إقـــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Building an Effective Stemmer for Arabic Language to Improve Search Effectiveness

تصميم مجذع كلمات للغة العربية لتحسين فعالية البحث

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي 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 أية 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: Ahmed Ibraheem Shaqalie **اسم الطالب: أحمد إبراهيم شقليه**

Signature : **التوقيع:**

Date: **التاريخ: 2015/2/9**

Islamic University, Gaza, Palestine
Research and Graduate Affairs
Faculty of Engineering
Computer Engineering Department

Building an Effective Stemmer for Arabic Language to Improve Search Effectiveness

By

Ahmed Ibraheem .J. Shaqalieh

Supervisor

Prof. Mohammad A. Mikki

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

1436 H (November,2014)

بسم الله الرحمن الرحيم

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

الرقم Ref: ج س غ/35/.....
التاريخ Date: 2014/11/25

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **أحمد إبراهيم جوده شقليه** لنيل درجة الماجستير في **كلية *الهندسة*** قسم **هندسة الحاسوب** وموضوعها:

تصميم مجذّع كلمات للغة العربية لتحسين فعالية البحث

Building an effective stemmer for Arabic language to improve search effectiveness

وبعد المناقشة التي تمت اليوم الثلاثاء 03 صفر 1436هـ، الموافق 2014/11/25م الساعة الثانية عشرة ظهراً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

أ.د. محمد أمين مكي	مشرفاً ورئيساً	.....
د. أيمن أحمد أبو سمرة	مناقشاً داخلياً	.....
أ.د. سامي سليم أبو ناصر	مناقشاً خارجيًا	.....

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 ولي 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

To my mother

To my wife

To my beloved son

To my father and mother in law

To my brothers

To my sisters

To my friends

Acknowledgements

Thanks to Allah for giving me the power and help to accomplish this research. Without the grace of Allah, I was not able to accomplish this work.

I would like to thank my parents and my wife very much for their pray, patience, motivation and continues support.

I am grateful to my supervisor, Prof. Mohammad Mikki, for his enormous support, valuable guide, and assistance throughout the work of this research.

Table of Contents

Dedication iii
Acknowledgements iv
List of Abbreviations vii
Abstract xi
Keywords xii
Chapter 1 Introduction 1
1.1. Background of stemmers in Arabic Language 1
1.2. Text Mining (TM) 3
1.3. Arabic Language 5
1.4.1. Complexity of Arabic Language 5
1.4.2. Arabic Corpus Problem 6
1.4. Research Motivation 8
1.5. Research Contribution 8
1.6. Thesis Structure 8
Chapter 2 : Related Work 10
2.1. Stemmers 10
2.1.1. Lookup Approach 10
2.1.2. Linguistic Approach 11
2.1.3. Combinational Approach 11
2.2. Applying Root Stemmer on Arabic Text 11
2.3. Applying Light Stemmer on Arabic Text 14
2.4. Applying Statistic Stemmer on Arabic Text 25
Chapter 3 : Background 29
3.1. Naive Bayes (NB) 29
3.2. Naive Bayes Multinomial (NBM) 30
3.3. Complement Naive Bayes (CNB) 31
3.4. Discriminative Multinomial Naive Bayes (DMNB) 32
3.4.1. Frequency Estimate 32
3.4.2. Discriminative Frequency Estimate 33
3.5. Support vector machines (SVMs) 34
3.6. Vector Space Model (VSM) and Term Weighting Schema 35
3.6.1. Term weighting equation 36
3.6.2. $\boldsymbol{Tf - Idf}$ example 38
Chapter 4 : Methodology 39
4.1. Stemming Algorithm(Shaqalieh Stemmer) 41

4.1.1. Dividing Text into Words 42
4.1.2. Normalization 42
4.1.3. Classify Stop Word precedes Nouns and Stop Word precedes Verbs 42
4.1.4. Delete All Stop Word 43
4.1.5. Searching in Irregular words 43
4.1.6. Applying Light Stemmer on Nouns 43
4.1.7. Applying Root Stemmer on Verbs 48
4.2. Morphological Analysis (Stemmer and Light Stemmer) 52
4.3. Text Preprocessing Tools (WEKA) 55
Chapter 5 : Corpora 60
5.1. Corpora Building Steps 61
5.2. Corpora Summary 62
5.3. BBC Arabic corpus 63
5.4. CNN Arabic corpus 63
5.5. OSAC corpus 64
Chapter 6 : Experimental results and analysis 65
6.1. Preprocessing time 65
6.2. Classifier Accuracy 67
6.3. Morphological Analysis and term pruning 71
Chapter 7 : Conclusion and Future work 73
7.1. Conclusion 73
7.2. Future Works 74
References 75

List of Abbreviations

SP_WOAL	Remove Suffix then Prefix then Match with Pattern
SVM	Support Vector Machine
TC	Text Categorization
tf	Term Frequency
IR	Information Retrieval
K-NN	K Nearest Neighbors
LSA	Latent Semantic Analysis
Min-F	Minimum Term Frequency
NB	Naive Bayes
Norm	Normalize data
OSAC	Open Source Arabic Corpus
ASR	Automated Speech Recognition
BBC	British Broadcasting Corporation
CNN	Cable News Network

List of Figures

Figure 1.1: Text mining 3
Figure 1.2: Text mining process 4
Figure 2.1: N-gram procedure 27
Figure 2.2: Frequency Distribution of Tri-gram 28
Figure 4.1: general diagram of the proposed Shaqalieh stemmer 41
Figure 4.2: Flowchart for updated light stemmer 46
Figure 4.3: root stemmer - ISRI 51
Figure 4.4: WEKA GUI Chooser 57
Figure 4.5: Browser to select arff file 57
Figure 4.6: Analysis data to view the file included in arff file 58
Figure 4.7: WEKA Arabic Stemmers including the proposal Root and Light stemmers "Shaqalieh Stemmer" 58
Figure 4.8: Browser to select classifier 59
Figure 5.1: Corpus Building Steps 62
Figure 5.2: The three corpora are available publically 62
Figure 6.1: Average time required to analyze the corpora 66
Figure 6.2: Preprocessing time of different term weighting schemes 67
Figure 6.3: Classifiers Average Performance 68
Figure 6.4: Khoja, Light and Proposed stemming time for Knn, SVM and NBM classifiers . 69
Figure 6.5: Khoja, Light and Proposed stemming accuracy classification for Knn, SVM and NBM classifiers 70
Figure 6.6: Recall/Precision 70
Figure 6.7: Shaqalieh stem vs. Stemming vs. light stemming vs. raw stem(Average Accuracy) 71

List of Tables

Table 1.1: different shapes of letter "ع" depending of its position in the word.....6
Table 4.1: Irregular words list.....43
Table 4.2: A word and its affixes.....43
Table 4.3: Arabic prefixes.....44
Table 4.4: Arabic suffixes.....44
Table 4.5: Sample the First type of patterns.....45
Table 4.6: Sample the Second type of patterns.....45
Table 4.7: Affix Sets.....49
Table 4.8: Arabic Pattern and Roots.....50
Table 4.9: WEKA String to word Vector options.....56
Table 5.1: bbc-arabic-utf8.....63
Table 5.2: cnn-arabic-utf8.....64
Table 5.3: osac-uft8.....64

بناء مجذع فعال للغة العربية لتحسين فعالية البحث

أحمد إبراهيم جوده شقليه

الملخص

إن إنشاء مجذع فعال للغة العربية يأتي من أهمية اللغة العربية باعتبارها اللغة السادسة الأكثر استخداما في العالم. وتنبع أهمية بناء مجذع فعال في التحسين في استرجاع المعلومات, استخراج البيانات, ومعالجة اللغة. وقد تم تطوير العديد من المجذعات اللغوية للغة العربية, ولكن لا يزال هناك الكثير من الضعف والمشاكل.

تقترح هذه الأطروحة خوارزمية فعالة وتعتمد على الفصل بين الأسماء والأفعال وذلك من خلال وضع قواعد للتمييز بين النوعين ومعالجة كل نوع باستراتيجية مختلفة عن الآخر مما أدى ذلك إلى زيادة كفاءة تحديد الكلمات. هذه الخوارزمية ستسهم في تعزيز كفاءة وسرعة استرجاع المعلومات ومحركات البحث. باستخدام هذه القواعد سيتم حل مشكلة عدم الوضوح.

تم تطوير المجذع المقترح باستخدام لغة البرمجة JAVA مع JDK1.6, وقد تم تطبيق المجذع المقترح على WEKA البيئة الحاضنة للعديد من المجذعات اللغوية لتقييمها.

استخدمت WEKA لإجراء اختبار للمجذع المقترح ومقارنتها مع مجذعات أخرى مثل Khoja و Light10, وقد أظهرت النتائج التي تم استخراجها باستخدام كلا من OSAC,CNN أن المجذع المقترح يزيد من دقة تصنيف النصوص إلى متوسط قدره 90.1% وهو أفضل من استخدام مجذع Khoja و Light10 اللذان يحققان متوسط قدره 88.2%, 85.17% على الترتيب.

Building an Effective Stemmer for Arabic Language to Improve Search Effectiveness

Ahmed Ibraheem Judah Shaqalieh

Abstract

Creating good stemming rules for the Arabic language comes from the importance of Arabic language as the sixth most used language in the world. Stemming is very important in information Retrieval , data mining , language processing . Many linguistic and light stemmers have been developed for Arabic language but still there are many weakness and problem.

This thesis proposes an efficient stemming algorithm that developed to solve the problems with several stemming approaches like ambiguity, broken plural problems, irregular words and confusion between nouns and verbs, a proposed stemming algorithm uses two stemming approaches, the root stemming for verbs and the light stemming for nouns.

The proposed algorithm will depend on separation between nouns and verbs by adding classification rules and addresses every part of words by special strategy, to increase efficiency of stemming words. Such algorithm will contribute to enhanced efficiency and speed of information retrieval and search engines, By using these rules, it can solve the ambiguity of words.

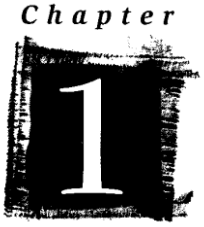
A new Arabic stemmer has been developed using Java Programming Language with JDK 1.6 and applied this stemmer on WEKA (Waikato Environment for Knowledge Analysis) for text preprocessing and classification and which it suitable environment for most stemmers to evaluation, it allows user to load any data set, choose from any included stemmers, select any included classifier and explorer the classification results like recall and precision ..etc.

WEKA used to test the proposed stemmer and compare it with other stemmers like Khoja stemmer and Light10 stemmer, the researcher compared the proposed stemmer with Khoja and Light10 by using OSAC (Open Source Arabic Corpus) and CNN (Ca-

ble News Network) corpus show that the proposed stemmer increase accuracy of text classification to an average of 90.1% which is better than using Light10 and Khoja which achieve an average accuracy of 88.2% and 85.17% respectively.

Keywords

Arabic Text Mining, Arabic text preprocessing / classification, Word Stemming, Arabic morphological analysis (Arabic /light stemming), Information Retrieval.



Chapter 1 Introduction

This chapter introduces background of stemmers in Arabic language, Text Mining (TM) and Information Retrieval (IR), describes Arabic Language, discusses the complexity of Arabic Language, and finally states the research motivation.

1.1. Background of stemmers in Arabic Language

The Arabic language differs from other languages syntactically, morphologically, and semantically. It is a Semitic language in which most words are built up from roots by following certain fixed patterns or measures. [1]

Such patterns can be thought of as templates conforming to Arabic grammatical rules and are applied by adding affixes to roots. Additional affixes (i.e., prefixes, infixes and suffixes) may be added to derive different grammatical usages such as possessives, plurals, definite forms, gender, etc. [2]

One main property of the Arabic language that makes it different from most of other languages is that it is a derivational, while most of the other languages, such as English, are concatenative.[3]For example, the removal of prefixes in Arabic words does not reverse the meaning of words, while in English, this operation could reverse the meaning of words, or even change their grammatical function. [4]

Computational Arabic morphology has drawn the attention of many researchers and many approaches were proposed to analyze the Arabic language at the morphological level. Stemming and morphological analysis techniques are computational processes that analyze the internal structure of words. The main objective of both techniques is to remove all of a word's affixes to end up with the word's stem or root. [4] [5] [6]Such techniques proved useful for many applications such as information retrieval, text categorization, dictionary automation,

text compression, data encryption, vowelization and spelling aids, automatic translation, and computer aided instruction. [1]

Stemming can be viewed as the process of normalizing word variations that share some semantic relation into one shared affix free stem. [7]

There are two types of stemming algorithms: stem based and root based algorithms. [8]

Stem based algorithms, such as Buckwalter's stemmer [9], remove suffixes and prefixes from Arabic words. Root based stemmers, on the other hand, produce roots from the generated stems (Examples: Beesley [1], Al Shalabi [2], and Khoja [10]stemmers).

Arabic stemming algorithms can be classified into four classes [11]: manually constructed dictionaries, light stemmers, morphological analyzers, and statistical stemmers. The manually constructed dictionaries generate dictionaries (tables) listing the Arabic prefixes, suffixes, and stems/roots such as Buckwalter's stemmer [10] . However, these dictionaries might not be exhaustive [12].

The light stemming algorithms remove the prefixes and suffixes to produce the original stems of words without the need to find the roots. They group semantically related words that share the same stem. The morphological analyzers attempt at restoring the original root of a word and group words sharing the same root. Note that words that are not semantically related may also be grouped into the same root class. The statistical stemmers, on the other hand, group word variants according to clustering techniques [13].

There are several root based stemming approaches that are applied for Arabic language. The researcher will briefly explain some of them before introducing our proposed approach. Al Shalabi [14]developed a system for extracting the roots of Arabic words. It first removes the longest prefix that precedes the first root letter in the input word. It then checks for the root in the new word formed by removing the prefix. Typically, the root would be within the first four or five letters. Al Shalabi, Kanaan and Muaidi [15]designed another stemmer to find the

roots of Arabic words by performing certain calculations on the word's letters after assigning numeric values to each letter.

The algorithm extracts the correct roots with an accuracy rate reaching 95%. Khoja [10] developed a stemmer algorithm, which uses the morphological patterns to detect roots of three or four letters. The algorithm initially removes the affixes of a given Arabic input word including prefixes, suffixes, and infixes. The algorithm then checks the validation of the word after the elimination of affixes and verifies whether the removed affix is part of the root or not. The resulting stem is then checked for correctness. It matches the remaining letters of the input word against a list of patterns of the same length in order to extract the root. The last step verifies whether the extracted root is valid by checking it against a list of around 4700 roots. An enhanced stemmer of Khoja's algorithm has been developed in [16].The algorithm extracts the roots in a similar way to the Khoja stemmer but without a root dictionary.

1.2. Text Mining (TM)

Data mining is the process of extracting patterns from data as shown in figure 1.1. Data mining is becoming an increasingly important tool to transform the data into information. Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text.

Figure 1.1: Text mining

High-quality information is typically derived through the divining of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the struc-

tured data, and finally evaluation and interpretation of the output as shown in figure 1.2. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness.

Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities) [17] [18].

The purpose of Text Mining is to process unstructured (textual) information, extract meaningful numeric indices from the text, and make the information contained in the text accessible to the various data mining algorithms. Information can be extracted to derive summaries for the words contained in the documents or to compute summaries for the documents based on the words contained in them.

Hence, The researcher can analyze words, clusters of words used in documents, etc., or The researcher could analyze documents and determine similarities between them or how they are related to other variables of interest in data mining. In the most general terms, text mining will "turn text into numbers" (meaningful indices), which can then be incorporated in other analyses such as predictive/descriptive data mining. [17] [19]

Text mining is well motivated, due to the fact that much of the world‘s data can be found in text form (newspaper articles, emails, literature, web pages, etc.).

Text mining processes include Document Collection, Retrieve and preprocess document, Analyze Text (Information Extraction, Clustering, Summarization), Management Information System and Knowledge [17] [19].

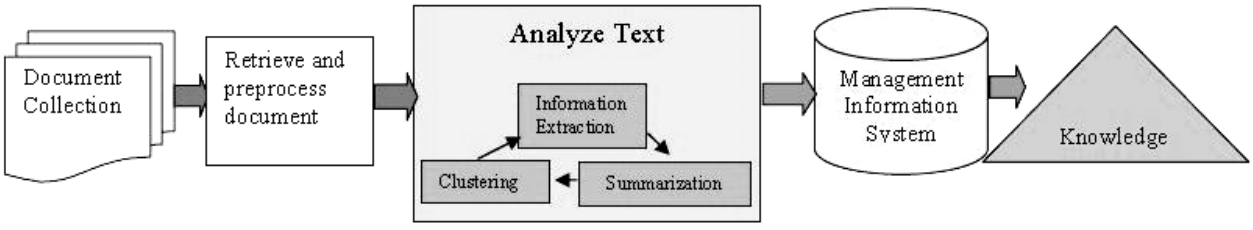


Figure 1.2: Text mining process

1.3. Arabic Language

Arabic is one of the most complex languages, in both its spoken and written forms. However, it is also one of the most common languages in the world as it is spoken by more than 400 million people as a first language and by 250 million as a second language [8]. Arabic Language belongs to the Semitic language family.

Arabic alphabet consists of 28 letters that structure the words; words are divided into three parts of speech: noun, verb, and particle. Nouns and verbs are derived from a closed set of around 11,311 roots distributed as follow:

- 115 two character roots (no derivation from them).
- 7198 three character roots.
- 3739 four character roots.
- 259 five characters roots.

These roots can be joined with several infixes to generate more patterns of the word [9], for example several forms can be derived from the root "فعل" of the morpheme "صنع" , the form "مصنع" can be found by adding the letter "م" to the morpheme "صنع".

The Arabic script has numerous diacritics (Damma, Fathah, Kasra, Shaddah)which decide how a word should be pronounced. Arabic has two genders (feminine and masculine),three cardinalities (singular, dual and plural), three grammatical cases (nominative, genitive and accusative) and two tenses (perfect and imperfect). Arabic nouns are formed differently depending on the noun gender, cardinality, and grammatical case [10].

1.3.1 Complexity of Arabic Language

Arabic is considered as one of the highly inflectional languages with complex morphology and considered as challenging language for a number of reasons:

- Morphological variation and the agglutination phenomenon, letters change forms according to their position in the word (beginning, middle, end and separate).

Table 1.1: different shapes of letter "ع" depending of its position in the word

Beginning	Middle	End	Separate
عـ	ـعـ	ـع	ع

- Plural form of irregular nouns (broken plural). In this case, a noun in plural takes another morphological form different from its initial form in singular.
- There is no space between a word and its prefix, postfix and pronoun; that makes the boundary between the word and the preposition invisible.
- The same word may have more than one meaning in different contexts, for example the word "ذهب"may refer to the word "gold" or "go" depending on the diacritics.
- Many words can refer to the same meaning that may lead to information mismatch in search process, example " ظهر – برز –بان".
- Arabic words may change according to their case modes (nominative, accusative or genitive); " مفاوضون – مفاوضين ".

1.3.2 Arabic Corpus Problem

Text data mining is a multidisciplinary field involving information retrieval, text analysis, information extraction, clustering, categorization and linguistics. Text mining is becoming of more significance, and efforts have been multiplied in studies to provide for fetching the increasingly available information efficiently .

Due to the Arabic language lacking of corpora, it is difficult to represent textual content and quantitative data of Arabic [6][7]. Corpus-based approaches to language have introduced new dimensions to linguistic description and various applications by permitting some degree of automatic analysis of text. The identification, counting and sorting of words, collocations and grammatical structures which occur in a corpus can be carried out quickly and accurately by computer, thus greatly reducing some of the human drudgery sometimes associated with linguistic description and vastly expanding the empirical basis [6][7].

Linguistic research has become heavily reliant on text corpora over the past ten years. Due to the increasing need of an Arabic corpus to represent the Arabic language and because of the trials to build an Arabic corpus in the last few years were not enough to consider that the Arabic language has a real, representative and reliable corpus, it was necessary to build such an Arabic corpus to support various linguistic research on Arabic [6][7].

One of the difficulties that encountered this work and other researches in the field of Arabic linguistics was the lack of publicly available Arabic corpus for evaluating text categorization algorithms . [6][7][15][16]Arabic corpus problem was posed by [6][7][15][16]. A survey by [6][7] confirms that existing corpora are too narrowly limited in source-type and genre, and that there is a need for a freely-accessible Corpus of Contemporary Arabic (CCA) covering a broad range of text-types.

There are rarely successful trials in compiling Arabic corpora, therefore, the third axis presents the technical design of the International Corpus of Arabic (ICA), a newly established representative corpus of Arabic that is intended to cover the Arabic language as being used all over the Arab world. The corpus is planned to support various Arabic studies that depends on authentic (اصيلة) data, in addition to building Arabic Natural Language Processing Applications. International Corpus of Arabic (ICA) is a big project initiated by Bibliotheca Alexandrina (BA).

