KEYPHRASE EXTRACTION FROM ARABIC SCIENTIFIC ARTICLES

FIRNAS HANCI

JANUARY 2015

KEYPHRASE EXTRACTION FROM ARABIC SCIENTIFIC ARTICLES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED

SCIENCES OF

ÇANKAYA UNIVERSITY

BY

FIRNAS HANCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF

COMPUTER ENGINEERING

JANUARY 2015

Title of the Thesis: **Keyphrase Extraction From Arabic Scientific Articles**

Submitted by **Fırnas HANCI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof.Dr.Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Müslim BOZYİĞİT
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist Prof. Dr. Gönenç ERCAN
Supervisor

Examination Date: 19.01.2015

Examining Committee Members

Assoc. Prof. Dr. Fahd JARAD	(T.H.K. Univ.)
Assist. Prof. Dr. Abdül Kadir GÖRÜR	(Çankaya Univ.)
Assist Prof. Dr. Gönenç ERCAN	(Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Fırnas, HANCI

Signature :

Date : 19.01.2015

ABSTRACT

KEYPHRASE EXTRACTION FROM ARABIC SCIENTIFIC ARTICLES

HANCI, Fırnas
M.Sc., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Gönenç ERCAN

January 2015, 41 pages

Keyphrases are very important tools for summarizing, clustering, indexing and searching documents. Many academic journals request from article authors a list of keyphrases summarizing their research articles. Despite the importance of keyphrases, unfortunately only a few of published Arabic articles contain them. Many algorithms and systems have been suggested and applied by automatically extracting keyphrases for many languages. In contrast to this rich literature, only a few articles have been written for the Arabic language.

In this thesis, an attempt will be made to extract keyphrases from Arabic articles, by making use of two methods; the first method uses a specialized stemming approach for extracting keyphrases. The second method splits the articles with respect to their main sections and determines the importance of the phrases in each section.

In this research a keyphrase extraction corpora for the Arabic language will be built, a new morphological processing strategy especially for keyphrase extraction will be implemented and this algorithm will be compared with two state-of-the-art algorithms, namely Kea and KP-Miner. The proposed morphological processing algorithm achieves superior results compared to these algorithms.

Keywords: Keyphrase Extraction, Arabic Scientific Articles, Kea, KP-Miner, Stemming.

ÖZ

ARAPÇA BİLİMSEL MAKALELERDEN ANAHTAR KELİME ÇIKARMA

HANCI, Fırnas
Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Yrd. Doç. Dr. Gönenç ERCAN

Ocak 2015, 41 Sayfa

Anahtar kelimeler, dokümanları özetleme, gruplandırma, indeksleme ve aramada çok önemli araçlardır. Birçok akademik dergi; makale yazarlarından, makaledeki çalışmalarını özetleyecek anahtar kelime listesini belirlemelerini ister. Anahtar kelimelerin önemine rağmen çok az yayınlanmış Arapça makale anahtar kelime içermektedir. Birçok algoritma ve sistem çeşitli dillerde anahtar kelime çıkarmak için kullanılır. Bu zengin literatüre rağmen, bu konuda sadece bir kaç makale Arapça dili için yazılmıştır.

Bu tez çalışmasında, iki yöntemden yararlanarak Arapça makalelerden anahtar kelime, çıkarma yapılacaktır. İlk yöntem; anahtar kelime çıkarmak için özel bir köklendirme yöntemi kullanılması, ikinci yöntem ise ana bölümlere göre makalelerin bölünmesi ve her bölümdeki anahtar kelimelerin öneminin belirlenmesidir.

Bu araştırmada Arapça dil için anahtar kelime çıkarmaya uygun korpus oluşturuldu. Anahtar kelime çıkarması için yeni bir morfolojik strateji uygulanacak ve bu algoritma anahtar kelime çıkarma konusunda en gelişmiş iki algoritmayla, yani Kea ve KP-Miner ile mukayese edilecektir. Önerilen morfolojik algoritma, bu algoritmalara göre daha verimli sonuçlar elde etmektedir.

Anahtar Kelimeler: Anahtar Kelimesi Çıkarma, Arapça Bilimsel Makale, Kea, KP-Miner, Köklendirme.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assist. Prof. Dr. Gönenç ERCAN for his supervision, special guidance, suggestions, and encouragement through the development of this thesis.

It is a pleasure to express my special thanks to my family for their valuable support.

To my brother Yılmaz HANCI for his advice at each stage when I consulted him and for his help in times of confusion.

To my friend Veli Hakan AKINCI for his encouraging and supporting me during writing this thesis

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGEMENTS vi

TABLE OF CONTENTS vii

LIST OF FIGURES ix

LIST OF TABLES x

LIST OF ALGORITHMS xi

CHAPTERS

1. INTRODUCTION 1

2. RELATED WORKS 3

2.1. Characteristics of the Arabic Language 3

2.2. Stemming Definition and Concept 4

2.2.1. Manual morphological processing 5

2.2.2. Delete affix 5

2.2.3. Statistical stemming 6

2.2.4. Morphological analysis 7

2.3. Keyphrase Extraction Algorithms 7

2.3.1. Genex: genitor and extractor 8

2.3.2. Kea: baseline characteristic set 9

2.3.3. KP-Miner: 12

2.3.4. Keyphrase extraction by neighborhood knowledge 13

2.3.5. Keyphrase extraction in scientific publications 14

3. THE METHOD 15

3.1. Phase I: Division of the Article Into its Main Sections 16

3.1.1. Section segmentation algorithm 18

3.2. Phase II: Stemming the Arabic Words 22

3.2.1. The first stage: deleting the diacritics 23

3.2.2. The second stage: identification of the stop words 24

3.2.3. The third stage: deleting the affixes 24

3.3. Phase III: Generating the Candidate Keyphrases 29

3.4. Phase IV: Selection the Appropriate Keyphrases. 31

3.5. Data Resources 32

4. RESULTS 34

4.1. Overview 34

4.2. Precision 34

4.3. Recall 35

4.4. Features of Candidate Keyphrases 35

4.4.1. Candidate keyphrases' features groups 36

4.5. Algorithm Results 38

5. CONCLUSION 41

REFERENCES R1

A LIST OF STOP WORDS IN THE ARABIC LANGUAGE A1

B LIST OF PATTERNS USED IN THE RESEARCH. A2

C CURRICULUM VITAE A4

LIST OF FIGURES

FIGURES

Figure 1 Processing phases to get a keyphrases list 16

Figure 2 Division structure for the sections of the Arabic articles......................... 16

Figure 3 The Arabic affixes classification 26

LIST OF TABLES

TABLES

Table 1 Different Grammatical Forms of Words in the Arabic Language............... 5

Table 2 Various Uses of the Term “Abstract” in Arabic Scientific Articles.......... 18

Table 3 Various Uses of the Term “Keyphrases” in Arabic Scientific Articles..... 19

Table 4 Various Uses of the Term “Introduction” in Arabic Scientific Articles.... 19

Table 5 Various Uses of the Term “Conclusion” in Arabic Scientific Articles..... 20

Table 6 Various Uses of the Term “Reference” in Arabic Scientific Articles....... 21

Table 7 Collection of Words in Arabic Under the Same Root.............................. 23

Table 8 The Letters Diacritics in Arabic Language... 24

Table 9 List of Symbols.. 30

Table 10 The General Classification for Articles that are Used in this Thesis......... 33

Table 11 Features of Candidate Keyphrases 36

Table 12 Candidate Keyphrases’ Features Gruops.. 37

Table 13 Results of the Algorithm 39

LIST OF ALGORITHMS

Algorithm 1 Deletion of the explicit prefixes 27

Algorithm 2 Deletion of the explicit suffixes 27

Algorithm 3 Deletion of the vague prefix 28

Algorithm 4 Generating the candidate keyphrases 31

CHAPTER 1

INTRODUCTION

Keyphrases are list of phrases which express the main concepts mentioned in a document [1]. As the amount of digital data and information content grows very rapidly, keyphrases can be utilized to manage the process and analysis of handling these large amounts of data. Keyphrases have an important role in content management systems, Internet contents and electronic libraries, especially for information retrieval and cataloging aims [2]. There are more than 200 million people in the world out of which about 3.8 percent speak Arabic [3] and many of the Arabic publications of scientific and non-scientific articles lack keyphrases assigned by their authors. Thus, inventing tools for extracting keyphrases automatically from published texts are essential. These tools can support several variety kinds of information over retrieval and analysis systems. So, these tools can supply automation in different areas:

- Summarizing documents for prospective readers. Keyphrases can represent a highly condensed summary of the document in question [4].
- Comparing the similarity between documents, and finding the potential to cluster and classify documents [5].
- Generating metadata that gives a high-level characterization of a document's contents. Providing additional information to tools for text-mining related tasks such as document and Web page retrieval.
- Highlighting important topics within the body of the text, to facilitate speed reading (skimming), which allows deciding whether it is relevant or not [2].

The keyphrases help the reader to have basic knowledge about the article and give him the ability to know if the document is within his interest. When they are added to the cumulative index of a journal, the goal is indexing. They enable the reader to quickly find a relevant article when the reader is searching for specific information.

When a search engine form has a field labeled keyphrases, the aim is to help the researcher conduct the search more effectively. Searching the requested query in full-text of the document takes long time and effort. However, when the query matches with one of the documents' keyphrases, the results will be better in both efficiency and effectiveness. Keyphrases can aid in these diverse scenarios, as they aim at producing a short list of words or phrases that capture the major meaning of the article.

Many prominent endeavors have been suggested and executed for automatically extracting keyphrases for articles in English or other languages [6][7]. In contrast, not much research done in the literature targeting articles written in the Arabic language [2]. Within this inadequate research, stemming algorithms designed for general use are applied, but stemming algorithms designed for the properties of keyphrases extraction problem are not used. In this master's thesis research, a novel stemming algorithm to extract keyphrases has been designed. Through the experiments to be stated hereafter, it will be concluded that this method achieves better results.

