A KEYPHRASE EXTRACTION ALGORITHM AND ITS POTENTIAL ON ARABIC

BY

MOHAMMED SULIMAN AL LOGMANI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In
INFORMATION AND COMPUTER SCIENCE

May 2013

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Mohammed Suliman Al Logmani** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

Dr. Husni Al-Muhtaseb
(Advisor)

Dr. Adel F. Ahmed
Department Chairman

Dr. Moataz Ahmed
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Wasfi Al-Khatib
(Member)

6/1/14
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

To my dear parents, my lovely wife, my sons Jamal and Rayan, and all my brothers and sisters.

ACKNOWLEDGMENTS

First and foremost, I would like to praise Allah the Almighty; Who has allowed this work to manifest into reality, by giving me the will, persistence, and guidance. Peace and blessing of Allah be upon his Prophet Mohammad (Peace Be upon Him).

I would like to thank King Fahd University of Petroleum and Minerals for creating an academic environment that aided me greatly in this work. I would also like to thank Saudi Aramco for encouraging me to continue growing intellectually.

I want to thank Dr. Husni Al-Muhtaseb, my thesis adviser for his invaluable input, advice, and guidance. Many thanks to my thesis committee members for their guidance and patience throughout this process; Dr. Moataz Ahmed for his help in narrowing down the topic, Dr. Wasfi Al-Khatib for encouraging me to pursue the Arabic keyphrase extraction research. Furthermore, I would like to thank Dr. Muhammad Shafique for his support prior starting the work on this thesis. I would also like to thank Omar Shaaban for providing me with some Arabic text that was used in my Arabic dataset. I would like to thank my brothers and sisters, especially my sister Azzizah, for taking the time to assist me in the review of the dataset documents.

On a personal note, I would like to thank my parents for their continuous prayers, and my wife for her never-ending support and encouragement. Finally yet importantly, I would like to thank all my friends and colleagues who helped me, in any way, to complete my thesis; especially my dear friend Mohammed Hamed.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ..... **V**

TABLE OF CONTENTS ..... **VI**

LIST OF TABLES ..... **X**

LIST OF FIGURES ..... **XI**

LIST OF ABBREVIATIONS ..... **XII**

ABSTRACT ..... **XIV**

ملخص الرسالة ..... **XV**

CHAPTER 1 INTRODUCTION **1**

1.1 Motivation **4**

1.2 Problem Definition **5**

1.2.1 Preparing an Arabic Dataset **5**

1.2.2 Enhance the Keyphrase Extraction Algorithm (KEA) **6**

1.3 Objectives **6**

1.4 Thesis Organization **7**

CHAPTER 2 LITERATURE REVIEW **9**

2.1 Introduction **9**

2.2 Classification of Automatic Keyphrase Extraction **9**

2.2.1 Classification based on targeted domains **9**

2.2.2 Classification based on the used approach **10**

2.2.3 Keyphrase Extraction Algorithm (KEA) **22**

2.3 Keyphrase Extraction and Arabic Language **25**

2.3.1 Arabic Automatic Keyphrase Extraction Algorithms 26
2.3.2 Commercial Keyphrases Extraction Solutions 27
2.3.3 Keyphrases Extraction Algorithms as Services 27
2.3.4 Arabic Text Summarization 28
2.4 Term Weighting 29
2.5 Summary 31
CHAPTER 3 DATASETS 32
3.1 Overview 32
3.2 English Dataset 32
3.3 Arabic Dataset 33
3.3.1 Sources 34
3.3.2 Selection and organization 34
3.3.3 Cleaning up 39
3.3.4 Manual Keyphrase Extraction 42
3.3.5 Keyphrase Verification 43
3.4 Summary 43
CHAPTER 4 AUGMENTED KEYPHRASE EXTRACTION ALGORITHM 44
4.1 Overview 44
4.2 Hypotheses 44
4.3 Comparative Analysis Setup 46
4.3.1 Stop Words 47
4.3.2 Stemmers 47
4.3.3 Controlled Vocabulary 48
4.4 Methodology 49

4.4.1 Comparative Analysis 1: Stemming Vs. No-Stemming 50
4.4.2 Comparative Analysis 2: Top Term Weighting Formula in KEA 50
4.4.3 Comparative Analysis 3: Last Occurrence and the Combined First and Last Occurrence 51
4.4.4 Comparative Analysis 4: Enhanced KEA vs. Other Algorithms 51
4.5 Evaluation Techniques 52
4.6 Summary 53
CHAPTER 5 EXPERIMENTAL RESULTS AND DISCUSSIONS 54
5.1 Overview 54
5.2 Experimental Results 55
5.2.1 Stemming Vs. No-Stemming Results for English dataset (1A) 55
5.2.2 Stemming Vs. No-Stemming Results for Arabic dataset (1B) 59
5.2.3 Top Term Weighting Formula Results for English Dataset (2A) 62
5.2.4 Top Term Weighting Formula Results for Arabic Dataset (2B) 64
5.2.5 Last Occurrence and Combined First and Last Occurrence Results for English Dataset (3A) 67
5.2.6 Last Occurrence and Combined First and Last Occurrence Results for Arabic Dataset (3B) 69
5.2.7 Enhanced KEA vs. Other Algorithms Results for English Dataset (4A) 72
5.2.8 Enhanced KEA vs. Other Algorithms Results for Arabic Dataset (4B) 74
5.3 Discussions 76
5.4 Summary 77
CHAPTER 6 CONCLUSION AND FUTURE WORK 79
6.1 Conclusion 79
6.2 Future Work 80
REFERENCES 81
APPENDIX A: EVALUATIONS OF THE USAGE OF VOCABULARY FILES 88

APPENDIX B: HELPER TOOLS 90

APPENDIX C CONTENT OF ENCLOSED CD-ROM 92

VITAE 94

LIST OF TABLES

Table 1- Some of automatic keyphrase extraction algorithms 13
Table 2 - Distribution of the documents in the English dataset 33
Table 3 - Distribution of the documents in the Arabic dataset 37
Table 4 - Manual & auto-generated keyphrases in file C-22 in stemming experiments .. 56
Table 5- Measures calculation on file C-22 in the stemming experiment on English 57
Table 6 - Stemming vs. no-stemming in English dataset (1A) 58
Table 7 - Measures calculation on file "Airport 01" in Arabic stemming experiment 61
Table 8 - Stemming vs. no-stemming in Arabic dataset (1B) 62
Table 9 - Results on English Dataset for comparative analysis 2A 63
Table 10- Summary of results obtained for the Arabic dataset in 2B 65
Table 11 - Summary of comparative analysis 3A on the English dataset 67
Table 12- Summary of the results of comparative analysis 3B on the Arabic dataset 70
Table 13- Results of comparative analysis 4A 72
Table 14- Expermentail results (4B) on Arabic dataset 74
Table 15 - Results on the SM2010 English dataset 88
Table 16 - Results for the evaluation on FAO-780 dataset 89

LIST OF FIGURES

Figure 1 - Overview of KEA 23
Figure 2- Organization of the Arabic Dataset 39
Figure 3 - Wikipedia table for Macau 41
Figure 4- Macau table after extraction 41
Figure 5- Gallery section of Quebec City (city04.txt) 41
Figure 6- A screenshot of the file heriarciy of three results 56
Figure 7 - A screenshot of the file hierarchy of the results for 1B 60
Figure 8 - Exact Matching Results for Top Term Weighting Formula on English (2A).. 64
Figure 9 – Precision, Recall, and F-Score Results for comparative analysis 2A..... 64
Figure 10- Exact Match Results for Top Term Weighting Formula on Arabic (2B) 66
Figure 11- Precision, Recall, and F-Score Results for comparative analysis 2B 66
Figure 12- Exact match results of the occurrence evaluation on English 3A 69
Figure 13- Precision, Recall, and F-Score Results for occurrence in evaluation 3A..... 69
Figure 14- Exact match results of occurrence evaluation on Arabic dataset (3B)..... 71
Figure 15- Precision, Recall, and F-Score Results for occurrence comparative (3B) 71
Figure 16- Exact match representation of KEA vs. other algorithms on English 4A..... 73
Figure 17- Precision, Recall, and F-Score Results for KEA vs. others on English 4A 73
Figure 18- Exact matching results for comparative analysis 4B 75
Figure 19- Precision, Recall, and F-Score Results for KEA vs. others on Arabic (4B) ... 75
Figure 20- CD-ROM Content 93

LIST OF ABBREVIATIONS

ACM : Association for Computing Machinery

AGROVOC : Agricultural Thesaurus

AI : Artificial Intelligence

API : Application Programming Interface

ASR : Automatic Speech Recognition

BM25 : Best Match 25

CMS : Content Management Systems

CRF : Conditional Random Fields

CRF : Conditional Random Fields

GROBID : **G**ene**r**ation **of Bi**bliographic **D**ata

IDF : Inverse Document Frequency

IDFP : Probabilistic Inverse Document Frequency

KEA : Keyphrase Extraction Algorithm

KeaKat : Web based tool for keyphrase assignment based on KEA

KERT : Keyphrase Extraction and Ranking by Topic

KP : Keyphrase

LDA : Latent Dirichlet Allocation

LOG : Logarithm

LZ78 : The Lempel-Ziv 1978 Algorithm (Lossless Data Compression Algorithm)

ML : Machine Learning

MLP : Multi-Layer Perception

NLP	:	Natural Language Processing
NN	:	Neural Networks
NP	:	Noun Phrases
PoS	:	Part of Speech
RST	:	Rhetorical Structure Theory
SaaS	:	Software as a Service
SEO	:	Search Engine Optimization
SKOS	:	Simple Knowledge Organization System
SQRT	:	Square Root
SVM	:	Support Vector Machine
TB	:	Terabyte
TF	:	Term Frequency
TF-IDF	:	Term Frequency - Inverse Document Frequency
UNESCO	:	United Nations Educational, Scientific & Cultural Organization

ABSTRACT

Full Name : Mohammed Suliman Al Logmani

Thesis Title : A Keyphrase Extraction Algorithm and its Potential on Arabic

Major Field : Information & Computer Science

Date of Degree : May 2013

Keyphrases are phrases that express the main topics discussed in a given document. Due to the unavailability of keyphrases in a large amount of digital content, the need for efficient Automatic Keyphrase Extraction algorithms has risen. Automatic Keyphrase Extraction aims to utilize the advancement in computation power and speed to resolve the problems of discoverability and assigning keyphrases without the costs (in efforts and time) associated with human indexers. In this research work, we studied some of the features that can be used to enhance the quality of extracted keyphrases and applied them on an algorithm called Keyphrase Extraction Algorithm (KEA). In addition, we carried out an analysis study for the enhanced KEA along with other state-of-the-art algorithms. The evaluation showed some enhancement. The evaluation effort is conducted using two datasets. The first one contains scientific papers from the research world. The second dataset, which contains documents in Arabic, is created as part of this work.

ملخص الرسالة

الاسم الكامل: **محمد سليمان اللقماني**

عنوان الرسالة: خوارزمية لاستنباط العبارات المفتاحية وإمكانية تطبقيها على اللغة العربية

التخصص: قسم علوم الحاسب الآلي والمعلومات

تاريخ الدرجة العلمية: مايو 2013

تُظهر العبارات المفتاحية في مستند ما المواضيع الأساسية المناقشة في ذلك المستند. ونظراً لعدم توفر العبارات المفتاحية في الكثير من مراكز المحتوى الرقمي، فقد أصبحت الحاجة ملحة إلى خوارزميات عالية الكفاءة لاستخراج العبارات المفتاحية.

تهدف برمجيات استخراج العبارات المفتاحية آليا إلى الاستفادة من التقدم في الحوسبة من حيث السرعة والكفاءة لحساب حل مشاكل استكشاف واستخدام العبارات المفتاحية دون التكاليف (في الجهد والوقت) المرتبطة بعمل البشر في تصنيف المستندات.

ندرس في هذا العمل البحثي بعضا من السمات التي يمكن استخدامها لتحسين جودة استخراج العبارات المفتاحية وتطبيقها على خوارزمية تدعى "خوارزمية استخراج العبارات المفتاحية". ونجري أيضاَ دراسة تحليلية للخوارزمية المحسنة مقارنة مع بعض الخوارزميات المستخدمة في نفس المجال.

ونستخدم في هذه الدراسة التحليلية مجموعتين من مجموعات البيانات. تحوي الأولى مستندات تمثل أبحاثا علمية باللغة الإنجليزية. بينما تحوي الثانية مستندات أنشأناها كجزء من هذا العمل تمثل وثائق باللغة العربية.

CHAPTER 1

INTRODUCTION

A keyphrase refers to a meaningful and significant expression. A keyphrase consists of one or more words in a document [1]. When choosing appropriate keyphrases for a document, we create a highly condensed summary for the document. This summary can be used to describe the document. For this reason, the usage of keyphrases is not limited to digital libraries but it is also applied to physical libraries to describe the library material [2]. The keyphrases can be used to label the document and tag important topics and ideas in the document. Many authors manually extract and assign keyphrases to their documents; especially in scientific articles like journal & conference papers. However, other authors do not assign keyphrases to their documents, especially articles from newspapers and magazines. Extracting keyphrases manually can be considered a time consuming, expensive, and error-prone process. These issues are clear when the number of documents to be analyzed is huge, as seen in journals and scientific repositories. To overcome this problem, several research studies discussed and proposed different methods to extract keyphrases automatically [3]. The process of extracting keyphrases is achieved systematically and with no or minimal human intervention.

Since keyphrases represent the meaning of a document in a compact way, extracting keyphrases has been applied in text mining research and many Natural Language

Processing (NLP) applications. Below are some examples of keyphrase extraction applications:

- Automatic document indexing [4] and [5].
- Document summarization [6], [7], [8], [9], [10], and [11]
- Documents classification, filtering and clustering [12], [13], and [14]
- Topic detection and tracking [15]
- Information visualization [16]
- Knowledge Management [17]

Additionally, metadata navigation and filtering is one of the new usages of keyphrases adopted by new technologies like Microsoft SharePoint Server 2013[1]. In this field, metadata (keyphrases) provides a way to filter the view of list content and enhances browsing experience. This approach is also effective with large amount of content.

Generally, the process of automatic keyphrase extraction is performed in two steps [18]:

1. Candidate Selection: where the terms and phrases are analyzed, correlated, and selected. There are several methods that are used as part of this step, e.g., stemming and stop words identification to eliminate unimportant words.

Some of the difficulties discussed in this step are identifying terms and phrases, ignoring non-terms, and dealing with phrase variations. The following examples illustrate some phrase variations:

- word order: "service quality" is not the same as "quality service"

[1] http://office.microsoft.com/en-us/sharepoint/

- word adjacency: "knowledge management" together is not the same as "knowledge" separated by words and then the word "management"

Additionally there are some difficulties faced in the candidate selection like:

- synonymy which refers to multiple ways to express a concept
- polysemy which means multiple meanings of a word/words.

2. Feature Engineering: In this step, all selected candidates from the previous step are assessed to filter out irrelevant candidates and rank the ones that can be used as phrases to describe the document. A difficulty noticed here is the specific domain features versus general domain features. If the features are created for a specific domain, then they can be written to suit that domain. However, they cannot be generalized for all domains. For example, the section related features mentioned in [18] give more weight to sections like abstract, related work, and conclusion. These sections are usually found in the scientific domain. These features are going to result in enhancing the quality of retrieved keyphrases in the scientific domain, but are not applicable for other domains such as news articles and meeting minutes.

We discuss the motivations behind this research work in section 1.1. Section 1.2 defines the problem under consideration. In Section 1.3, we present the objectives of this research work. The structure of the thesis is presented in Section 1.4.

1.1 Motivation

In many organizations document writers do not tend to assign keywords and keyphrases to their documented work. To illustrate the problem, Saudi Aramco[2] has 73,071,625 electronic documents distributed on different content management systems (CMS) and utilizing more than 22 terabyte (TB). Among these documents, there are more than 40 million documents which do not have any metadata assigned to them, except for the title. This makes identifying, retrieving, and classifying these documents very hard. Extracting keyphrases automatically would solve this problem and benefit Saudi Aramco greatly. Hence, utilizing automatic keyphrase extraction algorithms and tools would preserve the wealth of information and knowledge of previously published work. This is the biggest motivation behind this work as it is related to my own workplace.

