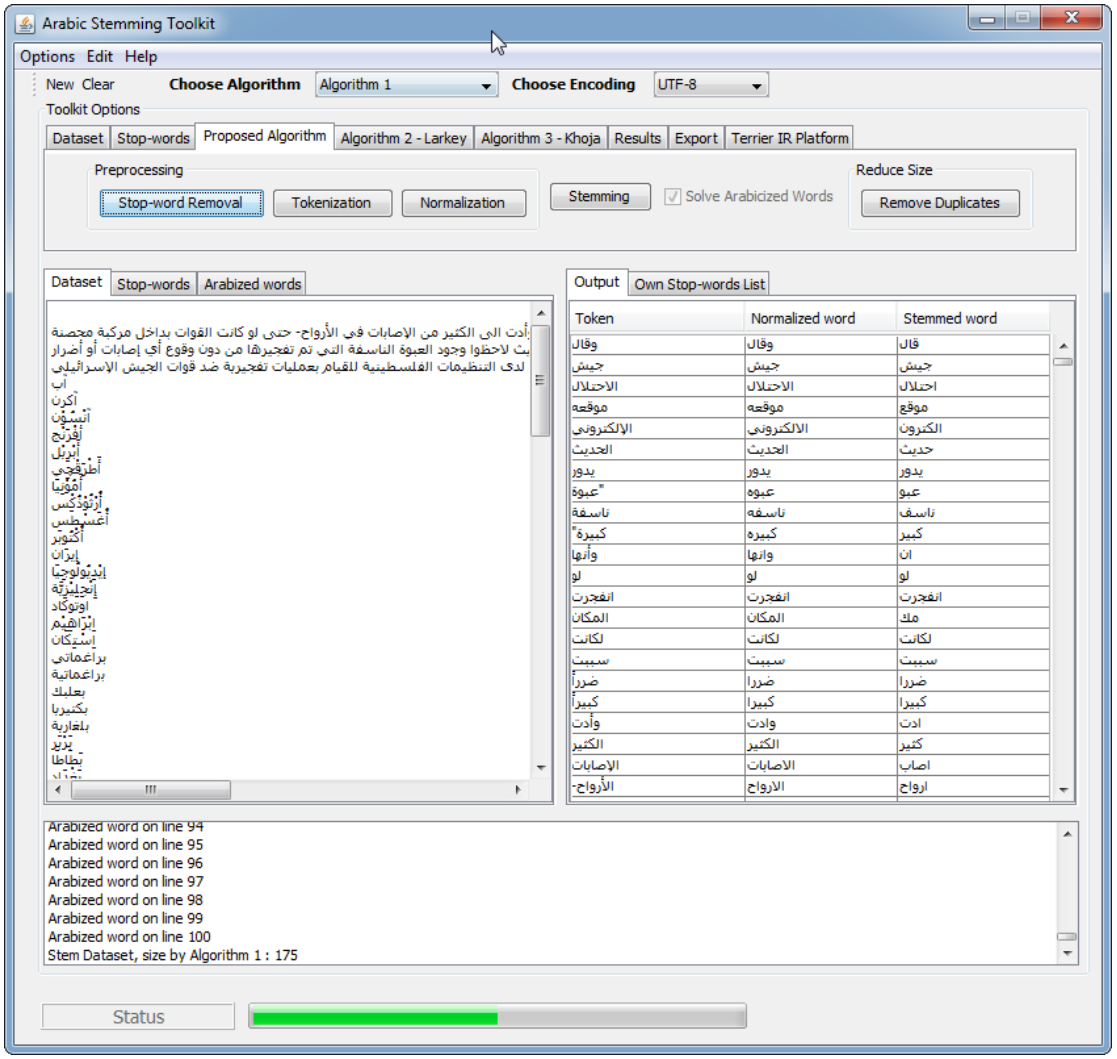
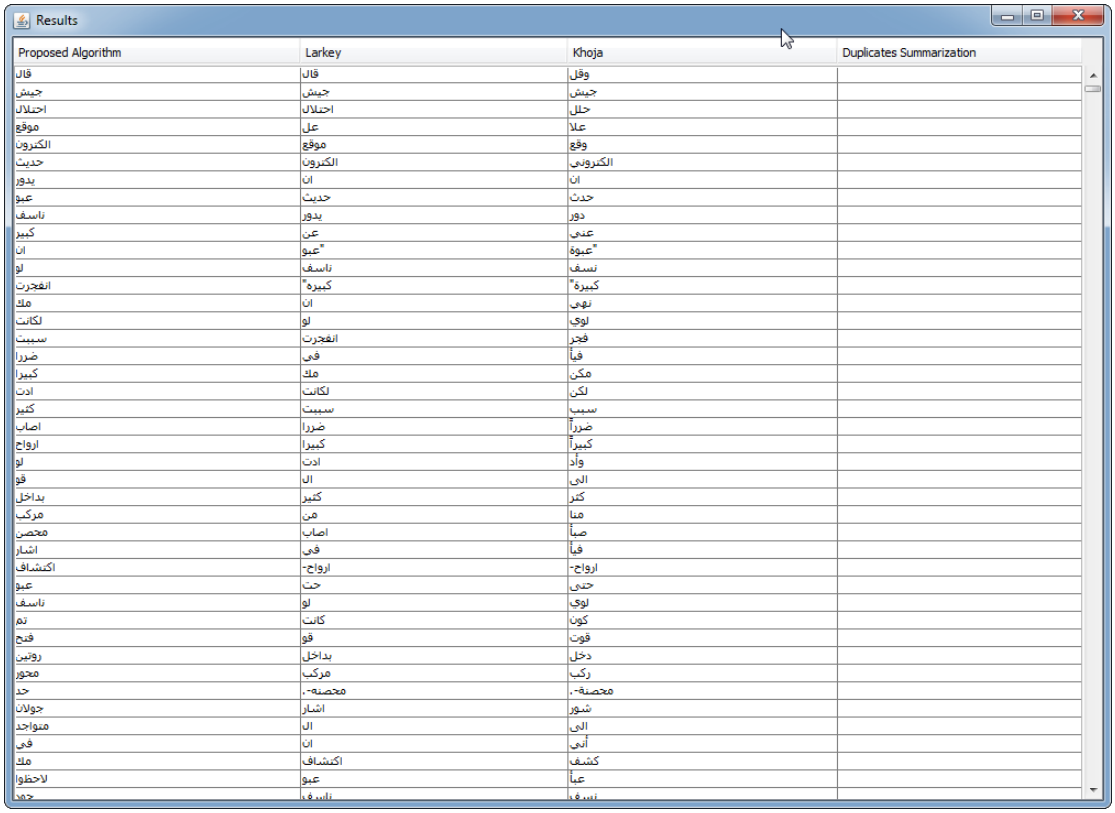


Figure 4-8: Preprocessing and Stemming

Larkey algorithm and Khoja algorithm are comparable to the proposed algorithm and introduced the importance of adding more algorithms to the toolkit to establish a standard approach in the Arabic stemming field.

The toolkit runs Larkey algorithm and Khoja algorithm with the same data set without enhancing normalization or any other improved features as in the proposed system, the difference in output length come due to the use of different stop-word removal,

especially in Khoja algorithm, the output of the three algorithms are shown in separated output table as shown in Figure 4-9.