BA is one of the international Egyptian organizations that play a noticeable role in disseminating culture and knowledge, and in supporting scientific research. ICA is a real trial to build a representative Arabic corpus as being used all over the Arab world to support research on Arabic [6][7] .

ICA corpus has been analyzed by Al-Nasry et. al. in [7], they shed light on the levels of corpus analysis e.g. morphological analysis, lexical analysis, syntactic analysis and semantic analysis. Al-Nasry also demonstrates different available tools for Arabic morphological analysis (Xerox, Tim Buck walter, Sakhr and RDI).

The morphological analysis of ICA includes: selecting and describing the model of analysis, pre-analysis stage and full text analysis stages. ICA is not publically available now and it expected to be released soon.

1.4. Research Motivation

Creating good stemming rules for the Arabic language comes from the importance of Arabic language as the sixth most used language in the world. Stemming is very important in information Retrieval, data mining, language processing. Many linguistic and light stemmers have been developed for Arabic language but still there are many weakness and problem like the absence of morphological rule, which helps to determine the correct affixes in the word, the irregular words, the broken plurals and the use of full root dictionary to extract the root. The main objective of this thesis is to propose a stemmer for Arabic language that solves all of the above mentioned problems.

1.5. Research Contribution

This thesis will contribute with the following:

1. Developing a new efficient stemmer depending on rule based techniques, show the effects of normalization and tokenization into stemming techniques.
2. Discrimination between nouns and verbs, In noun uses light stem but in verb uses root stem.
3. Adding some rule to stem plural noun.
4. Allow developers to add or modify to the stemmer as it is an open sources environment.
5. Adding the proposed stemmer to one of the most famous information retrieval platforms WEKA.

1.6. Thesis Structure

The rest of this thesis is organized as follow:

Chapter 2: Introduce the related work and applying 3 types of stemmers.

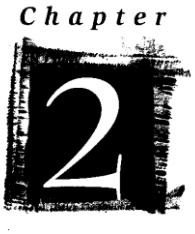
Chapter 3:Introduce the text classification with example for classification.

Chapter 4: Describe the methodology including the proposed stemmer "Shaqalieh stemmer" and morphological analysis and applied proposed stemmer on WEKA.

Chapter 5: Introduce different data set that may be uses to extract the result of stemmers.

Chapter 6: Show the results of the proposed stemmer.

Chapter 7: The conclusion of the research which will summarize the research.



Chapter 2 : Related Work

2.1. Stemmers

Many researches have been worked on building stemmers to support text classification in many language like English and European Language such as French, Spanish, German and in Asian languages such as Chinese and Japanese. That greatly affect the quality of information retrieval. However, Researches on Arabic language are quit limited. When categorizing text documents, not all features equally represent the documents semantics. In fact, some of these features may be redundant and add nothing to the meaning of the document. Others might be synonymous and therefore capturing one of them is enough to enhance the semantic for categorization purposes. Consequently, the effective selection of features words, which reflect the main topics of the text, is an important factor in text classification.[1][2][3][4] [5] [6][7][8][9][10] [20] [21].

Stemming techniques can be used in Arabic text preprocessing to reduce multiple forms of the word to one form (root or stem). Stemmers are commonly designed for each specific language. Stemmers design requires some linguistic expertise in the language itself. Many stemmers have been implemented for many languages including Malay, Latin, Indonesian, Swedish, Dutch, German, Slovene, Bulgarian and Turkish. Stemmers can be classified into table Lookup, Linguistic, and Combinational approaches.

2.1.1. Lookup Approach

Table Lookup approach utilize huge list that store all valid Arabic word a long with their morphological decompositions. This method does not use stemming process. For a given Arabic word, it access the list and retrieve the associated root/stem. Consequently, the stems obtained are guaranteed to be highly accurate. However, the a

availability of such a table that should include all the language words is practically impossible.

2.1.2. Linguistic Approach

Linguistic approach, attempts to simulate the behavior of a linguistic by considering Arabic morphological system and thoroughly analyzing Arabic words according to their morphological components. In such approach, prefix and suffix of a given list of affixes.

2.1.3. Combinational Approach

Combinational approach, a given word is used to generate all combinations of letters. These combinations are compared against predefined lists of Arabic roots. If match, stem and patterns are extracted.

2.2. Applying Root Stemmer on Arabic Text

Root is extracted from the word by means of morphological analysis. It attempts to restore original root of a word and group words accordingly. The basic two 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. This is accomplished by rule-based techniques, table Lookup [22], or by a mixture of two.

Very little research has been carried out on Arabic text. The nature of Arabic text is different than that English text and others languages, and preprocessing of Arabic text is more challenging stage in text classification (TC) particularly and Text Mining (TM) generally. The effect of the preprocessing tools on Arabic TC is one area of research.

Stemming can be defined as the process of removing any affixes(prefixes, infixes, or/and suffixes) from words to reduce these words to their stems or roots. A root can be defined as a word that cannot be created from other word, in other words, a word without prefixes, infixes, or suffixes. For example, the root of the Arabic word(الموحدون) is (وحد). While a stem is simply defined as a word without a prefix or/and suffix. For example, the stem of the Arabic word(الموحدون) is (موحد).

Uses morphological analysis to extract the root of a given Arabic word. Many algorithms have been developed for this approach.

Al-Fedaghi and Al-Anzi [23] algorithm tries to find the root of the word by matching the word with all possible patterns with all possible affixes attached to it.

Al-Shalabi [24] [25] [26] morphology system uses different algorithms to find the roots and pattern. This algorithm removes the longest possible prefix, and then extracts the root by checking the first five letters of the word. This algorithm is based on an assumption that the root must appear in the first five letters of the word. It analyzes Arabic patterns by grouping patterns with similar length sizes. Then extract general rules for each group, which presents the possible positions of letters in "" in order to consider them as excessive and thus remove them. does not use any dictionary, instead of removing prefixes and suffixes and using rules and pattern matching to extract the root.

Khoja [27] [28] stemmer is one of earliest techniques developed for root-based stemmers, has developed an algorithm that remove prefixes and suffixes, all the time checking that it's not removing part of the root and then matches the remaining word against the patterns of the same length to extract the root. it uses two dictionaries, one to match remaining letters against Arabic patterns, and the second is to confirm the correctness of the root.

Taghva, Elkoury and Coombs [29] developed a root-based Arabic stemmer, where Taghva stemmer does not depend on an Arabic root dictionary as Khoja Arabic stemmer. The performance of Taghva stemmer is equivalent to Khoja stemmer. Taghva stemmer is dedicated to non-vowelized Arabic text, since its starts by removing the short vowels(diacritical marks). Taghva, Elkoury and Coombs study deduced that using Arabic root dictionary has no effect on improving Arabic monolingual document retrieval. Also they conclude that building complicated effective Arabic stemmers will not lead to enhancing the effectiveness of the information retrieval.

Al-Kabi and Al-Mustafa [30] presented a new root-based stemmer. As its predecessors, this stemmer starts with normalization process which sometimes leads to conversion in the meaning of the normalized word. Stemmer presented in Al-Kabi and Al-Mustafa used a number of Arabic affixes less than those used by others like Larkey.

Ballesteros and Connell [31] Although this algorithm used more than 440 different Arabic verb patterns to extract the Arabic root, which is a high number of patterns,

but the evaluation tests conducted on this stemmer by [32] showed that its effectiveness is behind the Arabic stemmers presented in [30] [33] [34].

Al-Sarhan, Al-Shalabi and Kanaan [34] present a noval Arabic stemming algorithm which does not depend on Arabic verb patterns, and its depend on mathematical computation to extract the Arabic root of the inputted word. Also Al-Serhan and Ayesh [35]presented a nother Arabic stemming algorithm which uses neural networks.

Ghawanmeh et al [33] refer to the main problems facing modern Arabic stemmers. Their stemmer lack the capability to distinguish between those letters lay at the beginning or the end of the word and constitute part of its roots. Tests conducted on this algorithm show it's a relatively powerful than others [32].

Momani and Faraj [36] [37] presented another novel Arabic root-based stemmer to extract tri-literal Arabic roots with a 73% accuracy using more than 1500 Arabic words. They presented their algorithm with examples which is really interesting, but more tests have to be conducted to verify the effectiveness of their stemmer. It filters rootless words, then remove suffixes and prefixes. It removes excessive letters "" only of it occurs more than once in a word.

Sonbol [38] uses a rule-based technique to extract roots by dividing letters to a part of root, and others are further divided into sub-groups which are examined with well-defined rules to extract the final root, with no use dictionary.

Kchaou [39] uses two dictionaries, for roots and radicals. The main idea is to preserve the semantic of a word by using radical, which if changed, it will change the meaning of a word. Furthermore, an affix matching table is presented to check if a certain prefix, suffix, and infixes occurs in the same word or not. This is done to ensure correct affix removal.

Al-Serhan [19] uses back propagation neural network to extract five letters Arabic words roots. Encoded with binary digits, four types of input were created, one depends on the original letters, and the other three, classify the letters of "سألتمونيها" according to their frequency of occurring as an affix letters.

2.3. Applying Light Stemmer on Arabic Text

The key problem of the root detector algorithm in information retrieval is that many word variants do not have similar semantic interpretations. Although these words are different in meaning, they originate from one identical root. Thus, the root-based retrieval increases word ambiguities.

Inflected and derived words can have a strong impact on the retrieval effectiveness of any information retrieval system. Therefore, it is important to recognize the variants of word morphemes in highly inflected language such as Arabic. Word-sense disambiguation is essential to improve any Arabic information retrieval system.

Our main motivation is to develop a new light stemmer to minimize the sense ambiguity associated with the root-based retrieval. Stemming has multiple definitions. Shereen Khoja's definition [27] limits stemming for Arabic language to the root extraction process.

She has defined the stemming process as "Stemming is the process of removing all of a word's affixes to produce the stem or root. In Arabic this means the removal of prefixes, suffixes and infixes. The stemming component is the rule-based part". However, Leah Larkey [37] was more general in her definition.

She could fetch more techniques under the stemming umbrella. She defined stemming processes as "The researcher use the term stemming to refer to any process which conflates related forms or groups forms into equivalence classes, including but not restricted to suffix stripping ". This definition considers more stemmers than Khoja definition. For example, light stemmers and statistical n-gram methods to conflate words to same class are considered stemmers as well as stemmers that extract roots.

A close definition to Larkey's one is Al-Sughaiyer and Al-Kharashi definition. They defined the stemming as, " Stemming is a method of word standardization used to match some morphologically related words. The stemming algorithm is a computational process that gathers all words that share the same stem and have some semantic relation". There were also a lot of attempts to classify the existing stemmers.

Abdusalam Nwesri [45] has classified the stemmer in to heavy stemming, or root-based stemmer, and light stemming. Heavy stemming usually starts by removing well-known prefixes and suffixes. It aims to return the actual root of a word. Light stemming stops after removing prefixes and suffixes, and does not attempt to identify the actual root. And he further categorizes the light stemmers into three categories according to the way in which existing stemmers deal with particles; conjunctions and prepositions. He also mentioned that a stemmer can combine between any of these approaches:

- Match and Truncate (MT): the beginning of a word is removed if a match happens and the remaining words more than 3 letters length.
- Remove and Check (RC): the beginning of a word is removed if a match happens and the remaining word exists in the document collection.
- Remove With Other Letters (RW): removing a combination of particles and the definite article ال like فال, وال, كال and بال

Larkey [37] divided the Arabic stemmers into four classes:

- Manually constructed dictionaries
- Algorithmic light stemmers; which remove prefixes and suffixes
- Morphological analyses which attempt to find roots
- Statistical stemmers, which group word variants using clustering techniques.

Other classification is done by Al-Sughaiyer, Al-Kharashi and Al-Hajjar.

Al-Stem Stemmer [40] A stemmer developed by Kareem Darwish and modified by Leah Larkey from University of Massachusetts and further modified later by David Graff form LDC. It is intended for research purposes only. The original stemmer of Kareem Darwish removes the following prefixes: (وال , با , لا , فا , وا , في , لي , وي , لل ال , فم , كم , وم , لم , بم , نت , ست , وت , مت , لت , يت , بال , فال) and removes the following suffixes: (ات , وا , ون , وه , ان , تي , ته , تم , كم , هم , هن , ها , ية , تك , نا , ين , يه , ة , ي , ه,). In David Graff version of the stemmer the stemmer two additional prefixes are removed (تت , سي) and two additional suffixes are removed; (تا , ا). Kareem Darwish claims that the Al Stem is more Aggressive than Light10 stemmer.

Graff version of the stemmer has two modes for normalization light and aggressive.

The stemmer works as follows:

- Remove the following prefixes if exist from beginning of the word in the next order from right to left: (وال , فال , بال , بت , يت , بت , لت , مت , تت , ست, ال , لل, وي , لي , سي, في , وا , نا , لا , با , نت , بم , لم , وم , كم , فم)
- Remove the following suffixes if exist from the end of the word in the next order from right to left: (ات , وا , ون , وه , ان , تي , ته , تم , كم , هم , هن , ها , ية , تك , نا , يه , ة , ه, ي , ا , , ين)

Aljlayl Stemmer [41]

Mohammed Aljlayl developed a light stemmer used for his own information retrieval researches in TREC cross-language track. Aljlayl didn't mention the prefix or suffix list going to be removed from word rather he mentioned only that " to remove the most frequent suffixes and prefixes" then he said "The most common suffixation includes duals and plurals for masculine and feminine, possessive forms, and pronoun forms," and "The definite articles and prefixes that can be attached to the head of the definite article are considered the most common prefixes. In addition, the letter (و) is a commonly used letter to start the sentences within the Arabic language.

The algorithm of stripping the affixes is as follow:

- If word length is greater than or equal 3 characters, then remove the prefix و
- Remove the article from the beginning of the word if exist then normalize آ , ٠, أ from the beginning of the word to ا
- If the length of the remaining stem is greater than or equal 3 characters, then remove the suffixes form the stem using longest first strategy (remove the longest suffix first) only if remaining part of the stem is greater than or equal 3 characters.
- While length of the remaining stem is greater than 3 characters do,
    - Remove the prefixes form the stem only if remaining part of the stem is greater than 3 characters.

- Return the stem

Light8 Stemmer [42]

It is a light stemmer developed by Leah Larkey for the purpose of researching. The construction of the stemmer based on heuristics; try to remove strings which would be found as affixes far more often than they would be found as the beginning or end of an Arabic word without affixes. The stemmer removes the following prefixes: (ال , وال , كال, فال, و) and the following suffixes: (ة, ه , ية, يه , ين , ون , ات , ان, ها ي)

The stemmer works as follows:

- Remove و if the remainder of the word is 3 or more characters long.
- Remove any of the definite articles if this leaves 2 or more characters.
- Remove any of the following suffixes in order form right to left (ون , ات , ان, ها ي, ة , ه , ية , يه , ين,) if this leaves 2 or more characters.

Light10 Stemmer [43]

Light8, which has been developed by Leah Larkey, becomes Light10 after some modifications. Light10 was designed to strip off strings that were frequently found as prefixes or suffixes, but infrequently found at the beginning or ending of stems without intended to be exhaustive, as light8 did before. Light10 tries to improve the Information Retrieval (IR) performance. Larkey used heuristic as a strategy for developing here stemmer. And it did it. It outperforms most of the morphological analyzers in that time; Amira 1.0, Khoja, Buckwalter morphological analyzer, etc. This is important because some researcher claims that Light10 is not good because it doesn't return the right form of the word. The stemmer removes the following prefixes: (ال, وال, بال, كال, فال, لل, و) and it removes the following suffixes: (ها, ان , ات , ون , ين , يه , ية , ه , ة, ي). The Light10 stemmer removes the same set of suffixes as Light8. However, Light10 add (لل) to the prefix list to be removed. This addition made Light10 outperform Light8. Something that is notable in Larkey stemmers, Light8 and Light10, that they only remove definite articles. The stemmers don't remove any Arabic prefixes from words. Mohamed I. El-Disooqi, Waleed M. Arafa & Kareem M. Darwish

Light10 stemmer works as follows:

- Remove و if the remainder of the word is 3 or more characters long.
- Remove any of the definite articles if this leaves 2 or more characters.
- Remove any of the following suffixes in order form right to left (ها, ان , ات, ون , ة, ي , ه , ية , يه , ين) if this leaves 2 or more characters.

SP_WOAL Stemmer [69]

Al Ameed has reviewed multiple stemmers used in TREC 2001 and 2002 cross-language track. He reviewed the Al-Stem, Light8 stemmer and another stemmer he called it TREC-2001 stemmer; which is a modified version of Larkey's Light8 stemmer. Then he decided to enhance the performance of these stemmers in two ways. First enhancement is done by adding new affixes to the existing affixes of the mentioned stemmers. The second way is by modifying the sequence of algorithm components execution. Although the author was intending to develop a stemmer to improve the performance of information retrieval tasks, he didn't conduct any IR evaluation. He also said that his stemmer is much better than the stemmers that have developed for the TREC cross-language track claiming that his stemmer produces much more correct words than other stemmers and he neglected that it doesn't depend only on the correctness of the words to make an efficient retrieval process. The enhancement produced in SP_WOAL light stemmer. Although the user mentioned his prefixes list contain 17 two-characters, the list contains only 15. However, it contains 5 single-characters prefixes rather than 3. The stemmer removes the following prefixes (ال, وال, بال, كال, لل, ولل, ب, ل, فا, ست, با, ي, سي, لت, ت, لي, في, وبال, ن, كا, وست, لن, فت, وسن, فن, وسا, ولا, ولي, سا, سن, ولت, ولن, وسي) and removes the following suffixes (ا, ت, هن, ك, ته, تك, تن, و, ن, كن, تا, ما, يا, في, ين, ون, ات, ان, ي, ه, ها, هم, ة, يه, كم, نا, وا, تم). The stemmer is considered very aggressive stemmer. The stemmer works as follows:

- Remove the prefix لا from the beginning of the word
- Recursively remove the suffix from the end of the word starting with longest suffixes first.
- Non-recursively removes the prefix form the beginning of the word starting with longest prefixes first.

Berkeley light stemmer [17]

In 2002, University of California at Berkeley participated only in the cross-language track in TREC conference. They developed a light stemmer that made them among the best performers in the track. They used the standard Arabic data collection provided by Linguistic Data Consortium LDC to develop their light stemmer; they choose the affixes with the most frequently occurrence and that give highest performance when practically evaluated using the test collection. The stemmer removes 26 prefixes and 22 suffixes during the stemming processing. The list of the prefixes that should be removed is: (ال, وال, فال, كال, و, لل, وللل, ب, ل, فا, با, سي, وم, وت, وي, واا, لا, وب, ول, وس, كا, مال, االا, سال, لال) and the list of the suffixes that should be removed is: (ون, ات, ان, ي, ه, ها, هم, ة, ية, كم, نا, وا, تم, ت, هن, تن, كن, ما, يا, ني, ين). The Berkeley light stemmer works as follows:

1. If the word is at least five-character long, remove the first three characters if they are one of the following: (مال, االا, سال, لال, وال, بال, فال, كال, وللل).
2. If the word is at least four-character long, remove the first two characters if they are one of the following: (لل, فا, با, سي, وم, وت, ال, وي, وا, لا, وب, ول, وسي, كا)
3. If the word is at least four-character long and begins with و, remove it.
4. If the word is at least four-character long and begins with either ب or ل, remove ب or ل, only if, after removing the initial character, the resultant word is present in the Arabic document collection.
5. Recursively strips the following two-character suffixes in the order of presentation if the word is at least four-character long before removing a suffix: (ون, ات, ان, ين, تن, تم, كن, كم, هن, يا, ني, يا, وا, ما, نا, هم, ية, ها)
6. Recursively strips the following one-character suffixes in the order of presentation if the character is at least three-character long before removing a suffix: (ت, ي, ه, ة).

Kadri's linguistic-based stemmer [44]

The developing of the Kadri's linguistic-based stemmer depended on idea that the Arabic word consists of five part their order is; antefixes, prefixes, stem, suffixes and postfixes. The first part which is the antefixes is the prepositions and conjunc-

tions. However, the prefixes are the conjugations person of verbs. The suffixes are Termination of conjugation and numbers marks of the nouns. The postfixes are the pronouns that catch up with end of the word. The list of Antefixes is: وبال, وال, بال, كال, ولل, ال, وب, ول, لل, فس, فب, فل, وس, ك, ف, و,ب, ل and the list of prefixes is ت, ي, ن, ا. The list of suffixes is: تم, تا, وا, ين, ون, ان, ات, تان, تين, يون, تما, و ,ي, ا, ن, ت, نا, تن. And the list of postfixes is: ي, ه, ك, كم, هم, نا, ها, تي, هن, كن, هما, كما

The linguistic-based stemmer has two phases to work:

1. Training Phase:

   - A list of stems with its frequency occurrence is build for each word using corpus to avoid ambiguity that my happen when removing affixes.