Even though there has been a lot of work conducted in the area of automatic keyphrase extraction, yet there is a room for enhancement and improvement. Exploring some of the newly proposed improvements and applying them to other automatic keyphrase extraction algorithms were also extra drivers for this work.

Moreover, the Keyphrase Extraction Algorithm (KEA), which is one of the most worked on algorithms in the area of keyphrase extraction[19], relies only on four features: term frequency–inverse document frequency (TF-IDF), first occurrence, phrase length, and node degree. This leaves an area of improvement by adding new features like the ones mentioned by other researchers including Su Nam Kim et al. [18]. Su Nam Kim et al. suggestions were not implemented and evaluated on KEA, instead they were implemented with another algorithm [20].

[2] Source: Saudi Aramco, Corporate Applications Department March 29th , 2013

When we started to investigate the potential of extracting keyphrases from Arabic documents, we found that there are no available comprehensive Arabic dataset that can be used for Arabic keyphrase extraction. This motivated us to create an Arabic dataset that can be used in this work and future work, possibly as a benchmark.

1.2 Problem Definition

This research work addresses two problem areas: preparing an Arabic dataset to be used for keyphrase extraction research and enhancing the Keyphrase Extraction Algorithm.

1.2.1 Preparing an Arabic Dataset

Having a dataset is very critical to test the accuracy of different keyphrase extraction algorithms. In the Arabic domain, the Arabic datasets known to us before this study had a small number of documents in each set. For example, KP-Miner [21] used 100 documents where El-Shishtawy et al. [22] used only 50 documents. Additionally, those datasets were not publicly available for researchers.

We would like to prepare an Arabic dataset that contains a large number of documents along with their manually extracted keyphrases. This dataset will be available for future Arabic research in Keyphrase extraction and information retrieval.

In order to create our dataset, we are going to select 400 Arabic documents from different sources. This number can be considered sufficient to test keyphrase extraction algorithms. In the literature, we found that the work carried out in [18], [23], and [24] used a dataset consisted of 244 documents, Zhang et. al. [3] used 600 documents, You et al [25] worked on 500 documents, and Sarkar et al [26] tested their work using 150 documents. After the documents selection, we will classify and organize them into

different categories based on their topics. Finally, we will manually extract and verify the keyphrases for each document in the dataset. The manual assignment and verification for keyphrases is done to create the baseline that we are going to compare with the automated approaches later.

1.2.2 Enhance the Keyphrase Extraction Algorithm (KEA)

We would like to propose enhancements to a keyphrase extraction algorithm, which will lead to enhance the obtained results from the original algorithms.

KEA which is one of the most worked on algorithms in the area of keyphrase extraction relies only on four features i.e. TF-IDF, first occurrence, phrase length, and node degree. This leaves an area of improvement by enhancing the existing features or adding new features.

We will propose several possible enhancements to KEA and evaluate them. These enhancements will be done by changing the term weighting functions, term occurrence features, and a combination of them.

1.3 Objectives

This research work aims to meet the below objectives:

- Conduct a survey of the current automatic keyphrase extraction approaches and their classifications.
- A study for some features used in selecting keyphrases and propose new ones.
- Implement the studied techniques as part of Keyphrase Extraction Algorithm (KEA). This algorithm is discussed in section 2.2.3.

- Build a new corpus (dataset) that contains Arabic documents along with their manually extracted keyphrases and make it available for researchers.
- Find the outperformed variation of KEA after conducting comparison analyses among these variations. This outperformed variation will be considered our baseline.
- A comparison analysis to compare our baseline with existing algorithms found in the research like Keyphrase Extractor [27] and KP- Miner [21].

1.4 Thesis Organization

This section describes the structure of the remaining parts of the thesis.

Chapter 2 (literature review): In this chapter, we highlight the different keyphrase extraction algorithms found in surveyed studies. Due to the importance of Keyphrase Extraction Algorithm (KEA) in our work, we discuss it in more depth. We also, discuss Automatic Keyphrase extraction with Arabic along with some commercial keyphrase extraction solutions found for Arabic. The chapter also presents the use of keyphrase extraction algorithms as services.

Chapter 3 (Datasets): This chapter reports the datasets we used in our study. We used two datasets: an English dataset and an Arabic one. For the English dataset, we show the document sources and categorization. As for the Arabic dataset, we present the methodology of preparing it. We also show the sources used to collect the articles with the selection criteria. Additionally, we describe its organization and the process performed for the manual keyphrase extraction and keyphrase verification.

Chapter 4 (Augmented Keyphrase Extraction Algorithm): We discuss in this chapter our hypotheses planned to be tested on our experiments. We also explain the experiments setup including the preparation steps and parameters chosen. After that, we explain our evaluations on the Arabic and English datasets. Finally, we conclude the chapter with stating the evaluation techniques used in this study.

Chapter 5 (Experimental Results and Discussions): This chapter presents the experimental results for our comparative analysis explained in chapter 4. It discusses the results obtained from each alteration done on KEA as well as other algorithms chosen for the study. The chapter also includes observations noted from the overall experiments.

Chapter 6 (Conclusion and Future Work) the contribution of this research work to the field of phrase extraction is presented in this chapter. The chapter also offers some ideas for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Automatic keyphrase extraction aims to extract a set of phrases that are highly relevant and most descriptive phrases for the input text. The process of extracting keyphrases is achieved systematically and with no or minimal human interference. Keyphrase Extraction algorithms mine the data corpus to extract important phrases and label the documents with these phrases.

This chapter discusses background information about the field of automatic keyphrase extraction. Section 2.2 starts the review by studying some state-of-the-art keyphrase extraction algorithms along with their classifications based on their domains and used approaches. Section 2.3 lists some of the topics related to Arabic language including some research algorithms and commercial tools. Section 2.4 discusses term weighting formulae and their usage in the automatic keyphrase extraction.

2.2 Classification of Automatic Keyphrase Extraction

This section presents the classification of automatic keyphrase extraction algorithms based on the domains they are targeting and the used approaches.

2.2.1 Classification based on targeted domains

Generally, automatic keyphrase extraction algorithms are developed to target a general domain or a specific domain. The keyphrase extraction algorithms targeting general

domains tend to use generic features to extract the keyphrases without any focus on the nature of the input text. Examples of these algorithms are in the " KP-Miner" system presented by El-Beltagy and Rafea [21] and an algorithm that uses semantic information on terms gathered from a domain-specific thesaurus presented by Medelyan and Witten [28].

In the research field of automatic keyphrase extraction, several of the found work used scientific papers as their specific domain, such as Kumar and Srinathan [27] and Nguyen and Kan [20]. It is known that scientific articles are more organized than other documents in other domains. Moreover, the availability of datasets and verified keyphrases in scientific papers, make the evaluation and comparison among algorithms easier and faster than using other domains. According to Su Nam et al. [18], the position of the keyphrase plays an essential role in scientific papers i.e. important terms, as the abstract, come earlier. Other specific domains include extracting keyphrases from social networks like Twitter (see Bellaachia and Al-dhelaan[29] as an example) and videos lectures (see Balagopalan et al. [30]).

2.2.2 Classification based on the used approach

Generally, the approaches used in keyphrase extraction can be classified into two categories: free indexing based approach and controlled indexing based approach.

I. Free indexing based approach: this approach does not require linguistic or vocabulary support; instead, the selection of keyphrases relies on the utilization of

some Artificial Intelligence (AI) techniques. This approach can be classified into two further types:

a. Learning based approach: supervised or unsupervised machine learning approach where keyphrases can be extracted using examples. This approach requires the training data to be large and generic in order to get accurate results. The learning based approach was used by Zhang et al. in [3], Balagopalan et. al. in [31], and Medelyan and Witten in [2].

b. Non-Learning based approach: this approach relies on statistical methods, grammatical facts, or lexical information to extract keyphrases with no training data. Although the results from this approach are less accurate than the ones from the learning based methods, this approach is less costly to implement as it does not require any training data to be established. It was used by Nguyen and Kan in [20] and Kumar and Srinathan in [27].

II. Controlled indexing based approach: this approach uses a controlled vocabulary (e.g. dictionary, thesaurus, or a list of terms) to extract keyphrases. Using the vocabulary is applied in machine learning approaches. The work done by Medelyan and Witten in [2] and by Lopez and Romary in [23] are examples of this category.

Table 1 illustrates some of the automatic keyphrase extraction algorithms along with some classifications. The headers of the table include the following:

- **Indexing Approach:** the indexing approach used whether they are free indexing or using controlled dictionaries/vocabularies.

- **Machine Learning (ML):** an indication if they are utilizing machine learning techniques or not.
- **ML Type:** the type of machine learning approach: supervised or unsupervised.
- **Domain: the** domains that these algorithms were developed/used for.
- **Language:** Primary language used.

Table 1- Some of automatic keyphrase extraction algorithms

Work	Indexing Approach	Machine Learning (ML)?	ML Type	Domain	Language
KEA [31]	Free	Yes	Supervised (S)	Scientific	English
KEA++ [28] **,** [2]	Controlled	Yes	S +Unsupervised (U)	General	Multiple
Keyphrase Extractor [27]	Free	No	---	Scientific	English
CollabRank [1]	Free	Yes	U	News Articles	English
Nguyen & Kan [20]	Free	Yes	S (U after modification done by [18])	Scientific	English
Keyphrase Extraction using Conditional Random Fields [3]	Free	Yes		Scientific	Chinese
Keyphrase Extraction with a Refined Candidate Set [25]	Free	No	---	Scientific + Wikipedia	English
KP-Miner [21]	Free	No	---	General	English + Arabic
Keyphrase Extraction from Meeting Transcripts [32]	Free	Yes	S	Meeting videos/audios	English
Keyphrase Extraction for short technology text [33]	Free	No	---	Technical pages	Chinese
Keyphrase Extraction using NN [26]	Free	Yes	S	Scientific	English
Arabic Keyphrase Extraction [22]	Free	Yes	S	General	Arabic
WINGNUS [24]	Free	Yes	S	Scientific	English
HUMB [23]	Controlled	Yes	S	Scientific	English
Extended Keyword Extraction Method [34]	Free	No	---	General	Chinese
Keyphrase Extraction of Video Lectures [30]	Free	Yes	S	Videos	English
Graph-based Keyphrase Extraction [29]	Free	Yes	U	Twitter tweets	English
Microblog Keyphrase Extraction [35]	Free	Yes	U	Micro-blogs	Chinese
KERT[36]	Free	Yes	U	Short text	English

In the next few paragraphs, we describe some of the characteristics of each of the keyphrase extraction algorithms mentioned in “Table 1”:

Keyphrase Extractor [27] was based on a free indexing approach and it was targeting the domain of English scientific papers. It focused on the development of an algorithm that did not need a machine learning phase. The Keyphrase Extractor algorithm used an n-gram filtration technique, which used the words of input document to filter n-grams along with their weight. The first step of the n-gram filtration technique was done using LZ78 [37] data compression technique. LZ78, which was introduced by Lempel and Ziv [37], is a dictionary based compression algorithm used to create a dictionary of distinct n-gram candidates. In order to compress the data, LZ78 replaces repeated data mentioned in the input dictionary. After the utilization of LZ78, Keyphrase Extractor uses a refinement step to eliminate n-grams that do not fit the selection criteria. After that, a pattern filtration algorithm is used to get the candidates from the list resulted from previous step. The last step is the term weighting scheme that introduced the importance of the sentence position in the document and the phrase position in a sentence for documents of scientific domain. This is achievable since scientific domains are more organized than other domains.

CollabRank [1] was based on a free indexing approach and on unsupervised machine learning methods. The algorithm was implemented on papers of news article domain in English. CollabRank was implemented by first employing the clustering algorithm to obtain appropriate document clusters, it was noted that the algorithm performance relied on the quality of the document clustering. Then a graph-based ranking algorithm was used to select a group of words that can represent the cluster in whole for collaborative

single-document keyphrase extraction within each cluster. This was because multiple documents within an appropriate cluster context have mutual influences and usually contain useful clues to help extract keyphrases from each other. This idea was borrowed from human's perception that to understand a topic expressed in a document better, the user should read more documents about the same topic. In the end, they used a document-level keyphrase extraction algorithm which selects candidate phrases from the group of words that represent the cluster. The algorithm then evaluates the candidates for each single document in the cluster to find keyphrases for each document individually.

Nguyen & Kan [20] is another free indexing algorithm which capitalized on features that capture the logical position and additional morphological characteristics of keyphrases e.g., suffix sequence, part-of-speech (PoS) sequence, and Acronym. The algorithm is a domain-dependent algorithm specialized in scientific publications. The algorithm starts as a supervised machine-learning algorithm where all the classes used to classify the content are set. However, Su Nam Kim et al. [18] modified the original algorithm and made it as unsupervised algorithm. According to Su Nam Kim et al. [18], the modification outperformed the results obtained from the original algorithm and KEA.

Keyphrase Extraction algorithm using Conditional Random Fields [3] utilized the Conditional Random Fields (CRF) model to label the strings and then used them to extract the keyphrases. CRF model is a probabilistic method to segment and label the sequence of given data. It computes the probability distribution of the words based on identified features. The CRF model used in this algorithm relays on 22 features and uses them to label the keyphrases. Some of these features are length of the word, POS, and

next word. The dataset used in this algorithm consists of 600 academic papers in the Chinese language.

Keyphrase Extraction with a Refined Candidate Set is an algorithm proposed by Wei You et al. [25]. The main idea of this algorithm is to decrease the number of candidates to a small number that would result in enhancing the performance. In this algorithm, the core words (keywords) are used to identify potential keyphrases. This is based on the assumption that keyphrases usually contain an important word. After that, the keyword set is expanded to get the keyphrases. The expansion is done by evaluating the adjacent words to the keyword. To test this statistical algorithm, a dataset consisting of 500 scientific papers was used.

Keyphrase Extraction from Meeting Transcripts [32] is a supervised algorithm which investigates the transcripts of meetings to extract keyphrases. Beside the term frequency and position features, this algorithm uses term specificity features to capture specific words with precise meaning, such as the algorithm selects tiger and not animal. In addition, the algorithm uses decision-making sentences features to capture keyphrases. To illustrate this concept, let us take the sentence "we must buy an airplane". This algorithm would realize the word "must" as a decision indicator, and select the "airplane" as a keyphrase candidate. The used corpus included 29 transcripts written by human and generated by automatic speech recognition (ASR) tools.

Keyphrase Extraction for short technology text [33] is an algorithm to extract keywords from short paragraphs taken from different technological topics. The dataset used consisted of 3000 Chinese text from 20 fields like mathematical theory and farm

machine. To obtain the keywords, the authors of [33] used known statistical methods, such as frequency and PoS, to extract candidate keywords. Their contribution mainly was the refinement step at the end which aimed to detect unknown keywords and eliminate them from the final results. This step is done by calculating the maximum matching strings to the candidate keyphrase. The calculated number is called string frequency. The string with highest frequency would be considered a keyphrase. Candidates with low number would be eliminated.

Keyphrase Extraction using Neural Networks [26] is a supervised machine learning algorithm. It employs a Multi-Layer Perception (MLP) neural network model to categorize any candidate phrase as a keyphrase (positive) or not a keyphrase (negative). Sarkar et al. [26] designed the algorithm to start by identifying and tagging noun phrases (NP) from a given document. Then it uses features like phrase frequency, phrase position, word length, and the links between the candidate phrases to identify and rank the keyphrases. When experimenting with the algorithm, 150 documents (100 training and 50 testing) were used. The documents were journal articles from three fields: law, medical, and economics.

WINGNUS, by Nguyen and Luong [24], is an extracting system for a specific domain which is the English scientific articles. The algorithm uses two steps to complete the extraction process. The first step includes the utilization of a tool called “SectLabel” to classify the documents and extract their logical structure. The logical structure in a scientific paper includes title, authors, abstract, conclusion, and body text. There were two motivations of classifying the document based on the logical structure and they are:

1. to investigate which section has more keyphrases and
2. to minimize the computation time by limiting the search on selected sections.

The second step of WINGNUS is the selection criteria for candidate keyphrases that includes measures like:

- term frequency–inverse document frequency (TF-IDF) measure
- in-title: investigates if the candidate phrase is mentioned in the title or not.

The dataset used to test WINGNUS consisted of 244 scientific papers in the English language. It is worth mentioning that this dataset was chosen to be used in our work as the English dataset.