Due to the reasons mentioned above, a new Arabic keyphrase extraction corpus has been created and a special morphology tool for Arabic keyphrase extraction has been developed. In addition, the effect of the structure of Arabic scientific articles in keyphrase extraction has been investigated.

The second chapter contains a literature overview of keyphrases extraction, with a focus on keyphrase extraction from Arabic documents. The third chapter shows in detail the algorithms used in this thesis in two subsections. The first one presents the new stemming algorithm mentioned above, designed especially for keyphrase extraction features. The second subsection introduces our algorithms for dissecting an Arabic article into seven main sections and discusses its use in Arabic keyphrase extraction. The fourth chapter presents the results of the experiment and compares them with state-of-the-art algorithms in keyphrase extraction. Finally, chapter five concludes with a discussion related to the work done in this thesis.

CHAPTER 2

RELATED WORKS

This thesis deals with the stemming and keyphrase extraction algorithms used in the Arabic language. In the next chapters the focus will be on the stemming mechanism and keyphrase extraction algorithm that is developed through this thesis. For that reason, this chapter focuses on literature related to these two algorithms which will be used to compare with the new developed algorithm.

2.1 Characteristics of the Arabic Language

The Arabic language is a very rich and complex language. The Arabic language contains twenty- eight letters and nine diacritics written from right to left as opposed to European languages. In addition, the phenomenon and morphological difference in the Arabic language, causes the Arabic language to be considered even more complex in its morphological representation [8]. The morphological analysis process for the Arabic language is difficult. The Arabic language has 11,347 roots from which all the verbs and nouns are derived. These roots are mostly formed of three letters, but there are also roots made up of four or five letters. However, the number of three letter roots is more than the others.

Verbs, nouns and particles are three main classes for Arabic words. From roots, all nouns and verbs can be generated. Some of the root characters can be removed or updated through morphological derivation. In addition, when a word is followed by certain suffixes, the inflectional form can change. A similar change in the word is also observed when it is preceded by prepositions or certain prefixes. Moreover, a major change in the meaning of the word is observed when an infix is added to the middle of the word. Furthermore, the root characters can be removed or updated through morphological derivation [9].

Also, the major change in the meaning of the word can be observed when the diacritics of the word are changed. The changes in the diacritics of the word can change a verb to a noun and vice versa. Unfortunately, most of the words mentioned in the scientific articles are written without the diacritics on the letters. Instead, the readers try to detect the diacritics from the location of the word in the sentence.

2.2 Stemming Definition and Concept

The surface form of the word in the documents and articles may vary as affixes are used depending on the grammar rules of the language. A word can be written in many different forms; for example in English the word “go” can be written in forms like “go”, “ goes”, “going”, “went”, and “gone”. Despite this difference, all these forms share the same meaning “go”. Similar variations are observed in the Arabic language as well. For example, the word “كتاب” which translates as “book” can be written in other forms such as “كتابي – my book”, “كتب - books”, “كتبي –my books”, “الكتاب – the book” and “بالكتب – in the book”.

In order to know the number of available words that refer to the same concept (Lexeme) in one article, stemming or the lemmatization algorithms are used. Stemming is “a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes” [10]. Lemmatization means “using of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma” [10].

However lemmatization requires algorithms with extensive knowledge about the vocabulary, morphological and inflectional analysis of words, which is expensive both in terms of computation and resources. In contrast, stemming algorithms simply remove additional letters that may be in the affixes of the words without an in depth analysis, so that these words are transformed into the same form.

Table 1 shows the different forms of the word school and the difference between the word forms in the text and the Arabic language inflection of the word forms. On the other side, the word root in the Arabic language and the meaning of the word in the language dictionary is shown.

Different grammatical forms	Root word	Lemma
“مدرسة” - a school	“درس” - a lesson	“مدرسة” - a school
“مدرستان” - two schools		
“مدارس” – schools		
“المدرسة” - the school		

Table 1 Different Grammatical Forms of Words in the Arabic Language

2.2.1 Manual morphological processing

Manually constructed dictionaries were used in the early stages of the stemming of the Arabic texts. Al-Kharashi and Evens algorithms search article words using a manually built word-stem-root dictionary for each index term, to find the roots and patterns of the words for classification purposes [9]. This approach is not practical for processing large volumes of data because it takes a long time and manual labor to build the dictionary [11]. For this reason it is limited in terms of the size of the dictionary. Even for a small document set the total number of unique words is 1,125, the total number of stems is 725 and the total number of roots is 526 [12]. These numbers will rapidly increase as the number of documents increase.

2.2.2 Delete affix

The deletion process for the affix is commonly known as Light Stemming, and this approach does not make complex processes like deletion of infixes, and comparison

with patterns and finding the root of the word; it just scans the word for prefixes and suffixes. There are many scientific sources using Light Stemming technique but some authors do not mention how they delete the suffixes and prefixes. Also, they do not explain the algorithms used in the implementation of stemming technology [13] [14]. But the Light10, developed by Larkey and Ballesteros, uses a prefixes list (و، ال، وال، بال، كال، فال، لل) and a suffixes list (ها، ان، ات، ون، بن، يه، ية، ه، ة، ي) [15]. These prefixes and suffixes are not all the affixes used in the Arabic language. Darwish has published the Al-Stem Light Stemmer algorithm [16]. After comparing the Al-Stem Light Stemmer algorithm with the Light10 algorithm, Light10 got better results [15].
Algorithms differ from each other by the deleted affixes; for example Darwish uses a larger list of affixes [17]. In the Arabic language, a root or original word takes different forms in a sentence according to its context. For that reason, most of the algorithms contain mistakes in the conversion of different forms to the root or original word [15].

2.2.3 Statistical stemming

The statistical stemming approach is multilingual; which means it can be implemented for many different languages. The factors described in Section 2.1 about the complex compositions of words and linking mechanisms for the prefixes, suffixes and infixes in the Arabic language have negative impacts on the results of the statistical stemming technique. Results of the n-gram technique which uses the statistical stemming approach proved that this approach is not suitable for the Arabic language, because the infixes of the Arabic language complicate statistical data collection. [15].

Larkey has applied co-occurrence approach merged with n-gram, belonging to Xu and Croft [18] [15]. Larkey formed classes of words that mapped onto the same string if vowels were removed, and then the co-occurrence measures are used in order to further divide these classes to subclasses. The co-occurrence algorithms for the English and Spanish languages are modified with respect to properties of the Arabic language. Even with these modifications the co-occurrence method does not improve the results obtained with the Light10 algorithm [15].

2.2.4 Morphological analysis

Khoja and Garside have used the morphological analysis approach for stemming the Arabic texts [19]. Khoja and Garside applied several steps for stemming the words. The first step deletes the prefixes and suffixes from the words; the next step compares the results with a list of patterns and roots. If resulting strings match with patterns or roots, the approach adopts the results; otherwise in the case of a mismatch, it adopts the original word without a change. This stemming technique also deletes the 168 stop words [19].

When a root includes a weak character like ("أ"-alif, "و"-waw or "ي"-yah), it mostly changes the form of this character through derivation. Therefore, in order to deal with this, the stemmer will need to determine whether the weak character is in the right form. If it’s not, the right form for the weak character will be produced by the stemmer, then, the correct form for the root will be given [19]. Some words do not have roots; for example, the Arabic equivalents of “we,” “after,” “under” and so on. If the stemmer comes across any of these words, it leaves them as they are [19].

Occasionally a root character can be removed through derivation. This is mostly true for roots which contains repetitive characters like (the last two letters are the same). In this case the stemmer will determine this and retrieve the character which has been deleted. When a root includes a "ء"-hamza, this could change the form through derivation. This will be determined by the stemmer, then the original form of this Hamza will be retrieved [19].

This approach is efficient in its results but weak with foreign rooted words. [15]

2.3 Keyphrase Extraction Algorithms

After the stemming of the original text, the next stage is the extraction of keyphrases. There are different algorithms to extract keyphrases for many languages.

2.3.1 Genex: genitor and extractor

GenEx keyphrase extraction algorithm contains two components, the Genitor genetic algorithm [20] and the Extractor [21][22][23]. The documents are inputs for The Extractor and a list of words and phrases are the result of the extraction process. The Extractor generates output depending on a dozen of numerical parameters. The parameters and their values are set in order to optimize the overlap between the Extractor's output and authors' assignments. The Genitor is used to tune the Extractor in the training phase. Once the parameter values are learned, the Genitor is no longer needed.

It is important to decide whether the two phrases are a match or not, when we want to determine the similarity between the algorithm's phrases and the author's phrases. If the computer extracts the phrase "distributed computing" and the author uses the phrase "Distributed Computation" in a document, this should be accepted as equivalent phrases. Both Kea and GenEx utilize the same tactic for the identification of matches: by converting the phrases to lower case, stemming, and finally normalization. This two algorithms, using the iterated Lovins stemming algorithm, which applies this algorithm many times, until no changes happen in the word. [21][24].

By examining the input document containing series of one, two, or three successive words, the Extractor generates candidate phrases. The successive words have two conditions. The first condition is this words should not be isolated with any punctuation and the second condition is this words should not contain any stop word (like as "if", "to", "of", "and", "he", "the", etc.). The candidate phrases that generated by the algorithm are normalized in two steps. First step converting the candidate phrases to lowercase and in second step, apply the stemming algorithm on them.

GenEx has been explained in more detail in articles [22][23]. In this research, we did not focus on the GenEx algorithm's details but we focus on the features utilized to choose the candidate phrase for output.

GenEx requests from the user to determine the number of phrases that require in the result of algorithm. When the user requested N phrases as algorithm outputs, different features are utilized to compute a weight for all candidate phrase and the result of GenEx will be the first N candidate phrases that have highest weight. N in the current version of GenEx can be in a range from three to thirty. The last step after a candidate phrase has been chosen for the output is to get the original form by return all affixes and the original form of letter case.