2. The Stemming Phase:

   - The stemmer truncates possible affixes according to the above table.
   - If there an ambiguity raised for the stemmer (more than one combination was available), then stemmer selects the most appropriate candidate; according to corpus statistics computed in the training phase.

Restrict Stemmer [45][46]

Nwesri [45] focused on removing conjunctions, و, and ف, and prepositions, ل, و, ك, and ب, that come as prefixes in the beginning of the words. He (only in 2005) didn't mention other affixes such as articles and suffixes in general. He just tried to find a way to recognize the two types of affixes; the conjunctions and the prepositions. In [46] he developed an Arabic stemmer called Restrict. Its main idea is to retaining valid Arabic core words. This because he claimed that removing wrong affixes sometimes results in incorrect stem and in most cases reduces retrieval precision by conflating different words to the same class. Nwesri used in his proposed technique two things to improve his performance. The first was the Microsoft Office 2003 Arabic spellchecker to ensure that he extracted only correct words. The second was simple rules or heuristics exist in Arabic language to guarantee the correctness of the affixes removal. Although these rules don't guarantee correctness of hundred percent, they improve the information retrieval performance. The rules are removing a prefix keep the remaining word correct, adding the

waw or faa conjunction keeps the modified word correct, altering the a prefix with waw and faa keeps the modified word correct, and duplicating a particle result in a wrong word except for the lam case. He has a good justification for depending on correctness of words for improving the IR performance as follows, "Although correct words are not the main target of stemming, an incorrect stem can have a completely different meaning and correspond to a wrong index cluster."

The algorithm work as follows:

i. Dealing with لل, prefix:
   a) If the word is correct after removing the prefix لل , then remove it.
   b) Otherwise, The researcher add the letter ا, before the word, if the new word is correct The researcher drop one lam from the original word.
ii. Dealing with ل, particle when precedes definite article لا:
   a) The researcher replace the first lam with the letter ا, if the word exists in the lexicon remove the prefix without check lexicon (he depends on that the words that start with lam cannot preceded by Alef)
   b) The researcher remove the first letter and check to see whether The researcher can drop the first lam.
iii. (Here could be one of the three algorithms suggested by Nwesri)
iv. If a word starting with either waw or faa and after stemming has three or more characters that has either waw, kaf, baa, or lam as its first character
v. He suggested three algorithms to handle the conjunctions and the prepositions. All the algorithms depend on checking the words in the lexicon after removing the first letter as follows:
   - Remove and Check in Lexicon (RCL)
     - The prefix of a word is removed if the remaining word exists in the Arabic lexicon.
   - Replace and Remove (RR)
     - Remove the prefix and check the remaining word in lexicon
     - If exist, produce to instances of the remaining word by appending waw and faa to the beginning of the word and check them if they correct words

- If both of the new instances are correct then the prefix is removed
- Otherwise the original word is returned

- Replicate and Remove (RPR)
  - Remove the prefix and check the remaining word in the lexicon
    - ❖ If the word not exist go to the duplicate step
    - ❖ If the word exist return the original word
    - ❖ Duplicate the initial letter except the lam and check the new word in the lexicon
  - If the word exist, return the original word
  - Else, remove the prefix
  - For the words start with the letter lam, The researcher add both baa and kaf instead of replicating them
    - ❖ If both new instances are incorrect, The researcher remove the first lam.
    - ❖ Else keep the original word.

Stemming algorithm:

- Dealing with لل, prefix:
  - Replace the prefix لل, with ل, if the new word is in the lexicon remove the first ل
  - Else return the original word
- Use the RPR method to remove the conjunctions and prepositions.
- Remove the definite article لا, from the beginning of the word
- If the word starts with س ,ي ,ت ,ن and ا, generate two instances of the word by adding ك, and لا, to the beginning of the word, if either of the new words exist in the lexicon
  - Return original word
  - Else, remove the starting letter
- Repeat the previous step until the condition is not longer exist
- Remove the suffixes هن, هما, هم,ها
- If the word ends with ان, replace it with ين, and remove it only if the word exists in lexicon.
- If the word ends with ات, replace the suffix with ة only if:
  - Removing the suffix produces a word that exists in the lexicon, or

    - Replacing the suffix with ة, produces a word that exists in the lexicon
- If the word ends with ين, remove the suffix only if:
    - Replacing the suffix with ان, produces a word that exist in the lexicon, or
    - Replacing the suffix with ون, produces a word that exist in the lexicon
- If the word ends with ون remove the suffix only if replacing the suffix with ين, produces a word that exists in the lexicon,
- Else, remove it if the word start with ي, or سي,
- Remove the suffixes ة, ه, يه, ية, وا
- If the word ends with ي, remove the suffix only if:
    - Removing the suffix produces a word that exists in the lexicon, or
    - Replacing the suffix with ها, produces a word that exist in the lexicon, or
    - Replacing the suffix with ه, produces a word that exist in the lexicon

One limitation of his method is that it needs a lexicon contains all the forms of all the words in Arabic language which is very difficult to obtain. He used Microsoft office 2003 proof kit as resource of Arabic words. It contains about 15,500,000 Arabic words.

Beltagy Stemmer [47]

Samhaa El-Beltagy and Ahmed Rafea have proposed a stemming technique that not only removes prefixes and suffixes from the beginning and the end of the word, but also converts the irregular plural form of the word to its singular form. The stemmer also is a domain specific stemmer which conducts stemming according to the domain of the collection of text to be indexed. The domain specific idea is implemented using a stem list that contains the words and their stems. So, before accepting a stem that produce from a word using stem-based stemmer, the system check whether the produced stem exist in the list or not. This idea is helpful since words could have different stems on different domain. For example, the word "دقيقة" could have the following different meanings: minute and very fine. In the first sense, stemming is going to harm the word. However, stemming will be good choice for the second meaning. For simplicity, The researcher refer to the stemmer as Beltagy Stemmer. The stemmer first has to be built or trained then it can be used for stemming. The construction of the stemmer is done using a subset of the documents to be stemmed. The construction done as follow:

- A subset of documents from the corpus to be stemmed is selected for the stem list building.
- Through a user interface, the user checks the stem list to verify its correctness.
- The user can provides stems he wanted for specific words that he wants to stem them in a particular way.

In the second phase (operational phase),

- Stemming can be done by checking whether the possible stem exists in the stem list or not.
- The stemmer has two modes; restrict mode (the original word is returned if the word does not exist in the stem list or the corpus) and light stemming mode (the stem rather than the original word is returned if the word I not exist in the stem list or the corpus).

Stemming algorithm is done in the two phases. In the training phase the stemmed word checked only in the corpus, but in the stemming phase the word checked in the stem list and the in the document:

- Remove the following prefixes if the length of the remaining word is greater than or equal 2: ال, وال, بال, كال, فال, ولل, وكال, وفال
- Remove the prefixes و, ك, ف, ل, ب, only if the remaining word exists in the stem dictionary or in the input document collection.
- Remove the prefixes لا, if the length of the remaining word is greater than or equal 2
- Remove the suffixes يا, ات, اته,ت only if
  - The remaining word exists in the stem dictionary or in the input document collection, or,
  - Adding ة, to the remaining word and the modified word exist in the stem dictionary or in

the input document collection,

- Remove the suffixes ها, ون, وا, ين, ان, ية, يه, ي, ه, ة only if
  - The remaining word exists in the stem dictionary or in the input document collection, or,

- If the remaining word ends with ت, replace the ت, with ة and check if it exists in the stem dictionary or in the input document collection,

There are two limitations for this stemming technique. First, the stemmer has to be used with a specific domain. It cannot be used as a general stemmer. This means that there is no sense to be compared with the Light10 stemmer. The second limitation is the stem list which is built during the construction phase has to have the user intervention to edit the mistakes done by the stemmer.

2.4. Applying Statistic Stemmer on Arabic Text

Statistical techniques have widely been applied to automatic morphological analysis in the field of computational linguistics. For example, Gold Smith finds the best set of frequently occurring stems and suffixes using an information theoretic measure [48].

Oard et al. Consider the most frequently occurring word-final n-grams (1,2,3 and 4-grams) to be suffixes [49]. Although such systems can be used on many different languages, they cannot be expected to perform well on language like Arabic in which suffixing is not the only inflectional process.

Mayfield et al. have developed a system which combines word-based and 6-gram based retrieval, which performs remarkably well for many languages [50]including Arabic [51].

Al-Fares and De Roeck [52] used clustering on Arabic words to find classes sharing the same root. Their clustering was based on morphological similarity, using a string similarity metric tailored to Arabic morphology, which was applied after removing "a small number of obvious affixes". They evaluated the technique by comparing the derived clusters to "correct" classes. They didn't assess the performance in an information retrieval context.

Mustafa and Al-Radaideh [16] (have examined the performance of N-matching techniques for Arabic word-based string searching. They addressed the problem of finding morphologically related words in a text using two N-gram strategies: dia-

grams and trigrams. The results showed that diagrams strategy offered a better overall performance, in terms of conflation recall and precision ratios, than the trigrams strategy. However, when their performance values were evaluated using the sign test, the two methods were found significance.

De Roeck and Al-Fares [52] tested N-gram matching to assess its potential for clustering Arabic words sharing the same root. To do so, two strategies were tested: purely conventional N-gram matching and are fined two-stage approach in which potential affixes and weak letters were preprocessed to give relevance to root consonants only. The results gave a strong indication that the two-stage strategy involved improvement over Adamson algorithm. However, the small size of data sets used in the experiment should be considered as a limiting factor in these results.

Mayfied, Mc Namee, Costelb, Piakto and Banerjee [51] investigated the use of N-grams for Arabic retrieval in TREC 2001. They found that N-grams with N=4 were most effective. Using the same Corpus, this work was continued in TREC 2002 by McNamee, Piatko, and Mayfield with some modifications on the original research setup and tokenization. Plain 4-grams did quite well, but the hybrid scheme involving the combination of words plus N-grams of length 3,4 and 5 was the best performing approach.

Using the TREC Arabic corpus as a test bed, X4 et al. evaluated a number of search strategies for the retrieval of Arabic documents. Experimental results showed that spelling normalization and stemming could significantly improve Arabic monolingual retrieval. The best results were obtained with trigrams suggestion that bigrams carry too little contextual information while 4-gram and longer ones simply simulate word or stem-based retrieval.

Darwish [53] explored the effectiveness of OCR-based information retrieval using different Arabic index terms including three categories: character N-grams, terms obtained through morphological analysis, and combination of both. The results indicated that N-grams of length 3 and 4, and combinations of N-grams with lightly stemmed words were well suited for OCR-degraded Arabic text retrieval.

Probabilistic structured methods were shown in figure 2.1 is to improve N-gram matching. The same conclusion, in respect to N-grams of length 3 and 4, was pointed out earlier by Darwish and Oard.

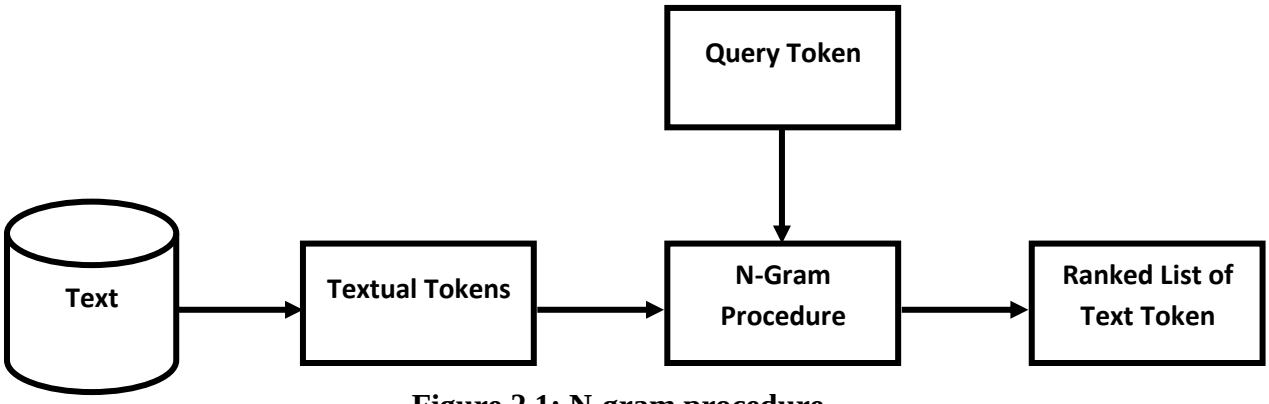


Figure 2.1: N-gram procedure

An N-gram [54] is an N-character slice of a string. The N-gram method is language independent and works well in the case of noisy-text (text that contains typographical errors). The researcher used tri-grams for text classification. The tri-grams of a string or token is a set of continuous 3-letter slices of the string. For example, the tri-grams for the word

[7]المودعين are :الم, لمو, مود, ودع, دعي , عين .In general, a word of length w has w – z tri-grams According to Zipf;s law [55]:

"The nth most common word in a human language text occurs with a frequency inversely proportional to n".

This has the implication that documents belonging to the same class or category will have similar N-gram frequency distributions.

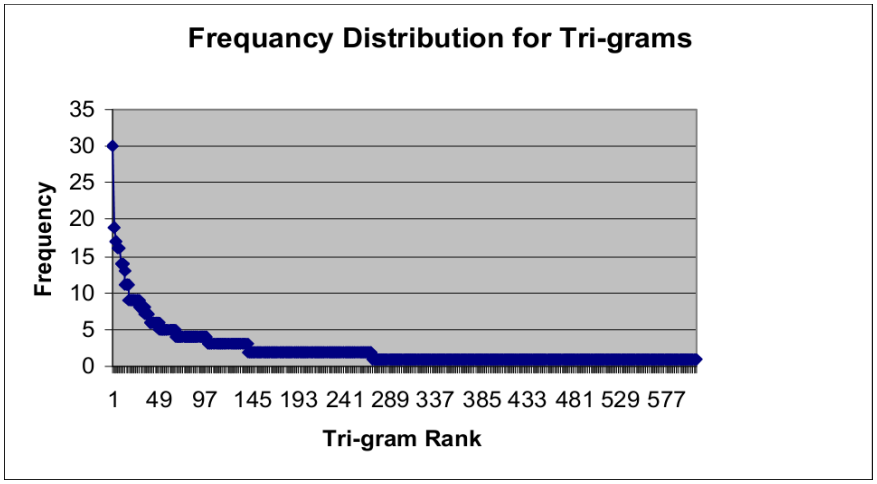


Figure 2.2: Frequency Distribution of Tri-gram

Figure 2.2 shows the Tri-gram frequency distribution for a text document belonging to the sports category from our corpus. It clearly shows that the frequencies of the most common Tri-grams are inversely proportional to their rank.

Chapter

3

Chapter 3 : Background

This chapter describes famous TC algorithms: Support Vector Machines (SVMs), K Nearest Neighbors (KNN), Decision Trees (DT), Naive Bayes (NB), and Naive Bayes variants (Multinomial Naive Bayes (MNB), Complement (CNB), and Discriminative Multinomial Naive Bayes (DMNB)). The followings are brief overview on the classification algorithms mentioned above.

The goal of classification is to build a set of models that can correctly predict the class of the different objects. The input to these methods is a set of objects (i.e., training data), the classes which these objects belong to (i.e., dependent variables), and a set of variables describing different characteristics of the objects (i.e., independent variables). Once such a predictive model is built, it can be used to predict the class of the objects for which class information is not known a priori. The key advantage of supervised learning methods over unsupervised methods is having an explicit knowledge of the classes [30] [56] [57][58].

3.1. Naive Bayes (NB)

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naïve) independence assumptions [30] [56] [57][58]. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple [30] [56] [57][58] .

3.2. Naive Bayes Multinomial (NBM)

The multinomial model of naive Bayesian classification algorithm captures the word frequency information in document. So it requires the word frequency that is not weighted and normalized [46] [57][58].Using multinomial probabilistic model as a Bayesian assumption tries to overcome the drawback of using multivariate Bernoulli model which represent a text document as a vector of binary attributes indicating which words occur and do not occur in the document. In Bernoulli model, the number of times a word occurs in a document is not captured. When calculating the probability of a document, one multiplies the probability of all attribute values, including the probability of non-occurrence for words that do not occur in the document [46][58].

On the other hand, multinomial probabilistic model is a uni-grams language model with integer word counts. The document is represented by a set of word occurrences from the document. The number of occurrences of each word in the document is captured. When calculating the probability of a document, one multiplies the probabilities of each word that occur.

The individual word occurrences can be understood as event and the document to be the collection of word events. This model is called multinomial event model. This approach is more traditional in statistical language modeling for speech recognition, where it would be called uni-grams language model [46][58].

Naive Bayes is a popular technique for this application because it is very fast and quite accurate. However, this does not take into account the number of occurrences of each word, which is potentially useful information when determining the category of a document. Instead, a document can be viewed as a bag of words a set that contains all the words in the document, with multiple occurrences of a word appearing multiple times (technically, a set includes each of its members just once, whereas a bag can have repeated elements). Word frequencies can be accommodated by applying a modified form of naive Bayes that is sometimes described as multinomial Naive Bayes. [58]

3.3. Complement Naive Bayes (CNB)

This approach use simple, heuristic solutions to some of the problems with naive Bayes classifiers. The approach addresses both systemic issues as well as problems that arise because text is not actually generated according to a multinomial model [59].

One systemic problem is that when one class has more training examples than another, naive Bayes selects poor weights for the decision boundary. This is due to an under-studied bias effect that shrinks weights for classes with few training examples. To balance the amount of training examples used per estimate, a complement class formulation of naive Bayes was introduced by Rennie et. al. [58].

Another systemic problem with naive Bayes is that features are assumed to be independent. As a result, even when words are dependent, each word contributes evidence individually. Thus the magnitude of the weights for classes with strong word dependencies is larger than for classes with weak word dependencies. To keep classes with more dependencies from dominating, the approach normalizes the classification weights [58].

In addition to systemic problems, multinomial naive Bayes does not model text well. Presenting a simple transform enables naive Bayes to instead emulate a power law distribution that matches real term frequency distributions more closely. Rennie et. al. [58] discussed two other pre-processing steps, common for information retrieval but not for naive Bayes classification, that incorporate real world knowledge of text documents (TF transformation, IDF transformation and normalization). They significantly boost classification accuracy.

The improved classification accuracy is worthwhile. Complement Naive Bayes (CNB) classifier made simple corrections to NB and it approaches the accuracy of the Support Vector Machines (SVMs) while being faster and easier to implement than the SVMs and most modern-day classifiers [58].

3.4. Discriminative Multinomial Naive Bayes (DMNB)

Learning Bayesian networks from data has two elements: structure learning and parameter learning. Given a fixed Bayesian network structure, parameters learning can take two different approaches: generative and discriminative learning. While generative parameter learning is more efficient, discriminative parameter learning is more effective. Discriminative Frequency Estimate DFE provides simple, efficient, and effective discriminative parameter learning method which learns parameters by discriminatively computing frequencies from data [60].

Empirical studies of [60] show that the DFE algorithm integrates the advantages of both generative and discriminative learning. DFE performs as well as the state-of-the-art discriminative parameter learning method, gradient descent based parameter learning [61], in accuracy, but is significantly more efficient. The motivation is to turn the generative parameter learning method, Frequency Estimate FE, into a discriminative one by injecting a discriminative element into it. DFE discriminatively computes frequencies from data, and then estimates parameters based on the appropriate frequencies.

The empirical studies show that DFE inherits the advantages of both generative and discriminative learning [60]. In the following, The researcher shall describe frequency estimate, and discriminative frequency estimate.

3.4.1. Frequency Estimate

Let the capital letters X be a discrete random variable. The lower-case letters x is used for the value taken by variable X, and $x$ijrefers to the variable Xi taking on its j^{th} value. Let the boldface capital letters X be a set of variables, and the boldface lower case letters x for the values of variables in X.

The training data D consists of a set of finite number of training instances, and an instance e is represented by a vector (x, c) where c is the class label. In general, the symbol hat‖ to indicate parameter estimates. A Bayesian network encodes a joint probability distribution $P(X, C)$ by a set of local distributions P for each variable. By forcing the class variable C to be the parent of each variable Xi, The researcher can compute the posterior probability $P(Cj|X)$ from eq. 3.1

$$P(Cj|X) = \alpha P(Cj) P(X_i|U_i) \quad 3.1$$

Where α is a normalization factor, and Ui denotes the set of parents of variable Xi. Note that the class variable C is always one parent of Xi. In naive Bayes, Ui only contains the class variable C. $P(C)$ is called the prior probability and $P(Xi|Ui)$ is called the local probability distribution of Xi.