HUMB [23] is based on a module called GROBID, stands for Generation Of Bibliographic. HUMB is a machine learning (ML) algorithm based on a combination of three ML models: Decision Tree, Multi-Layer Perception (MLP), and Support Vector Machine (SVM). The algorithm of HUMB starts by classifying and extracting the logical sections like header, citations, titles, figures, etc. Then, it selects candidate terms, and calculates the value of each feature independently to rank the keyphrases. When WINGNUS was compared with HUMB, WINGNUS outperformed slightly HUMB in respect of precision, recall, and F-score.

Extended Keyword Extraction Method, presented in [34] by Hong and Zhen, is a statistical-based algorithm which extends the Term Frequency (TF) feature to enhance the precision and recall. The extended TF employs a multi-step approach to calculate and identify the keyphrases. This approach includes the use of grammatical model to identify unary, binary, and ternary keywords. In addition, it includes considering some

characteristics of the keywords like position, frequency, part of speech (PoS), and morphology. The approach also uses Support Vector Machine (SVM) to classify and optimize the results. The work was done for the Chinese language only. The dataset used when experimenting with this work consisted of 400 articles collected from ACM digital library, Xinhua News Agency, and the internet. The experimental results showed an improvement of the extended TF over regular TF by 26% in precision and 16% in recall.

Automatic Keyphrase Extraction of Video Lectures [30] is a supervised machine learning algorithm that was developed with a focus on extracting keyphrases from videos transcripts in the English language. The domain for this work was classroom lectures including university video lectures and webcasts. The algorithm divides the lecture transcripts into different segments so each section of the video can be searched and viewed separately. The process of extraction keyphrases is similar to the one used in KEA (explained in 2.2.3). In addition to some of the features employed by keyphrase extraction algorithms, this algorithm uses features related to lectures and speeches in general. An example is “cuewords” which are used to capture important words that are mentioned rarely. In the lectures, such keyphrases come after cuewords like 'known as', 'titled as', or 'called as'. This algorithm was compared against KEA and an algorithm based mainly on C-Value (statistical approach extract keyphrase from nested collection) [38]. The used data contained close to 300 lectures transcripts in the field of computer science. The experiment results showed that the value of F-Score of this algorithm is 46% better than KEA and 33% higher than the C-Value based algorithm.

The research work of Bellaachia and Al-dhelaan proposed in [29] dealt with extracting keyphrases from Twitter. This **Graph-based Keyphrase Extraction** algorithm was

targeting large dataset of Twitter tweets. These tweets were short text messages of up to 140 characters. The aim of their research was to get some meaningful information out of these tweets based on analyzing of hashtags (tags or metadata identified by authors using the # symbol to mark topics or important words in their tweets) mentioned in these tweets. The proposed solution in this work relied on a graph based unsupervised approach. In this approach, first, the large dataset is clustered into topics using Latent Dirichlet Allocation (LDA) model [39]. LDA is a probabilistic Bayesian model which automatically discovers the topics in text corpus. The clustered topics are then passed to a learning model to find which hashtags can represent these topics. Hashtag frequency is one of the measures used in this analysis. The final step is to construct graph-based ranking model to rank the top keyphrases. One of the challenges faced in this work was the noise usually accompanied Twitter tweets. Tweets in many cases are written without proper grammar and they contain some symbols and slang words that cannot be used as keyphrases. The dataset used when experimenting with this algorithm consisted of 31,227 tweets with more than 4,079 hashtags. The results of the algorithm were compared against human evaluation and no other algorithms.

In a close domain to Twitter, Liao and Huang [35] developed an unsupervised algorithm targeted extracting keyphrases from micro-blogs which usually are smaller in content when compared with regular blogs. The dataset used in this work contained 25,660 documents collected from the Chinese micro-blogging website “weibo.com”. In order to extract keyphrases, context similarity features are used. These features are calculated for all candidate keyphrases by identifying their conceptual approximation in an online knowledge database called HowNet [40]. For each two-candidate phrases, a graph

structure is constructed based on the inter-concept and inter-attribute relations. The shorter the graph is, the greater the similarity between the phrases. To evaluate this algorithm, first, the collected micro blogs were divided into 15 datasets based on topics. Then three experiments were performed and the results, as reported by the author, were better than the ones obtained using another keyphrase extraction algorithm based on topical ranking.

KERT, stands for Keyphrase Extraction and Ranking by Topic, introduced by Danilevsky et. al. [36], is an unsupervised algorithm focuses on short texts (more specifically titles of papers). It aims to cluster a dataset and extract keyphrases based on the topics inside the dataset. To achieve its goal, KERT follows a three-step process: first, it uses Latent Dirichlet Allocation (LDA) [39] approach to cluster all words into many foreground topics and one background topics. Foreground topics include technical words mentioned in the titles where background topic contains common words used and found in titles from different topics. In the second step, keyphrases are generated by assigning a topic label to each word mentioned in the title. If several words frequently referenced that topic, then that topic is considered a candidate keyphrase topic. Third step is ranking the extracted keyphrases based on four criteria:

1- coverage in which it covers many documents,
2- purity: meaning it is frequent only in related topics,
3- phrase-ness which means that a collection of words frequently are used together, and
4- completeness which refers to the full length of the phrase and not just a subset of it. To test this algorithm, two experiments were conducted using a dataset of

scientific papers contained 33,313 titles and another one with 9,722 titles. The performance should an improvement of 50% than the used baseline. This algorithm can be categorized as a clustering algorithm more than keyphrase extractor algorithm since the focus is on the topics titles and not on the body of text.

2.2.3 Keyphrase Extraction Algorithm (KEA)

KEA, which is publicly available, was used in several research studies in the automatic keyphrase extraction area. Some examples can be found in [18], [41], [27], [42] , [34] ,[43] and [44]. KEA was proposed in 1999 as a supervised algorithm and was tested against scientific papers. KEA++ [28] and [45] had added the possibility of using controlled vocabulary in Simple Knowledge Organization System (SKOS) format [46]. SKOS is a W3C standard data model to allow the interoperability and sharing of knowledge organization systems, like classification schemes, thesauri, and taxonomies in the Web. For the purposes of keyphrase extraction, SKOS provides the Keyphrase Extraction algorithm with a dictionary that can be used to refine and enhance the quality of the extracted keyphrases. The latest version of KEA is 5.0. It combines controlled and free indexing. Figure 1, inspired from [47], illustrates how KEA works. The process is summarized as follows:

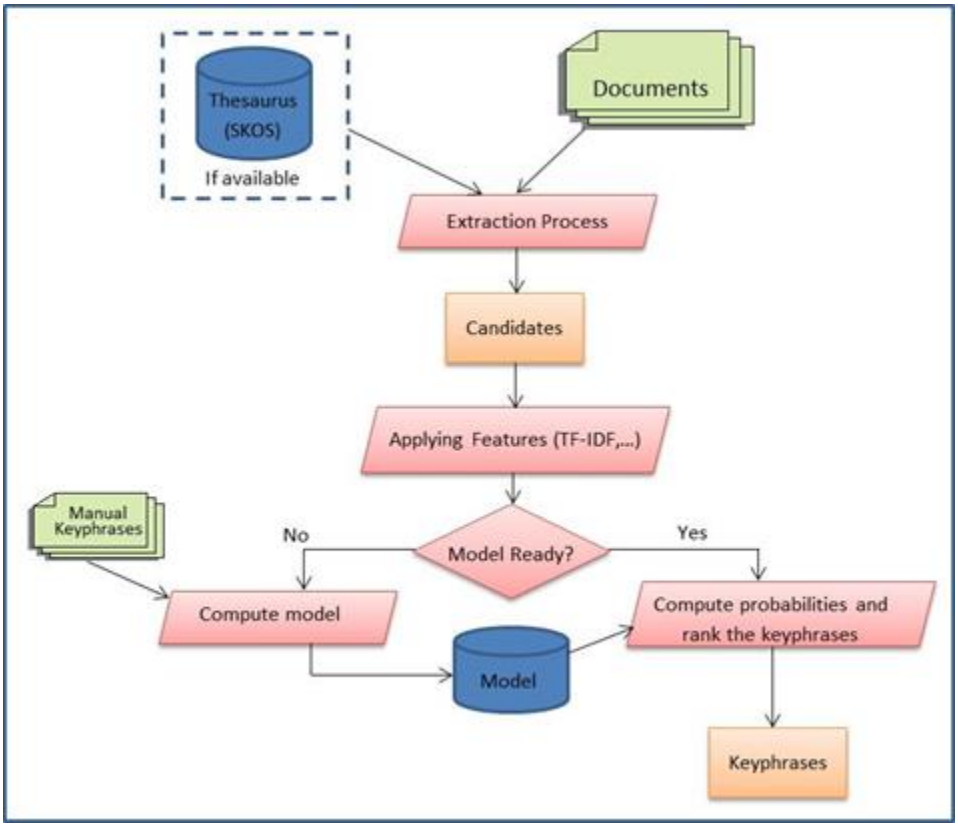


Figure 1 - Overview of KEA

I. KEA takes a directory as input; this directory contains all documents in which the keyphrases will be extracted from.

II. Starting with version 4.0, KEA can use a controlled vocabulary (thesaurus) in Simple Knowledge Organization Systems (SKOS) format.

III. KEA then starts extracting candidates. In case of not specifying a thesaurus file (supervised procedure), these candidates are n-grams of a predefined length and do not start or end with a stop word. In case of controlled indexing, the candidates are n-grams that match SKOS terms.

IV. Four features are computed on all candidates. These features are:

a. Term frequency–inverse document frequency (TF-IDF): which is calculated as

$$\mathbf{TF-IDF\ (T,D)} = \frac{freq(T,D)}{|D|} \times -log_2 \frac{count(T)}{N} \qquad \textbf{Formula 1}$$

Where freq(T,D) is the frequency of term (T) in document (D), |D| is the number of words in D, Count(T) is the number of occurrence of T in the global collection, and N is the number of all documents in the collection.

b. First occurrence of a candidate phrase: it is the phrase distance in words from the beginning of the document, normalized by the total number of words in that document. First occurrence of a candidate phrase emphasizes the location of the first occurrence of the phrase in proportion to the documents length.

c. Phrase length: it is the number of words in that phrase.

d. Node degree: it is the number of candidate phrases related semantically to this phrase. This feature is derived from the graph theory where it calculates the number of thesaurus connections between the tested phrase and other candidates. Phrases with the highest degree node value can be considered important to the document.

V. KEA builds the learning model from manually indexed documents. Each document in the training content should be associated with a file containing defined keyphrases. All candidates found from step (III) and found in the manually assigned phrases will be marked as positive samples where the rest will be marked as negative samples. The learning model then will be computed after analyzing the feature values gained from step (IV) for both positive and negative

candidates. The results will show how each feature value for each phrase is distributed.

VI. Keyphrases will be extracted by calculating the probability of each candidate to be a keyphrase using the learning model (step V) and feature values. Based on the number of required keyphrases, the top N candidates with the highest probabilities will be selected.

Irfan et al. [48] introduced a web based tool for keyphrase assignment based on KEA called KeaKAT. The motive for their work was to enhance user experience with KEA. Current implementations of KEA (including our work) are client based and require knowledge on preparing the environment to extract the keyphrases. KeaKAT is a web based targeting end users on 9 usability aspects like understandability, user-friendliness, and learnability. It follows the similar process as KEA except that there is refinement step in which users help in aligning the extracted keyphrases. According to Irfan et al [48], KeaKAT was tested by users and compared with two other systems: Maui [49] and Agrotagger II [50]. Users evaluated the three systems in the 9 usability characteristics and KeaKAT ranked top on all characteristics.

2.3 Keyphrase Extraction and Arabic Language

In this section, we discuss keyphrase extraction work related to Arabic in the research and commercial worlds. In addition, this section presents text summarization as one of the applications of Arabic keyphrase extraction.

2.3.1 Arabic Automatic Keyphrase Extraction Algorithms

We found few studies conducted on Extracting Arabic keyphrases. KP-Miner which was proposed by El-Beltagy et al. [21] was used to extract keyphrases from English text and Arabic text. It has several similarities to KEA [31] where both of them use TF-IDF and first occurrence when selecting candidates to be keyphrases. However, KP-Miner has some refinement steps which proved to make it outperform KEA version 3 [21], as reported by the authors. However, experiments of KP-Miner on Arabic were done on a set of 100 Wikipedia documents.

El-Shishtawy et al. [22] proposed an algorithm which uses both linguistic features as well as a supervised machine learning technique. The algorithm uses three stages to get the final keyphrases list. The first one is to preprocess the document and extract the PoS tokens. In the second stage, the candidate keyphrases are extracted based on syntactic rules. In the final stage, the candidates are ranked based on their linguistic and statistical scores. The system was experimented using dataset consisted of 50 documents. The results of these experiments were compared to results obtained from KP-Miner [21] and Sakhr [51] extractor. The comparison table showed some enhancement of this system over the two other systems.

In addition to the above algorithms, there are some research efforts that cover multiple languages including Arabic. Some language-independent examples are the work of Sudo et al. in [52] and Paukkeri et al. [53]. In [52], the text is translated first and then the keyphrases are extracted. Paukkeri et al. [53] introduced an extraction method called *Likey*. In this method, the phrases with highest frequency ratio are selected as keyphrases. The experiments were performed on 11 languages.

2.3.2 Commercial Keyphrases Extraction Solutions

In addition to the algorithms found in the research world, the keyphrases solutions/tools are available commercially. Sakhr Keywords Extractor [51] can analyze Arabic text and extract the keyphrases. Extractor [54] is another commercial tool which can be used by search engine optimization (SEO) and document management companies.

Also, some companies are offering commercial text mining products which feature a module to extract keywords. For example, BasisTech [55] offers Rosette Linguistics Platform which provides classification, analysis, indexing, and searching for unstructured text from different languages i.e. Asian, European and Middle Eastern (including Arabic). Another example is SmartLogic [56] which introduces a suite called Semaphore. It includes a Natural Language Processing (NLP) engine that accepts text files from applications and provides file type and text language recognition, Part of Speech (PoS) recognition, noun phrase extraction, and classification.

Since the above products are commercial, the technical aspects of how these products work are not released.

2.3.3 Keyphrases Extraction Algorithms as Services

There are several companies/websites offer extracting keyphrases/keywords as web services. This model is known as Software as a Service (SaaS). These services are exposed via application programming interface (API) provided by the services providers. To mention some examples, Yahoo provides developers with the ability to extract keyphrases by sending their text using HTTP calls. This service is free and is based on

Yahoo!'s Search API [57]. WordsFinder [58] provides their services free for personal use and require license for commercial use. Alchemy API [59] is a service utilizes machine learning algorithms and provides keyphrase extraction, named entity extraction (people, countries…), concept tagging, and topic categorization.

2.3.4 Arabic Text Summarization

Text summarization is one of the applications of keyphrase extraction. Text summarization programs aim to describe an article in few sentences or paragraphs. For the Arabic language, El-Shishtawy and El-Ghannam [8] proposed an algorithm which uses statistical and linguistic measures to summarize articles. Their approach reduces the problem into sub problems by extracting and focusing on sentences. These sentences then are scored based on the important phrases they have. After removing or minimizing the redundancy, the most informative sentences are selected to be part of the summary. In their work, the authors employed the Keyphrase Extractor [22] (explained in section 2.2.1) to extract keyphrases.

Azmi and Al-Thanyyan [7] proposed another algorithm for Arabic text summarization. This algorithm consists of two stages. In the first one, the algorithm extracts a preliminary summary based on the Rhetorical Structure Theory (RST) [60]. In the second stage, the system refines and scores the preliminary summary. The summarization process involves eliminating stop words, extracting words roots, calculating frequencies, scoring each sentence in the preliminary summary, and generating the final summary. An advantage that was reported in this work over the work mentioned in [8] is the possibility to allow the user to decide on the size limit of the result. In addition, the experimental

results claimed better performance when compared with other Arabic summarization algorithms.

2.4 Term Weighting

In the field of Information Retrieval (IR) and text categorization/classification, term weighting is used to improve retrieval performance. The performance is measured using known metrics like recall and precision. In this field, a collection of n documents which are indexed by m terms can be denoted as term-by-document matrix (m x n). In this matrix, each element (a_{ij}) is defined as the weighted frequency where the term i can be found in the document j. [61]

a_{ij} is composed of three different types of term weighting variables and can be defined as $a_{ij} = l_{ij}\ g_i\ d_j$, where l_{ij} is the local weight of term i in the document j, g_i is defined as the global weight for the term i in the document collection, and d_j is the normalization factor for the document j which signal whether the document j is normalized or not.