Proposed Algorithm	Larkey	Khoja	Duplicates Summarization
قال	قال	وقل	
جيش	جيش	جيش	
احتلال	احتلال	حلل	
موقع	عل	علا	
الكنرون	موقع	وقع	
حديث	الكنرون	الكنروني	
يدور	ان	ان	
عبؤ	حديث	حدث	
تاسف	يدور	دور	
كبير	عن	عني	
ان	"عبو	"عبوة	
لو	تاسف	نسف	
انفجرت	كبيره"	كبيرة"	
مك	ان	نهي	
لكانت	لو	لوي	
سببت	انفجرت	فجر	
ضررا	في	فياً	
كبيرا	مك	مكن	
ادت	لكانت	لكن	
كثير	سببت	سبب	
اصاب	ضررا	ضرراً	
ارواح	كبيرا	كبيراً	
لو	ادت	وأد	
قو	ال	الى	
بداخل	كثير	كثر	
مركب	من	منا	
محصن	اصاب	صباً	
اشار	في	فياً	
اكتشاف	ارواح-	ارواح-	
عبؤ	حت	حتى	
تاسف	لو	لوي	
تم	كانت	كون	
فتح	قو	قوت	
روتين	بداخل	دخل	
محور	مركب	ركب	
حد	محصنة-.	محصنة-.	
جولان	اشار	شور	
متواجد	ال	الى	
في	ان	أني	
مك	اكتشاف	كشف	
لاحظوا	عبو	عبأ	
حمد	تاسف	نسف	

Figure 4-9: Output Comparison

The output can be exported to different types of files to be studied later, which help to inspire other researchers to add more output format for different other available toolkit that need preprocessing.

The toolkit can be used as basic preprocessing toolkit for Arabic language and use the output with any TC or IR applications. The toolkit also, allow researcher to choose the appropriate stemming algorithm, also researchers may be interesting in developing root-based stemming and may use the normalization or tokenization alongside with stop-word removal.

Also in Figure 4-10 Terrier IR Platform is integrated in the system and can import the output of the proposed algorithm and the two other algorithms, which allows to clear previous index files that was generated by the Terrier IR platform when reading the imported dataset, this allows rapid testing and saves a lot of time for researchers.

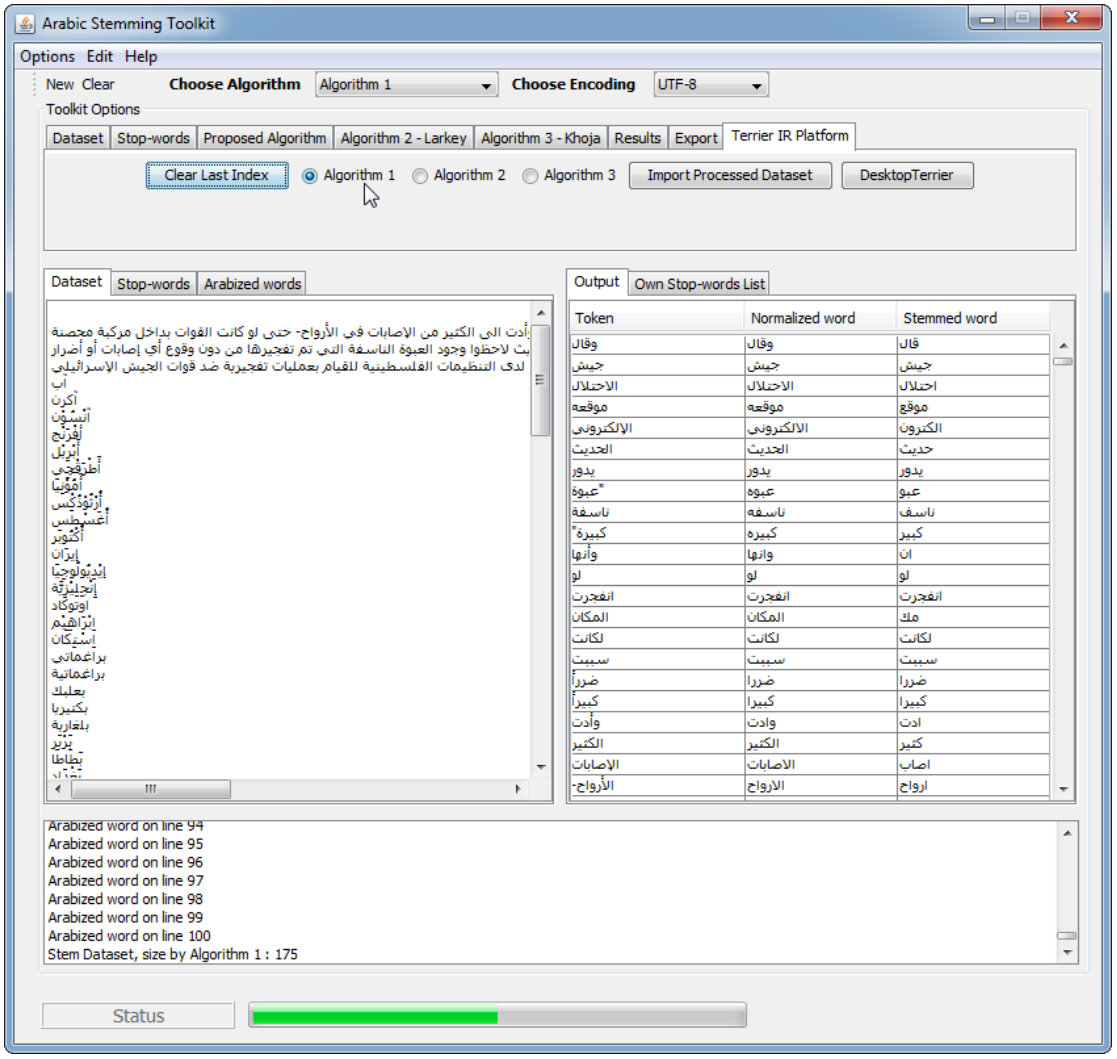


Figure 4-10: Terrier IR Integration

We improved the suffix list that the enhanced algorithm use, which is based on Larkey work, as this suffix is widely used, so it has been added to the list of the suffixes for better stemming of Arabic words, so the sequence of "ت", "ن" and "ا" characters are added to the enhanced suffixes list for best results, so a good stemming algorithm should deal with all possible occurrences of character sequence for both suffixes and prefixes.

4.5 Terrier IR

The output of the proposed system is fed into the Terrier IR, which is attached to the proposed system as shown in

Figure 4-11 to form a complete toolkit for Arabic IR system. This allows the availability of complete IR system to be studied and modified as researchers in IR system may studied the impact of every stem in the retrieval model start from tokenization alongside with normalization, stop-word removal, indexing to the scoring system.