3.4.2. Discriminative Frequency Estimate

Discriminative Frequency Estimate (DFE) is a discriminative parameter learning algorithm for Bayesian network classifiers. When counting a training instance in FE, simply increase the corresponding frequencies by 1. Consequently, The researcher do not directly take the effect on classification into account in computing frequencies. In fact, at any step in this process, The researcher actually have a classifier on hand: the classifier whose local probabilities are computed using the current entries (frequencies) in CPTs [60].

Thus, when counting an instance, The researcher can apply the current classifier to it, and then update the corresponding entries based on how well (bad) the current classifier predicts on the instance. Intuitively, if the instance can be classified perfectly, there is no need to change any entries. In general, given an instance e, The researcher can compute the difference between the true probability $P(c|e)$and the predicted probability) generated by the current parameters, where c is the true class of e, and then update the corresponding entries based on the difference. Furthermore, the FE process can be generalized such that The researcher can count each instance more than once (as many as needed) until a convergence occurs. This is the basic idea of DFE. More precisely, the DFE parameter learning algorithm iterates through the training instances. For each instance e, DFE firstly computes the predicted probability), and then updates the frequencies in corresponding CPTs using the difference between the true $P(c|e)$ and the predicted). Here M is a pre-defined maximum number of steps [60]. $L(e)$ is the prediction loss for training instance e based on the current parameters.

$$L(e) = P(c|e) - p\hat{}(c|e) \quad 3.2$$

In general, $P(c|e)$ are difficult to know in classification task, because the information The researcher have for c is only the class label.

Thus, The researcher assume that $P(c|e) = 1$ when e is in class c in the implementations. Note this assumption may not be held if data cannot be separated completely, and thus may introduce bias to our probability estimation.

3.5. Support vector machines (SVMs)

A support vector machine (SVMs) is a set of related supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two categories, SVMs training algorithm builds a model that predicts whether a new example falls into one category or the other.

Intuitively, SVMs model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [46] [57].

More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high dimensional space, which can be used for classification, regression or other tasks.

Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier [46] [57][58].

SVMs was derived from statistical learning theory by Vapnik, et al. in 1992 [46] [57][58].

SVMs became famous when, using images as input, it gave accuracy comparable to neural network with hand-designed features in a handwriting recognition task. Currently, SVMs is widely used in object detection and recognition, content-based image retrieval, text recognition, biometrics, speech recognition,

speaker identification, benchmarking time-series prediction tests. Using SVMs in text classification is proposed by [47], and subsequently used in [30][84].

The following summarizes SVMs steps:

- Map the data to a predetermined very high-dimensional space via a kernel function.
- Find the hyper plane that maximizes the margin between the two classes.
- If data are not separable find the hyper plane that maximizes the margin and minimizes the (a weighted average of the) misclassifications.

SVMs can be used for both linear and nonlinear data. It uses a nonlinear mapping to transform the original training data into a higher dimension. With the new dimension, it searches for the linear optimal separating hyper plane (i.e., decision boundary). With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyper plane. SVMs finds this hyper plane using support vectors (essential training tuples) and margins (defined by the support vectors)[46] [57][86].

SVMs is effective on high dimensional data because the complexity of trained classifier is characterized by the number of support vectors rather than the dimensionality of the data, the support vectors are the essential or critical training examples, they lie closest to the decision boundary, If all other training examples are removed and the training is repeated, the same separating hyper plane would be found. The number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVMs classifier, which is independent of the data dimensionality. Thus, an SVMs with a small number of support vectors can have good generalization, even when the dimensionality of the data is high [86][56] [57] [26].

3.6. Vector Space Model (VSM) and Term Weighting Schema

The aim of term weighting is to enhance text document representation as feature vector or vector space model (VSM). Popular term weighting schemes are Boolean model (which indicates absence or presence of a word with Booleans 0 or 1 re-

spectively), word count (wc), normalized word count, term pruning, Term Frequency (tf), and Term Frequency-Inverse Document Frequency ($tf - idf$).

Term frequency $tf(t, d)$ is the number that the term t occurred in the document d. Document frequency $df(t)$ is number of documents in which the term t occur at least once. The inverse document frequency can be calculated from document frequency using the formula log(num of Docs/num of Docs with word i). The inverse document frequency of a term is low if it occurs in many documents and high if the term occurs in only few documents. Term discrimination consideration suggests that the best terms for document content identification are those able to distinguish certain individual documents from the collection.

This implies that the best terms should have high term frequencies but low overall collection frequencies (num of Docs with word i). A reasonable measure of term importance may then be obtained by using the product of the term frequency and the inverse document frequency ($tf * idf$) [62] [63][49].

In many situations, short documents tend to be represented by short-term vectors, whereas much larger-term sets are assigned to the longer documents. Normally, all text documents should have the same importance for text mining purposes. This suggests that a normalization factor to be incorporated into the term-weighting to equalize the length of the document vectors [62] [63][49].

3.6.1. Term weighting equation

The term count in a given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term ti within the particular document dj. Thus The researcher have the term frequency, defined as follows [17] [19].

$$tf_{i,j} = \frac{n_{ij}}{\sum_n n_{k,j}} \qquad 3.3$$

Where ni,j is the number of occurrences of the considered term (ti) in

Term Frequency Transformation = $Log(1 + tf_{i,j})$

document dj, and the denominator is the sum of number of occurrences of all terms in document dj. A variation of tf is to apply log transformation to term frequency [17] [19].

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$idf_i = Log\ \frac{|D|}{|\{d_i t_j \in d\}|} \qquad 3.4$$

Where $|D|$ is the total number of documents in the corpus and is number of documents where the term ti appears (that is $nij \neq 1$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use 1+ [19]Then

$$(tf - idf)_{i,j} = tf_{i,j} \times idf_i \qquad 3.5$$

A high weight in $tf - idf$ is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. The $tf - idf$ value for a term will always be greater than or equal to zero [17] [19].

The $tf-idf$ weight (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Variations of the $tf-idf$ weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query [17] [19]. One of the simplest ranking functions is computed by summing the $tf - idf$ for each query term; many more sophisticated ranking functions are variants of this simple model [17] [19].

Suppose The researcher have a set of Arabic text documents and wish to determine which document is most relevant to the query "تبارك الذي بيده الملك" ("Blessed be the King in his hand"). A simple way to start out is by eliminating documents that do not contain all four words "تبارك", "الذي" و"بيده", and "الملك" but this still leaves many documents. To further distinguish them, The researcher might count the number of times each term occurs in each document and sum them all together; the number of times a term occurs in a document is called term frequency. However, because the terms "الذي" is so common, this will tend to incorrectly emphasize documents which happen to use the word "الذي" more, without giving enough weight to the more meaningful terms "تبارك","بيده" and "الملك". The term "الذي" is not a good keyword to distinguish relevant and non-relevant documents and terms like "تبارك","بيده" and "الملك" that occur rarely are good keywords to distinguish relevant documents from the non-relevant documents. Hence an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely [17] [19].

3.6.2. $Tf - Idf$ example

Consider a document containing 200 words wherein the word "تبارك" appears 8 times. Following the previously defined formulas, the term frequency (tf) for cow is then 0.04 (8 / 200).

Now, assume The researcher have 20 million documents and "تبارك" appears in four thousands of these. Then, the inverse document frequency is calculated as $log(20\,000\,000 / 4\,000) = 3.67$. The $tf - idf$ score is the product of these quantities: 0.04 × 3.67 = 0.1468.

The $tf - idf$ weighting scheme is often used in the vector space model together with cosine similarity to determine the similarity between two documents.

Chapter 4 : Methodology

In this research, The researcher propose a new method for stemming to solve many of the ambiguity problems related to light stemming and improve the performance of stemming that benefit for many application like information retrieval, classification, …etc.

Stemmers are basic elements in query systems, indexing, web search engines and information retrieval systems (IRS). Stemming offers the benefits of minimizing storage requirements by eliminating redundant terms, as well as increasing matching probability for document comparison and unifying vocabulary [1].

Unfortunately, stemming can cause errors in the form of over-stemming, mis-stemming and under-stemming. These errors decrease the effectiveness of stemming algorithms [2] however reducing one type of errors can lead to an increase of the other [3].

Over-stemming occurs when two words with different stems are stemmed to the same root. An over-stemming example is when the word "فعَلها" and "فعلها" 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.

Mis-stemming is defined as "taking off what looks like an ending, but is really part of the stem[4] for example, stemming the word "ولدين" to "لد" or the word "كرنك" to "رن".

The challenges associated with stemming are even more pronounced in Arabic. Arabic is one of the most complex languages, in both its spoken and written forms. However, it is also one of the most common languages in the world. The Arabic language exhibits a very complicated morphological structure.

In the first time, The researcher divide the words into noun and verbs this step is very important because the noun when return it to the root The researcher lost valuable information, for example (مكتبة) when return it to the root is (كتب) in the valuable information this differ is very important for information retrieval perhaps towards for Intention user. To distinguish the noun from verb, The researcher base on real information, which is that the preposition does not have to come after noun. The researcher note that a sub list of stop words preceding verbs and a sub list of stop words preceding nouns.

Stop words (functional words or structural word list[1]) are words that either carry no meaning or are very common[22] thus do not represent the document. Stop words list usually contains prepositions, pronouns, and conjunctions. Text processing often performs stop words removal early in the process, although there is currently no standardized list of Arabic Stop Words. The current available Arabic stop words list [9]introduces less than 200 words.

The researcher was able to define more than 2,200 stop words and categorize them into "useful" and "useless" stop words. Useless stop words are stop words that are used extensively and give no benefits to the subsequent words. On the contrary, useful stop words are words that can indicate the syntactical categories of the subsequent words. Unfortunately, due to the early removal of the stop words, this valuable information is lost. The researcher believe that the useful stop words can help us identify nouns and verbs and direct us into the appropriate stemming.

4.1. Stemming Algorithm(Shaqalieh Stemmer)

Figure 4.1 shows the general diagram of the proposed Shaqalieh stemmer. It consists of 8 steps.

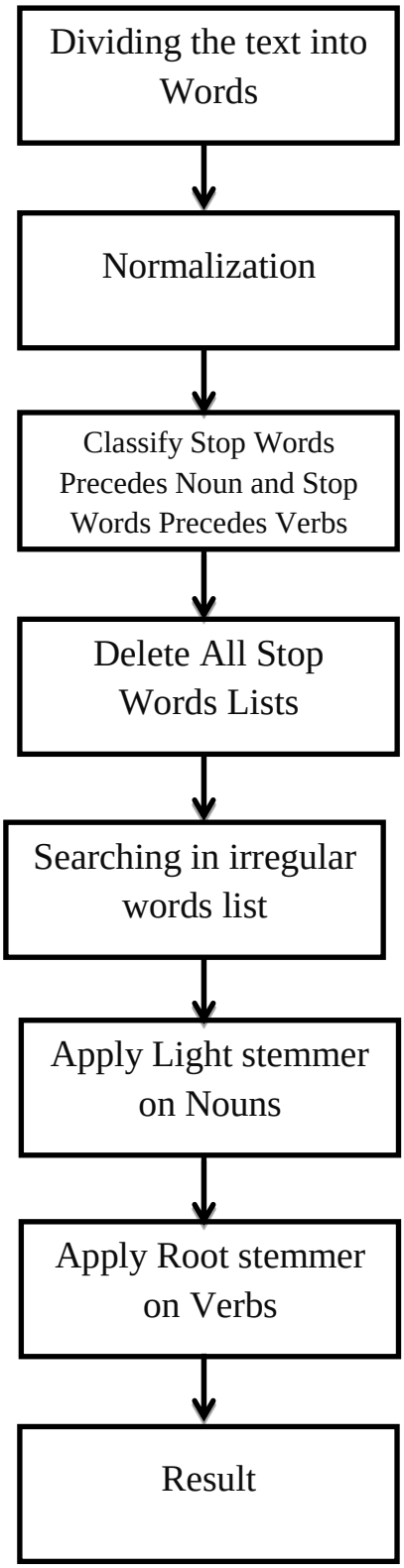


Figure 4.1: general diagram of the proposed Shaqalieh stemmer

The details of each step in the proposed stemmer will be discussed in next lines.

4.1.1. Dividing Text into Words

The first step of the stemmer is dividing the input text into words. Divide the text into words depends on the space after each word. For example, The stemmer will state the sentence "المهندس أحمد يبرمج المجذع" into 4 words. المهندس, أحمد, يبرمج, المجذع

4.1.2. Normalization

The second step in the stemmer is normalization of the words to set data more consistent. Normalization process in the proposed stemmer is the similar to the normalization process in light-10 stemmer which run as following :

1. Convert text to Unicode
2. Remove diacritics and punctuation
3. Remove non letter (for example, numbers)
4. Replace آ with double alif ا
5. Replace ى with ا
6. Replace initial إ with أ
7. Replace all Hamza forms ئ,ؤ,ء with أ

4.1.3. Classify Stop Word precedes Nouns and Stop Word precedes Verbs

The third step, The researcher know a stop words (functional words or structural words) are words that either carry no meaning or are very common thus do not represent the document. Stop words list usually contains prepositions, pronouns, and conjunctions. Text processing often performs stop words removal early in the process, although there is currently no standardized list of Arabic stop words. The current available Arabic stop words list.

The researcher able to categorize them into "useful" and "useless" stop words. Useless stop words are stop words that are used extensively and give no benefits to the subsequent words So The researcher delete all useless stop words. On the contrary, Useful stop words are words that can indicate the syntactical categories of the subsequent words. The researcher believe that the useful stop words can help us identify nouns and verbs where there is stop words precedes nouns and precedes verbs and direct us into appropriate stemming.

4.1.4. Delete All Stop Word

When finish the elicit nouns and verbs by using stop words these stop words becomes no benefits and existence in the documents cause consuming time without usefulness.

4.1.5. Searching in Irregular words

In this step, the stemmer will search for any word in a table of irregular words, to find out if the word exists in this table or not. If it exist, the stemmer returns the same word. The following table 4.1 shows sample of words the stemming it is the same word and contains twenty-six irregular words which were inserted in the table 4.1.

Table 4.1: Irregular words list

Irregular Word	Stem	Irregular Word	Stem
الله	الله	بريطانيا	بريطانيا
رضوان	رضوان	سوريا	سوريا
لبنان	لبنان	امريكا	امريكا
المانيا	المانيا	بغداد	بغداد
مليون	مليون	سنوات	سنوات
التلفزيون	التلفزيون	السعودية	السعودية
فرنسا	فرنسا	البيانات	البيانات

4.1.6. Applying Light Stemmer on Nouns

Light stemming is to find the representative indexing form of a word by the application of truncation of affixes.

The main goal of light stemming is to retain the word meaning intact and so improves the retrieval performance of an Arabic information retrieval system. Many light stemming methods like Larkey stemmer classifies the affixes to four kinds of affixes (antefixes, prefixes, suffixes and postfixes) that can be attached to its root.

The following example, Table 4.2, shows a sample of a word and its affixes

Table 4.2: A word and its affixes

Antefix	Prefix	Core	Suffix	Postfix
لـ	يـ	ناقش	و	هم

So from the above example The researcher see that if The researcher could remove all affixes of word then The researcher will get the stemmed word which is not the root but basic word without any affixes and so The researcher maintain the meaning of the word and improve the search effectiveness.

As The researcher see in related works, there are major problem in light stemming is that in many cases there is ambiguity. So The researcher will introduce a method for detecting such an ambiguity and to find if specific sequence is an affix or is part of the original word and thus The researcher solve this ambiguity issue that may lead to a completely unexpected behavior.

The weakness of Larkey is that its remove affixes that predefined in the list without checking if a remained is a stem. And in some cases, truncates it from the word and produces an erroneous stem.

The researcher are introducing a new method for stemming to solve many of the ambiguity problems related to light stemming as shown in figure 4.2.

Our method depends on set of possible affixes in which The researcher only have a prefix and suffix, in our prefixes The researcher combined all possible antefixes and prefixes to generate one complete list and in our postfixes and The researcher end up with the following list grouped by number of characters as shown in table 4.3 and table 4.4:

Table 4.3: Arabic prefixes

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

Table 4.4: Arabic suffixes

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

Before The researcher stem any word first The researcher match it against a set of all possible word patterns in Arabic, these patterns divided into two types the first is regular pattern and the other irregular patterns (plural forms of irregular nouns in Arabic language) when the word matched with the first type then return the same word on the other hand when the word matched with the second type then return the singular pattern as shown in the table 4.6.

The researcher get this list of patterns in Khoja stemmer with a list of word patterns provided by Marwan Albwab.

The researcher also added a set of patterns to this to have a complete list of Arabic word patterns. A sample list of patterns is as follows table 4.5:

Table 4.5: Sample the First type of patterns

The First type of patterns
فعل فاعل افعل تفعل تفاعل انفعل افتعل استفعل تفعيل فعال افعال تفعل تفاعل انفعال افتعال افعلال استفعال مفعل مفاعل مفعل متفعل متفعل متفاعل منفعل مفتعل مفعل مستفعل مفعول فعول مفعال فعال فعيل افعل فعلان فعلاء فعلى فواعل مفاعيل افاعل فعيل يفتعل يستفعل تفتعل فعائل

Table 4.6: Sample the Second type of patterns

The Second type of patterns	
Plural pattern	**Singular pattern**
مفاعل	مفعل
مفاعيل	مفعول
أفعال	فعل
فعلاء	فعيل
فعال	فاعل
أفعل	فعل
أفعلة	فعال
فواعل	فوعل

Matching a word against our Arabic patterns list solves the problem of prefix/suffix sequence ambiguity, so if The researcher have a word that starts with a possible prefix

but before The researcher truncate that prefix it matched one of the possible patterns then it's a valid word and The researcher will not truncate it.

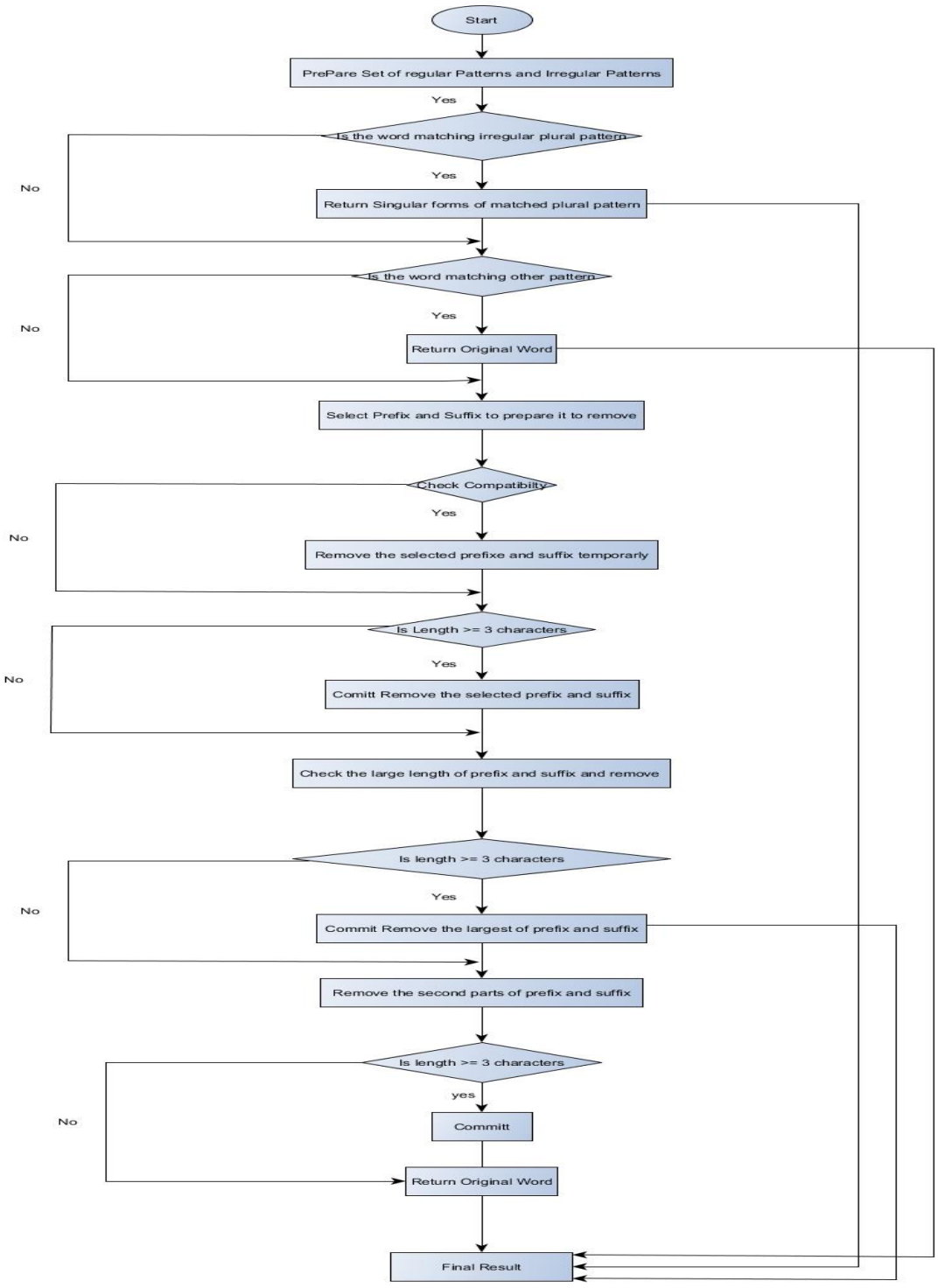


Figure 4.2: Flowchart for updated light stemmer

For example the word "كامل" it starts with a possible suffix "كا" but its part of the original word so removing it will lead to wrong word "" so The researcher detect that it is part of the word since it matches the pattern "", thus The researcher will not truncate it and return it as it is.