As mentioned in section 2.2.3, KEA uses TF-IDF as one of the criteria to extract keyphrases. In this thesis, we are investigating altering KEA by replacing the TF-IDF weighting mechanism with some of the found term weighting formulas like:

I. Best Match 25 (BM25) which was introduced by Robertson and Jones [62]. Later, several enhancements were proposed to modify and extend BM25. Some of which were the work of Mathias and Largeron in BM25t [63] and the work of Karkali in BM25H [64]. The IDF section in BM25 can be calculated as follows:

$$IDF(qi) = \log\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}\right) \qquad \text{Formula 2}$$

Where N is the number of documents, $n(q_i)$ is the number of documents which have the term q_i.

II. The probabilistic inverse (IDFP) is a global weight formula similar to IDF [65]. In the IDFP (Formula 3), terms appearing in few documents in the collection are awarded with high weight where terms appearing in many documents in the collection are assigned low weight. For terms found in more than half of the documents set, IDFP assign however, they differ because IDFP actually awards negative weight. The equation below shows how IDFP is calculated.

$$IDFP = \log\left(\frac{N - n_i}{n_i}\right) \qquad \text{Formula 3}$$

Where N denotes the total number of documents and n_i represents number of documents which contain the term i.

III. Logarithmic (LOGA) is a formula for the local term introduced in [65][66] and can be calculated as:

$$LOGA = 1 + \log f_{ij} \qquad \text{Formula 4}$$

Where f_{ij} represents the frequency of the term i occurred in document j.

IV. Augmented log (LOGG) is a local term formula. It was introduced by Erica Chisholm et al. [66] under the assumption that using log is better as a local weight than within-document frequency. The suggested formula is:

$$LOGG = 0.2 + 0.8 \log (f_{ij} + 1) \qquad \text{Formula 5}$$

Where f_{ij} represents the frequency of the term i occurred in document j.

V. Square root (SQRT) is another contribution by Erica Chisholm et al. [66]. The driver behind the SQRT formula was the similarity noticed between the graph of Formula 4) and the graph of $\sqrt{f_{ij}}$. They continued their experiments till they reached to below formula:

$$SQRT = \sqrt{f_{ij} - 0.5} + 1 \qquad \text{Formula 6}$$

Where f_{ij} represents the frequency of the term i occurred in document j.

2.5 Summary

In this chapter, we presented a survey about some of the keyphrase extraction algorithms and tools. The survey included a classification of the found algorithms based on the targeted domain and the used approach. We also described some topics related to Arabic and the keyphrase extraction field. This included a discussion about some found algorithms focused on Arabic like KP-Miner. In addition, we discussed the commercial use of keyphrase extraction tools and services. We also discussed two works of the Arabic text summarization, which is an application of keyphrase extraction.

In this chapter we also explained the KEA algorithm which is the base algorithm that our work is built upon. Two of the other keyphrase extraction algorithms discussed in this chapter, are used in our empirical study, which will be discussed in chapters 4 and 5. Finally, we described some of the term weighting formulas that we used in our comparative analysis found in later chapters.

CHAPTER 3

DATASETS

3.1 Overview

In the previous chapter, we discussed some of the related work to the area of keyphrase extraction. As noticed, having one or more datasets to test these algorithms is very critical to verify, experiment, and measure the efficiency of these systems. This chapter discusses the datasets that we have used in our work. First, in section 3.2, we describe the English dataset with its documents and categories. Second, in section 3.3, we present our Arabic dataset with its structure, organization, statistics, and the methodology used to prepare it. We present the summary of this chapter in Section 3.4.

3.2 English Dataset

The dataset chosen for this work is selected from the datasets submitted for the Workshop on Semantic Evaluation 2010 (Sem-Eval 2010) [41]. This dataset is part of a publicly available repository that contains datasets for automatic keyphrase extraction. All datasets of this repository are available at [67]. The chosen dataset contains 244 documents collected from the ACM Digital Library. The collection includes conference and workshop papers. The input papers ranged from 6 to 8 pages, including tables and pictures. To ensure a variety of different topics was represented in the corpus, the selected papers were chosen from four different research areas for the dataset. In particular, the selected articles belong to the following four 1998 ACM classifications: C2.4: Distributed Systems, H3.3: Information Search and Retrieval, I2.11: Distributed

Artificial Intelligence – Multi-agent Systems, and J4: Social and Behavioral Sciences – Economics. All training and test datasets had an equal distribution of documents from among the categories (see Table 2).

The collection contains of 144 documents as a training data and 100 articles as test data covering all four-research fields. All papers were converted from their original format, such as PDF, into (UTF-8) plain text using pdf-to-text tool.

Table 2 - Distribution of the documents in the English dataset

	Distributed Systems	**Information Search and Retrieval**	**Distributed Artificial Intelligence – Multi-agent Systems**	**Social and Behavioral Sciences – Economics**
Training	34	39	35	36
Test	25	25	25	25
Number of Documents	59	64	60	61

It is worth mentioning that the manual keyphrases for test data are available in stemmed format. In several cases, the stemming process was noticed to be overdone generating over-stemmed keyphrases. As part of this work, the regular or un-stemmed keyphrases are retrieved from ACM library and added to the dataset.

3.3 Arabic Dataset

One of the main contributions of this work is the new Arabic dataset we have prepared. This section explains the sources for the articles and the methodology used to create the dataset. Our dataset contains 400 documents distributed on 18 different categories.

3.3.1 Sources

We have collected the documents from two sources: Arabic Wikipedia [68] and King Abdullah Initiative for Arabic Content [69].

- Arabic Wikipedia: is the main source of articles used in creating this dataset. Arabic Wikipedia contains more than 198,349 pages. For our goal, we obtained 365 articles from Wikipedia. Out of the 365 articles, approximately 200 articles were obtained from a previous work by Shaaban [70] where the rest were collected using BzReader [71]. BzReader is an application that allows offline browsing of the Wikipedia dump files and displays the text-only version of Wikipedia pages. The dump files are available at [72] and they are saved compressed with *.xml.bz2* extension. More information about BzReader and other tools can be found in Appendix B.
- King Abdullah Initiative for Arabic Content (مبادرة الملك عبدالله لإثراء المحتوى العربي) [69]: is an initiative aims to enrich the Arabic content on the internet after noticing the small percentage of Arabic content. According to this initiative, the percentage of Arabic digital content does not exceed 0.3% out of the world content composed of other languages. For our goal here, we obtained 35 articles with focus on medical topics.

3.3.2 Selection and organization

The corpus covers different knowledge areas like religion, history, geography, technology, sciences, sports…etc. Selecting the documents from different fields would help future automatic keyphrase extraction algorithm to cover general domains

and not be tied to a specific domain like scientific papers. These documents vary also in size from 1 to 30 pages. The total number of words in these documents ranges approximately from 172 to 17,589 words. The number of words in the whole dataset is 1,708,168 words distributed on 288,191 lines. The unique number of words is 221,022. The documents are saved in text files with the extension (.txt). The Arabic files are saved using the Unicode format UTF-8. The file name of each document consists of the category of the topic (e.g. animals, economy, history...) followed by a serial number in that category. Examples: the name (*animals 01.txt*) represents the first document in the animals' category where *history 10.txt* represents the tenth document of history category. The largest category with regard to number of documents is the people category with 59 documents where the smallest one is the food category with 3 documents. From the perspective of total number of lines and number of words in each category, countries category came first with 73,757 lines and 427,205 words. In the same perspective, food category ranked last with 624 lines and 3,699 words. This is due to little number of documents in the food category. We also calculated the density percentage defined as the average number of words per file in each category. This measure shows the richness of a certain category based on the longest files they have and not based on the number of documents under that category. When calculating the density score, countries category scored the highest with 7,366. The next highest category is religion with 5,960 average words per file. This category contains 16 files. In this measure, food category scored last with 1,233. For the health and medicine category, the density score is 1,950, which is very small comparing to the number of files (51). The largest file in the Arabic dataset is from

the history category and it is about the Ottoman Empire (الدولة العثمانية) with 17,589 words and 3,094 lines. The smallest file belongs to the environment category and it discusses radioactive pollution (التلوث الإشعاعي) with 172 words and 32 lines.

Table 3 shows the 18 categories we have chosen to use for the categorization of the files in our dataset. It also shows the sub-categories, the number of files, the total number of words, the total number of lines, and the density percentage in each category.

Table 3 - Distribution of the documents in the Arabic dataset

#	Category	Sub-Category	Number of Files	Number of lines	Number of words	Density
1	History	History	**39**	32,976	4,991	4,991
2	Culture	Culture, Social, Cloths, Language, Buildings, palace, Festival, Flags, Ships	**22**	15,615	4,172	4,172
3	Countries	Country, City	**58**	73,757	7,366	7,366
4	Aviation	Airplane, Airport, Air Machine	**5**	1,954	2,450	2,450
5	Health & Medicine	Health, Medicine, Medical	**51**	16,605	1,950	1,950
6	Animals	Animal, Dinosaur , Zoology	**29**	26,606	5,459	5,459
7	War	Battles, War Machines	**21**	8,460	2,459	2,459
8	Technology	Technology, Software Engineering	**12**	8,631	4,237	4,237
9	Sciences	Chemistry, Electricity, Energy, physics, Law	**11**	6,136	3,165	3,165
10	Economy	Company, Economy	**10**	4,310	2,556	2,556
11	Environment	Environmental Issues, Pollution	**12**	4,904	2,353	2,353
12	Space	Space	**20**	10,621	3,258	3,258
13	Entertainment	Fiction, Movie, Music	**12**	7,418	3,766	3,766
14	Food	Fruit	**3**	624	1,233	1,233
15	Geography	Geography, Mountain	**8**	2,696	1,989	1,989
16	People	People	**59**	45,400	4,698	4,698
17	Religion	Religion	**16**	16,400	5,960	5,960
18	Sports	Sports	**12**	5,078	2,577	2,577
		Total Number of Files	**400**	288,191	1708168	

Figure 2 shows how the dataset folders and files are organized. Each category is represented by a folder. Each folder contains the text files along with their keyphrases. In the root folder, there is a spreadsheet titled "statistics" which contains statistical information about each category and a file reporting number of lines, words, and characters. Additionally, the dataset is also available as one folder with all files where keyphrases are listed with no categories. This dataset is provided in the enclosed CD-ROM (See Appendix C Content of Enclosed CD-ROM).

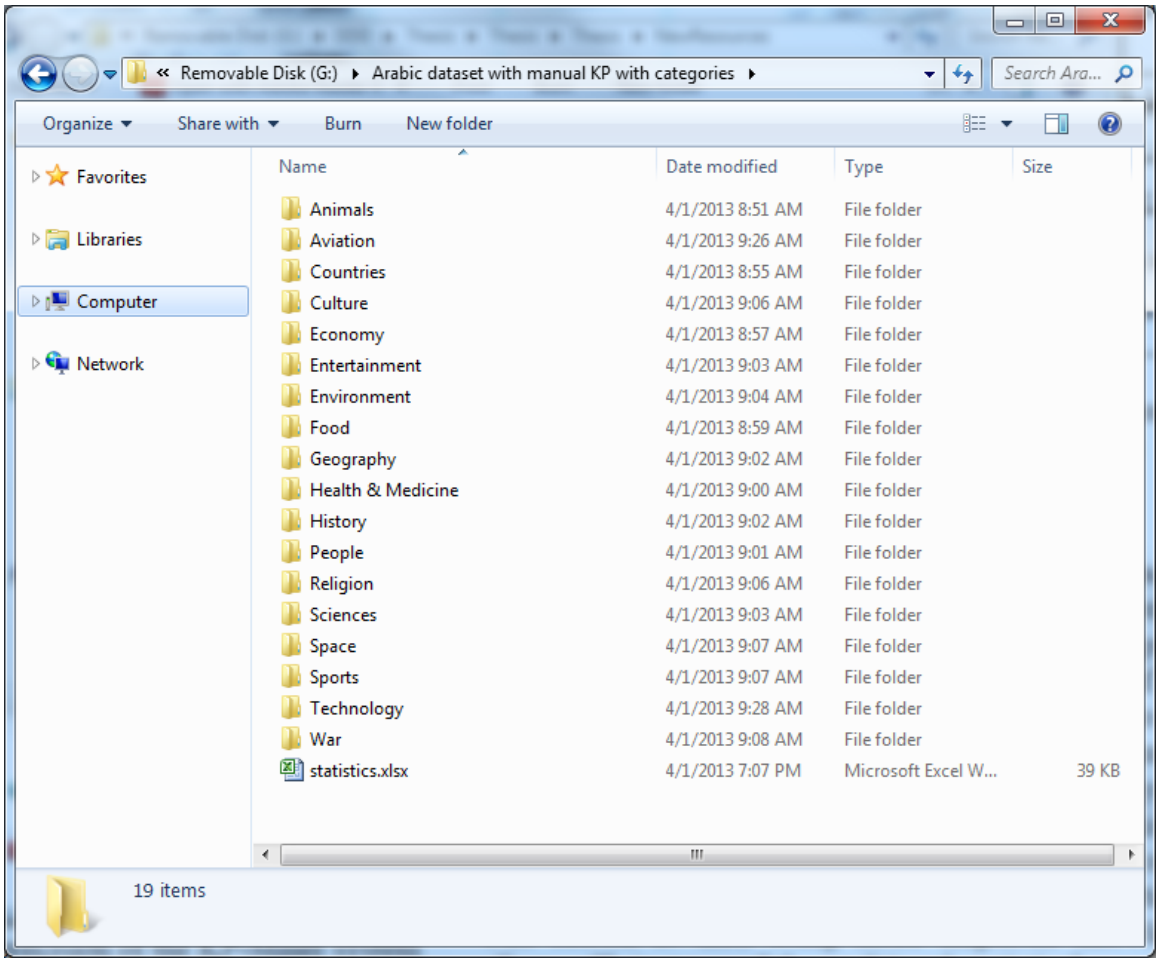


Figure 2- Organization of the Arabic Dataset

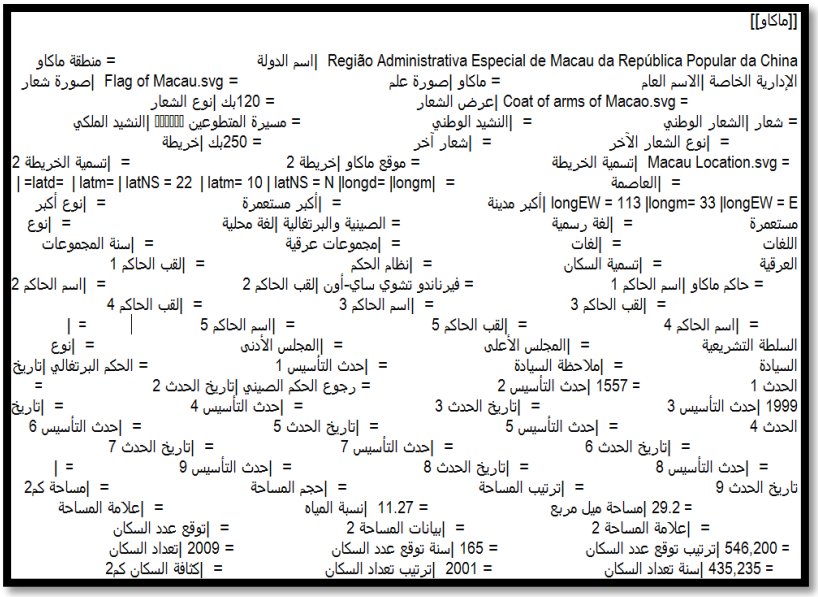
3.3.3 Cleaning up

When converting from Wikipedia pages to text format using BzReader, some clean up for the format was needed. The cleanup process included removing some text that may confuse the readers or make the articles hard to read. Text that was generated due to the conversion from HTML/ Rich text format was eliminated. This includes place holders of graphics, sounds, and videos. To illustrate this point by an example, if the article contains several images, then the word “png” or “jpg” will be repeated several times in the text version of the article. Hence, this will increase the chance of

selecting “png” or ”jpg” as a keyword. This is because many of Keyphrase Extraction algorithms (including the ones explained in this work) use term frequency as a factor when selecting candidates for keyphrases. The cleanup process included Wikipedia tables, side images captions, some references ...etc. Figure 3 and Figure 4 show an example of a table from Wikipedia page for Macau and how it looks after extracting the text via BzReader. Figure 5 shows the gallery section (بانوراما المدينة) of Quebec City (مدينة كيبك) available as city 04.txt in our dataset.