Different experiments indicated that GenEx capability is acceptable when it is trained on a single area and then tested on different area. [21][22][23]. With as few as fifty training documents, GenEx gets good results. The strength of matches between the phrases appointed by the articles authors and the GenEx result rely on the number of the phrases wanted by the user. When the user requests 7 phrases, about twenty percent of the GenEx algorithm results will match with phrases that appointed by the article's author.

This result is identical to the degree of accord amongst several people, appointed keyphrases to the same article [25] A more exact result is gained by requesting readers to rate the quality of the computer's result. For the specimen of 205 readers rating keyphrases for 267 Web pages, 20% were unrated, 18% were rated as "bad", 62% of the 1,869 GenEx's result phrases were rated as "good" [23]. It is considered more than enough when 80% of the phrases extracted by an application are considered agreeable (not "bad") to human readers.

2.3.2 Kea: baseline characteristic set

Kea create candidate phrases similar to the Extractor approach [26][27][28]. Kea after that, apply the Naïve Bayes algorithm to learn the classification for the candidate phrases [29]. In one version of Kea, candidate phrases are classified using only two features: TF - IDF and distance [26][27][28]. TF - IDF is known as "baseline feature set" [AAAfirnas].

"TF - IDF (Term Frequency times Inverse Document Frequency) is commonly used in information retrieval to assign weights to terms in a document" [30]. This numerical characteristic specifies a top rate to a phrase that is comparatively repeated in the given article (the TF part) which appears rarely in other article (the IDF part). The TF-IDF component calculated in Kea Algorithm as follows [26]:

$$TF(P,D) \times \mathrm{IDF(P,D)} = \frac{freq_d(P,D)}{size_d(D)} \times \log_2\left(\frac{\mathrm{freq_c(P,C)}}{\mathrm{size_c(C)}}\right) \qquad (2.1)$$

TF(P, D) is represent to the probability of the phrase P mentioned in the document D, evaluated by calculating the total number of P that mentioned in D, freq_d(P, D), and dividing by the total number of words in document D, size_d(D). IDF(P, C) is the negative log of the probability that phrase P mentioned in any document in corpus C, evaluated by calculating the total number of documents in corpus C that include phrase P, freq_c(P, C), and dividing by the number of documents include in corpus C, size_c(C)[31].

The freq_phrasefeaturein GenEx is similar to the TF component of TF-IDF in Kea. Turney has found in GenEx that the TF without IDF works fine for extracting the keyphrase[31]. It is probable which the relative_length characteristic in GenEx serves as a surrogate for IDF [32]. Then, TF TL can be used like alternate to TF-IDF, which the TL is Term Length. The reason behind why they want to replace TL with IDF is that longer words tend to have less frequencies than shorter words. A characteristic for the TL through IDF is that TL is easiest to compute. To avoid taking the logarithm of zero, in Kea, it's clear the freqc(P, C) and sizec(C) are increased by one, but TL does not need this type of adjustment [31].

Moreover, the same IDF value should be specified by Kea to whole out-of-corpus phrases, while the phrases of the same value are the same length, they will be specified by TL. To explain the distance characteristic in Kea is, for example in a specific phrase in a specific document, the number of words that precede the first occurrence of the phrase, will be divided by the number of words in the document. This matches with first_occur_phrase feature in GenEx. A candidate phrase in Kea, with a capitalization pattern that denotes a suitable noun is erased; it is not taken into consideration for the output. The proper_noun characteristic and the document_length attribute in GenEx tend to work jointly. Through training, GenEx tends to understand to obviate proper nouns just for long documents, but permit them for short documents [31].

Distance features and the TF-IDF are real-valued. Fayyad and Irani's algorithm were used by the Kea to discretise the features Minimum Description Length (MDL) technique used by this algorithm in order to partition the features into intervals in a way to minimize the entropy of the class with respect to the intervals [31].

The Bayes formula is utilized by the Naïve Bayes algorithm in order to calculate the probability of membership in a class. The “Naïve” is used to signify the assumption of statistically independence between the features. Assuming the <T, D> is a feature vector for the candidate phrase, where D is an interval of the discretised distance feature and T is an interval of the discretised TF-IDF characteristic. We can calculate the probability which the candidate phrase is a keyphrase, p(key| T, D) by utilizing Bayes formula and the independence assumption, as follows [26]:

$$p(key \,|\, T, D) = \frac{p(T \,|\, key).p(D \,|\, key).p(key)}{p(T,D)} \qquad (2.2)$$

The p(T | key) in this equation is probability which discretized TF-IDF characteristic own a estimation in the interval T, suppose that the candidate phrase is really a keyphrase. p(D | key) is the probability which the distance characteristic own a estimation in the interval D, suppose that the candidate phrase is really a keyphrase. The previous probability is the p(key) which the candidate phrase is a keyphrase. The normalization factor is p(T, D), for making the value of p(key | T, D) ranging from 0 to 1. p(T, D) is the probability of < T, D> while the class is not well-known. From the frequencies in the training data, these probabilitys can be readily evaluted

To evaluate the probability which a candidate phrase < T, D> is a keyphrase, the p(key | T, D) can be utilized after training. By the evaluated the probability p(key | T, D) that they belong to the keyphrase class, the Kea can ranks all of the candidate phrases. The Kea gives the top N phrases with the topmost probability as output when the user asks N phrases. The experiments showed there are no important diversity in performance when comparison the GenEx to Kea using the baseline feature [26]. Another tests showed the Kea can change for the better the browsing in a digital library, with automatically generating a keyphrase index [33], or by automatically generating hypertext links [34]. Kea 5.0, is a current version which is written in Java while The Kea source code is exist under the GNU General Public License.2 [35]. In the following experiences in this thesis Kea 1.1.4 has been used, which is a mixture of Perl, C, and Java.

2.3.3 KP-Miner:

KP-Miner (El-Beltagy&Rafea) [36] is an unsupervised machine learning algorithm which is dependent on three main steps: candidate keyphrase selection, candidate keyphrase weight calculation and finally keyphrase refinement. KP-Miner application was obtained from the Internet [37].

2.3.3.1 Candidate keyphrase selection

To configure a list of candidate keyphrases, phrases that are divided by signs of sanction or stop words will be specified. The list of stop words consists of 187 words. After the application of this principle, a long list of candidate key phrases will remain. To reduce this number, some additional rules will be used to filter some of the candidate phrases.

The first filter for the candidate keyphrases in the first stage is to check the frequency of the phrase, i.e. all candidate phrases appearing less than **Ω** times in the document are ignored. While in the English language documents **Ω**=3 is chosen, **Ω**=2 is chosen for documents written in the Arabic language; because of the differences between the two languages. The second condition depends on the location of the candidate keyphrases, which are mentioned in the beginning of the document, are more likely to continue to the second stage of the algorithm.

2.3.3.2 Candidate keyphrase weight calculation

Each candidate keyphrase is assigned a value according to the following equation

$$w_{ij} = Tf_{ij} \times Idf \times B_i \times Pf \qquad (2.3)$$

Where w_{ij} is the weight of term t_j in Document D_i, tf_{ij} is the frequency of term t_j in document D_i Idf is log 2N/n ,where N is the number of documents in the collection and n is the number of documents where term t_j occurs at least once. If the term is a compound, n is set to 1. B_i is the boosting factor associated with document D_i, Pf is the term position associated factor. If position rules are not used this is set to 1 .

2.3.3.3 Final keyphrase refinement

Kp-Minar, generates n number of keyphrases. The value of n will be determined at the beginning by the user. The final step in the algorithm is to sort candidate keyphrases depending on the weight of the candidate keyphrases. Finally, the top n keyphrases are returned from candidate keyphrases list.

2.3.4 Keyphrase extraction by neighborhood knowledge

Wan and Xiao have developed a method that would extract keyphrases from a single document. They want to extract key phrases from a single document, however, they did not confine all their knowledge to only a specific document; instead, they used a set of neighboring documents similar to the required document [38].

For example, they take two documents about the same subject "Health" where in these documents similar phrases, like "Health" and "Prevention" occur. Merging information from these documents provides more information yielding a better measure to extract notable keyphrases [38].

Wan and Xiao are using the graph-based ranking algorithm for praxis single document to extract the keyphrases. They use tow features, the word relationships in the specified document and the word relationships in the neighbor documents. The previous relationships point to the existing information in the specified document whereas the latter relationships indicate the preexisting information derived from the set of neighboring documents. In addition, they used TF-IDF equation to calculate the relationships. [38].

In the research for this thesis, a set of documents were worked with which were specifically collected to test and train the algorithm. For that, a collection of documents in order to calculate the TF-IDF equation were used, especially in calculating the IDF part. In contrast, Wan and Xiao used the neighbor documents to calculate the IDF part from the TF-IDF equation.

2.3.5 Keyphrase extraction in scientific publications

Nguyen and Kan'research focused in a specific domain. They extract keyphrases from scientific articles. They chose the Kea algorithm as the baseline framework for comparison. They added some extra features compatible with the Kea algorithm. The added features use the positions of phrases in the document related with sections. Also, they utilizes determined terminologically productive suffixes or added characteristics which capture notable morphological phenomena that exist in scientific keyphrases, like the candidate keyphrase being an acronym [39].

In the work done for this thesis, the same divisions for the scientific article are used. Different than previous studies, the focus here will be on the first location, frequency and the TF-IDF features of the phrases in the section.

CHAPTER 3

THE METHOD

For this thesis the method begins with a comparison of machine-created keyphrases and the human-created keyphrases. A human-created keyphrase is equal to a machine-created keyphrase when they equal to the same series of stems. A stem is what resides when the suffix of the word is deleted. According to this definition, “big windows” is equal to “big win”, but not equal to “windows”. The order in the series is important, so “mavi marmara” is not equal to “marmara mavi”.

One of the aims in this thesis is to build a keyphrase extraction algorithm. This algorithm will increase the number of keywords extracted from the articles that are equal to the keywords determined by the Authors of the articles. This algorithm consists of the phases shown in Figure 1.