In the next step if the word didn't match any of the patterns then The researcher need to truncate its prefix and suffix but before that The researcher find the compatibility between the prefix and suffix where some suffixes could not be combined with certain suffixes in the same word and this also help us solving some ambiguity problems, for example the prefix "ال" may not be combined with the suffix "ك" so The researcher cannot say "الكتابك" and thus if The researcher have a word like "الكرنك" The researcher will not remove the prefix and suffix which lead to the wrong word "كرن" but The researcher will detect that the last character "ك" is part of the original word and not a suffix and The researcher will only remove the prefix "ال" which lead to the correct stem "كرنك".

If the combination of the prefix and suffix is valid then The researcher count the letters of the word after removing the prefix and suffix since Arabic words other than conjunctions like "في","من" consists of at least 3 characters.
If the number was larger or equal to 3 The researcher remove prefix and suffix and return the truncated word but if the number of characters after truncation is less than 3 characters then The researcher roll back since The researcher know that The researcher removed sequences that is part of the original word, and try to remove only the suffixes and count the number again if it is larger than 3 then The researcher try to find if the word only has a dual or plural suffix like "ان","ون","ين" and return the truncated word.
If the number after removing suffix was less than 3 then The researcher roll back since again The researcher know that this sequences is part of the original word, then The researcher try to remove the prefix only.

If the count is larger than 3 The researcher return the truncated word else The researcher return the original word as it is. For example if The researcher have the word "ولدين" after removing the prefix and suffix The researcher will end up with wrong

word "د" which has only one character so The researcher roll back and remove only the suffix which is "ين" which re
turns "ولد" which has 3 characters and it is the correct so The researcher return it.

4.1.7. Applying Root Stemmer on Verbs

The root is the original form of the word before any transformation process [11], the main goal of a root stemmer is to extract the basic form for a word by applying morphological analysis for the word. A superior root-based stemmer is the Khoja's stemmer [4].

From the previous studies that mentioned in chapter 2 The researcher note the ISRI Stemmer is the best because it avoids the disadvantages of Khoja and faster in execution and gives more accurate results and better and does not need to Root Dictionary. The researcher will now explain the details of the algorithm ISRI Stemmer. The Information Science Research Institute's (ISRI) Arabic stemmer shares many features with the Khoja stemmer. However, the main difference is that no root dictionary is used.

To begin describe of ISRI Stemmer (as shown in figure 4.3), The researcher know that the ISRI Stemmer define sets of diacritical marks and affix classes in table 4.7. These are sets of marks which are removed by the stemmer. And define some pattern sets in table 4.8. Now The researcher provide a more general overview of the stemmer.

Stemming proceeds in the following steps :

Remove diacritics representing vowels.

1. Normalize the hamza which appears in several distinct forms in combination with various letters to one form(أ). This step is necessary in order to ensure terms such as تعكل (he is eating) and أؤكل (it is eaten) conflate to the same root after their prefixes are removed .
2. Remove length three and length two prefixes in that order.
3. Remove connector و if it precedes a word beginning with و.
4. Normalize إ ,أ ,آto ا Removing the hamza in this case does not affect the root.

5. Return stem if less than or equal to three. Attempting to shorten stems further results in ambiguous stems .
6. Consider four cases depending on length of the word:
    a) Length =4: If the word matches one of the patterns from PR4 (figure 2), extract the relevant stem and return . Otherwise, attempt to remove length one suffixes and prefixes from S 1 and P 1 in that order provided the word is not less than length three .
    b) Length =5: Extract stems with three characters for words that match patterns from PR53 . If none are matched , attempt to remove suffixes and prefixes, otherwise the relevant length-three stem is returned . If the word is still five characters in length , the word is matched against PR54 to determine if it contains any stem of length 4. The relevant stem is returned if found.
    c) Length =6. Extract stems of length three if the word matches a pattern from PR63 . Otherwise ,attempt to remove suffixes . If a suffix is removed and a resulting term of length five results , send the word back through step 7b. Otherwise, attempt to remove one character prefixes, and if successful ,send the resulting length five term to step 7b.
    d) Length =7. Attempt to remove one-character suffixes and prefixes. If successful, send the resulting length -six term to step7c .

Step 7 essentially takes longer words and successively attempts to trim single character affixes . If successful, it compares the resulting shorter term with various patterns at different levels until it either matches a pattern and extracts the relevant term , or becomes too short to be a viable stem.

Table 4.7: Affix Sets

Set	description	Examples
D	Diacritics-vowelization	سَ سِسُسْ سّ سًسٍسٌ
P3	Prefixes of length three	وللل , وال , كال , بال
P2	Length two prefixes	ال , لل
P1	Length one prefixes	ل , ب , ف , س , و ,ي , ت ن , ا
S3	Length three suffixes	تمل , همل , تان , تين , كمل

Set	description	Examples
S2	Length two suffixes	ون , ات , ان , ين , تن , كم , هن , نا , يا , يا , ها , تم , كن , ني , وا , ما , هم
S1	Length one suffixes	ن، ا , ت , ك , ي , ه , ة

Table 4.8: Arabic Pattern and Roots

SET	Description	Examples
PR4	Length four pattern	فاعل فعول فعلة فعال فعيل مفعل
PR53	Length five patterns and length three roots	تفاعل افتعل افعال افاعل فعالة فعلان فعولة تفعلة تفعيل مفعلة فاعول فواعل مفعال مفعول مفعيل افعلة فعائل منفعل مفتعل فاعلة مفاعل فملاع يفتعل تفتعل فعالى انفعل
PR54	Length five patterns and length four roots	تفعلل افعلل مفعلل فعللة فعلان فعالل
PR63	Length six patterns and length three roots	استفعل مفعالة افتعال افعوعل انفعل مستفعل
PR64	Length six patterns and length four roots	افنلل افعلال متفعلل

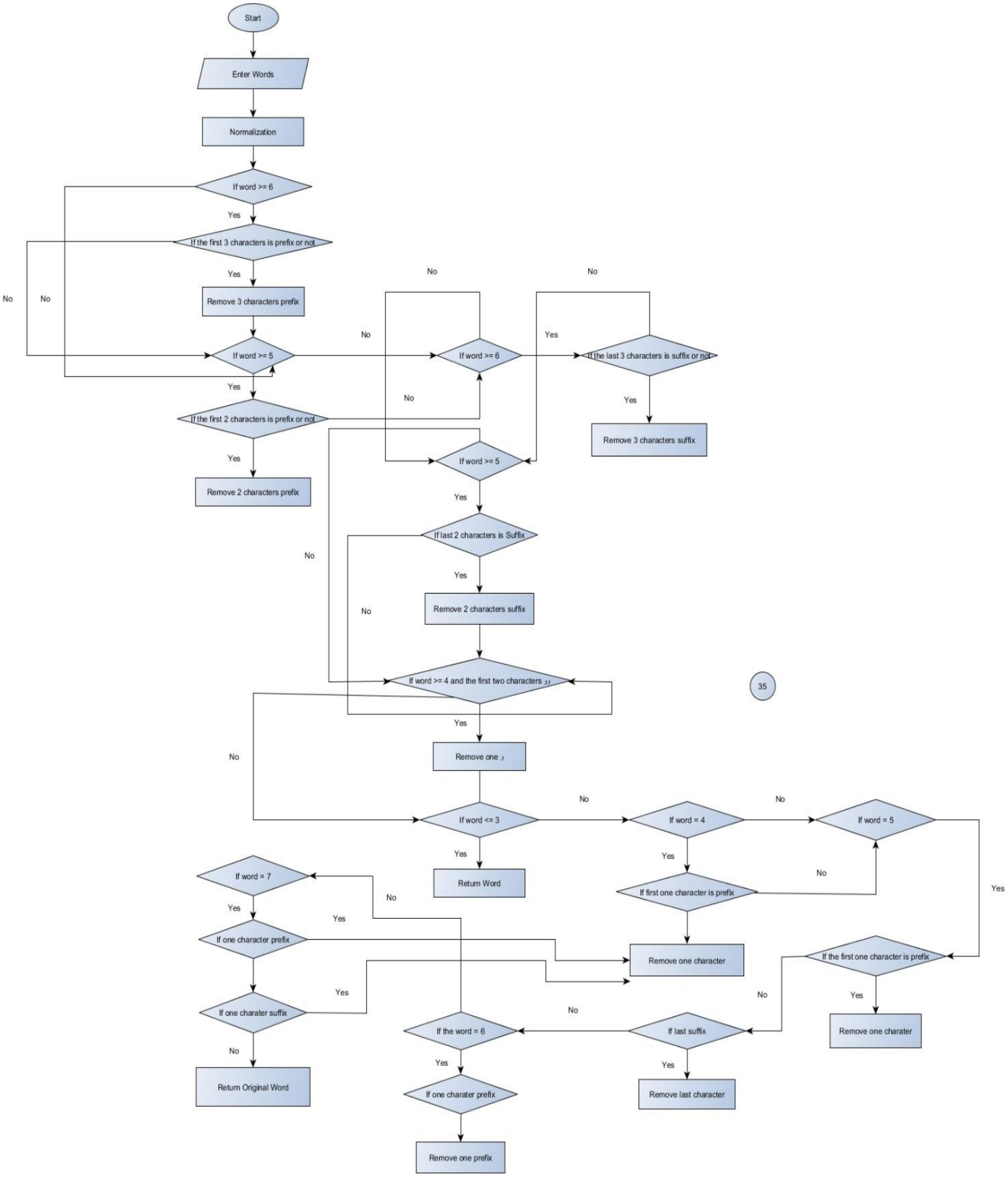


Figure 4.3: root stemmer - ISRI

4.2. Morphological Analysis (Stemmer and Light Stemmer)

In linguistics, morphology is the identification, analysis and description of the structure of morphemes and other units of meaning in a language like words, affixes, and parts of speech and intonation/stress, implied context (words in a lexicon are the subject matter of lexicology).

Morphological typology represents a way of classifying languages according to the ways by which morphemes are used in a language from the analytic that use only isolated morphemes, through the agglutinative ("stuck-together") and fusional languages that use bound morphemes (affixes), up to the polysynthetic, which compress lots of separate morphemes into single words [19].

While words are generally accepted as being (with clitics) the smallest units of syntax, it is clear that in most (if not all) languages, words can be related to other words by rules (grammars). For example, English speakers recognize that the words dog and dogs are closely related- differentiated only by the plurality morpheme "s" which is only found bound to nouns, and is never separate. Speakers of English (a fusional language) recognize these relations from their tacit knowledge of the rules of word formation in English. They infer intuitively that dog is to dogs as cat is to cats; similarly, dog is to dog catcher as dish is to dishwasher (in one sense).

The rules understood by the speaker reflect specific patterns (or regularities) in the way words are formed from smaller units and how those smaller units interact in speech. In this way, morphology is the branch of linguistics that studies patterns of word formation within and across languages, and attempts to formulate rules that model the knowledge of the speakers of those languages [19].

Terms have many morphological variants that will not be recognized by term matching algorithm without additional text processing. Stemming algorithms are needed in many applications such as natural language processing, compression of data, and information retrieval systems. In most cases, these variants have similar semantic interpretation and can be treated as equivalence in text mining. Stemming algorithm can be employed to perform term reduction to a root form [19].

In general, most of Arabic morphological tools face a problem with diacritics because most of them remove (normalize) diacritics. For example, the Arabic word (ذَهَبَ) which means (went) has identical form (without diacritics) to word (ذَهَبْ) which means gold. Diacritics distinguish between them, but unfortunately, most of Arabic morphological tools remove them as a first step [19].

For Arabic Language, there are two different morphological analysis techniques; stemming and light stemming. Stemming reduces words to their stems [56]. Light stemming, in contrast, removes common affixes from words without reducing them to their stems. Stemming would reduce the Arabic words (الكتاب الكاتب المكتبة) which mean (the library), (the writer), and (the book) respectively, to one stem (كتب), which means (write).

The main idea for using light stemming [32] [33] is that many word variants do not have similar meanings or semantics. However, these word variants are generated from the same root. Thus, root extraction algorithms affect the meanings of words. Light stemming aims to enhance the classification performance while retaining the words meanings. It removes some defined prefixes and suffixes from the word instead of extracting the original root [32][33].

Formally speaking, the aforementioned Arabic words (الكتاب الكاتب المكتبة) which mean (the library), (the writer), and (the book) respectively, belong to one stem (كتب) despite they have different meanings. Thus, the stemming approach reduces their semantics. The light stemming approach, on the other hand, maps the word (الكتاب) which means (the book) to (كتاب) which means (book), and stems the word (الكاتب) which means (the writers) to (كاتب) which means (writer). Another example for light stemming is the words (المسافرون والمسافرين) which mapped to word (مسافر). Light stemming keeps the words‘ meanings unaffected. The researcher previously described in section 1.3 that there are many words morphology have different meaning despite they have the same root.

Stemming algorithm by Khoja [56] one is of well know Arabic Stemmers. 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 di-

acritic characters, punctuation characters, definite articles, and stop words. However, the Khoja stemmer has several weaknesses .

First, the root dictionary requires maintenance to guarantee newly discovered words are correctly stemmed. Second, the Khoja stemmer replaces a weak letter with (و) which occasionally produces a root that is not related to the original word. For example, the word (منظمات) which mean (organizations) is stemmed to (ظما) which means (he was thirsty) instead of (نظم). Here the Khoja stemmer removed a part of the root when it removed the prefix and then added a hamza at the end. Third, by following a certain order of affixes, the Khoja stemmer will in some cases fail to remove all of them. For example, the terms (تستغرق) and (ركبتيه) are not stemmed although they are respectively derived from the two regular roots (غرق) and (ركب). Algorithm steps of Khoja Arabic stemmer as below. Al-Shalabi, Kanaan and Al-Serhan [14] developed a root extraction

1. Remove diacritics.
2. Remove Stop words, Punctuation, and numbers.
3. Remove definite article (الـ).
4. Remove inseparable conjunction(و).
5. Remove Suffixes
6. Remove Prefixes
7. Match result against a list of patterns
    - If a match is found, extracted the characters in the pattern representing the root.
    - Match the extracted root a against a list known "valid" roots.
8. Replace weak letters واي with و
9. Replace all occurrence of Hamza ,ؤ,ئ with ا
10. Two letter roots are checked to see if they should contain a double character. If so character is added to the root.

algorithm (tri-literal root extraction) which does not use any dictionary. It depends on assigning weights for a word‘s letters multiplied by the letter‘s position, Consonants were assigned a weight of zero and different weights were assigned to the letters grouped in the word (سألتمونيها) where all affixes are formed by combina-

tions of these letters. The algorithm selects the letters with the lowest weights as root letters.

Sawalhi and Atwell evaluated Arabic language morphological analyzers and stemmers. Authors reported Khoja stemmer achieved the highest accuracy then the tri-literal root extraction algorithm. The majorities of words have a tri-lateral root, in fact between 80 and 85% of words in Arabic are derived from tri-lateral roots [8] [28].

The rest have a quad-letter root, penta-letter root or hexa-letter root. Khoja stemmer works accurately for tri-literal roots, this why it achieved the highest accuracy. Sawalhi and Atwell also reported that most stemming algorithms are designed for information retrieval systems where accuracy of the stemmers is not important issue.

On the other hand, accuracy is vital for natural language processing. The accuracy rates show that the best algorithm failed to achieve accuracy rate of more than 75%. This proves that more research is required. The researcher cannot rely on such stemming algorithms for doing further research as Part-of-Speech tagging and then Parsing because errors from the stemming algorithms will propagate to such systems.

4.3. Text Preprocessing Tools (WEKA)

The researcher use WEKA (Waikato Environment for Knowledge Analysis) for text preprocessing and classification. WEKA is a popular suite of machine learning software written in Java, developed at the University of Waikato. It is free software available under the GNU General Public License.

WEKA provides a large collection of machine learning algorithms for data pre-processing, classification, clustering, association rules, and visualization, which can be invoked through a common Graphical User Interface. Using WEKA StringToWordVector tool options with different combinations, The researcher set-up the term weighting combinations presented in Table 4.9 to structure text data. Major combinations include Boolean, word count, tf, tdf, tf-idf, term pruning, and word count normalization options, these combinations have not been applied in the literature on Arabic text before.

The resulting combinations are listed in Table 4.9.

Table 4.9: WEKA String to word Vector options

WEKA String to word Vector options	
TF Transform	$\log(1 + f_{ij})$, where f_{ij} is the frequency of word I in document d_j.
IDF Transform	f_{ij} * log(num of docs / num of docs with word i) , where f_{ij} is the frequency of word I in document d_j.
TF-IDF Transformation	$\log(1 + f_{ij})$* log(num of docs / num of docs with word i), where f_{ij} is the frequency of word I in document d_j.
Min Term Freq	Sets the minimum term frequency (apply term running)
Normalize Doc Length	Sets whether if the word frequencies for a document should be normalized or not.
Output Word Counts	Output word count rather than Boolean 0 or 1 (indicating absence or presence of a word)
Stemmer	The stemming algorithm to be use on the words(Khoja Arabic Stemmer Algorithm)

Seven text classification algorithms (C4.5 Decision Tree (C4.5 DT), K Nearest Neighbors (KNN), Support Vector Machine (SVMs), Naive Bayes (NB), Naive Bayes Multinomial (NBM), Complement Naive Bayes (CNB), and Discriminative Multinomial Naive Bayes classifier (DMNB text)) are may be applied to classify text documents. Experimental results presented in chapter 6.

The researcher implement and integrate Arabic stemming and light stemming algorithms, into WEKA. The researcher adopt Arabic stop words list from for stop words removal. The complete package of integration is available publically .

A screenshot of Arabic stemmer / light stemmer integrated to WEKA is depicted in figure 4.4. to figure 4.9.