اللغة الرسمية	الصينية والبرتغالية
حكومة	
حاكم ماكاو	فيرناندو تشوي ساي-أون
التأسيس	
- الحكم البرتغالي	1557
- رجوع الحكم الصيني	1999
المساحة	
	29.2 كم²
	11.27 ميل مربع
السكان	
- تقدير 2009	546,200[1] (165)
- إحصاء 2001	435,235
- الكثافة السكانية	18,705/كم² (الأول)
	48,447/ميل مربع
الناتج المحلي الإجمالي	2006
(تعادل القدرة الشرائية)	
- الإجمالي	$17,600 (99)
الناتج المحلي الإجمالي (اسمي)	2007
- الإجمالي	$15,997 (94)
- للفرد	$36,357[2]
مؤشر التنمية البشرية (2004)	0.909[3] (مرتفع) (28)
العملة	باتاكا ماكاوية (MOP)
المنطقة الزمنية	(ت ع م+8)
جهة القيادة	يسار
رمز الإنترنت	.mo

Figure 3 - Wikipedia table for Macau



```
[[ماكاو]]

= منطقة ماكاو |اسم الدولة Região Administrativa Especial de Macau da República Popular da China
الإدارية الخاصة |الاسم العام = ماكاو |صورة علم = Flag of Macau.svg |صورة شعار
= Coat of arms of Macao.svg |عرض الشعار = 120بك |نوع الشعار
= شعار |الشعار الوطني = |النشيد الوطني = مسيرة المتطوعين |النشيد الملكي
= |نوع الشعار الآخر = |شعار آخر = 250بك |خريطة
= Macau Location.svg |تسمية الخريطة = موقع ماكاو |خريطة 2 = |تسمية الخريطة 2
= |العاصمة = |latd= | latm= | latNS = 22 | latm= 10 | latNS = N |longd= |longm
= أكبر مدينة |longEW = 113 |longm= 33 |longEW = E |نوع أكبر = أكبر مستعمرة |نوع
= اللغة رسمية = الصينية والبرتغالية |لغة محلية = |لغات مستعمرة
= |تسمية السكان = |مجموعات عرقية = |إسنة المجموعات اللغات العرقية
= |نظام الحكم = |القب الحاكم 1 = حاكم ماكاو |اسم الحاكم 1
= فيرناندو تشوي ساي-أون |القب الحاكم 2 = |اسم الحاكم 2
= |القب الحاكم 3 = |اسم الحاكم 3 = |القب الحاكم 4 = |اسم الحاكم 4
= |القب الحاكم 5 = |اسم الحاكم 5 = | السلطة التشريعية
= |المجلس الأعلى = |المجلس الأدنى = |نوع السيادة
= |ملاحظة السيادة = |حدث التأسيس 1 = الحكم البرتغالي |تاريخ الحدث 1
= 1557 |حدث التأسيس 2 = رجوع الحكم الصيني |تاريخ الحدث 2 =
1999 |حدث التأسيس 3 = |تاريخ الحدث 3 = |حدث التأسيس 4 = |تاريخ الحدث 4
= |حدث التأسيس 5 = |تاريخ الحدث 5 = |حدث التأسيس 6
= |تاريخ الحدث 6 = |حدث التأسيس 7 = |تاريخ الحدث 7 = |حدث التأسيس 8
= |تاريخ الحدث 8 = |حدث التأسيس 9 = | تاريخ الحدث 9
= 29.2 |مساحة ميل مربع = |ترتيب المساحة = |مساحة كم2
= |علامة المساحة = 11.27 |نسبة المياه = |إبيانات المساحة 2 = |علامة المساحة 2
= 546,200 |ترتيب توقع عدد السكان = 165 |سنة توقع عدد السكان = |توقع عدد السكان
= 435,235 |سنة تعداد السكان = 2001 |ترتيب تعداد السكان = 2009 |تعداد السكان
= |كثافة السكان كم2
```

Figure 4- Macau table after extraction

```
== بانوراما المدينة ==

ملف:Centre-ville de Québec vu depuis Val-Bélair.jpg|الذي يقع في الشمال Val-Bélair المدينة واسعة: مركز المدينة مصور من حي
ملف:Quebecfml3.JPG|الغربي من المدينة. ملف:Gates to نهر سان لورون، ومن بعيد، جزيرة أورلينز، أخذت من المقر العام للحاكم بكيبك
Quebec City.jpg|زنقة شمبلن الصغير ملف|Quebec City 02.jpg:باب سانت جون، أحد الأبواب الثلاثة المؤدية للحي التاريخي عبر الأسوار
ملف:Promenade محطة القصر بكيبك ملف|GarePalaisQuebec.JPG:نظرة لحي شمبلن ملف|Quebec City 03.jpg:في كيبك القديمة
Samuel-De Champlain-A.JPG|QuébecSaintePrétonille.JPG:نزهة صمويل شمبلن ملف صورة لكيبك من سانت بتروني بجزيرة اورلينز
ملف:MarcheVieuxPortQuebec.jpg|السوق في الميناء القديم ملف|DaudelinGarePalais.jpg:أمام محطة القصر انها نافورة شارل دودلان
ملف:ExMachina.jpg|اكس ماشينا في الميناء ملف|Québec - 231.jpg:زقاق شمبلن الصغير ملف|Bassin Brown-Québec.JPG:حوض
براون ملف:Cap-Blanc, Cap Diamant.JPG|رأس الماس والرأس الأبيض

== المصادر ==

* [//www.criacc.qc.ca/climat/suivi/CumulPluieNeige/CumulNeige20072008_f.html كرياك]
```

Figure 5- Gallery section of Quebec City (city04.txt)

3.3.4 Manual Keyphrase Extraction

All documents were assigned to readers to read and extract 10 keyphrases from each file. These keyphrases are stored in separate files with '.key' extension. This format is the one used by KEA and some other Automatic Keyphrase Extraction tools. In the '.key' files, each row represents a Keyphrase. They are sorted based on their importance in the article from high importance to low importance. The '.key' file name is matching exactly the '.txt' file. This is done to help the algorithm to locate the files in the training phases and help in organizing the dataset.

One of the challenges we faced was spelling mistakes in the articles. This includes writing the word wrongly or using incorrect format. Many times the word is written in several wrong formats within the same article. Some examples of these spelling mistakes are:

- using "ي" instead of "ى" or "ئ"
- using "أ" instead of "إ" ,"ء" ,"ا", or "آ"
- using "ه" instead of "ة"

In addition, it was noticed that some spelling mistakes were based on the accent of the author. For example, some authors write the Arabic word denoting to Syria using "ة" to become like this "سورية" or using "ا" and be in this form "سوريا". Sophisticated search engines can take care of such cases. We corrected all observed mistakes.

3.3.5 Keyphrase Verification

The final step in the methodology of preparing the Arabic dataset is the verification step. It included proofreading the articles and adjusting or concurring with the extracted keyphrases. This step also included reviewing and correcting the spelling mistakes, the number of keyphrases, and the '.key' files format.

3.4 Summary

In this chapter, we discussed the datasets that we have used in this research work. We presented the English dataset which contained 244 documents from ACM digital library. We contributed to enrich this dataset by adding the missing manual keyphrases assigned by authors. Additionally, we explained one of our main contributions i.e., the preparation of an Arabic dataset. Our Arabic dataset contains 400 documents along with their keyphrases. The sources of the documents were the Arabic Wikipedia and King Abdullah Initiative for Arabic Content. We also explained the structure, classification, and organization of the files in this dataset. This chapter also discussed steps taken to clean up and verify the extracted keyphrases.

CHAPTER 4

AUGMENTED KEYPHRASE EXTRACTION ALGORITHM

4.1 Overview

We have discussed in chapter 3 the datasets that we have used in our experiments. In this chapter, we present the methodology that we have followed to identify some possible ways to outperform algorithm in keyphrases extraction to introduce an augmented keyphrase extraction algorithm. This includes the experiment setup, plans, and evaluated hypotheses. This chapter is structured as follows: in section 4.2 we list our hypothesis. Section 4.3 describes the setup of our experiments. Section 4.4 presents the performed comparative analysis for both Arabic and English datasets. The evaluation techniques are mentioned in section 4.5. Finally, the chapter concludes with a summary in section 4.6.

4.2 Hypotheses

As we have seen in Chapter 1 and Chapter 2, KEA is an algorithm for automatic keyphrase extraction. It is developed based on Naïve-Bayes machine learning approach. It can be used as a free indexing algorithm or using controlled vocabulary files.

We designed our experiments to test the following four hypotheses:

First, stemmers reduce the size of documents by simplifying the words to their original roots. In general, this would improve the quality of the retrieval process by reducing noise when representing the text in different forms.

"Using stemmers will increase the number of correct extracted keyphrases"

(H1)

Second, since KEA utilizes the simple form of term frequency represented by TF-IDF, other and more advanced formulas to calculate the frequency with different term weighting could enhance the results obtained by KEA. When testing for this hypothesis, we have used some of the known weighting formulas mentioned in section 2.4. We have also introduced the following new term weighting methods:

I. LOGG- BM25: we suggest a new way of term weighting based on combining the power of two formulas, LOGG and BM25. LOGG is going to be used as the local term factor; where the IDF part of BM25 is going to be used as a global term.
II. LOGA-IDFP is another contribution based on LOGA as the local weight formula, and IDFP as the global weight.
III. LOGA-BM25 is the final contribution aimed to utilize the power of LOGA as the local weight formula and IDFP as the global weight.

The rationale behind combining previous weighting methods is to combine the strength of the logarithmic formulas with the augmentation that we are getting from BM25 and IDFP formulas. We assume that the results obtained after combining these formulas would outperform the results of each formula separately.

"The results gained from KEA could be enhanced by modifying the TF-IDF measure to include better weighting term formulas" *(H2)*

Additionally, the second feature that is used by KEA is the "first occurrence". It is implemented with the reasoning that important phrases are usually mentioned in the beginning of the documents. Su Nam et al. [18] suggested this idea in their work. Analogically, we claim that the last occurrence of a candidate keyphrase could indicate that it is an important keyphrase. Some examples include sections like conclusion and discussions where they come at the end of scientific papers.

Using "Last Occurrence" as a feature in KEA could enhance the obtained keyphrases. *(H3)*

The fourth hypothesis is derived from the idea that *we usually tend to mention important keywords in the beginning and ending of our articles*. For example, a scientific paper has an abstract and an introduction at the beginning of the paper and a conclusion at the end of it. These sections tend to have the main ideas behind the work.

"Using the combined first and last occurrence as a feature would enhance the keyphrases extracted by KEA" *(H4)*

4.3 Comparative Analysis Setup

Before we have started to run our comparative analyses and modify KEA, we went through data preparation processes. These processes included selecting training and test sets from the datasets, collecting and deciding on the stop word list, and studying the possibility of using stemmers.

Concerning the English dataset, we used the same categorization mentioned in section 3.2. The English collection contains 144 documents as training sets and 100 articles as test sets. For the Arabic dataset, the 400 documents are divided into two sets: a training set and a test set. The training set contains 300 documents where the test set consists of 100 documents. The documents of the test set were selected from all categories to allow testing the algorithm on all topics. Naturally, we selected more files from the larger categories like *people* and *medical* (See Section 3.3). The sub-categories with one document (like *cloths* and *zoology*) were also selected to be part of the test set for the similar reasoning.

4.3.1 Stop Words

For Arabic stop words, we have studied and examined the work done in [73] and [74]. We decided to choose the stop words list generated in [73] due to the large coverage and the quality of selected words. We also did some experimental tests to support this decision. For the English dataset, we elected to use the list provided by KEA. It is comprehensive and sufficient for English.

4.3.2 Stemmers

Stemmers aim to automatically map morphological variants of terms to their root or a single representative word known as a stem. For the Arabic dataset, the selected stemmer is a modified version of Khoja stemmer [75]. In this version, we fixed some memory issues we faced when stemming large files. We also enabled the stemmer to stem a directory of files and not one file at a time. For the English dataset, the selected stemmers are Porter [76] and Lovins [77] due to their popularity in English language.

4.3.3 Controlled Vocabulary

The existing version of KEA, which we plan to work on, supports the use of controlled vocabulary files. These files come in the format of SKOS which provides a mechanism to express the basic content and structure of concepts (e.g. definitions and taxonomies). KEA uses the vocabulary file to assign the keyphrases [28]. Since our Arabic and English datasets are generic in nature; covering several topics, we investigated the possibility of using a SKOS which covers different domains. However, there are no such SKOS files dedicated for Arabic language, up to our knowledge. In the English language, we found several explaining specific domains. Examples include the AGROVOC Agricultural Thesaurus [78] and the UNESCO nomenclature for fields of science and technology (UNESCO6) [79]. To illustrate the importance of having the vocabulary files relevant to the tested datasets, we conducted a preliminary evaluations using AGROVOC and UNESCO6 vocabulary files and compared the results against the default KEA behavior. This evaluation showed a decrease in the quality of returned keyphrases by 76.7% when we used AGROVOC and 83.3% with UNESCO6. Furthermore, we conducted the same evaluation with a third dataset. This dataset contains a subset of the FAO-780 dataset [80]. For this evaluation, we used a dataset contains 175 documents from the agriculture field. Since the dataset is relevant to the AGROVOC dataset, we noticed an increase of the quality of the results by approximately 148% over the default KEA settings without a vocabulary usage. In this evaluation, the results decreased by 73.3% when we used UNESCO6. More information about the two evaluations can be found in Appendix A.

From the previous evaluations, we found that vocabulary files are effective when they are relevant to the tested dataset. Since we do not have relevant files, we decided to ignore the configuration of controlled vocabulary in our work.

4.4 Methodology

The first step in our methodology is evaluating KEA quality of with and without stemmers for both Arabic and English datasets. For this step, we aim to test the first hypothesis (H1) using two comparative analyses. Following this step, we develop twelve enhancements to KEA: eight of these enhancements are designed using term weighting schemes explained in section 2.4 and 4.2. The remaining four enhancements are developed using a combination of the last occurrence and the combined first and last occurrence measures. These enhancements are planned to be evaluated using six more comparative analyses: three analyses on the English dataset and three on the Arabic dataset. The outperformed variation of the first comparative analysis is used in the next one. Then the top performed variation from the second comparative analysis is selected as a baseline to be compared with two other algorithms in the fourth comparative analysis. In each analysis, we extracted the top keyphrases and compared them with the manual keyphrases. For the case of English dataset, the keyphrases were provided by authors. For Arabic dataset, readers who helped in preparing the dataset provided the keyphrases. For the comparative analysis on English-dataset, we are selecting the top four candidate keyphrases from each file. On Arabic-dataset, we are selecting the top ten keyphrases. The reason behind this selection is the availability of the manual keyphrases. Authors usually pick four keyphrases or less for their papers. Although our algorithm can select the top ten keyphrases and then compare them with the four chosen by authors,

however, this gives the algorithm unfair chance to get the right keyphrases. For the Arabic dataset, we are going to pick the top ten keyphrases and compare them with our ten keyphrases selected for each file. The following sections describe our comparative analyses.

4.4.1 Comparative Analysis 1: Stemming Vs. No-Stemming

In this analysis, we are evaluating the usage of KEA with stemmers and without them to test hypothesis H1. The comparative analysis is divided into two analyses i.e. comparative analysis 1A for English dataset and comparative analysis 1B for Arabic dataset.

4.4.2 Comparative Analysis 2: Top Term Weighting Formula in KEA

In this analysis, we are evaluating different formulas to test hypothesis H2 (see Section 4.2). The comparative analysis is divided into two analyses i.e. comparative analysis 2A for English dataset and comparative analysis 2B for Arabic dataset. Below approach summarizes the steps taken in this comparative analysis:

I. Modify KEA to incorporate one of the five term weighting formulas mentioned in section 2.4 and our suggested formulas: LOGA-BM25, LOGA-IDFP, and LOGG-BM25. For the runs, no stemmers and no vocabularies are used.
II. Run KEA on the training documents to create the model.
III. Run KEA on the 100 test documents and identify the top four keyphrases and the top ten keyphrases on each document for English and Arabic, respectively.
IV. Conduct an evaluation study to find the outperformed implementation.