The keyphrases extraction algorithm designed for this thesis is made up of four phases. In the first phase the article is divided into its main sections. In the second phase the stemming algorithm is applied to the Arabic words. In the third phase the candidate keyphrases are generated and the statistical features are calculated. In the fourth phase the appropriate keyphrases are selected.

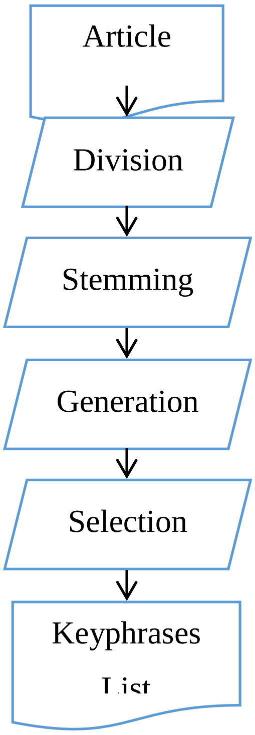

Figure 1 Processing phases to get a keyphrases list

3.1 Phase I: Division of the Article Into its Main Sections

The basis of this research depends on the adoption of the division of the article into several main sections and weighing these sections depending on the strength of each section and the range of the keyphrases mentioned in it. Each article is divided into 7 main sections as shown in Figure 2.

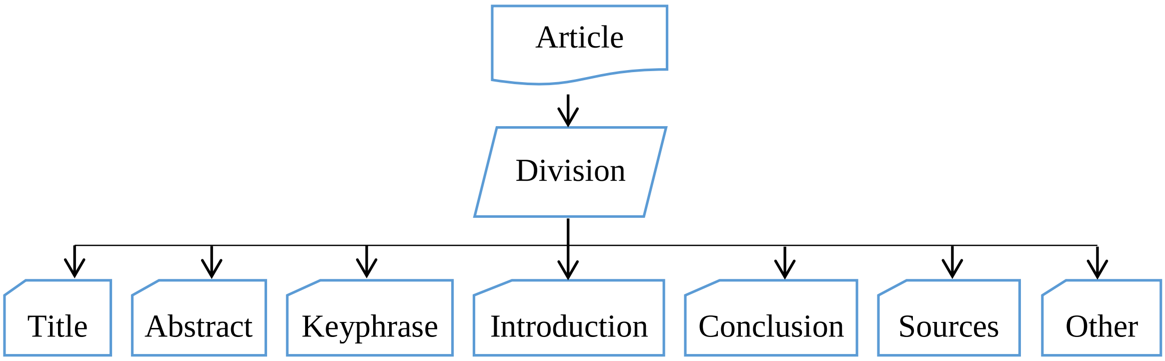

Figure 2. Division structure for the sections of the Arabic articles.

The main reason for the division of the article into its sections is to clarify the content of the scientific article. Where the title consisting of one phrase contains the purpose of the article, the abstract includes a brief description of the article in a few paragraphs. The keyphrases section contains the keyphrases that have been assigned by the author of the article. It is a set of varying number of phrases, each with a varying number of words mentioned in the article. The introduction section shows the importance and the reasons for writing the analyzed article. It consists of several paragraphs which do not go into the exact details about the article or the details of the mechanism of the work, but just serves to introduce the key points and structure of the article.

The conclusion section states the results of the scientific research and the discussions that are focused on the resulting recommendations in the research. Its number of paragraphs is more than the abstract and is usually less than the introduction. In addition, the references section lists the scientific sources used in the article, consisting of only article titles and author names. The rest of the article consists of the sections not included in the previously presented sections. This unclassified section depends on the context of the article and contains more words than the previously classified sections and is considered as the largest section in size and detail. This section for the method of division for this thesis is named as the “Other section”.

The “Other section”, mentions the extraction details about the research, work mechanism and the scientific and analytical comparisons about the results for the research topic.

The reasons for the division of the scientific article to these seven sections can be summarized by considering these sections to serve a specific purpose in the scientific articles, as these sections are common among scientific articles in general.

The fact is that the content of these sections differ much among themselves in terms of style. On the other hand, these sections have different number of words which has a significant impact in determining the keyphrases of the article. In this research, algorithms and mechanics revolve around proving this theory.

3.1.1 Section segmentation algorithm

The algorithm that is adapted to segment articles to its sections depends on the title of the sections in the article. The only exception to this is the title section which is located in the beginning of the scientific article in a single line. This simple rule was enough to determine the title section of the articles.

The abstract section, starts from the "abstract" term in the article. The articles used as the input data for the research topic showed that the Arabic articles do not depend on a specific term to determine abstract section. Researchers used nine different terms as title of this section (Show in Table 2) The abstract section ends with the beginning of the "keyphrases" term, and this term has six different forms like the "abstract" term (Show in Table 3).

Abstract Term	**Meaning in English**	**Abstract Term**	**Meaning in English**
الخلاصة	The Abstract	مستخلص	Extract
مستخلص البحث	Thesis extract	المستخلص	The Extract
خلاصة البحث	The Thesis Abstract	الملخص	The Summary
ملخص البحث	The Thesis Summary	الملخّص	The Summary
ملخص	Summary		

Table 2 Various Uses of the Term “Abstract” in Arabic Scientific Articles

KeyphrasesTerm	Meaning in English	Keyphrases Term	Meaning in English
الكلمات الدالة	The Function words	كلمات مفتاحية	Key words
الكلمات المفتاحية	The Key words	كلمات جوهرية	Substantial words
الكلمات الجوهرية	The Substantial words	كلمات دالة	Function words

Table 3 Various Uses of the Term “Keyphrases” in Arabic Scientific Articles

The introduction section starts from the "Introduction" term, which has eight different forms (Show in table 4). All the paragraphs that make up this section were compiled into one paragraph for this research, in order to determine the end of this section so that the end of the paragraph would follow the title of this section and would be considered the body of the introduction section. Due to the diversity in the style of writing by the writer, no specific method or even several methodologies could be applied to automatically detect sections following the introduction section.

Introduction Term	Meaning in English	Introduction Term	Meaning in English
المقدمة	The Introduction	تقديم	Presentation
مقدمة	Introduction	المقدّمة	The Introduction
توطئة	Foreword	التمهيد	preface
المدخل	Entrance	الإطار العام للدراسة	The general framework of the study

Table 4 Various Uses of the Term “Introduction” in Arabic Scientific Articles

The conclusion section starts from the beginning of the "Conclusion" term, which has twenty different forms like "Conclusion", "discussion", "conclusions" and "recommendations" in general, but in different ways such as plural or singular form (Show in table 5). The end of this section is the start of the sources section.

Conclusion Term	Meaning in English	Conclusion Term	Meaning in English
الخاتمة	Conclusion	الاستنتاجات	Illation (s)
الاستنتاجات والتوصيات	Conclusions and recommendations	النتائج والتوصيات	Results and recommendations
التوصيات والنتائج	Recommendations and Results	النتائج والمناقشة	Results and discussion
النتائج والتوصيات والمقترحات	Results and recommendations and proposals	خاتمة البحث	Conclusion research
النتائج والخلاصة والتوصيات	Results and Conclusions and Recommendations	نتائج وتوصيات البحث	Results and recommendations of the research
النتائج والاقتراحات	Results and Suggestions	عرض النتائج وتفسيرها	Results and interpretation
الخلاصة والاستنتاجات	Summary and Conclusions	عرض نتائج البحث	Display search results
الاقتراحات والتوصيات	Suggestions and recommendations	نتائج الدراسة	Results of the study
نتائج الدراسة ومناقشتها	Results of the study and discussion	التوصيات والمقترحات	Recommendations and proposals
توصيات الدراسة	Recommendations of the study	ملخص نتائج الدراسة	Summary results of the study

Table 5 Various Uses of the Term “Conclusion” in Arabic Scientific Articles

The sources section begins with the "sources" term, which has seventeen different forms like "references" and "margins" in general, but in different ways such as in plural or singular form (Show in Table 6).

Reference Term	Meaning in English	Reference Term	Meaning in English
مراجع ومصادر البحث	References and sources of research	قائمة المراجع	References list
المصادر والمراجع	Sources and references	مصادر البحث	Research sources
المصادر العربية والأجنبية	Arab and foreign sources	الهوامش والمصادر	The margins and sources
هوامش البحث	Research margins	المصادر	The sources
المراجع المعتمدة	Approved References	المرجع	Reference
مصادر البحث وهوامشه	Research sources and margins	المراجع	The references
قائمة المصادر والمراجع	List of sources and references	الهوامش	The margins

Table 6 Various Uses of the Term “Reference” in Arabic Scientific Articles

The other section is the “Section”, which is located between the “Introduction” section and the “Conclusion” section. At the end of this process, each article is divided into seven different sections for the next steps.

All these manual steps are done just to collect data for testing the algorithm.

3.2 Phase II: Stemming the Arabic Words

The referring of the light stemming can be describing as process on the small set of prefixes to make it stripping off, without the affixes deletion, or finding the roots of the patterns recognizing [15]. Another expressions, like “shallow” stemming [15] or “elementary” stemming [40], are utilized for the same significance. The concept of light stemming like “S” stemming algorithm is described by Harman, just as little popular word endings were deleted: “ies”, “es”, and “s” (with specific exclusion) [15]. Like the word “light” proposes, the term is utilized to refer the inverse of heavy stemming that the all set of the probable suffixes and prefixes are deleted. All of the two strategies have their own weaknesses and strengths.

Many stemming methods to stem Arabic words are varied through their degrees from light to heavy stemming. As mentioned in Section 2.1 the Arabic words are generated from their three lettered roots. This leads to the collection of many words under the same root. Table (7) shows an example of these combinations gathered under the same root. This example demonstrates that the adoption of heavy stemming leads to incorrect results. Turney and the Kea Algorithm for the English language and the Khoja algorithm for the Arabic language are examples of heavy stemming [19]. Despite the success of this algorithm for the English language, using this algorithm in the Arabic language gives incorrect results. Accordingly, for this thesis light stemming methods have been relied upon mainly for the morphological processing of Arabic words in the keyphrase extraction.