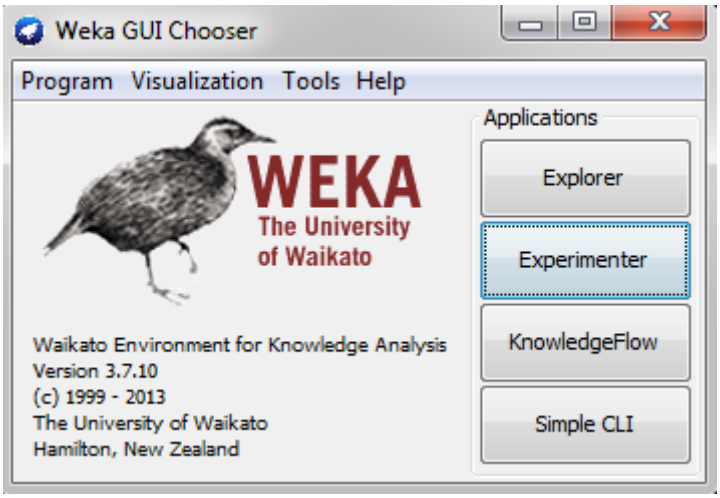


Figure 4.4: WEKA GUI Chooser

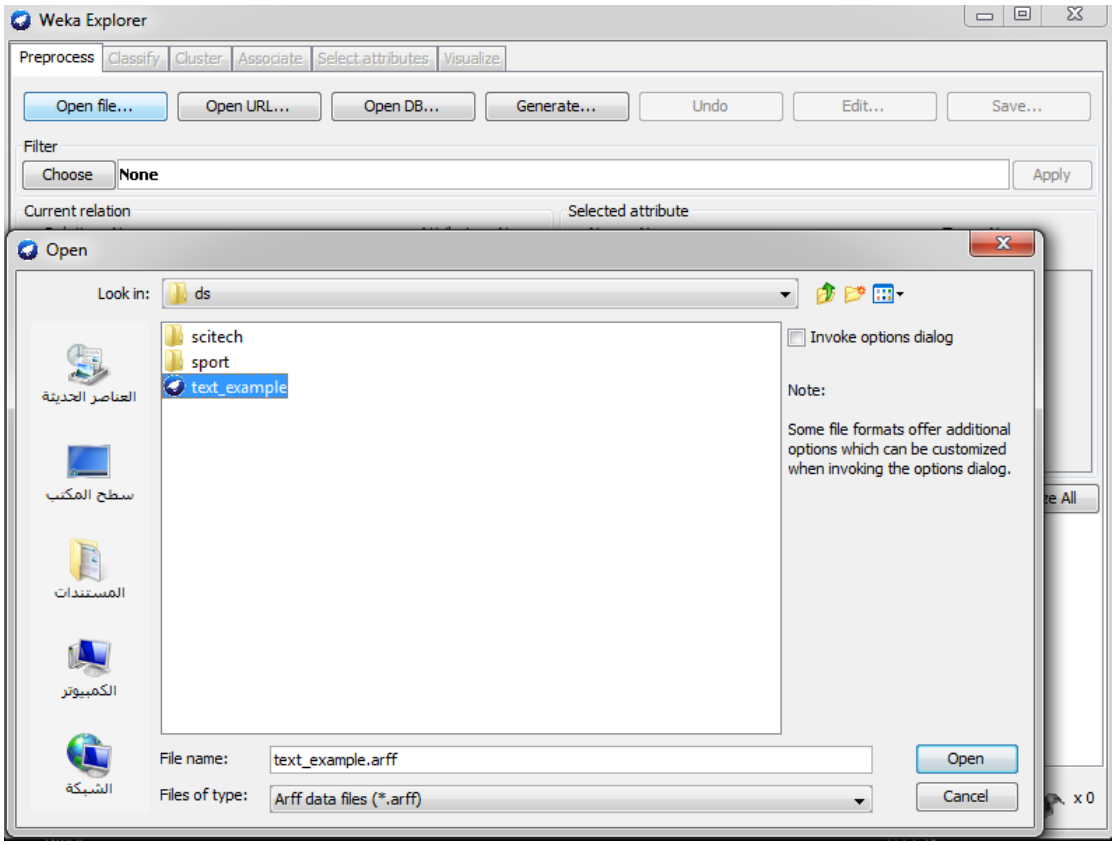


Figure 4.5: Browser to select arff file

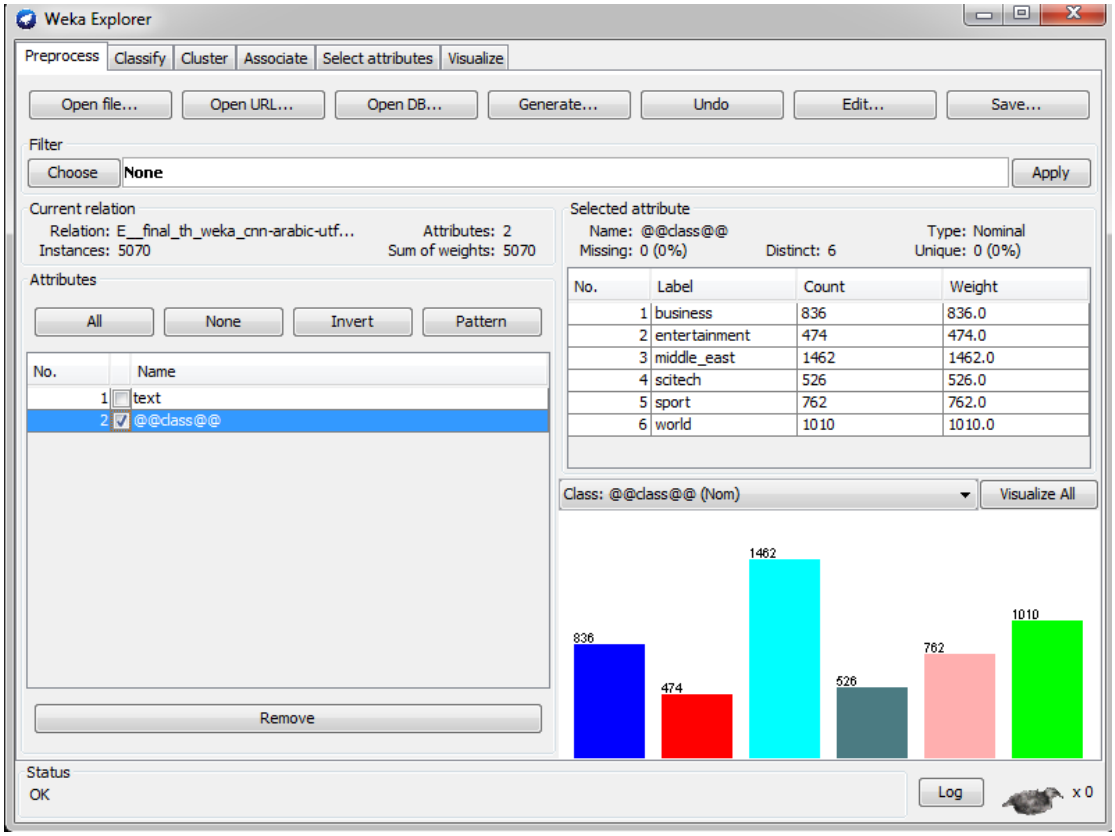


Figure 4.6: Analysis data to view the file included in arff file

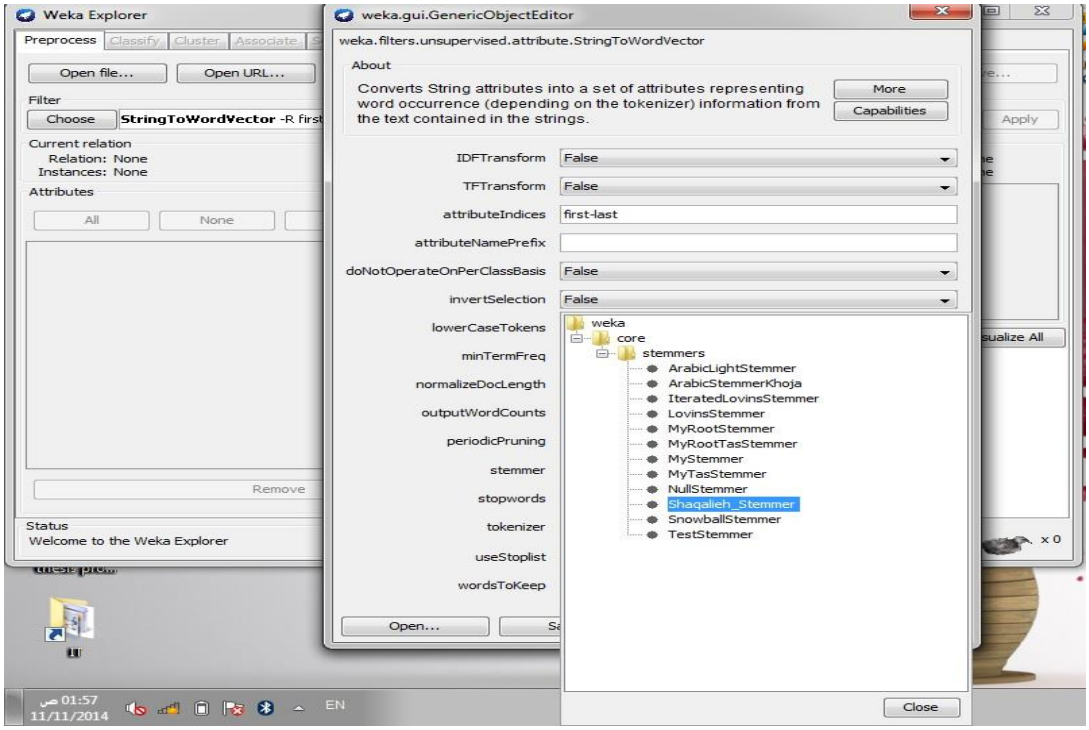


Figure 4.7: WEKA Arabic Stemmers including the proposal Root and Light stemmers "Shaqal-ieh Stemmer"

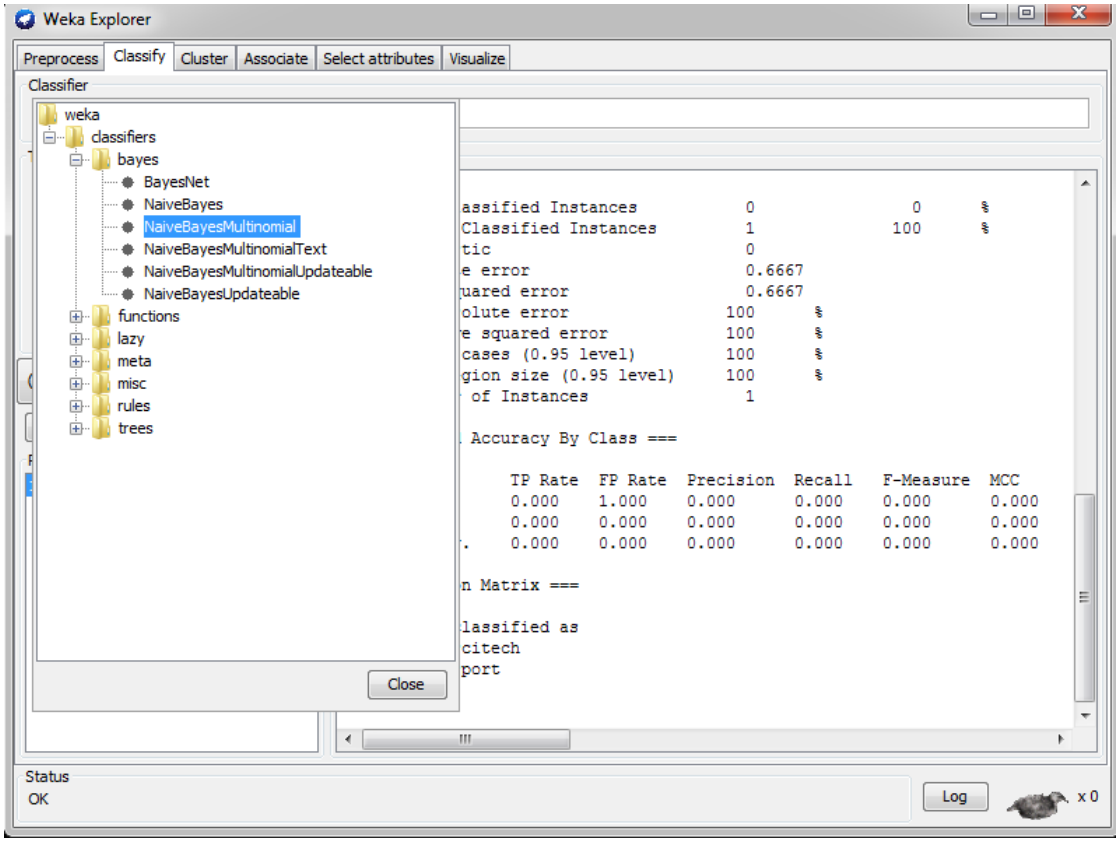


Figure 4.8: Browser to select classifier

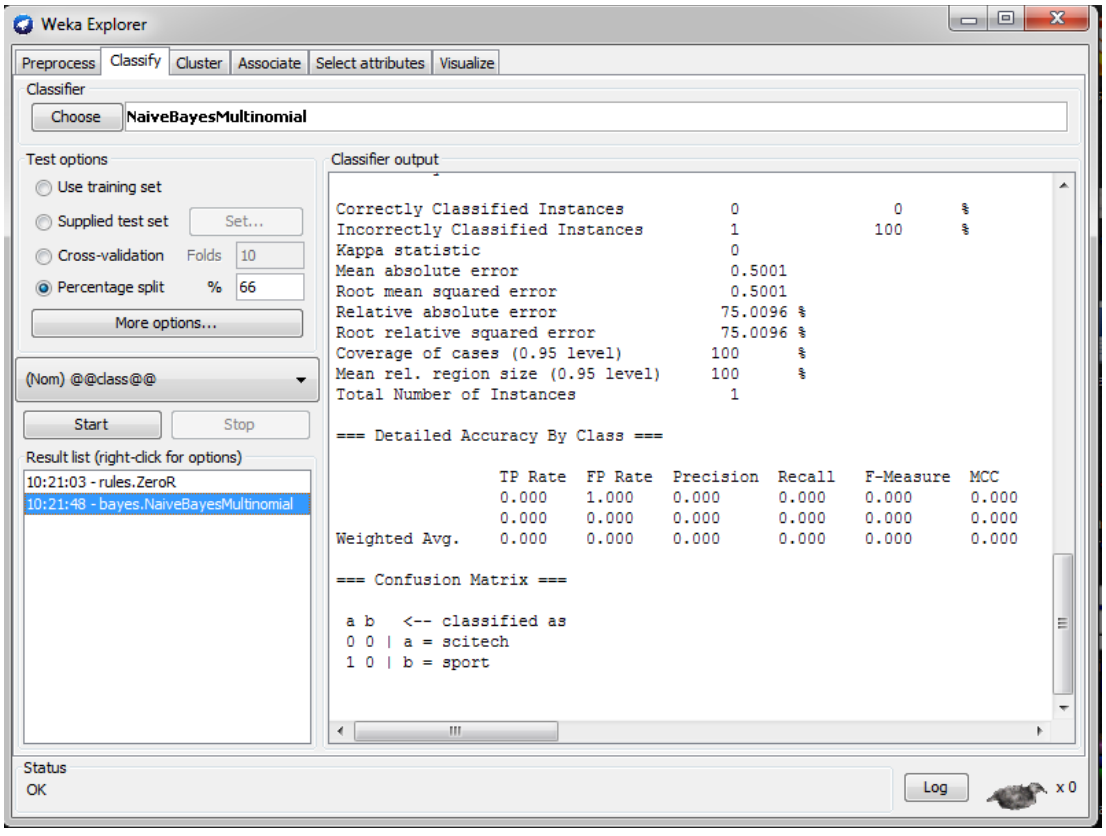


Figure 4.9 :To view the classifier output

Chapter 5 : Corpora

One of the difficulties that encounter this work and other researches in the field of Arabic linguistics was the lack of publicly available Arabic corpus for evaluating text categorization algorithms. Different training data sets are available for text classification in English. Reuter‘s collections of news stories are popular and typical example.

The Linguistic Data Consortium (LDC) provides two non-free Arabic corpora, the Arabic NEWSWIRE and Arabic Gigaword corpus. Both corpora contain newswire stories. One of the aims of this research is to compile representative training datasets for Arabic text classification that cover different text genres which can be used in this research and in the future as a benchmark. Therefore, three different datasets were compiled covering different genres and subject domains.

There is a need for a freely-accessible corpus of Arabic. There are no standard or benchmark corpora. All researchers conduct their researches on their own compiled corpus. Arabic language is highly inflectional and derivational language which makes text mining a complex task. In Arabic TC research field, there are some published experimental results, but these results came from different datasets, it is hard to compare classifiers because each research used different datasets for training and testing[4] [5] [6][7][8][9][10][11][12] [13] [34][38][63] [64] [65] [66] [67] [68] [69].

Sebastiani stated at [54]"The researcher have to bear in mind that comparisons are reliable only when based on experiments performed by the same author under carefully controlled conditions".

Corpus sizes for the same topics written in Arabic and other different languages are not the same. In fact, the size of the corpus extracted from the French newspaper Le model from the period of 4 years, is 80 million words [22].

Moreover, the size of corpus extracted from the period of almost 7 years of Associated French Press (AFP) Arabic Newswire, and released in 2001 by LDC is 76 million tokens [4] [5] .

This gap between the two sizes is justified by the compact form of the Arabic words. Formally speaking, the English word write is equivalent to one Arabic word كتب. But the group .He writes, made up of two words, and also corresponds to one Arabic word يكتب. And the Arabic equivalent of the sentence . He will write is the only one word سيكتب. Moreover, the word سيكتبه amounts to the group of words .

He will write it. Another example is the Arabic word (وبنفوذها) and its equivalence in English (4 words) and with her influences. This makes segmentation of Arabic textual data different and more difficult than Latin languages. This gives an explanation of the gap between the two corpuses size, if The researcher make into consideration the difference of data extraction period [2][3].

On the other hand, the required amount of storage (disk or RAM) for Arabic corpus is twice of English corpus for the same size of characters because Arabic characters require 2 bytes to be saved in Unicode format. This implies that feature reduction for Arabic text is necessary to consider storage limit.

5.1. Corpora Building Steps

The first phase in construction process is to build a text dataset which involves compiling and labeling text documents into corpus. The researcher collect web documents from internet using the open source offline explorer. The process also includes converting corpus html/xml files into UTF-8 encoding using Text Encoding Converter by Web Key Soft. The final step is to strip/remove html/xml tags as shown in Figure 5.1. There are Java program that strip / remove html/xml tags. The program is available publically [70].

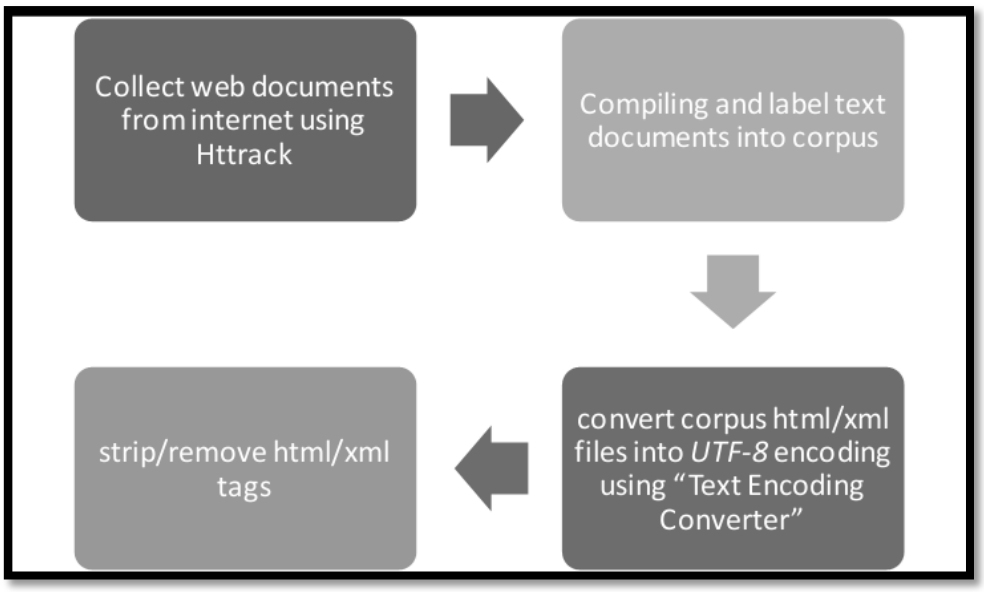


Figure 5.1: Corpus Building Steps

5.2. Corpora Summary

The researcher use various corpora to perform our experimentations, the corpora variations include small/large size corpus, with few and more categories. The used corpora have been collected by us and by other researchers. The researcher collected three corpora, The researcher collect them from: BBC Arabic, CNN Arabic, and the third corpus was collected from multiple websites, The researcher shall call the third corpus as Open Source Arabic Corpus (OSAC).

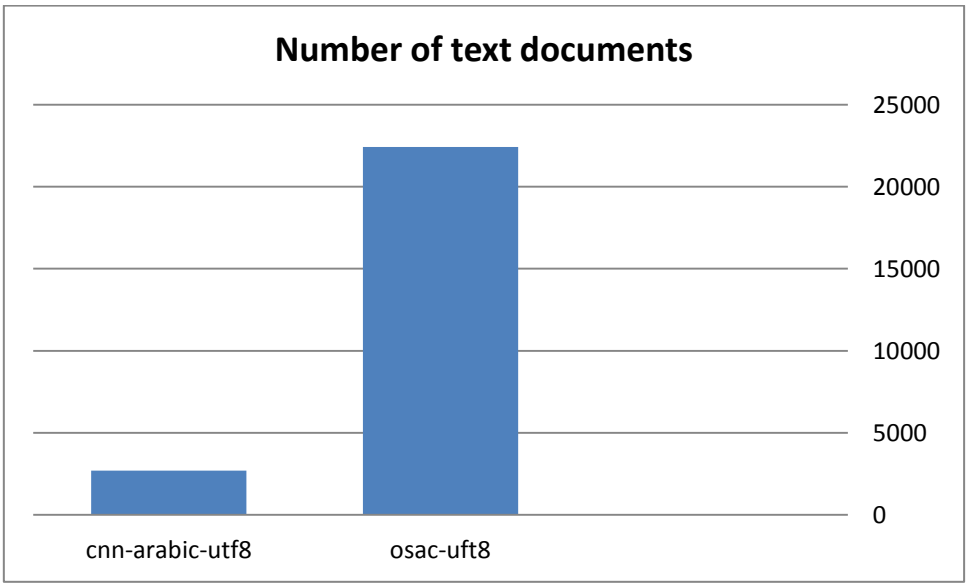


Figure 5.2: The three corpora are available publically

Figures 5.2 present the corpora The researcher used in this research, the used corpora have various keywords size and various number of documents. Figure 5.2 shows the number of text documents for each corpus. Despite CNN corpus has a small number of text documents (2390 text documents) but it has relatively a large number of keywords (95,350 keywords), the reason is that CNN corpus has large size text documents (long documents), also, the corpus covers broad range of text genre. OSAC corpus has the largest number of text documents and largest vocabulary. In the following, The researcher shall describe each corpus in details.