4.4.3 Comparative Analysis 3: Last Occurrence and the Combined First and Last Occurrence

In this comparative analysis, we are evaluating hypotheses H3 and H4 (see Section 4.2). We have implemented last occurrence and combined first and last occurrence as a feature instead of the default first occurrence. The enhancement of KEA was done using the default TF-IDF and the top performed KEA alteration resulted from comparative analyses 2A and 2B. Like comparative analysis 2, this analysis is divided into two analyses: comparative analysis 3A for English dataset and comparative analysis 3B for Arabic dataset. The methodology to perform this comparative analysis is similar to the one explained in 4.4.2.

4.4.4 Comparative Analysis 4: Enhanced KEA vs. Other Algorithms

This evaluation analysis includes comparing the outperformed enhancement from comparative analysis 3 (baseline) with existing algorithms. The targeted algorithms are Keyphrase Extractor developed by Kumar [27] and KP-Miner which was developed by El-Beltagy and Rafea [21]. The reason for selecting these two algorithms is that they are relatively new and they are publically available to experiment with. Keyphrase Extractor is available as a desktop application where you put two inputs i.e., the text file and the number of keyphrases required. The text file can be just an abstract or the full length. For our empirical study, we select the full length. KP-Miner is available as a web-based tool at [81]. The input for this tool includes the language, number of desired keyphrases, and the targeted text file.

As in comparative analyses 1, 2 and 3, comparative analysis 4 is divided into two analyses for our English (4A) and Arabic (4B) datasets.

4.5 Evaluation Techniques

We measure the performance for the developed variations of KEA using the exact matching scheme [31]. In the exact matching scheme, the retrieved keyphrase is considered correct if and only if it is fully matching one keyphrase from the manual keyphrases list. Additionally, the average precision, recall, and F-Score (F-Measure) are calculated to find the outperformed implementation. Formulas 7, 8, and 9 explain the calculation of the mentioned measures [82]

$$precision = \frac{Correct\ keyphrases\ retreived\ by\ KEA}{All\ keyphrases\ retreived\ by\ KEA} \qquad \text{Formula 7}$$

$$Recall = \frac{Correct\ keyphrases\ retreived\ by\ KEA}{Manual\ keyphrases} \qquad \text{Formula 8}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad \text{Formula 9}$$

F-measure (F-score) is a harmonic average also known as the weighted average, which takes in consideration the values of the precision and recall. The closer the value of F-score is to one the better, correspondingly, the closer its value is to zero the worst.

The evaluation process will be performed over the top four candidate keyphrases for the English dataset and for the top ten candidate keyphrases for the Arabic dataset. In order to conduct the evaluations, we developed a new class and added it to KEA library. This class takes as an input two folders: the folder contains '.key' files hosting manual keyphrases and the folder hosting the automatically generated keyphrases. The output

will be the calculation of the total of correct matched keyphrases as well as average precision, recall, and f-measure. The results are analyzed using Microsoft Excel.

The literature indicates the averages of precision and recall in automatic keyphrase extraction are not high due to the difficulty of the keyphrase extraction problem. Leininger [83] clarified that the consistency in indexing among expert indexers is very hard to obtain. In free indexing, the indexing consistency was found to be ranged from 4% to 67% with an average of 27.05%.

4.6 Summary

In this chapter, we presented the methodology we have followed while experimenting with KEA. We discussed our hypotheses that we have implemented and tested. We also described the environment setup which contains the used datasets and KEA configuration. Additionally, we presented the details of our eight comparative analyses planned for our KEA enhancements on the English and Arabic datasets: stemming vs. no-stemming, the top weighting formula in KEA, last occurrence and the combined first-last occurrence, and the evaluation against state-of-the art keyphrase extraction algorithms. Finally, we highlighted the evaluation techniques we use to analyze the results. These techniques are exact matching, precision, recall, and F-score. In the next chapter, we present the use of these techniques to analyze and evaluate the experimental results.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSIONS

5.1 Overview

As we have seen from the previous chapter (Chapter 4), we have augmented KEA algorithm with 12 enhancements. These enhancements have been designed, developed, implemented, and tested. In this chapter we present the results of the experiments using these enhancements. We have performed the comparative analysis on both English and Arabic Datasets. The empirical study included comparative analysis 1A and 1B aim to investigate the usage of KEA with and without stemmers. In addition, this empirical study involved the investigation after modifying TF-IDF measure to use different term weighting schemes. The top variations with regard to exact matching values are used in our next analyses: 3A and 3B. In these analyses, we implemented last occurrence and the combined first and last occurrence using the top performed formulas resulted from the first empirical study. Then we compared the results to select the top KEA variations that we used in the third comparative analysis as our baseline to compare with KP-Miner and Keyphrase Extractor.

This chapter is organized as follows: in section 5.2, we discuss the experimental results. Section 5.3 discusses the overall evaluations. Finally, we conclude this chapter with our finding summary in section 5.4.

5.2 Experimental Results

In each of the eight comparative analyses, the tested algorithm generated 100 '.key' files that contain the extracted keyphrases. These files are analyzed and compared against the manual keyphrases for both Arabic and English datasets.

5.2.1 Stemming Vs. No-Stemming Results for English dataset (1A)

In this comparative analysis, we tested KEA using the Porter and Lovins stemmers and without them on the English dataset. We tested our KEA three times using the mentioned settings. Each test generated 100 keyphrase files. Then we compared the manual keyphrases we have and the generated keyphrases in each file. First, we identify the exact matches in the two files. Then we calculate the recall, precision, and F-score using the exact match, number of extracted keyphrases (4 in this experiment), and the number of the manual keyphrases in that file. To illustrate how these measures are calculated, we take one of the files (i.e. C-22) as an example and show the extracted keyphrases with stemming and with no stemming. Figure 6 shows the file hierarchy of the ".key" files for each KEA configuration as well as the three versions of "C22.key" resulted from the three experiments.

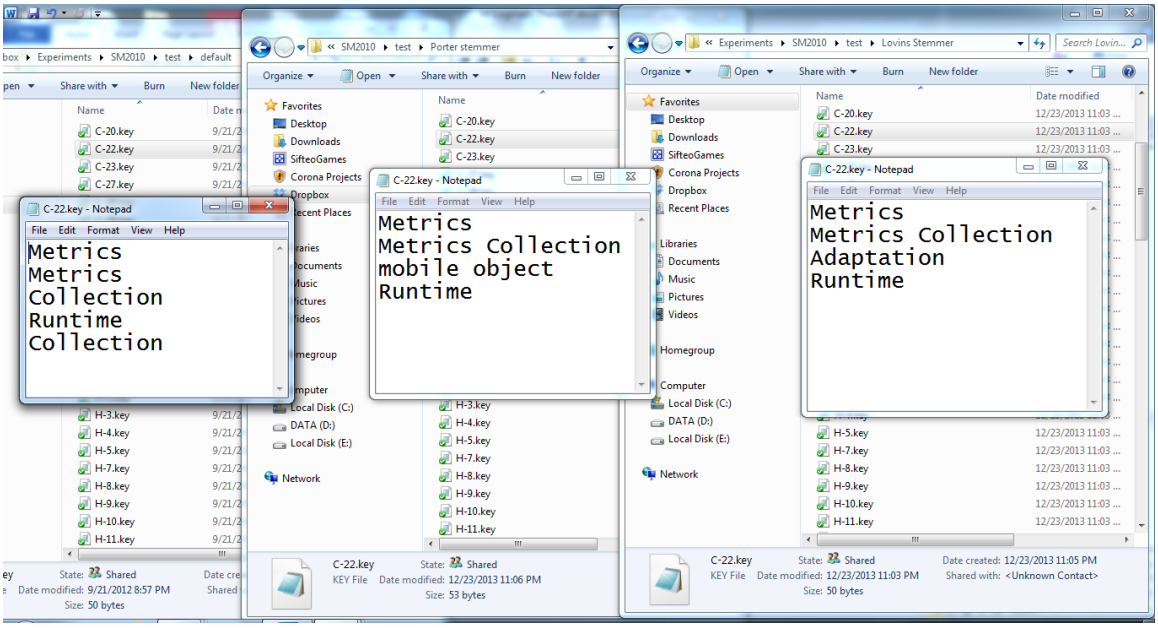


Figure 6- A screenshot of the file heriarciy of three results

Table 4 - Manual & auto-generated keyphrases in file C-22 in stemming experiments

Manual	No stemmer	Porter	Lovins
MobJeX adaptation metrics collection mobile objects	Metrics Metrics Collection Runtime Collection	Metrics Metrics Collection mobile object Runtime	Metrics Metrics Collection Adaptation Runtime

In table 4, we see that when KEA ran with no stemmer, the results have one exact match i.e. “Metrics Collection”. Hence, the precision is $\frac{1}{4}$ where 4 is the number of keyphrases retrieved by KEA. The recall was $\frac{1}{4}$ where 4 is the total of the manual keyphrases. F-

score was calculated and found to be 0.25. These values are close to the average of human indexing which is 27.05% as mentioned in section 4.5. Moving to Porter stemmer, we can see that the exact match was also one keyphrase, which is "Metrics Collection". The values of other measures were the same as with no stemmer, hence both gave the same results for this file. For Lovins Stemmer, two exact matchers were returned and they are "Metrics Collection" and "Adaptation". Now we can see the precision and recall increased to be 0.5 (meaning 50%) and this indicates that for this particular file, Lovins stemmer was the outperformed variation of KEA. The values of our measures on each run for the file C-22 can be found in table 5.

Table 5- Measures calculation on file C-22 in the stemming experiment on English

Measure	Default	Porter Stemmer	Lovins Stemmer
Exact Match	1	1	2
Precision	0.25	0.25	0.5
Recall	0.25	0.25	0.5
F-Score	0.25	0.25	0.5

We repeated this process to all 100 test files for each configuration. As a result, Table 6 - Stemming vs. no-stemming in English dataset (1A) summarizes the values for the total exact matched keyphrases, average precision, average recall, and the average F-score on all 100-test files. For the precision, recall, and F-score, as stated in the previous example, we used the numbers from the exact matching scheme. To calculate the precision average, we add all obtained precisions from all files and divide the sum by the total

number of documents (100 files). The same process is done for the recall averages. The F-score average is calculated using the recall average and precision average.

We observed that in the overall results the default KEA with no stemmer and Porter stemmer scored the highest results in the Exact Matching Scheme with 60 matched keyphrases. However, Lovins stemmer could not score the same results. This shows why Porter stemmer has more popularity than Lovins and why it is the default stemmer in KEA. For the precision measure where we focus on the number of correct keyphrases over the incorrect, we saw that KEA with no stemmer and Porter scored the same result, which is 0.15 (15%). This result is better than Lovins stemmer but 15% less than the average for consistency in manual keyphrase extraction. We believe that one of the reasons of having this average is our extraction targeted four keyphrases only. If the evaluation was conducted to extract larger number e.g., the top ten keyphrases, the results of all measures is going to be enhanced. This will be seen later when testing on Arabic where we extract ten keyphrases. Similar analogy was reported in the literature. Examples can be seen in Kumar et. al. [27] and Kim et. al. [18].

Table 6 - Stemming vs. no-stemming in English dataset (1A)

Measure	No-Stemmer	Porter Stemmer	Lovins Stemmer
Exact Match	**60**	**60**	57
Precision	0.150	0.150	0.143
Recall	0.156	0.150	0.134
F-Score	0.153	0.150	0.138

5.2.2 Stemming Vs. No-Stemming Results for Arabic dataset (1B)

In this comparative analysis, we tested KEA with the modified Khoja stemmer and without it. Khoja was not integrated as part of KEA; therefore, we performed some preparation to complete this evaluation. First, we performed the training of KEA against the training dataset in their normal format (i.e. with no stemming). Then we extracted the keyphrases and did the comparison with manual test dataset. For the stemming part, we run Khoja against the manual training and test datasets to convert the text into stemmed form. Then we used the stemmed dataset as training for KEA. Finally, we compared the extracted keyphrases against the manual test dataset in its stemmed format. As in previous evaluation, each run generated 100-keyphrase files. In this evaluation, KEA was configured to extract the top ten keyphrases. To illustrate the work done in this evaluation and how we calculated our measures, we use one of the files (i.e. airport 01. The topic of the file is "Cairo International Airport") and show the extracted keyphrases with stemming and with no stemming. Figure 7 shows the file hierarchy of the ".key" files for the two KEA configurations (stemmed and not stemmed) as well as the two manual datasets (stemmed and not stemmed). Additionally, figure 7 shows the content of the file "airport 01.key" in the stemmed and not stemmed settings.

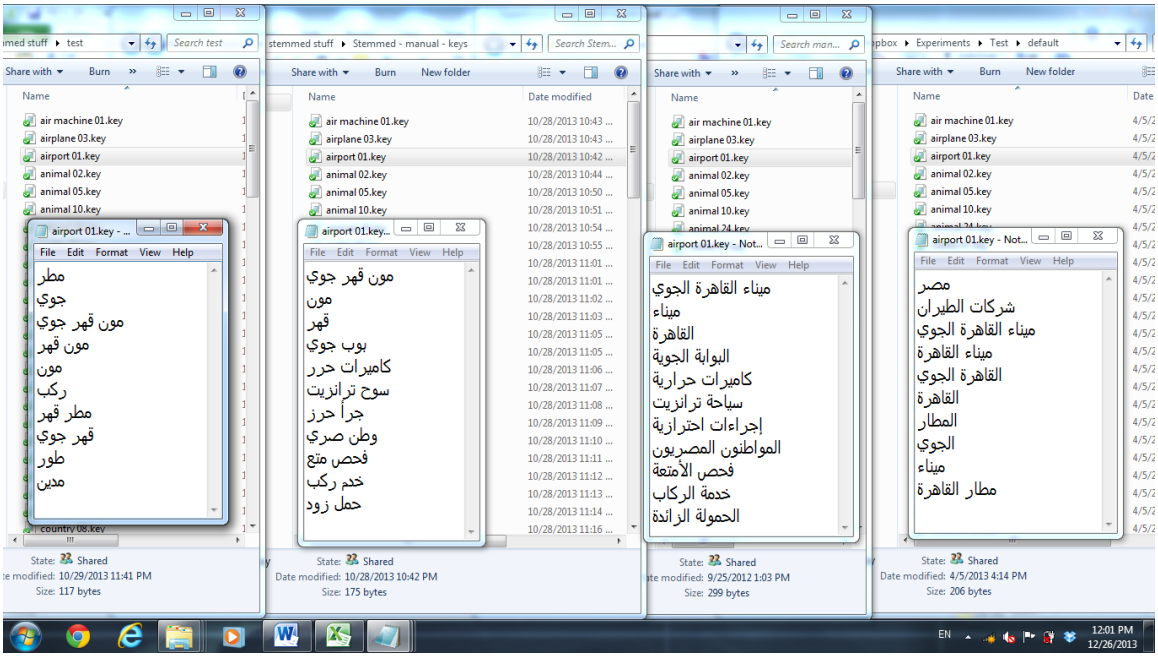


Figure 7 - A screenshot of the file hierarchy of the results for 1B

In Table 7, we show the values of our measures where we see the default KEA usage with no stemmer returned three correct keyphrases out of the retrieved ten keyphrases. The correct keyphrases are: "ميناء القاهرة الجوي", "القاهرة", and "ميناء". This gives us a precision of $\frac{3}{10}$ = 0.3 (30%), where 3 is the correct keyphrases and 10 is the number of keyphrases retrieved by KEA. The recall was $\frac{3}{11}$ = 0.27 (27%), where 3 is the correct keyphrases and 11 is the total manual keyphrases. For the stemmer evaluation on this file, KEA managed to retrieve two correct keyphrases and they are "مون قهر جوي" and "مون". This meant the precision would be 0.2 (20%) with recall of 0.182 (18.2%). Hence, in this particular file the performance of KEA with no stemmer outperformed KEA with stemmer.

The same process performed on this file is repeated for all files in the Arabic dataset. We calculate the average precision by adding the precision value for each file and divide the

total by 100 which are the total test files. The average recall is calculated the same way by considering recall values. To calculate the average F-measure, we apply formula 9 using average precision and recall.