In Table (7) an example for group words that have different meanings deriving from a single root are shown. In addition, the pronunciations of these words are also shown in the table. In the last column of the table, the acceptable results for stemming these words are shown.

Arabic Original Word	Meaning in English	Arabic word with diacritics	Root of the word	Expected results
الجمع	Add	الجَمْعُ	جمع	جمع
الجماع	Intercourse	الجِمَاعَ	جمع	جماع
الجامع	mosque	جَامِعَ	جمع	جامع
المجموعة	group	المَجْمُوعَةْ	جمع	مجموعة
الجامعة	university	الجَامِعَةْ	جمع	جامعة
تجمع	Combines	تَجْمُعَ	جمع	تجمع
جموع	Crowds	جُمُوعَ	جمع	جموع
مجمع	Complex	مُجَمَعَ	جمع	مجمع
الجمعة	Friday	الجُمُعَة	جمع	جمعة
الإجماع	Consensus	الإجْمَاع	جمع	اجماع

Table 7 Collection of Words in Arabic Under the Same Root

3.2.1 The first stage: deleting the diacritics

Aforementioned in section 2.1 the fact that most of the words that exist in the scientific articles are written without diacritics on the letters, therefore deleting all the diacritics that are rarely available in some words is chosen as a method for this thesis.

Table 8 shows the list of Arabic letters diacritics. These diacritics are used to assist the pronunciation of the letters. These diacritics are not used in general, because the pronouncement of the words can be derived from the context of the sentence. For this reason they are usually omitted in Arabic texts.

Diacritics	In Arabic	Symbol
Fathah	الفتحة	◌َ
Kasrah	الكسرة	◌ِ
Dammah	الضمة	◌ُ
Sukoon	السكون	◌ْ
Tanween	التنوين	◌ٍ ◌ٌ ◌ً
Shaddah	الشدة	◌ّ
Maddah	المدة	~

Table 8 The Letters Diacritics in Arabic Language

3.2.2 The second stage: identification of the stop words

At this stage, all the stop words used in the Arabic language, 185 words, are listed. For example, “and-و”, “on-على”, “in-في” are some of these stop words. The complete list is given in Appendix (A). At the beginning these words were not deleted because they played an important role in determining the candidate keyphrases. In the following paragraph details are given about the role of the stop words in determining the keyphrases .

3.2.3. The third stage: deleting the affixes

Most stemming algorithms are designed to strip off strings that were frequently found as prefixes or suffixes, but seldom the ones found at the beginning or ending of stems. Most of the stemming algorithms use a part of the Arabic list of prefixes and suffixes.

For example, Light10 deletes a list of prefixes ("لل", "فال", "كال", "بال","وال","ال") and a list of suffixes ("ات", "ان","ها", "ة", "ي","ه","يه","ية", "ين","ون") [11].
As shown in Figure 3, in this thesis, the prefixes and suffixes have been classified into two groups. In the Arabic morphology, the prefix "ال" is added to the Lemma directly, i.e. prefixes between the Lemma and the prefix "ال" are not allowed. According to this rule, a list of prefixes have been created that can precede "ال". This list will be referred hereafter as the explicit prefixes. The explicit prefixes list is formed of ("وبال", "ولل", "وكال", "فبال", "فلل", "فكال," "وال," "فال," "كال", "لل", "بال"," ال").

The other prefixes used in the Arabic language are collected in a list ("ف", "و", "ب", "ل", "ك") under the name "vague prefixes", as they are ambiguous and can be a part of the lemma. For that reason, two control mechanisms are used before deleting these prefixes in order to avoid deleting any letters from the lemma. The first control mechanism controls the length of the word before deleting any vague prefixes. If the word length is equal or less than three characters, this means that the vague prefixes that are found in the word is part of the lemma. The second control mechanism compares the word after deleting the vague prefixes with the pattern list. If none of the patterns match the word, this means that a part from the lemma has been deleted. In this case the deletion of the vague prefixes is reversed to its original by undoing.

In this work, the suffixes are classified into two groups. The first group includes explicit suffixes which are subcategorized into two, namely the feminine suffixes ("اتهما","اتهم", "اتها", "تان", "تين", "تها", "ات", "ته", "تا") and the third person suffixes ("هما", "ها", "هم"). If an original word contains a feminine suffix, it will be replaced with the feminine letter ("ة"). When an original word contains a third person suffix, it will be deleted immediately.

The second group of suffixes are called the "vague suffixes" and include ("ون", "ين", "ت", "ي","ه","ا"). In contrast to the explicit suffixes they can be a part of the lemma. So, vague suffixes are treated in a similar way as vague prefixes are treated.

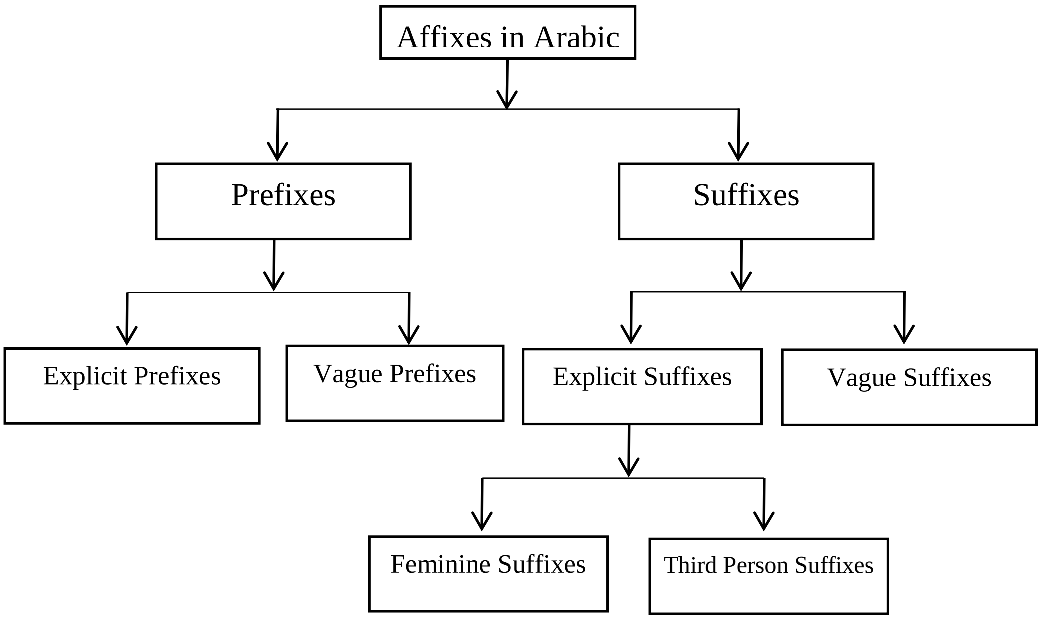

Figure 3. The Arabic affixes classification

3.2.3.1. Delete the explicit prefixes

In this step, the explicit prefixes are deleted from the original words. First, the explicit prefixes are searched from the beginning of the original word. When an explicit prefix is found, it is deleted from the original word. After the deletion of the prefix, a mark indicating “there is no more prefix” is added to the word. As after the explicit prefix is removed, the Lemma is started from the beginning of what is left from the word. Algorithm 1 shows the prefix deletion strategy.

3.2.3.2. Delete the explicit suffixes

In this step, the explicit suffixes are deleted from the original words. Firstly the feminine suffixes list is searched at the end of the original word. If the suffix is found, it is deleted from the original word. The feminine letter is added to the word to keep the gender of the word after deleting the suffix. Also, the “there is no more suffixes” mark is added to the word, as the feminine suffix is a part of the Lemma. When the feminine suffix is deleted, it means that the lemma without the feminine letter is at the end of the word and no more suffixes exist.

Secondly, words that do not contain any marks are searched in order to find any third person suffix. If any suffix is found, it is deleted from the original word. After the suffix is deleted, the "word does not contain any more suffixes" mark is added to the word. (Shown in Algorithm 2).

```
For each prefix in explicit Prefix List
    If( explicitPrefixesList[i] = beginning of the word) then
        Delete the prefix from beginning of the word
        Add the "there is no more prefix" mark to the word
        Break loop
    End if
End of loop
```

Algorithm 1 Deletion of the explicit prefixes

```
For each suffix in Feminine Suffixes List
    If (feminineSuffixesList [i] = end of the word) then
        Delete the suffix from end of the word
        Add feminine letter to the word
        Add the "there is no more suffix" mark to the word
        Return
    End if
End of loop 1
Ifworddoesn't contain "there is no more suffix" mark then
    For each suffix in Third Person Suffixes List
        If (thirdPersonSuffixesList [i] = end of the word) then
            Delete the suffix from end of the word
            Add the (There is no suffix)'s mark to the word
            Return
        End if
    End of loop 2
End if
```

Algorithm 2 Deletion of the explicit suffixes

3.2.3.3. Delete the vague prefix

In this stage, the vague prefixes are deleted from the words that do not contain any marks. The removal of the vague prefixes depends on additional controls in order to avoid the removal of some parts of the lemma. To test that these letters are not within the lemma; the words are compared with a set of Arabic patterns. If the results give more than one match, it means that this word has another prefix or suffix (mechanism of the pattern match is explained in section 3.2.3.5). In this case, the word will be searched for a vague prefix. If a vague prefix is found, it will be deleted. After the deletion of the prefix, the word will be compared with the Arabic patterns. If the word does not match with any of the Arabic patterns, it means that a part of the lemma is deleted, and the prefix is not removed. In this case the search continues with the next prefix from the vague prefix list.

```
If the worddoes not contain (There is no prefix)'s mark then
    SetnumberOfMatch = 0
    For each pattern in the patternList
        If (patternMatch(word)) then
            numberOfMatch = numberOfMatch + 1
        end if
    end Loop
    if(numberOfMatch> 1 ) then
        For each prefix in the vaguePrefixList
            If (vaguePrefixList [i] = beginning of the word) then
                Delete the prefix
                SetnumberOfMatch = 0
                For each pattern in the Patterns List
                    If (patternMatch(word)) then
                        numberOfMatch = numberOfMatch + 1
                    end if
                end Loop
                if (numberOfMatch =0 ) then
                    Add the deletion part
                End if
            End if
        End of loop
    End if
End if
```