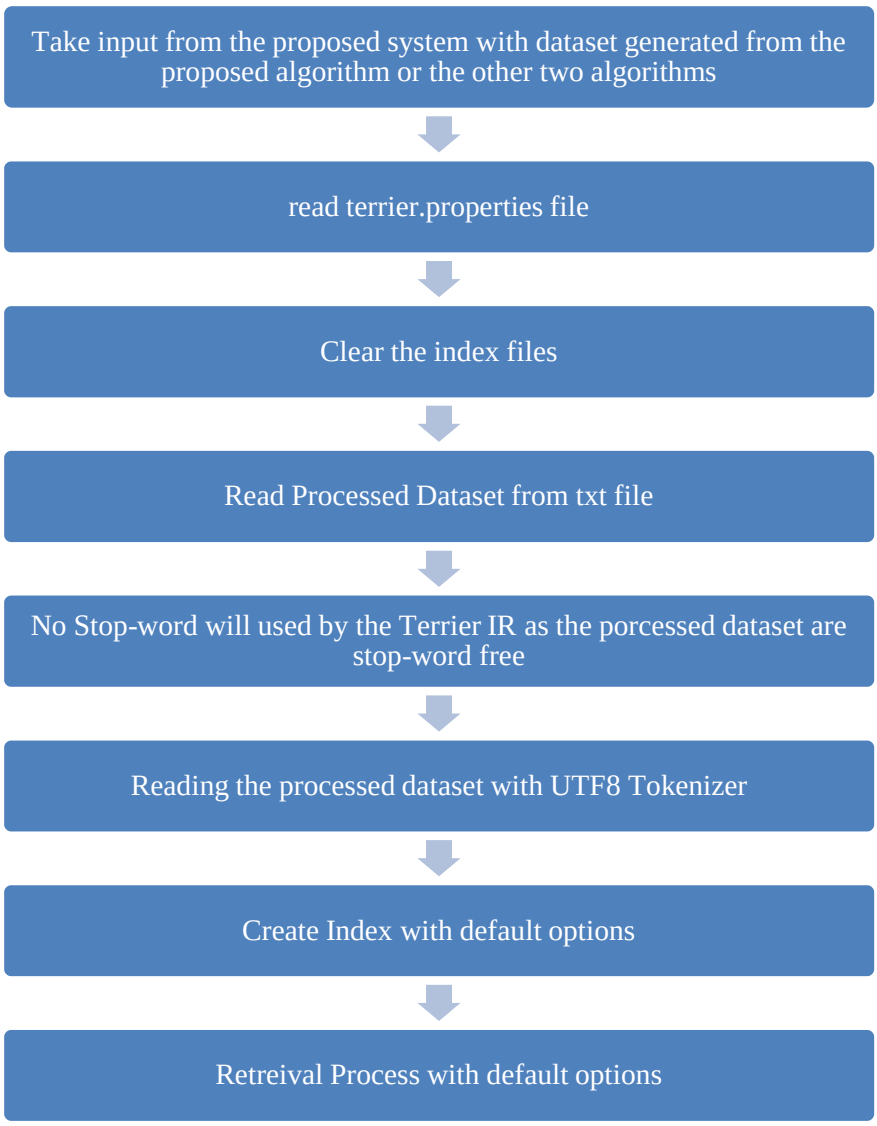


Figure 4-11: Terrier IR Platform Arabic Retrieval Integration

A Desktop instance used in the proposed system to call Desktop-Terrier application as shown in Figure 4-12, this application uses the output of the toolkit after stemming and then, using queries we can retrieve desired information.

The Terrier desktop search can read many type formats and tokenize them, but as informed before there no Arabic stemming in the Terrier IR platform, so the proposed system will offer the enhanced Arabic light stemming as stemming algorithm for the Terrier IR platform since light stemming is more accurate than other types

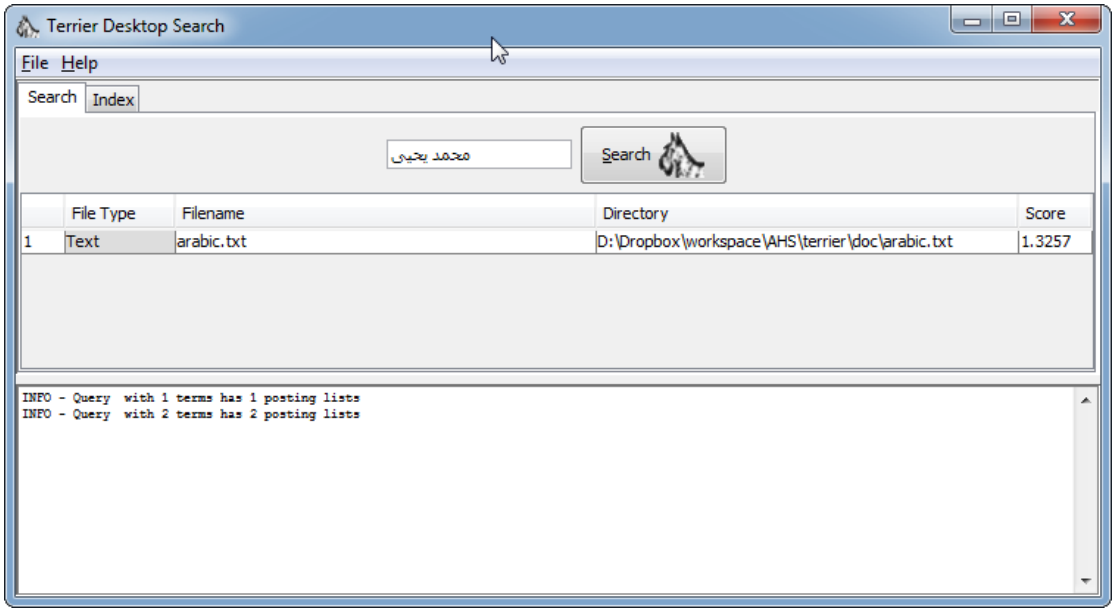


Figure 4-12: Terrier Desktop Search Attached to the toolkit

The Terrier IR platform shows many statistics such as posting lists and score alongside the location of the document that contain the term match by the query. In Figure 4-13 we see the indexing process done by the Terrier IR platform and other statistics.

The Terrier IR platform will be used to test the new enhanced Arabic light stemming algorithm and to help other researchers interested in Arabic IR by adding Arabic stemming to this completed IR platform

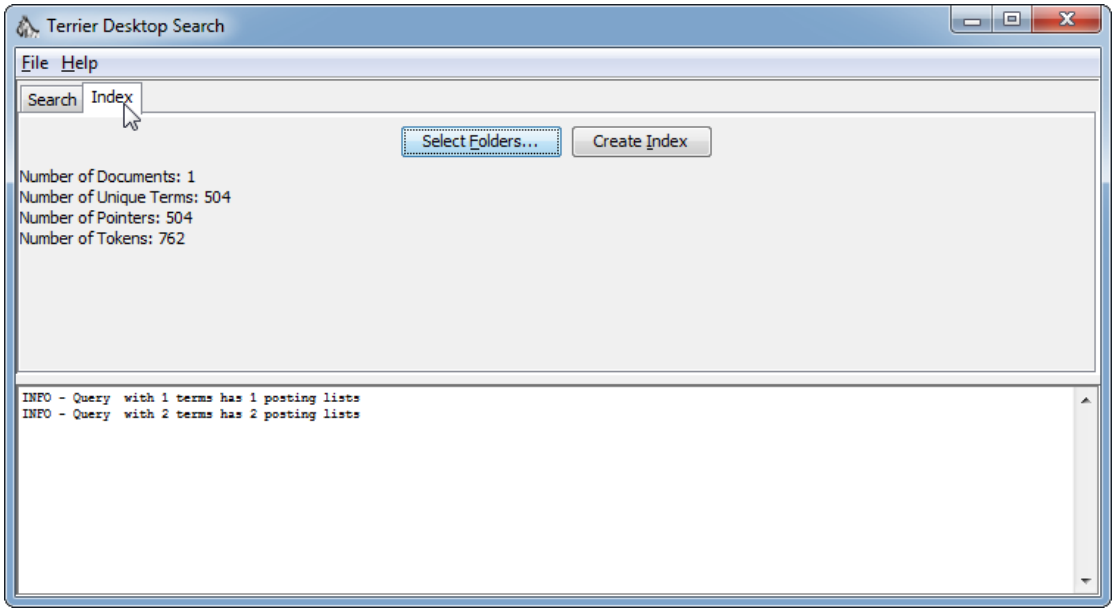


Figure 4-13: Terrier Desktop indexing

We modified the following settings in Terrier source code, particularly in a properties file that contains all the parameters for this platform to be able to handle Arabic language and proposed stop-words list and successfully tokenize Arabic text with UTFTokeniser Class and index them in the Terrier IR platform by using the following properties:

- tokeniser=UTFTokeniser
- trec.encoding=UTF-8

Also Added proposed Arabic stop-word lists to the Terrier IR platform, by setting the properties:

- stopwords.filename=intensive-arabic-stop-words.txt
- stopwords.filename=light-arabic-stop-words.txt

For the Final integration of the Terrier IR platform and proposed toolkit the following properties are set:

- terrier.home=D:\\Dropbox\\workspace\\AHS\\terrier
- terrier.share=\\share
- terrier.etc=\\etc
- terrier.doc=\\doc
- terrier.index.path=\\index
- termpipelines=

“terrier.home” properties tell the system where to find the default location of the Terrier IR platform, “terrier.share” tells the system the location of stop-word lists as

shown above, “terrier.etc” tells the system where to find terrier.properties file that contains all the options and parameter of the Terrier IR Platform, as for “terrier.doc” we inform the system where to look for the dataset, this could be any file type that the Terrier IR platform supports not only txt file, for example DOC, XLS, PDF and others could be used beside txt files.