Table 7 - Measures calculation on file "Airport 01" in Arabic stemming experiment

Measure	Default	Khoja Stemmer
Exact Match	3	2
Precision	0.3	0.2
Recall	0.27	0.182
F-Score	0.29	0.19

In Table 8, we list the results of the complete evaluation that contains the values of the total of exact matching measure, average precision, recall, and F-measure. The overall number of exact matching scheme shows that KEA with Khoja stemmer could not reach to the same level as KEA with no-stemmer. Using stemmer, KEA scored 150 exact-matches. With no stemmer, it scored 189. We believe the reason is due to the effectiveness of the stemmer and the difficulty of stemming Arabic language. Table 8 also shows that with no stemmer the average precision was 0.189 with recall of 0.191. With stemmer, KEA could not achieve these results and it scored 0.152 precision with a recall value of 0.155. We noticed that the values of precision and recall were very close to each other. This is because we are retrieving ten keyphrases and the majority of our manual documents contained ten keyphrases. So when calculating the precision and recall, we are dividing the number of correct keyphrases over ten. The harmonic mean of

both precision and recall (F-measure) showed that with no-stemmer, KEA reached 0.19 which still less than the average human indexer but it was better than the results obtained from the evaluation conducted on the English language. We believe this is because we are retrieving more keyphrases in this evaluation.

Table 8 - Stemming vs. no-stemming in Arabic dataset (1B)

Measure	No-Stemmer	Khoja Stemmer
Exact Match	189	150
Precision	0.189	0.152
Recall	0.191	0.155
F-Score	0.190	0.153

5.2.3 Top Term Weighting Formula Results for English Dataset (2A)

In this comparative analysis, we created eight KEA enhancements (variations) using different term weighting schemes. Table 9 presents the values for the total exact matched keyphrases, average precision, average recall, and the average F-score. For the precision, recall, and F-score, we used the numbers from the exact matching scheme. Figure 8 and Figure 9 provide a graphical representation of the results. We observe that default TF-IDF scored the top results in the Exact Matching Scheme with 60 matched keyphrases and followed by BM25 with 59. IDFP came last with 33 matched keyphrases. When observing other measures for both TF-IDF and BM25, we see that KEA with TF-IDF achieved 0.15 (15%) precision and 0.156 (15.6%) where BM25 scored 0.148 (14.8%) precision and 0.159 (15.9%) recall. With higher recall value, this means BM25 worked better than TF-IDF with files with less number of manual keyphrases. Both variations scored 0.153 (15.3%) in f-measure which indicates the average of both variations was the same.

Table 9 - Results on English Dataset for comparative analysis 2A

Measure	Default	BM25	IDFP	LOGA	LOGG	SQRT	BM25+LOGG	LOGA+IDFP	LOGA+BM25
Exact Match	**60**	59	33	53	47	34	56	34	50
Precision	0.150	0.148	0.083	0.133	0.118	0.085	0.140	0.085	0.125
Recall	0.156	0.159	0.091	0.139	0.132	0.092	0.155	0.092	0.131
F-Score	0.153	0.153	0.086	0.136	0.124	0.088	0.147	0.088	0.128

As we have moved further with our analysis, we found that for the articles related to distributed systems (category C) BM25 and TF-IDF ranked first with 18 exactly matched keyphrases. In the second field which is information search and retrieval (category H) LOGG scored the best with 21 exact matches and it was followed by LOGG-BM25 with 20 exact matches. In the Distributed Artificial Intelligence and Multi-agent Systems group (category I) BM25 was the best with 11 exact matched keyphrases. We have noticed that in this category, the retrieval from all variations was worse than the rest of categories. Finally, for the social and behavioral sciences – economics area (category J) the default TF-IDF ranked first with 17 exact matched keyphrases. LOGA variation did not perform well in all categories except for this category and scored 15 matches. We observed that the documents in this category were relatively longer than other categories.

As for the newly suggested term-weighting formulas, LOGG-BM25 came third after TF-IDF and BM25 with 56 exact matches and average F-measure of 14.7%. On the other hand, the results of KEA with LOGA-BM25 and LOGA-IDFP were not promising. LOGA-IDFP ranked just before last with 34 exact matches and 8.8% F-measure. The assumption was when combining the power of two types of term weighting schemes they would outperform the other known formulas.

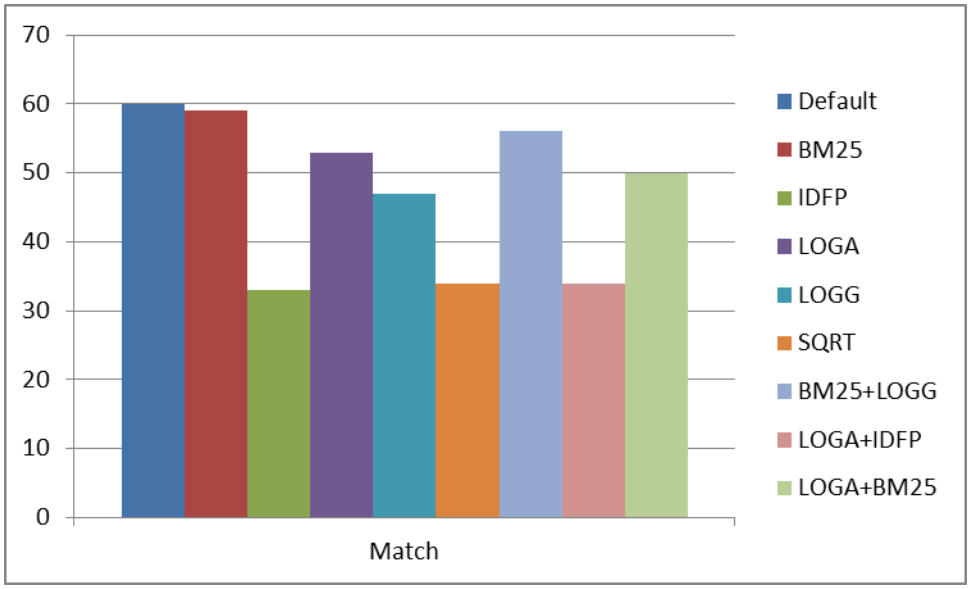


Figure 8 - Exact Matching Results for Top Term Weighting Formula on English (2A)

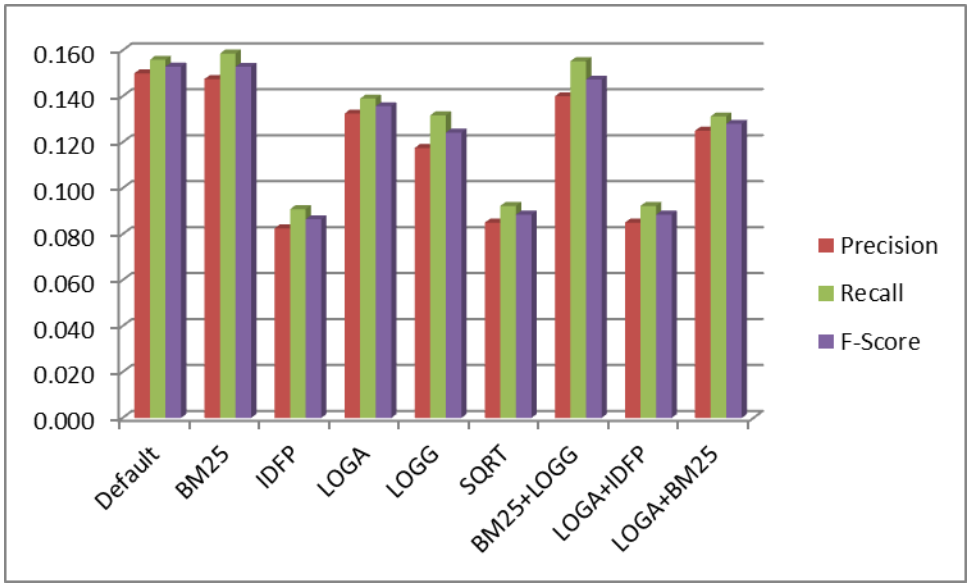


Figure 9 – Precision, Recall, and F-Score Results for comparative analysis 2A

5.2.4 Top Term Weighting Formula Results for Arabic Dataset (2B)

This comparative analysis is similar to the one pursued in the 2A comparative analysis, except the dataset used was the Arabic dataset in this experiment. Here, we dropped the LOGA-IDFP measure for the poor results obtained from comparative analysis 2A. Table

10 shows the results obtained for the total exact matched keyphrases, average precision, average recall, and the average F-score. In this evaluation, BM25 scored the top results in the Exact Matching Scheme with 191 matched keyphrases. It was followed by the suggested weighting scheme LOGA-BM25 with 190 matched keyphrases. IDFP and SQRT scored the least in the exact matching measure with only 154. Since we are using the exact matched results when calculating the precision, recall, and F-score, consequently the values of these measures are directly proportional to the exactly matched measure. In this analysis, BM25 scored 0.191 as precision, 0.193 as recall, and 0.192 as F-measure. These values are better than the ones obtained in the English test and this is due to the fact that we extract more keyphrases. SQRT and IDFP results were last with 0.154 as a precision, 0.157 as recall, and 0.155 as F-measure. Since there is consistency with human indexers, these results considered acceptable.

In this evaluation, the suggested term weighting scheme LOGA-BM25 extracted more correct keyphrases than the ones retrieved by the default TF-IDF. Additionally, the second suggestion i.e., BM25-LOGG scored average and better than its performance in the top term weighting comparative analysis conducted on English dataset.

Table 10- Summary of results obtained for the Arabic dataset in 2B

Measure	Default	BM25	IDFP	LOGA	LOGG	SQRT	BM25+LOGG	LOGA+BM25
Exact Match	189	**191**	154	187	173	154	177	**190**
Precision	0.189	0.191	0.154	0.187	0.173	0.154	0.177	0.190
Recall	0.191	0.193	0.157	0.189	0.176	0.157	0.180	0.193
F-Score	0.190	0.192	0.155	0.188	0.174	0.155	0.178	0.191

Figure 10 and Figure 11 provide a graphical representation of the exact matching results as well as other measures evaluated in this comparative analysis.

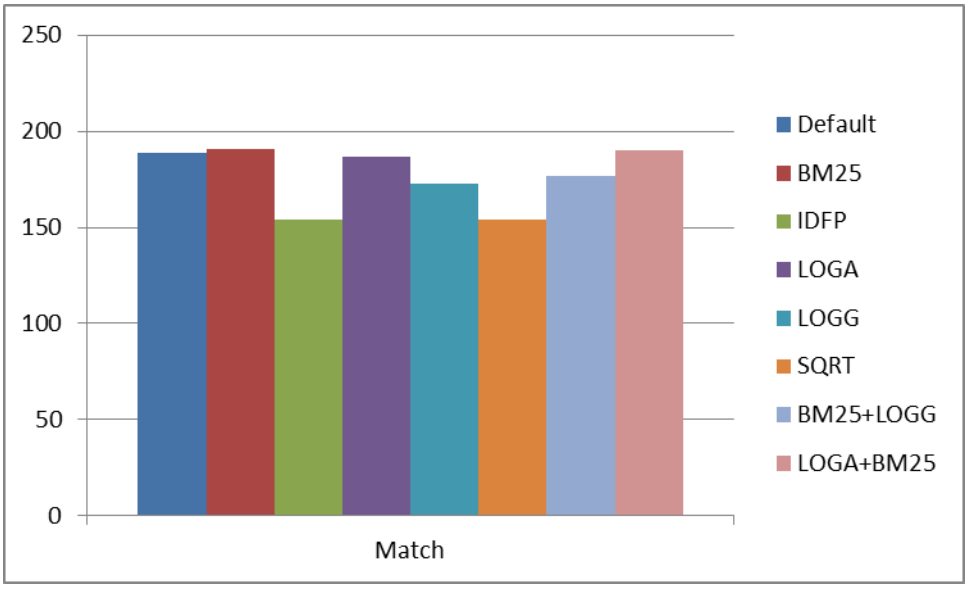


Figure 10- Exact Match Results for Top Term Weighting Formula on Arabic (2B)

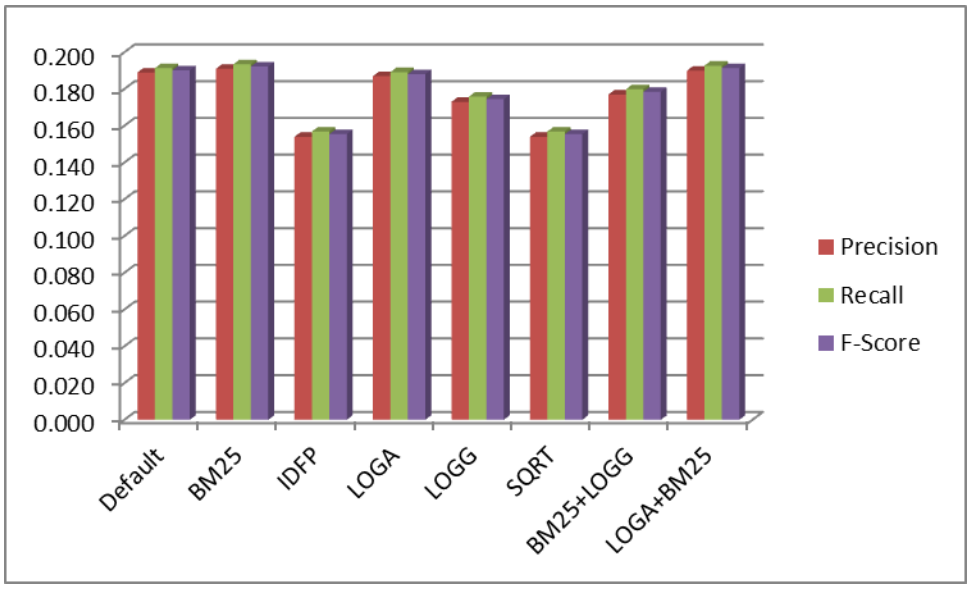


Figure 11- Precision, Recall, and F-Score Results for comparative analysis 2B

5.2.5 Last Occurrence and Combined First and Last Occurrence Results for English Dataset (3A)

In this comparative analysis, we implemented the last occurrence and the combined first and last occurrences. For the frequency feature, we selected the top weighting measures in (2A) evaluation i.e., BM25 and the default TF-IDF. Table 11 displays the results obtained from this comparative analysis. The variations are:

- the default KEA behavior including TF-IDF and first occurrence
- the last occurrence with TF-IDF and BM25
- the combined first and last occurrences with TF-IDF and BM25.

For the exact match, the default KEA settings came first with 60 exactly matched keyphrases. The two variations of the combined first and last occurrence came second with 58 exact matches for the TF-IDF and 57 for the BM25. The last occurrence variations came last with 48 exact matches for the TF-IDF and 47 for the BM25. The results for other measures followed this sequence. The average mean for the default feature of KEA ranked top with 15.3% and it was followed by the combined first and last occurrence with BM25 which scored 14.6%.

Table 11 - Summary of comparative analysis 3A on the English dataset

Measure	Default (TF-IDF and first occurrence)	First-Last Occurrence with TF-IDF	First-Last with BM25	Last occurrence with TF-IDF	Last with BM25
Exact Match	60	58	57	48	47
Precision	0.150	0.145	0.143	0.120	0.118
Recall	0.156	0.143	0.149	0.117	0.112
F-Score	0.153	0.144	0.146	0.118	0.115

Looking at the results obtained for each category of the English dataset, we see that the default KEA behavior that uses TF-IDF with focus on first occurrence came top in three categories. The categories are distributed systems (category C) with 18 exactly matched keyphrases, information search and retrieval (category H) with 15 keyphrases, and the Distributed Artificial Intelligence and Multi-agent Systems group (category I) with 10 exact matched keyphrases. In category I, the variation of the combined first and last occurrence with BM25 scored also 10 exact matches. Again, we noticed that in category I, the retrieval from all variations was worse than the rest of categories. Finally, for the social and behavioral sciences – economics area (category J) the combined first and last occurrence with TF-IDF ranked first with 18 exact matched keyphrases.

We observed that giving more weights to the last occurrence lowered the scores. Scientific papers usually tend to have important words in the abstract and end with references and appendices, which may contribute with some noise.