Algorithm 3 Deletion of the vague prefix

3.2.3.4. Deletion of the vague suffixes

In this stage, the deletion of the vague suffixes will be processed with the same steps applied in the vague prefix deletion. (See section 3.2.3.3)

3.2.3.5. Detecting a match with the pattern list

The Arabic language uses patterns in standard forms; for example, the word "كتب - wrote" is a root and has 3 letters. This root can be written by its pattern by changing all the letters in the root from right to left with "ف-Faa", "ع-Ayn" and "ل-Laam". In this step the word "فعل - verb" is the pattern for "كتب" root and for all roots with 3 letters. If a complex word like "سأكتب - I well write" is used and needs to be converted into its pattern, the same steps that are explained above needs to be followed. As the result, "سأفعل- I well do" is obtained. In other words, when "ف-Faa", "ع-Ayn" and "ل-Laam" are changed in any patterns by any three letters from any root, it results in a new word with a new meaning which is indirectly related with the meaning of the root. In comparing the match between any word and the patterns in the newly designed algorithm for this thesis, all elements in the pattern list are converted into standard regular-expression. This conversion is made by changing"ف-Faa", "ع-Ayn" and "ل-Laam" letters in the pattern with the letter ".". After converting the pattern into the regular-expression, the match between the word and the regular-expressions is found.

3.3 Phase III: Generating the Candidate Keyphrases

After stemming the Arabic text in the article, digits, diacritics of the letters, stop words and punctuation are removed. The generation stage of the candidate keyphrases is started, which generates a list of candidate phrases that could be a considered as a keyphrase. The list of keyphrases is limited to at most three words in this research. Keyphrases that consist of more than three words are not to be used, because most keyphrases are less than three words.

The candidate keyphrases are generated in accordance with a set of exclusion rules, so that some phrases that cannot be keyphrases are never added as candidates. In this step, some rules that are common with the KP-miner algorithm are applied. The phrases containing punctuations or digits in the middle are also excluded. Table 9 lists all the symbols and signs that have been treated as punctuation. The “three words” principle will be adopted as the maximum number of words in each candidate keyphrase.

Symbol Name	Symbol	Symbol name	Symbol
at symbol	@	opening brace	{
exclamation point	!	closing brace	}
percent sign	%	Period	.
ampersand	&	Arabic question mark	؟
slash	/	double quotes	"
opening parenthesis	(	Colon	:
closing parenthesis	)	End of Line	\n
minus sign - hyphen	-	backslash	\
Underscore	_	less than sign	<
en dash	–	greater than sign	>
Comma	,	plus sign	+
Arabic comma	،	question mark	?
opening bracket	[	asterisk	*
closing bracket	]	number sign	#
dollar sign	$	single quote	‘
Semicolon	;	equal sign	=
caret – circumflex	^		

Table 9 List of Symbols

As the first step, all words that are not in the ‘stop word’ list are added to the candidate keyphrases. In the second step, two words are taken and these words are checked to make sure they are not ‘stopping words’ or contains no punctuation between the two words. In the third step, three consequent words are processed to add in candidate keyphrases. The exclusion is done for phrases that include a stop word at the beginning or the end of the phrase. While the exclusion is not done for phrases that have a “stop word” in the middle. (Shown in Algorithm 4)

```
// phrases with one word
For each word in Words List
    word = wordsList[i]
    If (word is not a ‘stop word’ or a punctuation) then
        Add word to the keyphrases list
    End if
End Loop

// phrases with two words
For each word in Words List-1
    word = wordsList[i]
    word2 = wordsList[i+1]
    If (word and word2 is not a ‘stop word’ or a punctuation) then
        Add two words as one phrase to the keyphrases list
    End if
End Loop

// phrases with three words
For each word in Words List-2
    word = wordsList[i]
    word3 = wordsList[i+2]
    If (word and word3 is not a ‘stop word’ or a punctuation) then
        Add three words as one phrase to the keyphrases list
    End if
End Loop
```

Algorithm 4 Generating the candidate keyphrases

3.4 Phase IV: Selection the Appropriate Keyphrases.

After generating a list of candidate keyphrases, the features of these candidate keyphrases, are identified depending on the concepts explained in the previous sections. TF-IDF will be one of these features. Moreover, the values for TF-IDF are generated for each candidate. The values needed by TF-IDF’s equation are the frequency of the word in the article (TF) and the frequency of the word in other articles (IDF), calculated from articles included in the keyphrase corpora. TF-IDF is calculated as shown in Equation 2.1.

$$TF(P,D) \times \mathrm{IDF(P,D)} = \frac{freq_d(P,D)}{size_d(D)} \times \log_2\left(\frac{\mathrm{freq_c(P,C)}}{\mathrm{size_c(C)}}\right) \qquad (2.1)$$

The first place that the candidate keyphrase is mentioned in the article is used as a feature. Therefore, the candidate keyphrase mentioned at the beginning of the article leads to an increased probability of the candidate being a keyphrase.

As mentioned in the previous chapters, the basis of this research depends on the division of a single article into seven main sections. These sections are the title, the abstract, the introduction, the results, the conclusion, the resources and others. The place value for each of these sections will be generated. Thus, in the conclusion for each candidate keyphrase founded in the list of candidate keyphrases 32 features were generated.

3.5 Data Resources

The scientific articles keyphrase extraction corpus created for this thesis is gathered from different universities in Arabic countries including Iraq, Syria and Egypt. Some of these universities are University of Mosul, University of Babylon, University of Basrah, University of Anbar, Ain shams University and Tishreen University. Because some of the collected articles from these universities are in PDF format and the others are in Microsoft Word document format, the files in PDF format are converted to Microsoft Word document format. Afterwards, all the files in the Microsoft Word format are converted to text files with the extension (txt).

The corpus consists of 111 different articles. The articles are selected from different subjects and scientific fields. Table 10 shows the categories and statistics for the corpus used in this thesis. As the result, the average number of keyphrases per article is about 4.261.

Category Name	Number of Articles	Total of Keyphrases	Average of Keyphrases	Standard Deviation
Arabic	10	40	4	1,2
Sociology	40	162	4,05	1,118
Art	3	18	6	1,667
Economy	34	150	4,412	1,464
Education	18	77	4,278	1,037
Others	6	26	4,333	0,667
Total	111	473	4,261	1,192

Table 10 The General Classification for Articles that are Used in this Thesis

CHAPTER 4

RESULTS

4.1 Overview

In this thesis the three algorithms; the Kp-miner algorithm, the KEA algorithm, and the algorithm developed in this thesis, are evaluated on the data described in the Section 3.5. The effectiveness of the algorithm designed for this thesis is compared with the state-of-the-art Arabic keyphrase extraction algorithms.

In addition, in order to investigate which features are more effective in the article for keyphrase extraction, the features are divided into groups. Then, the results of these groups are compared with each other using Precision and Recall. As a result, the best feature set in keyphrase extraction is determined.

Precision and recall are defined in terms of a set of retrieved keyphrases that are the result of the algorithm of this thesis and a set of relevant keyphrases assigned by the authors of the articles.

4.2 Precision

The precision is the fraction of the keyphrases which are relevant to the number of extracted phrases.

$$Precision = \frac{|\{RelevantKeyphrases\} \cap \{\ RetrievedKeyphrases\}|}{|\{\ RetrievedKeyphrases\}|}$$

Whole the keyphrase that retrieved will be taken by the precision into account but the precision at a given cut-off rank can also be evaluated, considering just the superior results get back with the algorithm. This measure is termed precision at n.

Whereas, with the recall the precision can be utilized, which is the percent for whole the relevant keyphrases which are get back with the algorithm. To supplied a single measurement to the algorithm, the two measures are utilized with one another in the f-measure [4].

4.3 Recall

Recall is the fraction of the keyphrases that are relevant to the extraction that are successfully retrieved.

$$Recall = \frac{|\{RelevantKeyphrases\} \cap \{\ RetrievedKeyphrases\}|}{|\{RelevantKeyphrases\}|}$$

Recall is defined sensitivity in the binary classification. Therefore, the recall can be considering as the probability that the extraction is retrieve a relevant document [4].

It is unassuming to be realize 100% recall by get back whole keyphrases in reply for any extraction.So, recall sole is not sufficient but one requires to measure the number of non-relevant keyphrases too.

4.4 Features of Candidate Keyphrases

Each keyphrase has group of features extracted from the article processed, and the other articles that exist in the data set. These features are explained in Table 11.

Name	Description
FAt(x)	The first location that mentions the phrase in the (title, abstract, introduction, conclusion, reference, other) sections or the article.
fregIn(x)	The frequency of the phrase in the (title, abstract, introduction, conclusion, reference, other) sections or the article.
CValue	The text value of the phrase after stemming.
iOA	The frequency of the phrase in the other articles in the data set.
length	Number of words in the phrase
prob	Probability of the vicinity of the words in the phrase
value	The original text value of the phrase.
tfidf(x)	The value of the Tf-Idf for the (title, abstract, introduction, conclusion, reference, other) sections or the article.

Table 11 Features of Candidate Keyphrases

4.4.1 Candidate keyphrases' features groups

The features are divided into groups as shown in Table 12. Using the Weka data mining library, supervised learning using these groups are evaluated, to determine the effectivenes of each group.