5.3. BBC Arabic corpus

The researcher collected BBC Arabic corpus from BBC Arabic website bbcarabic.com, the corpus includes 2390 text documents. Each text document belongs 1 of to 7 categories shown in the table 5.1 (Middle East News 1178, World News 745, Business & Economy 148, Sports 110, International Press 25, Science & Technology 123, Art & Culture 61). The corpus contains 1,860,786 (1.8M) words and 106,733 district keywords after stop words removal. The researcher converted the corpus to utf-8 encoding and stripped html tags. The corpus is available publically at [70].

Table 5.1: bbc-arabic-utf8

No.	Label	Count	Weight
1	أخبار الشرق الاوسط	1178	1178.0
2	اخبار العالم	745	745.0
3	اقتصاد واعمال	148	148.0
4	رياضة	110	110.0
5	عرض الصحف	25	25.0
6	علوم وتكنولوجيا	123	123.0
7	منوعات	61	61.0

5.4. CNN Arabic corpus

The researcher collected CNN Arabic corpus from CNN Arabic website cnnarabic.com, the corpus includes 2,689 text documents. Each text document belongs 1 of to 6 categories shown in the table 5.2, (Business 435, Entertainments 248, Middle East News 766, Science & Technology 282, Sports 412, World News 546). The corpus contains 2,241,348 (2.2M) words and 144,460 district key-

words after stop words removal. The researcher converted the corpus to utf-8 encoding and stripped html tags. The corpus is available publically [70].

Table 5.2: cnn-arabic-utf8

No.	Label	Count	Weight
1	Business	435	435.0
2	Enter attainment	248	248.0
3	Middle east	766	766.0
4	SciTech	282	282.0
5	Sport	412	412.0
6	world	546	546.0

5.5. OSAC corpus

The researcher collected OSAC Arabic corpus from multiple websites as presented in Table 5.3, the corpus includes 22,429 text documents. Each text document belongs 1 of to 10 categories shown in the table 5.3, (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. The researcher converted the corpus to utf-8 encoding and stripped html tags. The corpus is available publically [70].

Table 5.3: osac-uft8

No.	Label	Count	Weight
1	اقتصاد	3102	3102.0
2	تاريخ	3233	3233.0
3	تربية واسرة ومرأة	3608	3608.0
4	دين وفتاوى شرعية	3171	3171.0
5	رياضة	2419	2419.0
6	صحة	2296	2296.0
7	فلك	557	557.0
8	قانون	944	944.0
9	قصص	726	726.0
10	وصفات واكلات	2373	2373.0

Chapter 6 : Experimental results and analysis

In this chapter, The researcher present and analyze experimental results. Text Classification algorithms (Support Vector Machines (SVMs), Naive Bayes (NB), Naive Bayes Variants (Naive Bayes Multinomial (NBM), Complement Naive Bayes (CNB), Discriminative Multinomial Naive Bayes (DMNB))) are described in chapter 3, methodology is described in chapter 4, and corpora are described in chapter 5. The researcher split each corpus to 2 parts (66% of the corpus for training and the remaining 34% for test).

The researcher could not run any classifier in batch mode because the corpora size is very large and did not fit to memory. All classifiers were run in incremental mode on 64-bit machine with 6 GB RAM. The researcher use method provided by WEKA for experiments.

Experimental results investigate preprocessing time, classifiers accuracy and recall/precision.

The researcher could not generate all text representation for OSAC corpus because it does not fit to memory. Also, The researcher could not run all text classifiers on this corpus for the same reason.

6.1.Preprocessing time

Text preprocessing includes morphological analysis and term weighting. Raw text requires string tokenization + stop words removal + term matching (to add word as a count to existing feature or to add it as a new feature). Stemming/light stemming preprocessing requires the same steps in addition to one additional step after stop word removal which is stemming/light stemming.

Figure 6.1 shows the average time required to analyze the corpora morphologically (stemming, light stemming and proposed algorithm stemming), the Figure 6.1 also shows the average time required to process raw text (without morphological analysis).

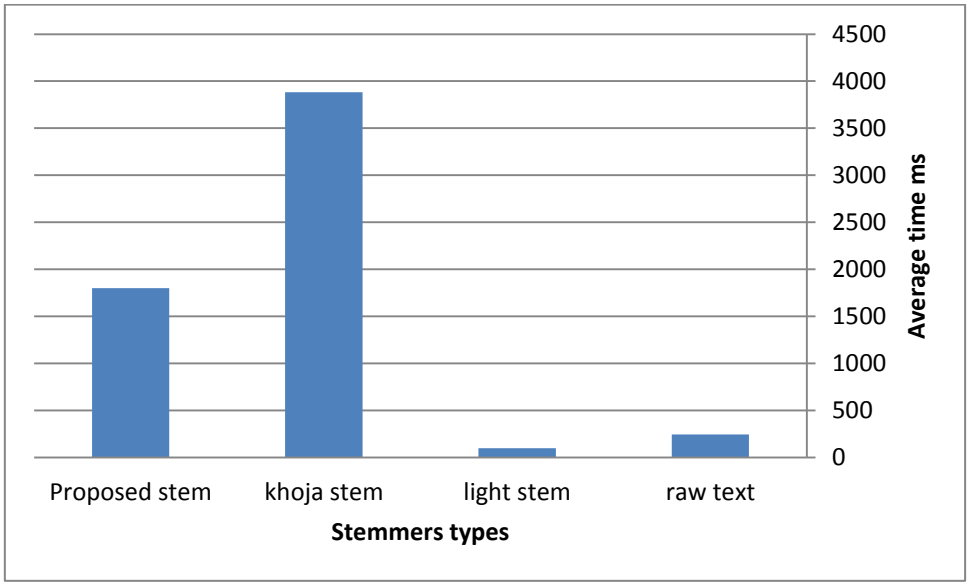


Figure 6.1: Average time required to analyze the corpora

Proposed algorithm stemming requires the least time to preprocess text data after light stemming, even less than raw text preprocessing time, this is explained by two reasons:

1. Light stemming algorithm step is fast (just normalizes word and removes suffixes and prefixes), but the proposed stemming algorithm save the meaning of words so more time but increase accuracy.
2. Proposed stemming reduces the original raw text to 50% with term pruning, in other words, despite raw text does not preprocess text morphologically, it needs more time than Proposed stemming preprocessing time , raw text preprocessing takes long time to search in large feature/dictionary for match terms.

The results by Duwairi [32][33] show that preprocessing and classification time of stemming is the least. The result also states that the difference of stemming and light stemming preprocessing and classification time is slight. The reason is

that Duwairi used stemmer by Al-Shalabi [14] which does not match pattern and validated extracted root is stemming process.

Preprocessing time of different term weighting schemes is shown in figure 6.2 In average, all term weighting schemes have approximately similar preprocessing time despite each term weighting scheme has different counting formula. The least preprocessing time is achieved when applying term pruning because it leads to save the required time to look up into feature/dictionary to match terms.

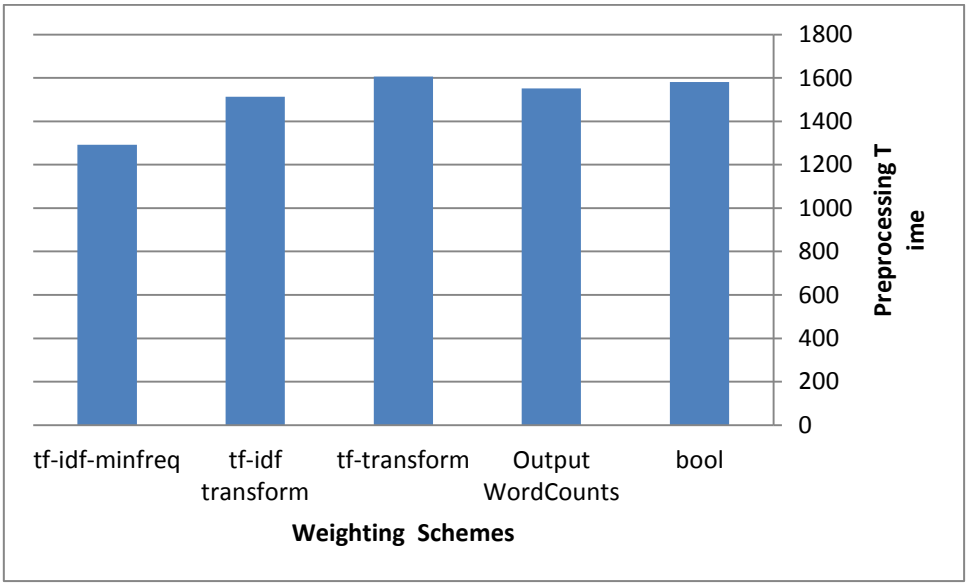


Figure 6.2: Preprocessing time of different term weighting schemes

6.2. Classifier Accuracy

Among seven classifiers applied on seven corpora, SVMs achieved the highest average accuracy (98.11%), then DMNB with average accuracy of 95.33%. KNN was the worst with average accuracy of 72.48%. Figure 6.3 shows the classifiers average performance.

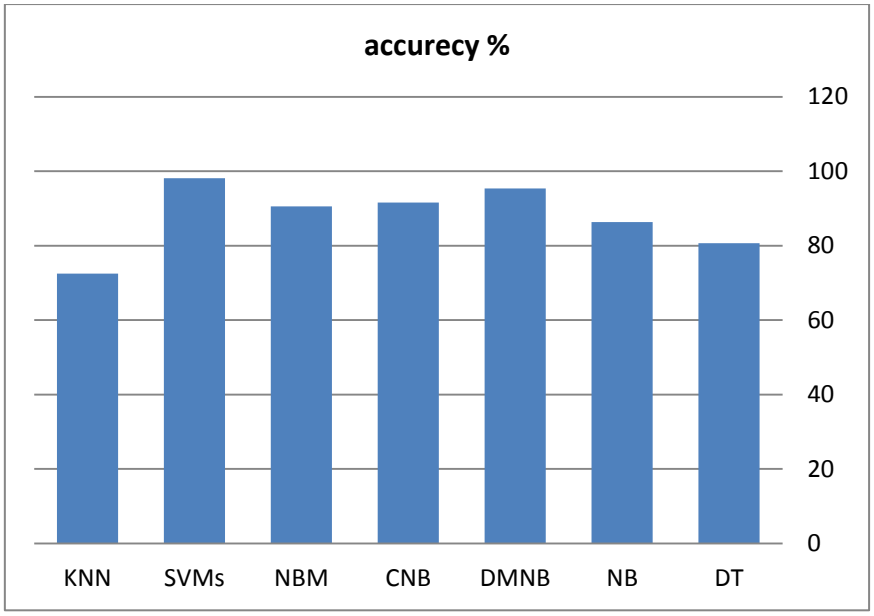


Figure 6.3: Classifiers Average Performance

Generally, SVMs and NB variants achieved the best average classification accuracy. SVMs achieved the best accuracy because it is a robust classifier.

Text dataset requires considerations like language model, decision boundary for imbalanced text dataset, good parameters estimation, and word dependency. These considerations have been taken into account in NB variant classifiers, this makes them achieve the best average accuracy. Furthermore, NB variant classifiers inherit NB property of naive assumption of independent features which make them simple and achieve respectable effective performance. The researcher have described text classification consideration and the corrections to NB in details in chapter 3.

DT is not scalable and it requires very long training time [57][60]. Additionally, term weighting schemes have a direct impact on KNN because it depends on distance function. Distance functions are not scalable.

KNN achieves high performance using (tf-idf + normalization + term pruning) term weighting schemes and light stemming feature reduction and term pruning as The researcher will see in Figure 6.4 shows the average accuracy of classifiers applied on OSAC corpus.

Figure 6.4, Figure 6.5 shows the applied KNN text classifier, SVM classifier and NBM Classifier. On OSAC corpus with using new stemmer "Shaqalieh stemmer", Khoja stemmer, Light stemmer.

The researcher note that the new stemmer is the best because the new stemmer consume high performance. This is due because separation between nouns and verbs where most of the nouns back without any processing. And The researcher note that the Light10 stemmer is the least time.

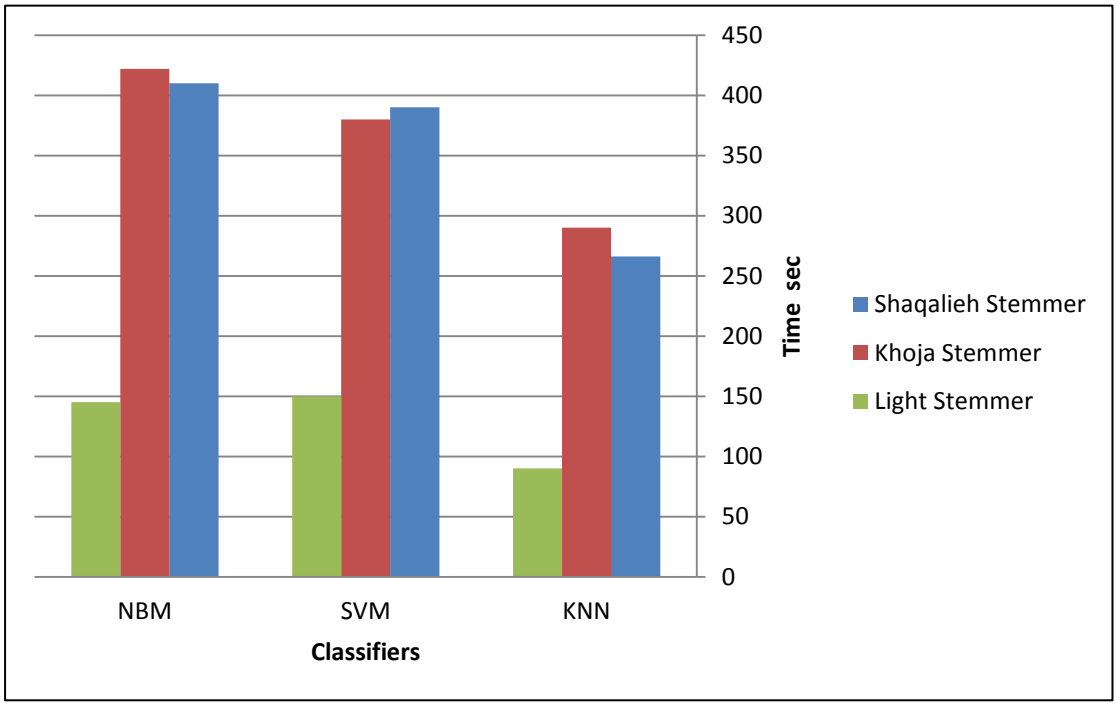


Figure 6.4: Khoja, Light and Proposed stemming time for Knn, SVM and NBM classifiers

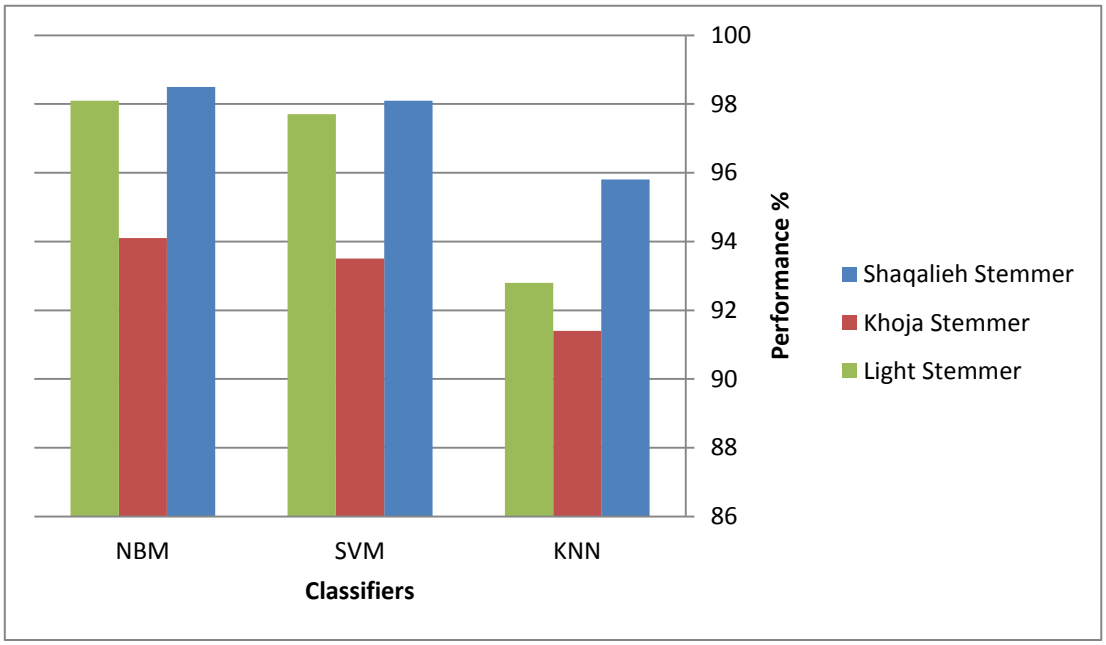


Figure 6.5: Khoja, Light and Proposed stemming accuracy classification for Knn, SVM and NBM classifiers

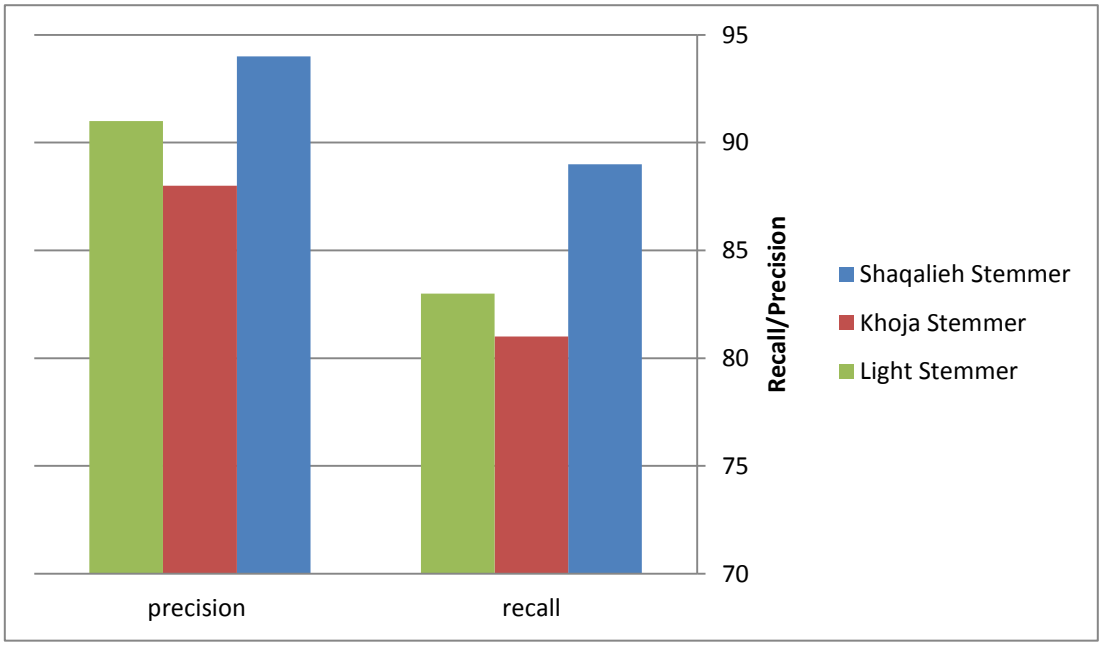


Figure 6.6: Recall/Precision

Figure 6.6 shows the effect of using the proposed stemmer on recall and precision values. The proposed stemmer will increase the value of recall and precision to 87% and 92% respectively which is better than using Khoja or Light10.