“termpipelines” is the most important properties which tells the system to use stop-word removal or not, or to use stemming or not, the stop-word removal and stemming will be done by the proposed system and the output of the processed dataset will be indexed by the Terrier at location specified by “terrier.index.path” this will add Arabic stemming and preprocessing toolkit as extension to Terrier IR platform that now support Arabic language and could be used by Arabic IR researchers for developing IR component that could be added as extensions just as the proposed system.

4.6 Evaluation

The terrier IR platform uses many weighting models for Term weighting. Term weighting helps us to locate important terms in a document collection for ranking purposes. There are several term weighting schemes, we will use Terrier IR default weighting which is Term Frequency (TF) and it is variants Inverse Document Frequency (IDF) and Term Frequency-Inverse Document Frequency (TF-IDF).

4.6.1 Term Frequency

Term frequency TF (*t, d*) is the number that the term *t* occurs in the document *d* .The TF measures the importance of term t_i within the particular document d_j can be calculated by equation:

$$TF_{i,j} = \frac{n_{i,j}}{\sum n_{k,j}} \quad \text{..... (4.1)}$$

Where

$n_{i,j}$: The number of occurrences of the considered term (t_i) in the document d_j.

$\sum n_{k,j}$: Sum of number of occurrences of all terms in document d_j.

4.6.2 Inverse Document Frequency

The inverse document frequency (IDF) is one of the most widely used term weighting schemes for estimating the specificity of term in a document collection. It is based on the idea that if a term appears in only a few documents in the collection, then such a term is expected to be a good discriminator of these documents. The IDF weight of a term t can be calculated from document frequency using the formula:

$$IDF_t = \log(\frac{N}{n}) \quad \text{..... (4.2)}$$

Where

N: number of documents.

n: number of documents with term t.

The IDF of a term is low if it occurs in many documents and high if the term occurs in only a few documents.

4.6.3 Term Frequency-Inverse Document Frequency

We will focus on Term Frequency and Inverse Document Frequency (TF-IDF), because it is a popular method of ranking documents in the IR community. TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. The TF-IDF calculated by using the formula:

$$\text{TF-IDF} = TF_{i,j} . \ IDF_t \quad \text{..... (4.3)}$$

Terrier IR support many weighting and scoring models specially for TREC likewise dataset and give many details statistics and results when applying and running Terrier IR platform on these TREC like dataset.

Arabic Language needs an updated TREC like Dataset that will help researcher to improve and investigate Arabic IR, these TREC like dataset is available publicly, but not free. On the other hand, we use datasets, as mentioned earlier that is mainly for TC applications and these datasets does not contain any queries that belong to the same dataset to be tested and used alongside the same data set, we test a sample of

each dataset and extract queries randomly from these samples and used with TF-IDF scoring and evaluation model.

4.7 Summary

In this section we describe the pseudo code about the whole enhancement process that this thesis is following and suggesting for improving Arabic light steaming and contribute to open source projects, especially in IR field, the pseudo code is described in Figure 4-14.

Algorithm 4.2: Enhance Arabic IR Algorithm

Purpose: Enhance Arabic light stemming and integrate with IR applications

Input:

- Dataset
- Light stop-word list or Intensive stop-word list
- Arabized-word list

Output: Results Documents scored with TF-IDF

Procedure:

1. Read input files
2. Apply normalizing Algorithm 4.1
3. Stop-word removal as shown in section 4.2.4
4. Apply Stemming Algorithm with an enhanced suffix list and enhance normalizing
5. Compare results with other Algorithm 3.1 and 3.2
6. Parse output stemmed dataset from different stemming algorithms to Terrier IR
7. Scoring with TF-IDF options.

Figure 4-14: Pseudo Code of Proposed System

Chapter 5 Experimental Results

This chapter describes the results and analyses of the proposed system with enhanced stemming techniques on the selected Arabic datasets and the integration of Terrier IR platform with the proposed toolkit.

In our experiments we introduce Arabized words and show how stemming should avoid them, also the enhanced normalizing and tokenization with the choice of appropriate encoding is introduced, in addition to that two stop-word lists are introduced, one with intensive stop-word removal and the other with a more light list for testing and allowing wide choices for different researcher in the future.

A toolkit is presented with three stemming algorithms and preprocessing capabilities to make a base toolkit for future Arabic stemming and preprocessing improvements, this toolkit has a wide choice to allow different running scenarios for experiments and examinations.

5.1 Implementation Environment

The proposed system and the integration with the Terrier IR platform are implemented using Terrier IR platform and Java programming. We use Java programming language to build the proposed system and the toolkit alongside with preprocessing components and the GUI, also a Java framework programming language called Apache commons is used for its advance features.

In our implementation we use the hardware of a custom build PC with the following specs:

- CPU: Intel Core i7 and speed of 3.4 GHz,
- Memory: 8 GB
- OS: Microsoft Windows 7 64bit

For software we use the following:

- Programming Language: Java with JDK7
- Frameworks: Apache Common

- Platforms: Terrier IR Platform
- IDE: Eclipse

5.2 Datasets

Arabic is the official and religious language of millions of Arabs and Muslims around the globe, which make Arabic one of the most important Language in the world.

Most IR research was extended out in English and fueled by the annual Text Retrieval Conferences (TREC) sponsored by NIST (the National Institute of Standards and Technology). NIST has collected large quantities of standard data (text collections, inquiries, and relevance judgments) so that IR researchers can compare their techniques on common data sets. Most cases of these datasets are not publicly usable.

5.2.1 Open Source Arabic Corpora

For Arabic language preprocessing test, the dataset Open Source Arabic Corpora (OSAC) collected by Saad and Ashour [9] will be applied. The corpora include:

- BBC Arabic corpus: described in Table 5-1 and collected from bbcarabic.com, includes 4,763 text documents. Each text document belongs to 1 of 7 categories (Middle East News 2356, World News 1489, Business & Economy 296, Sports 219, International Press 49, Science & Technology 232, and Art & Culture 122). The corpus contains 1,860,786 (1.8M) words and 106,733 district keywords after stop-words removal.

Table 5-1 BBC Arabic corpus details

Number	Category	Number of text document
1.	Middle East News	2356
2.	World News	1489
3.	Business	296
4.	Science & Technology	232
5.	Sports	219
6.	Entertainments	122

7.	World Press	49
Total		**4,763**

- CNN Arabic corpus: described in Table 5-2 and collected from cnnarabic.com, includes 5,070 text documents. Each text document belongs to 1 of 6 categories (Business 836, Entertainments 474, Middle East News 1462, Science & Technology 526, Sports 762, and World News 1010). The corpus contains 2,241,348 (2.2M) words and 144,460 district keywords after stop-words removal.

Table 5-2 CNN Arabic corpus details

Number	Category	Number of text document
1.	Business	836
2.	Entertainments	474
3.	Middle East News	1462
4.	Science & Technology	526
5.	Sports	762
6.	World News	1010
Total		**5070**

- Open Source Arabic Corpus (OSAC): described in Table 5-3 and collected from multiple sites, includes 22,429 text documents. Each text document belongs to 1 of 11 categories (Economics, History, Entertainments, Education & Family, Religious and Fatwas, Sports, Heath, Astronomy, Low, Stories, Cooking Recipes). The corpus contains about 18,183,511 (18M) words and 449,600 district keywords after stop-words removal

Table 5-3 OSAC Arabic corpus details

Number	Category	Number of text document
1.	Economic	3102
2.	History	3233
3.	Education and family	3608
4.	Religious and Fatwas	3171
5.	Sport	2419
6.	Health	2296
7.	Astronomy	557
8.	Law	944
9.	Stories	726
10.	Cooking Recipes	2373
Total		**22,429**

5.2.2 Watan-2004 Corpus

For evaluation using the TF - IDF score, the Watan-2004 corpus based on Abbas [50] work is used, and contains about 20000 articles talking about the six following topics: Culture, Religion, Economy, Local News, International News, and sports. In this corpus, punctuation has been omitted intentionally. Table 5-4 describes the Watan-2004 dataset.