Name	Description
All Features	All the candidate keyphrases properties.
Basic Kea	FAtArt, tfidf.
Sections Features	FAtAbs, FAtCon, FAtInt, FAtOth, FAtRef, FAtTit, freqInAbs, freqInCon, freqInInt, freqInOth, freqInRef, freqInTit.
Basic Kea – Other (BKO)	FAtArt, tfidf, iOA.
Basic Kea – Vicinity (BKV)	FAtArt, tfidf, prob, length.
Sections tfidf	tfIdfTitle, tfidfAbs, tfidfCon, tfidfInt, tfidfOth, tfidfRef

Table 12 Candidate Keyphrases' Features Groups

After applying the new algorithm to all of the groups that the result is that the BKVgroup achieves a better result compared to others. These results can be seen in Table 8. Also it can be seen that, the Basic Keagroup is the closer one to the BKVgroup. The difference between the BKVgroup and the Basic Keagroup is the probability of the neighborhood of the phrases' words and the number of words in the phrase.

When the sections features group is looked at, it can be seen that it contains all the features that are related to the sections. By separating the features in this group from other features in the candidate keyphrases, the efficiency of division article to sections is improved. This group, contradicting the expectations, did not improve the results.

The last row refers to the "Without apply Weka algorithm" group. In this group, the weight of the candidate keyphrases are sorted out and larger values are obtained. The weight of the candidate keyphrases is explained in equation 4.1. This equation has two features, the TF-IDF and the first location that mentions the phrase.

$$Wck_i = \frac{TF-IDF}{First\ location\ of\ the\ phrase} \qquad (4.1)$$

The number of keyphrases in the articles (relevant keyphrases) in all cases that are mentioned in Table 13 is 473 because the data set is the same. The algorithm was applied three times with all groups and different numbers of keyphrases were extracted. The number of keyphrases that were extracted each time was (5,10,15). To illustrate the results, the number of retrieved keyphrases in Table 13 is divided into three groups with values (555, 1110, 1665).

4.5 Algorithm Results

Group Name	**Right extracted keyphrasese**	**Precision**	**Recall**	**F-measure**
KP-minar	52	0,094	0,11	0,101
	83	0,075	0,175	0,105
	107	0,064	0,226	0,1
KEA	88	0,158	0,186	0,171
	127	0,114	0,268	0,160
	146	0,087	0,308	0,136
Kea with our stemming algorithm	**98**	**0,176**	**0,207**	**0,190**
	137	**0,123**	**0,289**	**0,173**
	166	**0,0997**	**0,350**	**0,155**
All Features	11	0,02	0,023	0,021
	57	0,051	0,121	0,072
	99	0,059	0,209	0,092
Basic Kea	35	0,063	0,074	0,068
	88	0,079	0,186	0,111
	138	0,083	0,292	0,129

Group Name	Right extracted keyphrasese	Precision	Recall	F-measure
Sections Features	33	0,059	0,07	0,064
	74	0,067	0,156	0,094
	106	0,064	0,224	0,1
Basic Kea – Other (BKO)	35	0,063	0,074	0,068
	88	0,079	0,186	0,111
	138	0,082	0,291	0,128
Basic Kea – Vicinity (BKV)	**57**	**0,102**	**0,121**	**0,111**
	95	**0,086**	**0,201**	**0,120**
	142	**0,085**	**0,300**	**0,132**
Sections tfidf	22	0,04	0,047	0,043
	46	0,041	0,097	0,058
	62	0,037	0,131	0,058
Without apply Weka algorithm	43	0,077	0,090	0,083
	90	0,081	0,190	0,113
	130	0,078	0,275	0,122

Table 13 Results of the Algorithm

The first algorithm divides the scientific articles into seven sections. These sections are: the title, the abstract, the keyphrases, the introduction, the conclusion and the results, the resources, and other. These sections were considered to be popular sections that are frequently used in the scientific articles. The expectation was that the differences in these sections, the writing format and the number of words would positively affect the results in extracting keyphrases.

But when the results seen in Table 13, especially the “Sections Features” group that contains all data belonging to the division operation, were compared with other results, the ineffectiveness of this method in keyphrases extraction was approved. In addition, this algorithm cannot be applied automatically without intervention and continuous updating for the method of sections determinations. The reason for this is that the Arabic articles do not use standard phrases as titles for the sections which are mentioned above.

The second algorithm uses a special stemming for filtering the Arabic words from suffixes and prefixes, which are added to the words. So, this algorithm allows to determine the similar words in the meaning and disjoins them from words that have different meanings. Thus, through this algorithm the statistic data for the words that are used in the Arabic scientific articles can be determined.

Different algorithms were also used in previous researches. In order to approve the efficiency and strength of the algorithm designed for this thesis, it was compared with famous algorithms which were used in the keyphrases extraction field. The comparison of the Basic Kea – Other (BKO) and Basic Kea – Vicinity (BKV) group of results mentioned in Table 13 with the KP-minar results approved that the algorithm of the thesis can extract a larger number of keyphrases than the KP-minar group results.
But the KP-minar is different from the thesis algorithm in two aspects. The first is the stemming aspect and the second is calculating the candidate keyphrases weight aspect. Therefore, the comparison made proved to be imperfect, because there was a difference in the most important aspect: the keyphrase extraction.

The KEA algorithm was used in the comparison in terms of its similarity to the calculation operation for the keyphrases weight from where mechanism of the Basic Kea group can be seen in Table 13. However, using the KEA algorithm lacks the stemming algorithm particularly designed for the Arabic language. Therefore the Sremoval Stemmer algorithm was also used in , in the basic Kea gruop and comper it with our stemmer algorithm after integrate it with Kea algorithm. This algorithm enables a comparison of the results of the thesis algorithm in terms of the stemming, so that the result turns out to be positive.

CHAPTER 5

CONCLUSION

This research has aimed at evaluating the development of two different algorithms in the Arabic scientific articles. The purpose was to extract the maximum number of corresponding keyphrases with the keyphrases also assigned by the authors of the articles.

Through this research a new method to extract keyphrases has been developed. But this method has necessitated many manual interventions in order to apply it. Regardless of this, the results of this method have been below the expected level.

On the other hand, in this thesis a new algorithm for stemming Arabic words has been developed. This algorithm was applied to the Arabic words before extracting the keyphrases and gave promising results for extracting keyphrases.

The results of the application in this thesis showed that the common use in extracting the root of the word in the Arabic language is not the most optimized way of stemming words that are to be used in the extraction keyphrases. So, detecting the simplest form of Lemma is the best way for stemming words for the extraction of keyphrases.

It can be concluded that stemming algorithms for the Arabic Language can still be improved to achieve better results. Merging various words that have the same root in one word is the most important deterrent for stemming in the Arabic language especially for the extraction of keyphrases. The focus in this respect may be one of the main reasons for the improved results in the extraction keyphrases.

REFERENCES

1. **Turney P. D., (1999),** "*Learning Algorithms for Keyphrase Extraction",* Information Retrieval, Canada, pp. 1-46.

2. **El-shishtawy T. A., Al-sammak A. K., (2009),** "*Arabic Keyphrase Extraction Using Linguistic Knowledge and Machine Learning Techniques*", The MEDAR Consortium, Cairo, Egypt, pp. 1-8.

3. **Weber G., (2008),** "*Top Languages*", AATF National Bulletin, vol. 24, pp. 22-28.

4. **Avanzo E. D., Magnini B., (2005),** "*A Keyphrase-Based Approach to Summarization: the LAKE System at DUC-2005".* Proceedings of Human Language Technology DUC Workshop, pp.1-6.

5. **Karanikolas N., Skourlas C., (2006),** "*Text Classification: Forming Candidate Keyphrases from Existing Shorter Ones*", Facta Universitatis – Series: Electronics and Energetics, vol. 19, pp. 439-451.

6. **Pala N., Çicekli I., (2007),** "*Turkish Keyphrase Extraction Using Kea*", The 22nd International Symposium on Computer and Information Sciences, Ankara, Turkey.

7. **Erbs N., Gruevych I., Rittberger M., (2013),** "*Bringing Order to Digital Libraries: From Keyphrase Extraction to Index Term Assignment*", D-Lib Magazine, vol. 19, pp. 1-13.

8. **Abdusalam N., Seyed T., Falk S., (2005),** "*Stemming Arabic Conjunctions and Prepositions*", The 12th International Conference on String Processing and Information Retrieval, Heidelberg, pp. 206-217.

9. **Al-Kharashi I., Evens M. W., (1994)**, "*Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System*", JASIS, vol. 45, pp. 548-560,

10. **Manning C. D., Raghavan, P., Schütze, H., (2008),** "*Introduction to Information Retrieval",* Cambridge: Cambridge University Press, vol. 1, p. 32.

11. **Larkey L. S., Ballesteros L., Connell M. E., (2007),** "*Arabic Computational Morphology: Knowledge-Based and Empirical Methods*", Springer, Netherlands, pp. 3-14.

12. **Hegazi M., Elsharkawi A. A., (1985),** "*An Approach to a Computerized Lexical Analyzer of Natural Arabic",* Kuwait Institute for Scientific Research, Kuwait vol. 1.

13. **http://trec.nist.gov/pubs/trec10/papers/IIT-TREC10.pdf,**
(Data Download, Date : 10.11.2014).

14. **DeRoeck A. N., Al-Fares W. A., (2000),** "*Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots*", The 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, pp.199-206.

15. **Larkey L. S., Ballesteros L., Connell M. E., (2007),** "*Light Stemming for Arabic Information Retrieval*", In Arabic Computational Morphology, Springer Netherlands, pp. 221-243.

16. **Darwish K., Oard D. W., (2002)**, "*Evidence Combination for Arabic-English Retrieval*", In TREC 2002. Gaithersburg: NIST, pp. 703-710.

17. **Eldesouki M. I., Arafa W., Darwish K. (2009),** "*Stemming Techniques of Arabic Language: Comparative Study from the Information Retrieval Perspective*", The Egyptian Computer Journal, vol. 36, pp. 30-49.

18. **Xu J., Croft W. B., (1998),** “*Corpus-Based Stemming Using Co-Occurrence of Word Variants*”, ACM Transactions on Information Systems, vol. 16, pp. 61-81.