The reason that the precision rate is greater than recall is that the data is not distributed equally to the classes, and that the selection of training data depends on the number of files for each class.

6.3. Morphological Analysis and term pruning

Morphological analysis tools (stemming / light stemming) can be used to reduce features as described in chapter 1 and 4. In addition, term pruning can be used for the same purpose. The researcher discussed the impact of morphological analysis tools on feature reduction, in this section; The researcher shall discuss the impact of morphological analysis tools on classification accuracy.

The impact of morphological analysis and term pruning on different corpora is depicted in Figure 6.7. The figure shows that the average classification performance for raw text, stemming, light stemming and proposed stemming are convergent because the morphological analysis and term pruning have slight impact on most classifiers.

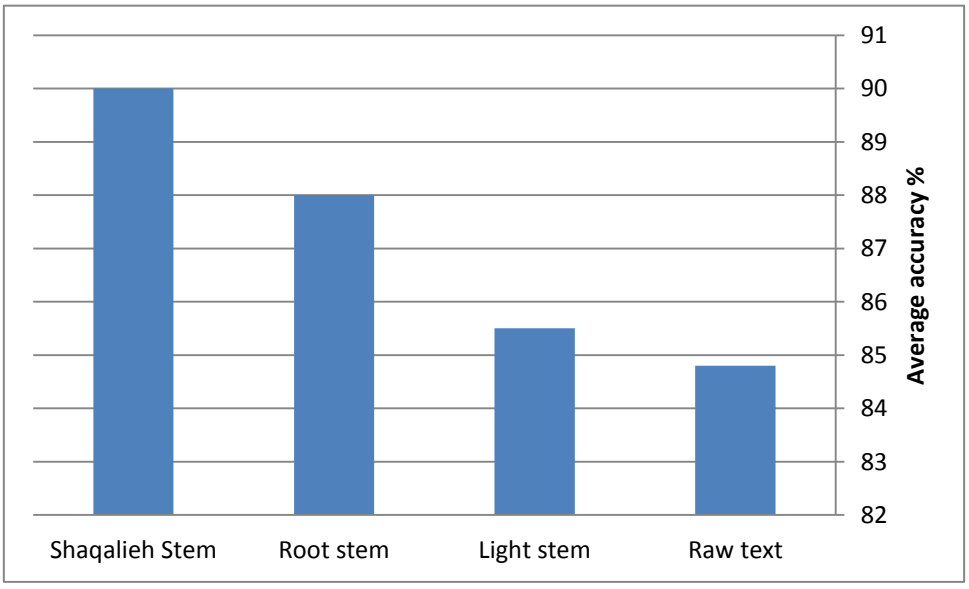


Figure 6.7: Shaqalieh stem vs. Stemming vs. light stemming vs. raw stem(Average Accuracy)

The researcher recommend collecting light stemming and root stemming as Arabic morphological analysis tool to improve average classification performance; The researcher recommend it on nouns because save meaning of words and light stemming is more proper than stemming from linguistics and semantic view point and it has the least preprocessing time. The researcher note the root stem is

better with verbs. because stemming has the slight classification performance than light stemming is that the majorities of Arabic words have a tri-lateral root, in fact between 80 and 85% of verbs in Arabic are derived from tri-lateral roots [8] [28]. The rest have a quad-letter root, penta-letter root or hexa-letter root. Khoja stemmer works accurately for tri-literal roots, this why it achieved the highest accuracy.

Chapter 7 : Conclusion and Future work

7.1. Conclusion

Text mining is on the cross road of information retrieval and machine learning. Arabic text mining is promising research field due to the complexity and problems in different aspects:

The lack Arabic corpus, lack of language tools, and lack of stemming on Arabic text.

In this research, The researcher presents a new method for stemming to solve many of the ambiguity problems related to light stemming and improve the performance of stemming that benefit for many application like information retrieval, classification, …etc.

Experimental results showed that The researcher cannot avoid feature reduction for Arabic language to reduce complexity for classifiers, reduce storage requirements and to save time. Stemming / light stemming greatly reduced features to an average of 30% and 50% of the original feature space respectively. The researcher conclude that light stemming with term pruning is the best feature reduction technique with nouns because save meaning of words and light stemming is more proper than stemming from linguistics and semantic view point, and it has the least preprocessing time, it also has superior average classification accuracy. The root stem is better with verbs. because stemming has the slight classification performance than light stemming is that the majorities of Arabic words have a trilateral root, in fact between 80 and 85% of verbs in Arabic are derived from trilateral roots. The rest have a quad-letter root, penta-letter root or hexa-letter root

SVMs is a robust classifier even in high dimensions. Language consideration in NB variants improved performance. SVMs and NB variant have superior performance and achieved the best classification accuracy.

Term indexing and weighting aim to represent high quality text. The High quality in text mining usually refers to some combinations of relevance, novelty, and interestingness. Several approaches are used to index and weight terms but all of them share the following characteristics: The more the number of times a term occurs in documents that belong to some category, the more it is relative to that category. The more the term appears in different documents representing different categories, the less the term is useful for discriminating between documents as belonging to different categories. Term weighting schemes have direct impact on distance based classifiers. Distance based classifiers also affected by the used distance metric

7.2. Future Works

In the future works, The researcher shall work on extending and elaborating BBC Arabic corpus, CNN Arabic corpus, and OSAC corpus. Elaborations include performing extensive corpus analysis and tag them with Part of speech tags. The researcher also open the door for other researchers and contributors to elaborate the open source corpora.

The researcher shall develop a classifier that classifies any text document based on set of keywords, this will save the time to preprocess test text documents. Keywords are ranked based on different Language aspects based on semantic. Hierarchy classification is also will be supported by our future classifier.

References

[1] Aask Eikvil L, *Text Categorization: A survey*, Computing Center ed., Technical Report, Ed. Norwegian, 1999.

[2] A Abdelali, J Cowie, and H Soliman, *Building a modern standard Corpus, Workshop on computational modeling of Lexical Acquisition*, split ed.: In the Split Meeting, 2005.

[3] A Abdelali and J Cowie, *Regional Corpus of modern standard Arabic*.: Arabic language Engineering , 2005.

[4] S Al-Ansary, M Nagi, and N Adly, *Building an International Corpus of Arabic (ICA): Progress of Compilation Stage*. Alexandrina, Egypt: Bibliotheca Alexandrina, 2008.

[5] S Al-Fedaghi and F Al-Anzi, *A new algorithm to generate Arabic root-pattern forms*, 1989th ed. Dhahran, Saudi Arabia: In Proc. of the 11th National Computer Conf. King Fahd University of Petroleum & Minerals.

[6] S Al-Harbi, A Almuhareb , A Al-Thubaity, M Khorsheed , and A Al-Rajeh, *Automatic Arabic Text Classification*. France: JADT'08, 2008.

[7] A Al-Marghilani, H Zedan, and A Ayesh , *A general framework for multilingual text mining using self-organizing maps*. Innsbruck, Austria: In the 25th IASTED Int. Multi-Conf: Artificial Intelligence and Applications (AIA'07), 2007.

[8] A Al-Marghilani, H Zedan , and A Ayesh , *Text Mining Based on the Self-Organizing Map Method for Arabic-English Documents*. Cincinnati, USA: Midwest Artificial Intelligence and Cognitive Science Conf, 2008.

[9] R Al-Shalabi, G Kannan , and H Gharaibeh , *Arabic text categorization using KNN algorithm*.: In the Proc. of Int. multi conf. on computer science and information technology CSIT06, 2006.

[10] A El-Halees, *A Comparative Study on Arabic Text Classification*.: Egyptian Computer Science Journal 20(2), 2008.

[11] S Ghwanmeh, *Applying Clustering of Hierarchical K-means-like Algorithm on Arabic Language*.: In the Int. Journal of Information Technology, 3(3), 2005.

[12] G Kanaan , R Al-Shalabi, and Ghwanmeh S, *A comparison of text-classification techniques applied to Arabic text*.: Journal of the American Society for Information Science and Technology, 60(9), pp. 1836 – 1844, 2009.

[13] S Khoja and R Garside , *Stemming Arabic text*. Lancaster, UK: Computer Science Department, Lancaster University, 1999.

[14] S Mustafa and Q Al-Radaideh, *Using N-grams for Arabic text searching*.: Journal of the

American Society for Information Science and Technology, 55 (11), pp. 1002–1007, 2004.

[15] D Said, N Wanas , N Darwish , and N Hegazy , *A Study of Arabic Text preprocessing methods for Text Categorization*. Cairo, , Egypt, : In the 2nd Int. conf. on Arabic Language Resources and Tools, 2009.

[16] K Taghva, R Elkhoury, and J Coombs, *Arabic stemming without a root dictionary*.: Information Technology: Coding and Computing, ITCC, Vol. 1, pp. 152 – 157, 2005.

[17] K (Beesley, 1996) Beesley, *Arabic finite-state morphological analysis*.: In COLING-96: Proceedings of the 16th international conference on computational linguistics, vol. 1, pp.8994,1996. (Chen & Gay, 2002) Chen, A.,and Gey, F. Building an Arabic stemmer for information retrieval.In TREC 2002.Gaithersburg: NIST, pp 631-639, 2002..

[18] R Feldman and J Sanger , *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*.: Cambridge University Press, 2007.

[19] H AI-Serhan and A Ayesh, *A Triliteral Word Roots Extraction Using Neural Network For Arabic*.: The 2006 International Conference on Computer Engineering and Systems, , 5-7 Nov. 2006, pp.436-440., 2006.

[20] *ARABIC WORD FREQUENCY COUNTS*.: [Online]. Available: qamus.org/wordlist.htm, (2010, August).

[21] H Zhang , *The Optimality of Naive Bayes*.: In FLAIRS2004 conference, 2004.

[22] G Kanaan, AI-Shalabi, M Ababneh, and A AI-Nobani, *Building an effective rule-based light stemmer for Arabic language to improve search effectiveness*.: International Conference on Innovations in Information Technology, lIT 2008., 16-18 Dec., 2008.

[23] S AI-Fedaghi and F AI-Anzi, *A new algorithm to generate Arabic root-pattern forms*.: In proceedings of the 11 th national Computer Conference and Exhibition. PP 391-400., March 1989.

[24] R A1-Shalabi and M Evens, *A computational morphology system for Arabic*.: In Workshop on Computational Approaches to Semitic Languages, COLING-ACL98., August 1998.

[25] R Alshalabi, *Pattern-based Stemmer for Finding Arabic Roots*.: Information Technology Journal, vol. 4, no. I, pp. 38-43, 2005.

[26] R AI-Shalabi, G Kanaan, S Ghwanmeh, and F.M. Nour, *Stemmer Algorithm for Arabic Words Based on Excessive Letter Locations*.: 4th International Conference on Innovations in Information Technology lIT '07., 18-20 Nov. 2007, pp.456-460.

[27] S Khoja, *Stemming Arabic Text*.: Lancaster, U.K., Computing Department, Lancaster University., 1999.

[28] S Khoja and R Garside, *Stemming Arabic text*.: Computer Science Department, Lancaster University, Lancaster, UK, 1999.

[29] K Taghva, R Elkhoury, and J Coombs, *Arabic Stemming Without A Root Dictionary*.: in Proceedings of the ITCC 2005 International Conference on Information Technology: Coding and Computing, 2005.

[30] M Al-Kabi and R Al-Mustafa, *Arabic Root Based Stemmer*. Irbid, , Jordan, : in Proceedings of the International Arab Conference on Information Technology (ACIT 2006), 2006.

[31] L. S. Larkey, L Ballesteros, and M. E. Connell, *Improving stemming for Arabic information retrieval:*. Tampere, Finland, : in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02), 2002.

[32] M. N. Al-Kabi, Q. A. Al-Radaideh, and K. W. Akkawi, *Benchmarking and Assessing the Performance of Arabic Stemmers*.: Journal of Information Science, vol. 37, no. 2, pp. 111-119, April 2011.

[33] H Al-Sarhan, R Al-Shalabi, and G Kanaan, *New Approach for Extracting Arabic Roots*. Alexandria, Egypt: in Proceedings of the 2003 Arab conference on Information Technology (ACIT2003)., 2003.

[34] S Ghawanmeh, R Al-Shalabi, G Kanaan, K Khanfar, and S Rabab'ah, *An Algorithm for Extracting the Root for the Arabic Language*. Cairo, Egypt: in Proceedings of the 5th International Business Information Management Association Conference (IBIMA), 2005.

[35] H Al-Serhan and A Ayesh, *A Triliteral Word Roots Extraction Using Neural Network For Arabic*. Cairo, Egypt: in Proceedings of the 2006 International Conference on Computer Engineering and Systems, Nov. 2006.

[36] M Momani and J Faraj, *A Novel Algorithm to Extract Tri-Literal Arabic Roots*.: in Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications, AICCSA '07, May 2007.

[37] (Larkey, 2002) Larkey, S Leah, Ballesteros, Lisa, and Connell, *Improving Stemming for Arabic Information Retrieval*. Tampere, Finland: Light Stemming and Co-occurrence Analysis In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), August 11-15, 2002.

[38] R Sonbol, N Ghneim, and M.S. Desouki, *Arabic Morphological Analysis: a New Approach, in Proc. 3rd International Conference on Information and Communication*

Technologies.: From Theory to Applications ICTTA., 7-11 April 2008.

[39] Z Kchaou and S Kanoun, *Arabic stemming with two dictionaries*.: International Conference on Innovations in Information Technology IIT 2008, 16-18 Dec. 2008.

[40] (Darwish, 2002b) Darwish and D.W Oard, *CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval.*: In TREC 2002. Gaithersburg: NIST, pp 703-710, , 2002.

[41] (Aljlayl, 2002) Aljlayl and O Frieder, *On Arabic search: Improving the retrieval effectiveness via light stemming approach*. New York, USA: In Proceedings of the 11th ACM International Conference on Information and Knowledge Management, Illinois Institute of Technology (pp. 340–347).

[42] (Larkey, 2002) Larkey, S Leah, Ballesteros, Lisa, and Connell, *Improving Stemming for Arabic Information Retrieval*. Tampere, , Finland, : Light Stemming and Co-occurrence Analysis In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), August 11-15, 2002.

[43] (Larkey, 2005) Larkey, L Ballesteros, and Connell, *Light stemming for Arabic information retrieval*.: Arabic Computational Morphology: Knowledge-based and Empirical Methods. (LDC, 2001) LDC, Linguistic Data Consortium.

[44] Kadri (Kadri & Nie, 2006) and J. Y. (2006) Nie, *E_ective stemming for Arabic information retrieval*. London, , UK: The challenge of Arabic for NLP/MT Conference, The British Computer Society.

[45] A (Nwesri, 2005) Nwesri, S.M.M Tahaghoghi, and Falk Scholer, *Stemming Arabic Conjunctions and Prepositions,*. Buenos Aires, Argentina: In Mariano Consens and Gonzalo Navarro (eds.), Lecture Notes in Computer Science – Proceedings of the Twelfth International Symposium on String Processing and Information Retrieval (SPIRE'2005).

[46] A (Nwesri, 2007) Nwesri, S.M.M. Tahaghoghi, and Falk Scholer, *Arabic Text Processing for Indexing and Retrieval*.: Proceedings of the International Colloquium on Arabic Language Processing, Rabat, Moroc, 18-19 June, 2007. (proofing, 2003) Microsoft Office 2003 proofing toolkit,http://www.microsoft.com/middleeast/arabicdev/office/office2003/proofing.asp.

[47] S (El-Beltagy & Rafea, 2009) El-Beltagy and A Rafea, *A FRAMEWORK FOR THE RAPID DEVELOPMENT OF LIST BASED DOMAIN SPECIFIC ARABIC STEMMERS*.: Proceedings of the Second International Conference on Arabic Language Resources and Tools, 2009.

[48] J Goldsmith, *Unsupervised learning of the morphology of a natural language*.:

Computational Linguistics, 27 (2), pp. 153-198, 2000.

[49] D. W. Oard, G.-A. Levow, and Cabezas, *C. I. CLEF experi-ments at Maryland: Statistical stemming and backoff transla-tion.*: In Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop, C. Peters, Ed.: Springer Verlag, pp. 176-187, 2001.

[50] P McNamee, J Mayfield, and Piatko, *C. A language-inde-pendent approach to European text retrieval*.: In Cross-language information retrieval and evaluation Proceedings of the CLEF 2000 workshop, C. Peters, Ed.: Springer Verlag, pp.129-139, 2000.

[51] J Mayfield, P McNamee, C Costello, C Piatko, and A Banerjee, *JHU/APL at TREC 2001: Experiments in fil-tering*.: A. JHU/APL at TREC 2001: Experiments in fil-tering and in Arabic, video, and web retrieval. In TREC 2001. Gaithersburg: NIST, 2001., 2001.

[52] A. N. De Roeck and W Al-Fares, *A morphologically sensitive clustering algorithm for identifying Arabic roots*. Hong Kong,: In Pro-ceedings ACL-2000, 2000.

[53] Darwish and Waleed M. Arafa , *Linguistic Data Consortium*.: uckwalter Morphological Analyzer Version1.0,LDC2002L49,2002.http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49 Mohamed I. El-Disooqi,The Egyptian Computer Journal , Vol. 36 No. 1, June 2009 49.

[54] M Damashek, *Gauging Similarity with n-grams: Language-Independent Categorization of Text*.: Science 267, pp 843 –848, 10, February 1995.

[55] G. K. Zipf, *Human Behavior and the Principle of Least Effort, an Introduction to Human Ecology*.: Addison-Wesley Reading, Mass., 1949.

[56] G Kanaan, R Al-Shalabi, and M Al-Kabi, *New approach for extracting quadrilateral Arabic roots*.: Abhath Al-Yarmouk, Basic Science and Engineering vol. 14, no. 1, pp. 51–66, 2005., 2005.

[57] Al-Kharashi (Al-Kharashi & Evens, 1994) and Evens, *M. W. Comparing words, stems, and roots as index terms in an Arabic information retrieval system*.: JASIS, 45 (8), pp. 548-560, 1994.

[58] I Witten and E Frank , *Data Mining: Practical machine learning tools and techniques*.: (2nd Ed), Morgan Kaufmann, San Francisco, 2005.

[59] Janssen, *A. Segmentierung Franzosischer Wortformen in Morphe ohne Verwendung eines Lexikons*.: In Computatio linguae, U.Klenk, Ed. Stuttgart: Steiner Verlag, pp. 74-95, 1992. pp. 548-560, 1994.

[60] T Hill and P Lewicki, *STATISTICS Methods and Applications*.: (1st Ed), StatSoft, Tulsa,

OK., 2007.

[61] S Khoja , *An RSS Feed Analysis Application and Corpus Builder*.: [Online]. Available: www.elda.org/medar-conference/pdf/73.pdf , (2010, August).

[62] K Darwish , H Hassan , and O Emam, *Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval*.: Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 25–30, Ann Arbor, June 2005.

[63] U Klenk, *Verfahren morphologischer Segmentierung und die Wortstruktur im Spanischen*.: In Computatio linguae, U. Klenk,Ed. Stuttgart: Steiner Verlag, 1992.

[64] H (Abu-Salem, 1999) Abu-Salem, K Al-Omari, and M Evens, *Stemming methodologies over individual query words for Arabic information retrieval*.: JASIS, 50 (6), pp. 524-529, 1999. (Al-Ameed et al., 2005) Al-Ameed k. Hayder, Al-Ketbi O. Shaikha, Al-Kaabi A. Amna, Al-Shebli S. Khadija, Al-Shamsi F. Naila, Al-Nuaimi H. Noura, Al-Muhairi S. Shaikha,Arabic Light Stemmer: A new Enhanced Approach, The sec, 2005.

[65] F. C. (Gay & Oard, 2002) Gey and D. W. Oard, *The TREC-2001 cross-language information retrieval track Searching Arabic using English, French, or Arabic queries*.: In TREC 2001. Gaithersburg: NIST, 2002.

[66] S (Khoja, 1999) Khoja and R Garside, *Stemming Arabic text*.: Computing Department, Lancaster University, Lancaster, 1999.

[67] K Taghva, R Elkhoury, and J Coombs, *Arabic stemming without a root dictionary*.: Information Technology: Coding and Computing, ITCC, Vol. 1, pp. 152 – 157, 2005.

[68] *ARABIC WORD FREQUENCY COUNTS*.: [Online]. Available: qamus.org/wordlist.htm , (2010, August).

[69] H Zhang, *The Optimality of Naive Bayes*.: In FLAIRS2004 conference, 2004.

[70] G Flenner, *Ein quantitatives Morphsegmentierungssytem fur Spanische Wortformen*.: In Computatio linguae II, U. Klenk, Ed. Stuttgart: Steiner Verlag, pp. 31-62, 1994.