Table 5-4 Watan-2004 corpus details

Topic	Corpus Size (Number of documents)
Culture	2782
Religion	3860
Economy	3468
Local News	3596
International News	2035
Sports	4550
Total number of docs	**20291**

5.3 Stop-word Lists Analysis

In this subsection we examine the role of the two proposed stop-word lists intensive stop-word list and light stop-word list using 3102 documents from OSAC dataset, particularly the economy section and notice how these stops-words will melt off the size of tokens that require to be treated by the other systems like IR or TC systems.

We use two properties from Terrier IR platform properties file which are "termpipline" and "stopwords.filename" the first one tells the Terrier IR system to process the term in the given series of preprocessing techniques, for example, you may set the value to stop-words, then follow with your stemmer name which tell the system to remove stop-words first then apply the specific stemming, in Table 5-5 we did not use any stop-word list.

Table 5-5 Stop-word Analysis: non-used

Trec.properties	Value
termpipline	Empty
stopwords.filename	Empty
Number Of Documents: 3102 **Number of Tokens:** 7318033	

We notice that the number of tokens without the use of any stop-word removal is 7318033 token that need to be processed by the stemming algorithm.

Table 5-6 Stop-word Analysis: intensive

Trec.properties	Value
termpipline	stopwords
stopwords.filename	Intensive-stop-words
Number Of Documents: 3102 **Number of Tokens:** 1810121	

From Table 5-6 and after use the intensive sop-word list, the number of document tokens become 1810121 token, which is much better value to be processed by the algorithm and Terrier IR system.

Table 5-7 Stop-word Analysis: light removal

Trec.properties	Value
termpipline	stopwords
stopwords.filename	light-stop-words
Number Of Documents: 3102 **Number of Tokens:** 7318029	

As for Table 5-7 and after use the light stop-word list no significant improvements are noticed, but we include the light stop-word list as intensive stop-word removal may affect the recall for IR systems, any standard stemming algorithm should include both intensive and light stop-word list to allow the wide possibility for researchers to investigate the effects of stop-word removal on precision and recall among other things.

5.4 Advance Normalizing Analysis

We demonstrate the enhanced normalization preprocessing step, and how the proposed system use regular expression for advance removal of unwanted data, for example punctuations marks because this may affect some applications that use preprocessing like Text Classification and IR that may process ambiguous and poor words due to weak normalization. In Table 5-8 we choose a random sample from the sport section available in OSAC dataset that contains data that was collected from web pages which may contain many punctuation marks and other special characters, any good preprocessing algorithm must take care of these characters as it may be fed into IR or TC systems.

Table 5-8 Normalization Demonstration

Token	Normalized word by Proposed Algorithm	Larkey
تحديث:	تحديث	تحديث:
السبت,	السبت	السبت,
أغسطس/	اغسطس	اغسطس/
آب,	اب	اب,
(فيفا)	فيفا	(فيفا)
المقبل.	المقبل	مقبل.
دولارا،	دولارا	دولارا،
العمل،	العمل	عمل،

As we saw in Table 5-8 the algorithm has successfully removed all unnecessary characters like punctuation marks and special characters that may change the meaning of the words or the form of the word, in some retrieval or classification applications, these glitches may have a bad impact on the results because computer machines does not know the difference and take the whole word as it is.

Stemming algorithms should handle punctuation marks and special characters using regular expression as we saw in chapter four, as regular expression has a powerful way to remove unnecessary characters from a word, so in our proposal for creating rules and guidelines for standard Arabic stemming algorithm a rule with the use of regular expression for normalizing step should be added.

5.5 Arabized Word List Comparison

In this section we compared the proposed Arabized word list through different stemming algorithms that can be found in the proposed toolkit which are the proposed enhanced light stemming algorithm, Larkey light stemming algorithm and Khoja root-based algorithm.

In Table 5-9 we demonstrated samples of the Arabized word list and notice how every algorithm we used deals with them, in the proposed algorithm we chose to skip these words because we cannot apply Arabic stemming rules on non-Arabic words, but we apply normalization techniques to unify all the terms.

Table 5-9 Arabized Word List Comparison

Original Word	Proposed Algorithm	Larkey	Khoja
أَفْرَنْج	افرنج	افرنج	افْرَنْج
أَبْرِيْل	ابريل	ابريل	اَبْرِيْل
أَرْثُوْذَكْس	ارثوذكس	ارثوذكس	اَرْثُوْذَكْس
أَغَسْطَس	اغسطس	اغسطس	اُغَسْطَس
أَكْتُوبَر	اكتوبر	اكتوبر	اَكْتُوبَر
إِيرَان	ايران	اير	ايرَان
إِيْدِيُولُوجِيَا	ايديولوجيا	ايديولوجيا	ايْدِيُولُوجِيَا
إِنْجِلِيْزِيَّة	انجليزيه	انجليز	اِنْجِلِيْزِيَّة
اوتوكاد	اوتوكاد	اوتوكاد	وكد
اِبْرَاهِيْم	ابراهيم	ابراهيم	اِبْرَاهِيْم
براغماتي	براغماتي	براغمات	رغم
بعلبك	بعلبك	بعلبك	علب
بكتيريا	بكتيريا	بكتيريا	بكتيريا
بلغارية	بلغاريه	بلغار	بلغارية
بَرْبَر	بربر	بربر	بَرْبَر
بَطَاطَا	بطاطا	بطاطا	بَطَاطَا
بَغْدَاد	بغداد	بغداد	بَغْدَاد
بَنْشَرَ	بنشر	بنشر	بَنْشَرَ
بُسْتَان	بستان	بست	بُسْتَان
تَمُّوْز	تموز	تموز	تَمُّوْز
تُوْنِس	تونس	تونس	تُوْنِس
تِلْفَاز	تلفاز	تلفاز	تِلْفَاز
جُغْرَافِيَة	جغرافيه	جغراف	جُغْرَافِيَة
خَان	خان	خان	خَان
دِيسَمْبَر	ديسمبر	ديسمبر	دِيسَمْبَر
سَانْدوِيش	ساندويش	ساندويش	سَانْدوِيش
سَبْتَمْبَر	سبتمبر	سبتمبر	سَبْتَمْبَر
سَلَطَة	سلطه	سلط	سَلَطَة
سِرْوَال	سروال	سروال	سِرْوَال
شَرْشَف	شرشف	شرشف	شَرْشَف
شَطْرَنْج	شطرنج	شطرنج	شَطْرَنْج
شَنْطَة	شنطه	شنط	شَنْطَة
صَالُوْن	صالون	صال	صَالُوْن
صُوفَا	صوفا	صوفا	صُوفَا
طَاوِلَة	طاوله	طاول	طَاوِلَة

طَرْبُوْش	طربوش	طربوش	طَرْبُوْش
عَرَّاب	عراب	عراب	عَرَّاب
غَوْغَلَ	غوغل	غوغل	غَوْغَلَ
غَوْغَلَة	غوغله	غوغل	غَوْغَلَة
فَارِس	فارس	فارس	فَارِس
فَبْرَايِر	فبراير	فبراير	فَبْرَايِر
فَرَنْج	فرنج	فرنج	فَرَنْج
فَرَنْسَا	فرنسا	فرنسا	فَرَنْسَا
فَهْرَسَ	فهرس	فهرس	فَهْرَسَ
فُرْس	فرس	فرس	فُرْس
فُسْتَان	فستان	فست	فُسْتَان
فَنْدُق	فندق	فندق	فَنْدُق
فِهْرِس	فهرس	فهرس	فِهْرِس
فِيزْيَاء	فيزياء	فيزياء	فِيزْيَاء
قَابِيْل	قابيل	قابيل	قَابِيْل
قَلَنْسُوَة	قلنسوه	قلنسو	قَلَنْسُوَة
كَمُّوْن	كمون	كم	كَمُّوْن
كَنَبَة	كنبه	كنب	كَنَبَة
كُشْك	كشك	كشك	كُشْك
كُورَة	كوره	كور	كُورَة
كُورِيَّة	كوريه	كور	كُورِيَّة
كِشْك	كشك	كشك	كِشْك
كِيمْيَاء	كيمياء	كيمياء	كِيمْيَاء
مَارِس	مارس	مارس	مَارِس
مَايُو	مايو	مايو	مَايُو
مَغُول	مغول	مغول	مَغُول
مَكَّرَوْنَة	مكرونه	مكرون	مَكَّرَوْنَة
مُهَنْدِس	مهندس	مهندس	مُهَنْدِس
مُوْسَى	موسي	موس	مُوْسَى
نُوفَمْبَر	نوفمبر	نوفمبر	نُوفَمْبَر
نِسْرِين	نسرين	نسر	نِسْرِين
هَمْبَرْگَر	همبركر	همبركر	هَمْبَرْگَر
هَنْدَسَة	هندسه	هندس	هَنْدَسَة
هَوْنَدَا	هوندا	هوندا	هَوْنَدَا
وِيْكِيْبِيْدِيَا	ويكيبيديا	يكيبيديا	وِيْكِيْبِيْدِيَا
يَنَايِر	يناير	يناير	يَنَايِر
يَنْسُوْن	ينسون	ينس	يَنْسُوْن
يَوْنِيَوْ	يونيو	يونيو	يَوْنِيَوْ
يُولِيُو	يوليو	يوليو	يُولِيُو
Defected words	**None**	**32**	**5**