19. **Khoja S., Garside R., (1999),** “*Stemming Arabic Text*”, Lancaster U.K.

20. **Whitley D., (1989),** “*The GENITOR Algorithm and Selective Pressure*”. The Third International Conference on Genetic Algorithms, California, pp. 1-6.

21. **Turney P. D., (1997),** “*Extraction of Keyphrases from Text: Evaluation of Four Algorithms*”, National Research Council, Canada, pp. 1-29.

22. **Turney P. D., (1999),** “*Learning to Extract Keyphrases from Text. National Research Council*”, National Research Council, Canada, pp. 1-43.

23. **Turney P. D., (2000),** “*Learning Algorithms for Keyphrase Extraction*”, National Research Council, Canada, vol. 2, pp. 1- 46.

24. **Lovins J. B., (1968),** “*Development of a Stemming Algorithm*”, Mechanical Translation and Computational Linguistics, vol. 11, pp. 22-31.

25. **Furnas G., Landauer T., Gomez L., Dumais S., (1987),** “*The Vocabulary Problem in Human-System Communication*”, Communications of the ACM, vol. 30, pp. 964-971.

26. **Frank E., Paynter G. W., Witten I. H., Gutwin C., Nevill-Manning C. G., (1999),** “*Domain-Specific Keyphrase Extraction*”, the Sixteenth International Joint Conference on Artificial Intelligence, California: Morgan Kaufmann, pp. 668-673.

27. **Witten I. H., Paynter G. W., Frank E., Gutwin C., Nevill-Manning C. G., (1999)**, “*KEA: Practical Automatic Keyphrase Extraction*”, the Fourth ACM Conference on Digital Libraries, pp. 254.

28. **Witten I. H., Paynter G. W., Frank E., Gutwin C., Nevill-Manning, C. G., (2000),** "*KEA: Practical Automatic Keyphrase Extraction*". the Fourth ACM Conference on Digital Libraries, University of Waikato pp. 255.

29. **Domingos P., Pazzani M., (1997),** "*On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss",* Manufactured, Netherlands, vol. 29, pp. 103-130.

30. **Turney P., (2003),** "*Coherent Keyphrase Extraction via Web Mining*", the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico. pp. 434-439.

31. **Turney P.D., (2002)**, "*Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from Labeled and Unlabeled Data, National Research Council, Institute for Information Technology*", National Research Council, Canada, pp. 1-34.

32. **Banko M., Mittal V., Kantrowitz M., Goldstein J., (1999),** "*Generating Extraction-Based Summaries from Hand-Written Summaries by Aligning Text Spans*", the Pacific Rim Conference on Computational Linguistics, pp. 1-5.

33. **Gutwin C., Paynter G. W., Witten I. H., Nevill-Manning C. G., Frank E. (1999),** "*Improving Browsing in Digital Libraries with Keyphrase Indexes*", Journal of Decision Support Systems, vol. 27, pp. 81-104.

34. **Jones S., Paynter G. W., (1999),** "*Topic-Based Browsing Within a Digital Library Using Keyphrases*", the Fourth ACM Conference on Digital Libraries, pp. 114-121.

35. **https://code.google.com/p/kea-algorithm/downloads/list,** (Data Download, Date : 10.01.2015).

36. **El-Beltagy S. R., Rafea A., (2008),** "*KP-Miner: A Keyphrase Extraction System for English and Arabic Documents. Information systems*", Elsevier, vol. 34, pp. 132-144.

37. **http://www.claes.sci.eg/coe_wm/apis/KP.jar,** (Data Download, Date: 10.01.2015).

38. **Wan X., Xiao J., (2008),** "*Single Document Keyphrase Extraction Using Neighborhood Knowledge*", the Twenty-Third AAAI Conference on Artificial Intelligence, vol. 8, pp. 855-860.

39. **Nguyen T. D., Kan M. Y., (2007),** "*Keyphrase Extraction in Scientific Publications*", Springer-Verlag Berlin Heidelberg, pp. 317-326.

40. **Harman D., (1991),** "*How Effective is Suffixing?"*, Journal of the American Society for Information Science, vol. 42, pp. 7-15.

APPENDIX A

LIST OF STOP WORDS IN THE ARABIC LANGUAGE

إلى	الى	من	منهما	منهم	ومنهما	ومنهم	ومنه	ومن	منها
في	فى	وفى	فيما	وفيما	وفي	على	هذا	وهذا	بهذا
بهذه	هذه	ذلك	تلك	هؤلاء	أولئك	اولئن	لعل	و	أو
او	بين	كل	ككل	بكل	لكل	منذ	عن	عمن	عنها
عند	مع	ومع	معه	وعن	و	لي	هم	هو	هي
وهي	وهو	هن	إنهم	انت	أنت	متى	كيف	التي	التي
والتى	والتي	والذي	الذي	الذين	كان	كون	يكن	كونه	كأن
كانت	وكانت	فكان	وكان	فكانت	سيكون	تتكون	وتتكون	فإن	ليس
ليست	لماذا	ماذا	أن	إن	ان	إنها	أنها	انها	لا
الا	إلا	لن	لم	لك	له	لها	لهن	منهم	ألا
إلا	الا	ما	كما	حيث	بعض	كل	عند	أما	قد
وقد	تم	وتم	تمت	يتم	ثم	بها	بنا	هنا	وهنا
هناك	فقط	به	لو	اي	أي	هل	اين	أين	كلما
إذ	اذ	الا	إليه	اليه	بل	غير	فان	فلقد	فهي
فهي	فيه	فيها	لا	لاسيما	لان	لدى	لذلك	لقد	لكن
لما	لنا	لهذا	لهذه	نحن	نحو	هاتين	وانما	وعلى	ال
وعليه	عليها	عليه	وكذلك	ولا	ولذا	ولقد	ولكن	ولم	وليس
ومما	وهذه	وهكذا	هاتان	هذان	معاً	ربما	ذات	ذو	ذي
بينهما	بين	بينهم	بينهن	وما	عد				

APPENDIX B

LIST OF PATTERNS USED IN THE RESEARCH.

8 Letter patterns

إستفعالت، استفعالت

7 Letter patterns

استفعال، اعفلاوى، فاعولاء، مفعولاء، يفاعلاء، فعيلياء، إفعيلاء، إستفعال، إفعالية،
افعالية، مفعالية

6 Letter patterns

استفعل، فعيلاء، فيعلاء، فعنلاء، إفعيلى، تفاعيل، يفاعيل، مفاعيل، افاعيل، فاعولى،
فعاويل، فعاييل، فعلوان، فعليان، فعلايا، فعلياء، فعلوتى، إفعلان، أفعلان، افعلان،
تفعلاء، افعلاء، فنعلاء، فاعلاء، فعالاء، فوعلاء، مفعلاء، فعولاء، فعاعيل، فوعالى،
فعالين، فعالان، فيعلان، فوعلان، مفعلانفواعيل، فياعيل، فنعليل، أفاعيل، فعالية،
فعنلوة، فعلنية، فعنليل، فيعلول، فعلويل، مفعوعل، مستفعل، متفتعل، متفعئل، متفعنل،
متفعول، متفعيل، متفوعل، متفيعل، متمفعل، مفتعئل، مفعنمل، مفتعأل، مفتعلى، متفعلى،
متفعأل، مفعنلى، مفونعل، مفعالة، مفاعلة، تفعيلة، فعولية، إنفعال، انفعال، يتفاعل،
تستفعل

5 Letter patterns

افتعل، فاعلة، فاعول، مفنعل، فعالة، مفعال، مفعلة، فعيلى، فعالى، مفعلى،
يفعلى، تفعال، فعلول، فعليا، فعلان، إفعال، افعال، أفعال، فعالي، تتفلة،
فعنلة، فعلية، فعلتة، فعلوة، أفنعل، يفنعل، مفاعل، يفاعل، تفاعل، فيعول،
فوعال، فاعال، فيعال، فنعال، فعنلى، فعيلأ، فعولى، إفعيل، افعيل، أفعيل،
مفعيل، مفعول، تفعيل، يفعيل، فنعلو ، فوعلى، فيعلى، أفعلى، إنفعل، انفعل،
أنفعل، فعوعل، فناعل، فواعل، فعوال، فعيال، فعايل، فعائل، فعامل، فعلنى،
علوى، فعلين، فعلاء، مفعلس، مفعلم، مفعلن، مفعمل، مفتعل، مفعتل، مفعنل،
مفمعل، مفهعل، مفوعل، مفيعل، ممفعل، منفعل، مهفعل، متفعل، مفعهل، مفلعل،
مفعأل، مسفعل، مفعفل، مفعلت، مفنعل، ميفعل، افعول، يتفعل، يفتعل

4 Letter patterns

تفعل، فعال، مفعل، فعيل، أفعل، إفعل، يفعل، نفعل، فاعل، فيعل،
فأعل، فنعل، فعول، فعأل، فعلة، فعلى، فعلن، فعلم، فعلا، معفل، معهل

3 Letter patterns

فعل

APPENDIX C

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: HANCI, Firnas

Date and Place of Birth: 08 May 1979, Kirkuk-Iraq

Marital Status: Single

Phone: +90 533 139 74 91

Email: firnashanci@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya University, Computer Engineering	2015
B.Sc.	Foundation of Technical Education, College of Technology / Kirkuk, Software Engineering Techniques	2007
Diploma	Foundation of Technical Education, Kirkuk Technical Institute, Computer Systems	1999
High School	Al-Tameem High School	1997

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2011- Present	Havelsan Company	Technical Adviser
2008-2011	Türkmeneli Vakfı, Türkmeneli Kültür Merkezi	Programmer

FOREIN LANGUAGES

Advanced English, Advanced Arabic

PROJECTS

1. Finance Project for Anadolu Sigorta 2013- Present
2. E-government Project for Iraq government, land Registry Project, 2011-2013

HOBBIES

Coding, Travel, Books, Paint.