Larkey algorithm was the most algorithm that were affected by these Arabized words, these words does not follow any Arabic rules because it came to the Arabic language

from other languages and cultures and affect the results if we apply stemming rules on them, so for any future standard stemming algorithm, these words must be handled correctly and this list must be updated and maintained as a standard for general guideline for other researchers and developer.

5.6 Terrier IR Results and Analysis

In this subsection we demonstrated the results of the successful integration of our toolkit that contains the enhanced preprocessing techniques with the enhanced stemming algorithm alongside the other two Arabic stemming algorithms, we used samples from the suggested datasets OSAC, BBC, and CNN datasets to test the proposed toolkit.

The proposed toolkit allows you to use the output of the other two stemming algorithms as input for the Terrier IR so researchers can add their own algorithms to the toolkit as same as the enhanced stemming algorithm proposed in our work.

One of the limitations we faced in our experiments that the dataset is not specialized in the IR field, the OSAC, BBC, and CNN datasets does not contains any query list or samples since it is mainly focuses on TC, IR on the other hands, needs queries to be tested alongside the dataset and these queries must be extracted from the dataset.

The terrier IR supports TREC dataset standard and offer more customizing options for these dataset types. Unfortunately these datasets are not free despite of being publicly available, Arabic language needs up to date TREC dataset to be used with IR applications to enrich the field of Arabic IR development. We suggest more research to create a TREC dataset that can be standardized.

In the following tables, we tested the integration of our toolkit with the Terrier IR using data and queries samples form OSAC, BBC, and CNN datasets, also, we considered TF-IDF as a scoring model for the retrieval results.

For the OSAC dataset we use three samples of the Health section. In Figure 5-1 we presented the output of the toolkit when applying the preprocessing steps with light stop-words lists and use the output file with the integrated Terrier IR platform.

```
Start of Algorithm
Read Dataset, Size  : 140
Read Stopset, Size  : 13957
Remove Stop-words, Match  : 33
Tokenize Dataset, size: 107
Normalize Dataset, new size: 103
Stem Dataset, size by Algorithm 1 : 103
```

Figure 5-1: OSAC sample # 1 – Toolkit output using intensive stop-word list

In Figure 5-2 and 5-3 we tested the other two samples with the toolkit, both figures show statistics of each sample run and how the researchers may use light or intensive light stop-word.

```
Start of Algorithm
Read Dataset, Size  : 539
Read Stopset, Size  : 119
Remove Stop-words, Match  : 33
Tokenize Dataset, size: 506
Normalize Dataset, new size: 454
Arabized word on line 89
Stem Dataset, size by Algorithm 1 : 454
```

Figure 5-2: OSAC sample # 2 - Toolkit output using light stop-word list

We notice that sample #3 as shown in Figure 5-3 contains Arabized word and the enhanced algorithm we proposed successfully detect it.

```
Start of Algorithm
Read Dataset, Size  : 353
Read Stopset, Size  : 119
Remove Stop-words, Match  : 24
Tokenize Dataset, size: 329
Normalize Dataset, new size: 309
Arabized word on line 89
Stem Dataset, size by Algorithm 1 : 309
```

Figure 5-3: OSAC sample # 3 - Toolkit output using light stop-word list

In Table 5-10 we demonstrated the output of the enhanced proposed algorithm that we proposed with sample #1 of the OSAC dataset.

Table 5-10 OSAC sample #1 – Preprocessing demonstration

Token	Normalized word	Stemmed word
العيادة	العياده	عياد
الالكترونية	الالكترونيه	الكترون
مشاكل	مشاكل	مشاكل
الجلد	الجلد	جلد
الشائعة	الشائعه	شائع
ممارسة	ممارسه	ممارس
الرياضة	الرياضه	رياض
الأحد,	الاحد	احد

The output of the Terrier IR platform with OSAC samples is shown in Figure 5-4, for the query we choose query randomly since the dataset does not contain any query samples and the three datasets OSAC, BBC and CNN used mainly for TC.

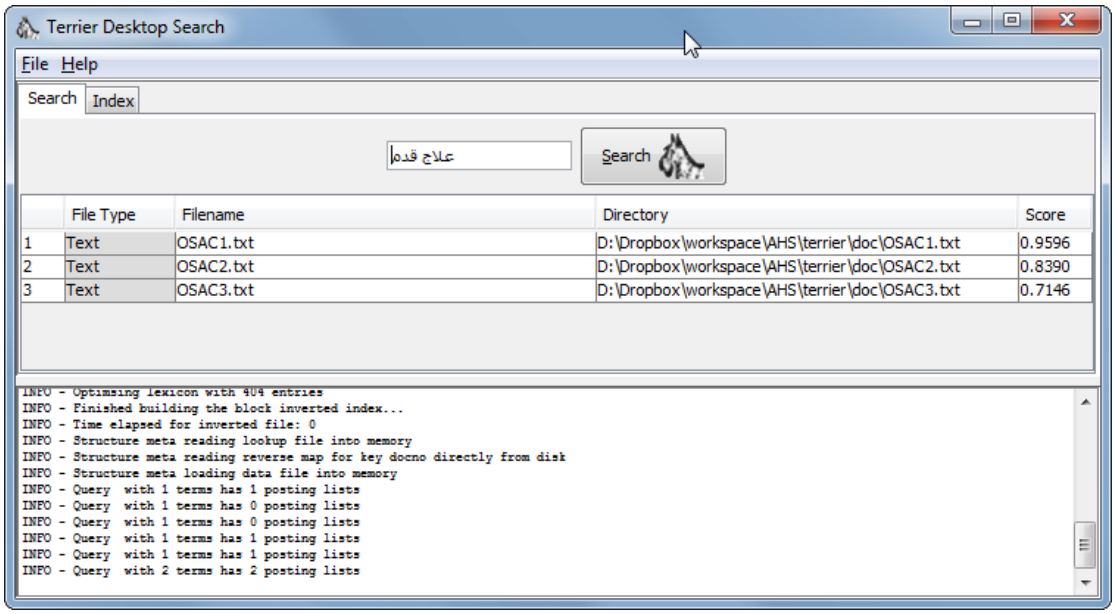


Figure 5-4 OSAC samples score with TF-IDF using enhanced light stemming

Once more, we executed our program with BBC dataset, we preferred three samples randomly from technology and science. In Figure 5-5 we demonstrated a sample of BBC dataset, executed through the preprocessing.

```
Start of Algorithm
Read Dataset, Size  : 303
Read Stopset, Size  : 119
Remove Stop-words, Match  : 33
Tokenize Dataset, size: 270
Normalize Dataset, new size: 263
Arabized word on line 84
Arabized word on line 84
Stem Dataset, size by Algorithm 1 : 263
```

Figure 5-5: BBC sample # 1 - Toolkit output using light stop-word list

As for Figure 5-6, we used a sample randomly chosen from the business section in CCN dataset that was extracted from HTML files.

```
Start of Algorithm
Read Dataset, Size  : 299
Read Stopset, Size  : 119
Remove Stop-words, Match  : 21
Tokenize Dataset, size: 278
Normalize Dataset, new size: 258
Arabized word on line 38
Arabized word on line 99
Stem Dataset, size by Algorithm 1 : 258
```

Figure 5-6: CNN sample # 1 - Toolkit output using light stop-word list

Due to the huge size of the datasets and that these datasets mainly used with TC applications we choose to work with random selection and apply our proposed preprocessing steps and enhanced algorithm on them.

The lack of queries also creates a challenge for us since the datasets used for TC research and does not contain queries, we then choose queries randomly to show and demonstrate that Terrier IR platform support Arabic language and get scoring results.

In Table 5-11 we demonstrated a comparison of the enhanced proposed algorithm that with the using of light stop-word removal and the Larkey stemming algorithm, we used 10 documents extracted from the Watan-2004 dataset. We use 10 random query samples.

Table 5-11 TF-IDF Comparison

Document number and size		**Proposed Algorithm with light stop-word list**		**Larkey Algorithm Implementation**		
		New size	**TF-IDF**	**New size**	**TF-IDF**	**Normalizing error percentage**
1	**105**	85	1.3257	104	1.3257	5.76%
2	**367**	338	1.3273	359	1.3246	9.19%
3	**614**	575	2.0514	613	1.9675	1.79%
4	**163**	140	1.3238	159	1.3238	5.03%
5	**325**	306	1.9902	320	1.9874	2.50%

From Table 5-11 we noticed a slight improvement in TF-IDF measure, and a significant improvement in normalizing process. We use light stop-word list with the proposed algorithm, new size column describes the size of the dataset after stop-word removal, and for the stop-word list we used the proposed light stop-word list.

Normalizing affects the results of the TF - IDF, since the terms could be increased by punctuation marks, and the indexer considered them as terms, these punctuation marks increase the size of the terms and affect the TF-IDF score.

Chapter 6 Conclusions

6.1 Summary and Concluding Remarks

In this dissertation, we introduce an enhancement to the Arabic IR field by using an existing open source application that already supports other languages, then, adding Arabic language support to it. Also we improved one of the most significant steps in IR field which is preprocessing step, this step affects the outcomes of any IR system.

The advantages of Terrier IR Platform motivate us to add Arabic language to it, to help researchers focus on applying the features of this platform for researching and improving the Arabic IR field, the lack of Standard Arabic stemming algorithm and preprocessing steps such as encoding, normalizing, stop-word removal and stemming also motivates us to bring out these instruments.

The thesis suggests a group of principles and guidelines for a standard Arabic stemming algorithm and Arabic preprocessing steps, since there are no standardization for Arabic normalizing, stemming, or stop-word removal.

We proposed a toolkit that handles Arabic preprocessing steps which covers encoding, normalizing, tokenization, stop-word removal and stemming. GUI toolkit can be applied and customized by future research or to inspire researchers and developers to make a layout of tools for Arabic preprocessing that can be used in many fields like natural language processing, TC, and IR.

The proposed GUI toolkit has many options including reading and writing dataset files, display output in tables, and produce statistics about preprocessing steps. This toolkit could considered as a first step through a standard and could be adapted widely in the Arabic language preprocessing, and Arabic IR systems.

We used UTF-8 encoding and suggested to employ it with all upcoming tools and applications, the encoding is important and offer great options specially communicating with other schemes, so we proposed to consider it as perquisite for Arabic language preprocessing.

We improved normalizing step by using regular expression since it allows for powerful removal and cleaning of unnecessary characters that should not enter the preprocessing steps, we add a rule to the set of the existing rules of the available

normalizing implementations, and since there is no normalizing standard we hope to consider these rules as a standard form.

We also introduced the use of two stop-words list for any preprocessing based application, since stop-words removal affect the results of the TC or IR specially recall.

We introduced light-stop-word list that contains 119 words and an intensive stop-word list manually collected and merged from three other stop-words list and contains 13957 stop-words. These options allows more options for researchers to test and improve the effect of stop-words removal on different application like TC, or IR.

Too, we ushered in the use of Arabized words and clear how these words must not abide by any Arabic stemming rules since these words are not Arabic words, we collected 100 Arabized words and include them in our enhanced light stemming algorithm, and this will simplify improving the effects of any stemming algorithm.

Too, we compared the yield of a dataset that contains Arabized words with two popular Arabic stemming algorithms Larkey that failed with 32 Arabized words and Khoja stemmers that failed with 5 Arabized words, we understood that the proposed enhanced stemming algorithm output the best result that consider Arabized words, this also should be regarded as a standard feature for any upcoming Arabic stemming algorithm.

Preprocessing for IR systems is enhanced by the proposed toolkit, and the modular design of the proposed toolkit allows easy developing and integration by other interested groups.

Terrier IR now supports Arabic language using the proposed toolkit, and offers wide options for preprocessing data before indexing it.

6.2 Recommendations and Future Work

In this dissertation, we add Arabic support to a popular IR platform that can be used in training or researching the Arabic IR field, a toolkit for preprocessing also introduced alongside with two stop-words list each for specific usage and different effects.

We noted that the employment of light stop-word list improve scoring, on the other hand the using of intensive stop-word list reducing the size of the index and tokens greatly.

Arabized words should not be stemmed by any means since stemming will change the structure of the word completely and these words came from different languages and cannot follow Arabic stemming rules.

The toolkit and its integration with Terrier IR platform may set a foundation for Arabic preprocessing and IR research, and can be employed to improve many specific fields in IR systems like normalizing, stop-word removal, encoding, tokenization, stemming, indexing, retrieval, scoring models, and datasets.

Granting to the results of experiments and the limitations that we plant during our work, our future work will be given to the following tips:

1- Improve Terrier IR platform Arabic support and implements normalizing and stemming API alongside the stop-word removal into Terrier IR APIs
2- Improving the toolkit towards more modular design to allow other researchers to integrate or use the preprocessing tools that the thesis used.
3- Maintain and update light stop-word list and intensive stop-word list for optimizing and improving overall results.
4- Produce a set of standards and guidelines from this thesis and discuss them with other interested group.
5- Make a snowball like project for languages such as Arabic to be considered by third party application developers and the companies as a standard Arabic stemming algorithm including light based or root based stemming.
6- Develop a TERC like Arabic dataset because of the popularity of TREC conferences and the broad support of AI applications for these types of dataset.
7- Create open source dataset with the query sample to be used with IR applications and research.
8- Using different scoring models other that TF-IDF.

References

[1] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, New York: Cambridge University Press, 2008.

[2] M. R. Al-Maimani, A. Naamany and A. Z. A. Bakar, "Arabic information retrieval: techniques, tools and challenges," in *GCC Conference and Exhibition*, Dubai, 19-22 Feb. 2011.

[3] "Internet World Stats," [Online]. Available: http://www.internetworldstats.com. [Accessed 5 March 2013].

[4] M. Al-Nashashibi, D. Neagu and A. A. Yaghi, "An improved root extraction technique for Arabic words," in *Computer Technology and Development (ICCTD)*, Cairo, 2-4 Nov. 2010.

[5] A. Chen and F. Gey, "Building an Arabic stemmer for information retrieval," in *Proceedings of TREC 2002*, 2002.

[6] M. Aljlayl 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.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, 2009.

[8] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz and T. Euler, "Yale: Rapid prototyping for complex data mining tasks," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, 2006.

[9] M. K. Saad and W. Ashour, "Arabic Morphological Tools for Text Mining," in *EEECS'10 the 6th International Symposium on Electrical and Electronics*

Engineering and Computer Science, Cyprus, 2010.

[10] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald and C. Lioma, "Terrier: A high performance and scalable information retrieval platform," in *Proceedings of the OSIR Workshop*, Seattle, Washington, 2006.

[11] M. Porter, "Snowball: A language for stemming algorithms," 2001. [Online]. Available: http://snowball.tartarus.org. [Accessed 5 March 2013].

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

[13] E. T. Al-Shammari and J. Lin, "Towards an error-free Arabic stemming," in *Proceedings of the 2nd ACM workshop on Improving non english web searching*, New York, NY, 2008.

[14] C. Morris, "A review of recent developments in term conflation approaches for Arabic text information retrieval," in *ACM Conference'10*, Month 1-2, 2010.

[15] T. M. T. Sembok and B. A. Ata, "Arabic Word Stemming Algorithms and Retrieval Effectiveness," in *Proceedings of the World Congress on Engineering*, London, July 3-5, 2013.

[16] M. A. Otair, "COMPARATIVE ANALYSIS OF ARABIC STEMMING ALGORITHMS," *International Journal of Managing Information Technology*, vol. 5, no. 2, 2013.

[17] A. El Salam Al Hajjar, M. Hajjar and K. Zreik, "A System for Evaluation of Arabic Root Extraction Methods," in *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on*, Barcelona, 9-15 May 2010.

[18] G. Kanaan, R. Al-Shalabi, M. Ababneh and A. Al-Nobani, "Building an effective rule-based light stemmer for Arabic language to inprove search

effectiveness," in *Innovations in Information Technology, 2008. IIT 2008. International Conference on*, Al Ain, 16-18 Dec. 2008.

[19] H. Al Ameed, S. Al Ketbi, A. Al Kaabi, K. Al Shebli, N. Al Shamsi, N. Al Nuaimi and S. Al Muhairi, "Arabic light stemmer: a new enhanced approach," in *Proceedings of the second international conference on innovations in information technology*, Al Ain, 9 - 11 November, 2005.

[20] L. S. Larkey, 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*, Tampere, Finland, 2002.

[21] S. Khoja and R. Garside, "Stemming Arabic Text," Computing Department, Lancaster University,, 22 September 1999. [Online]. Available: http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps.

[22] T. Buckwalter, "Buckwalter arabic morphological analyzer version 2.0.," in *Linguistic Data Consortium*, Philadelphia, 2004.

[23] A. Goweder, M. Poesio and A. De Roeck, "Broken plural detection for Arabic information retrieval," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Sheffield, United Kingdom, 2004.

[24] K. Taghva, R. Elkhoury and J. Coombs, "Arabic stemming without a root dictionary," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, 4-6 April 2005.

[25] A. Al-Omari, B. Abuata and M. Al-Kabi, "Building and Benchmarking New Heavy/Light Arabic Stemmer," in *International Conference on Information and Communication Systems*, Irbid, Jordan, 2013.

[26] M. N. Al-Kabi, "Towards improving Khoja rule-based Arabic stemmer," in *Applied Electrical Engineering and Computing Technologies (AEECT), 2013*

IEEE Jordan Conference on, Amman, 3-5 Dec. 2013.

[27] K. Darwish and A. M. Ali, "Arabic Retrieval Revisited: Morphological Hole Filling," *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics*, 2012.

[28] Boudlal, Abderrahim, B. Rachid, L. Abdelhak, M. Azzeddine, M. Abdelouafi and M. Bebah., "A Markovian approach for Arabic root extraction," *The International Arab Journal of Information Technology 8*, no. 1, pp. 13-20, 2009.

[29] T. Dilekh and A. Behloul, "Implementation of a New Hybrid Method for Stemming of Arabic Text," in *International Conference on Communications and Information Technology ICCIT* , Beirut, 2012.

[30] A. Boubas, L. Lulu, B. Belkhouche and S. Harous, "GENESTEM: A novel approach for an Arabic stemmer using genetic algorithms," in *Innovations in Information Technology (IIT), 2011 International Conference on*, Abu Dhabi, 25-27 April 2011.

[31] Y. Alhanini and M. J. A. Aziz., "The Enhancement of Arabic Stemming by Using Light Stemming and Dictionary-Based Stemming," *Journal of Software Engineering and Applications*, vol. 4, no. 9, pp. 522-52, 2011.

[32] M. Hadni, A. Lachkar and S. A. Ouatik, "A new and efficient stemming technique for Arabic Text Categorization," in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, Tangier, 10-12 May 2012.

[33] M. Y. Al-Nashashibi, D. Neagu and A. Yaghi, "Stemming techniques for arabic words: A comparative study," in *Computer Technology and Development (ICCTD)*, Cairo, 2-4 Nov. 2010.

[34] H. M. AlSerhan, S. Alqrainy and A. Ayesh, "Is paice method suitable for evaluating Arabic stemming algorithms?," in *Computer Engineering \&*

Systems, 2008. ICCES 2008. International Conference on, Cairo, 25-27 Nov. 2008.

[35] A. Goweder, H. Alhami, T. Rashed and A. Al-Musrati, "A hybrid method for stemming Arabic text," *International Arab conference on information technology (ACIT),* pp. 16-18, 2008.

[36] B. E. Rogerson, *An evaluation of existing light stemming algorithms for Arabic keyword searches,* Dissertation, University of North Carolina, 2008.

[37] I. A. El-Khair, "Effects of stop words elimination for Arabic information retrieval: a comparative study," *International Journal of Computing \& Information Sciences,* vol. 4, no. 3, pp. 119-133, 2006.

[38] J. Atwan, M. Mohd and G. Kanaan, "Enhanced Arabic Information Retrieval: Light Stemming and Stop Words," in *Soft Computing Applications and Intelligent Systems,* Springer Berlin Heidelberg, 2013, pp. 219-228.

[39] R. Al-Shalabi, G. Kanaan, J. M. Jaam, A. Hasnah and E. Hilat, "Stop-word removal algorithm for Arabic language," *Proceedings of 1st International Conference on Information and Communication Technologies: From Theory to Applications,* pp. 545-550, 2004.

[40] H. R. B. Froud, A. Lachkar and S. A. Ouatik, "Stemming and similarity measures for Arabic Documents Clustering.," *I/V Communications and Mobile Network (ISVC),* pp. 1-4, 2010.

[41] A. Fraser, J. Xu and R. Weischedel, "TREC 2002 Cross-lingual Retrieval at BBN," *TREC 2002 Proceedings,* 2002.

[42] L. Khreisat, "Arabic text classification using N-gram frequency statistics," *Proceedings of the 2006 International Conference,* pp. 78-82, 2006.

[43] "Windows 1256," Microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/goglobal/cc305149. [Accessed 5 January

2014].

[44] "Unicode 6.3 Character Code Charts," Unicode, [Online]. Available: http://www.unicode.org/charts/. [Accessed 20 August 2013].

[45] I. Ounis, G. Amati, P. V., B. He, C. Macdonald and D. Johnson, "Terrier Information Retrieval Platform," *In Proceedings of the 27th European Conference on IR Research* , vol. 3408, 2005.

[46] "Arabic Stop Words," [Online]. Available: http://arabicstopwords.sourceforge.net/. [Accessed 5 March 2013].

[47] J. Savoy, "Jacques Savoy Dataset," [Online]. Available: http://members.unine.ch/jacques.savoy/clef/. [Accessed 2013].

[48] M. Saad, "Motaz Saad Website," [Online]. Available: https://sites.google.com/site/motazsite/. [Accessed 2013].

[49] Wiktionary, "Arabized Words," [Online]. Available: http://ar.wiktionary.org/wiki/%D8%AA%D8%B5%D9%86%D9%8A%D9%81:%D9%83%D9%84%D9%85%D8%A7%D8%AA_%D9%85%D8%B9%D8%B1%D8%A8%D8%A9. [Accessed December 2013].

[50] M. Abbas, K. Smaili and D. Berkanis, "Evaluation of Topic Identification Methods on Arabic Corpora," *Journal of Digital Information Management,* vol. 9, no. 5, pp. 185-192, 2011.