Figure 12 and Figure 13 demonstrate the exact matched, average precision, average recall, and average F-score results. The default KEA outperformed the other alterations.

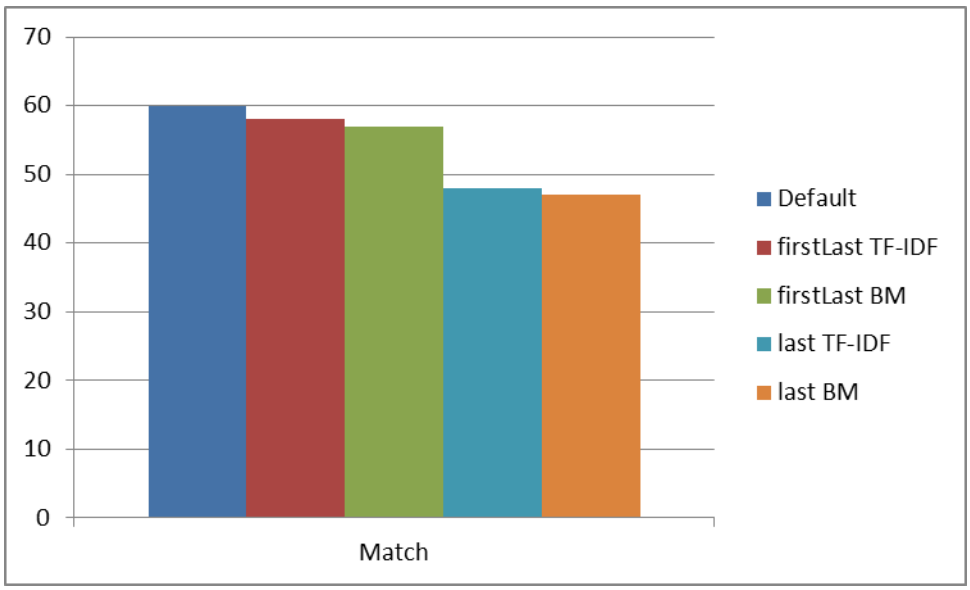


Figure 12- Exact match results of the occurrence evaluation on English 3A

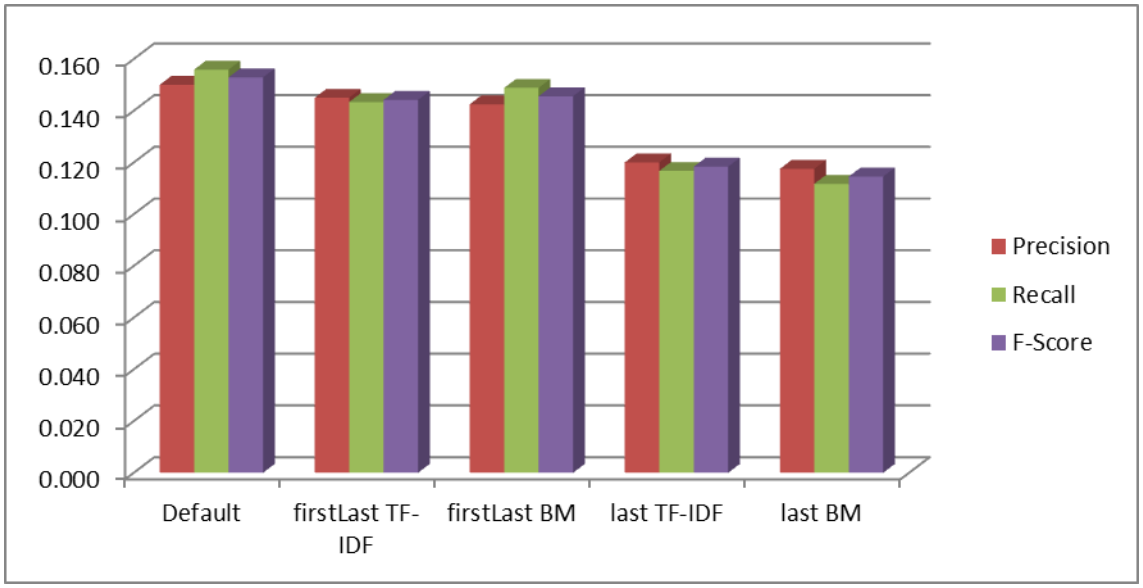


Figure 13- Precision, Recall, and F-Score Results for occurrence in evaluation 3A

5.2.6 Last Occurrence and Combined First and Last Occurrence Results for Arabic Dataset (3B)

In this comparative analysis, we repeated the previous analysis but on Arabic dataset. The tested five variations included the default settings of KEA with TF-IDF and first

occurrence, the last occurrence with TF-IDF and BM25, and the combined first and last occurrences with TF-IDF and BM25. We present the results in Table 12 and the graphical representation for the exact match as well as other measures in Figure 14 and Figure 15. In these experiments, the default settings of KEA with TF-IDF and first occurrence ranked top with 189 exact matched. Our variation consisted of the combined first and last occurrence with BM25 scored the second topping other variation with 185 exact matches and 18.6% f-measure. The last occurrence variations ranked the least with 158 matches for the BM25 and 153 for the TF-IDF variation.

Table 12- Summary of the results of comparative analysis 3B on the Arabic dataset

Measure	Default (TF-IDF and first occurrence)	First-Last Occurrence with TF-IDF	First-Last with BM25	Last occurrence with TF-IDF	Last with BM25
Exact Match	189	185	187	153	158
Precision	0.189	0.185	0.187	0.153	0.158
Recall	0.191	0.188	0.190	0.156	0.161
F-Score	0.192	0.186	0.188	0.154	0.159

When we analyzed the Wikipedia articles that were a major source for our Arabic dataset, we found that they follow the same pattern which starts with a definition and a summary about the topic, and then the details will follow with no conclusions. Hence, we saw the first occurrence, which focuses on and give more weight to the candidate keyphrases mentioned in the beginning of the article, ranked first. We believe that our suggestion for the combined first and last occurrences was effected by the noise generated by giving some weight for the words mentioned at the end of the article. For the same reason, we saw that last occurrence ranked last.

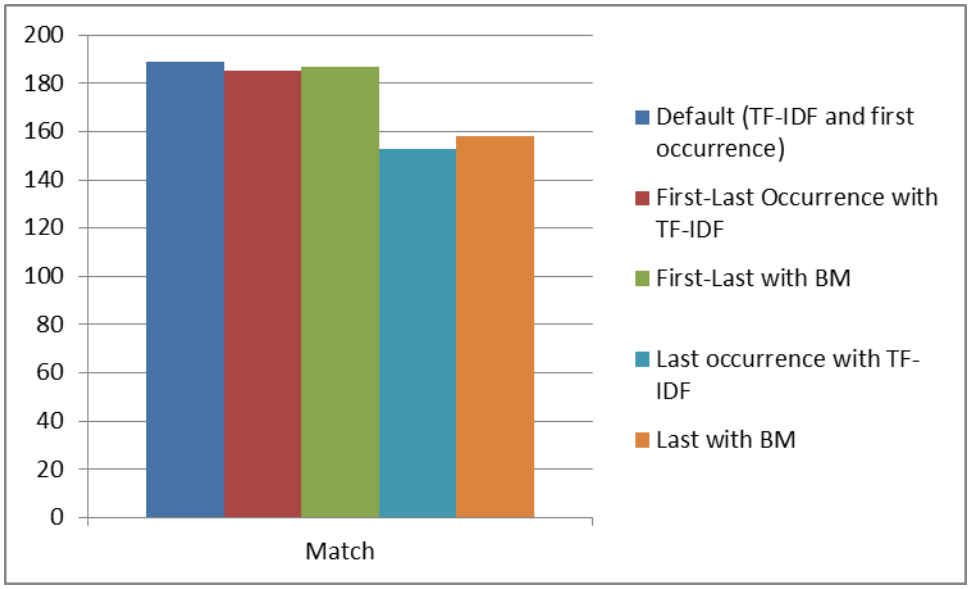


Figure 14- Exact match results of occurrence evaluation on Arabic dataset (3B)

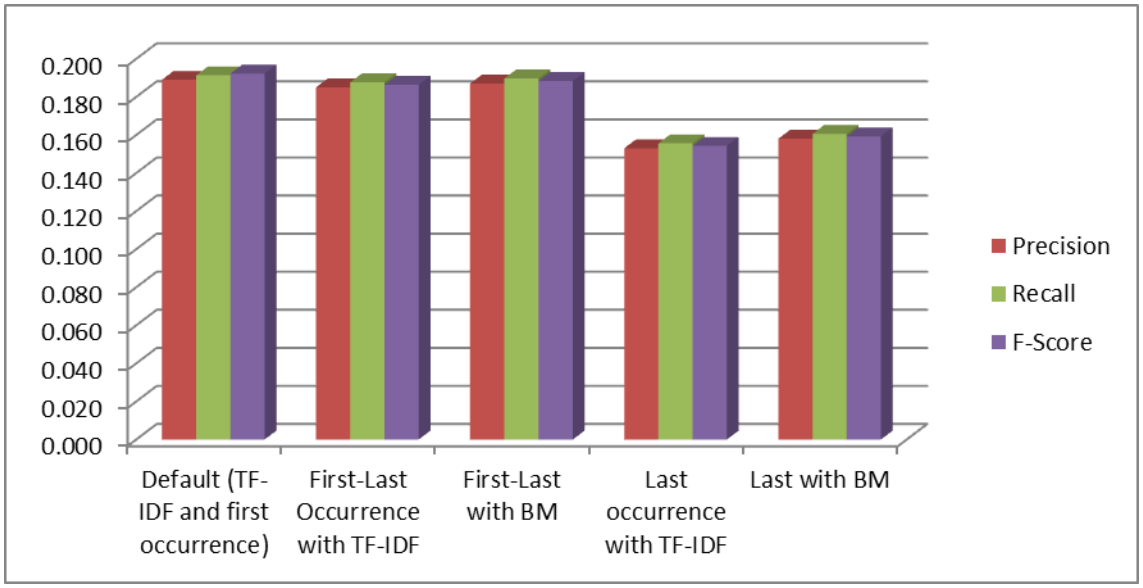


Figure 15- Precision, Recall, and F-Score Results for occurrence comparative (3B)

5.2.7 Enhanced KEA vs. Other Algorithms Results for English Dataset (4A)

In this experiment on the English dataset, we selected the two top performed variations from the 3A comparative analysis i.e., the default KEA configuration with TF-IDF and our first-last occurrence with TF-IDF weighting measure. Our baselines were compared against the results obtained from KP-Miner and Kumar's Keyphrase Extractor. Table 13 shows the obtained results from this comparative analysis. In addition, the results for exact matching and other measures are illustrated in Figure 16 and Figure 17.

Our two KEA variations scored top in the exact matching scheme with 60 to the default and 58 to the first-last occurrence. KP-Miner came third with 52 matches where Keyphrase Extractor ranked last with poor results of 4 matched keyphrases and 1% precision which is lower than the minimum value of the consistency by human indexers. The original KEA as well as our alteration performed well with an average of precision of 15% and 14.5%.

Table 13- Results of comparative analysis 4A

Measure	Default	First Last TF-IDF	KP-Miner	Keyphrase Extractor
Exact Match	60	58	52	4
Precision	0.150	0.145	0.130	0.010
Recall	0.156	0.143	0.136	0.009
F-Score	0.153	0.144	0.133	0.009

Furthermore, we found that the default KEA ranked first on three of the four categories i.e.: distributed systems (category C) with 18 exact matches, information search and retrieval (category H) with 15 matches, and the Distributed Artificial Intelligence and Multi-agent Systems group (category I) with 10 keyphrases. For the fourth category i.e.,

and the social and behavioral sciences – economics area (category J), KEA with first-last occurrence scored top with 18 exact matched keyphrases.

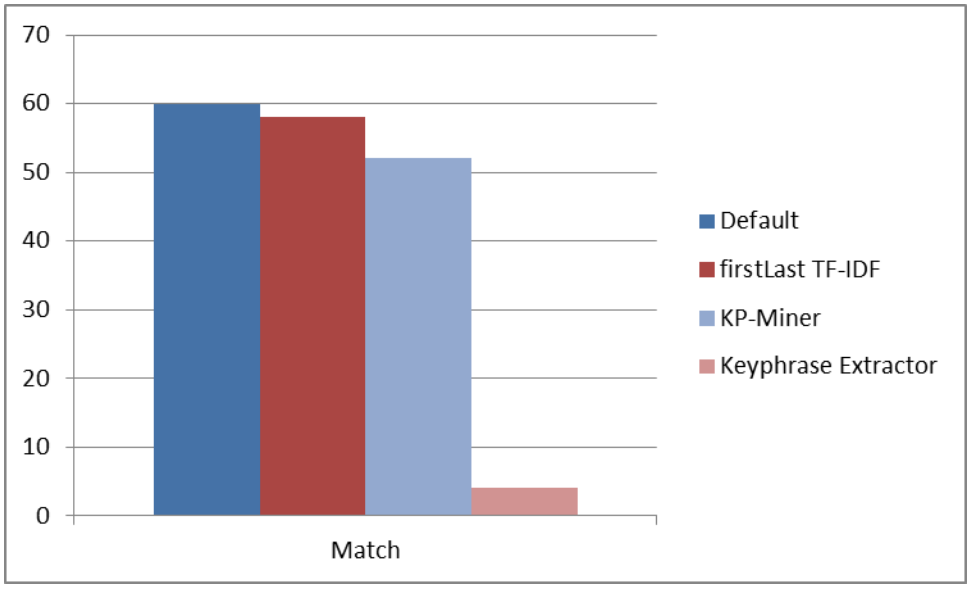


Figure 16- Exact match representation of KEA vs. other algorithms on English 4A

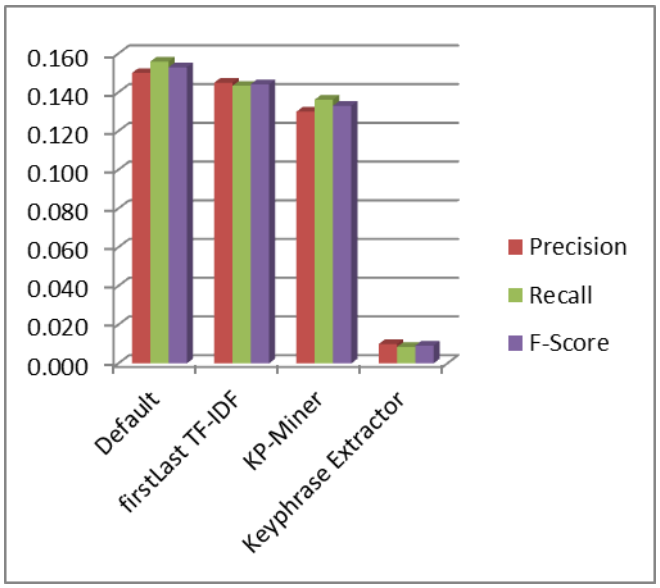


Figure 17- Precision, Recall, and F-Score Results for KEA vs. others on English 4A

5.2.8 Enhanced KEA vs. Other Algorithms Results for Arabic Dataset (4B)

In this analysis on the Arabic dataset, we compared two of our enhancements (variations) against KP-Miner and Kumar's Keyphrase Extractor. Our KEA variations include the BM25 configuration and our suggested first-last occurrences with TF-IDF term weighting measure. Table 14 presents the results obtained from these experiments. Figure 18 and Figure 19 demonstrate the results for exact matching scheme and the average precision, recall, and F-measure.

In this evaluation, our alterations for KEA scored top with 191 exact matches for BM25 and 187 for the combined first-last occurrence. The harmonic mean for our top alteration was 19.2% and it was the highest in our testing. Even though, it is less than the average for professional indexers, but they are considered acceptable when we look to the value that we are gaining with automatic keyphrase extraction. KP-Miner was relatively close to KEA with 167 exact matched keyphrases and an F-score of 16.8%. Keyphrase Extractor ranked last with 95 keyphrases and 9.6% f-score.

Table 14- Expermentail results (4B) on Arabic dataset

Measure	BM25	First Last Default	KP-Miner	Key Phrase Extractor
Exact Match	191	187	167	95
Precision	0.191	0.187	0.167	0.096
Recall	0.193	0.190	0.169	0.097
F-Score	0.192	0.188	0.168	0.096

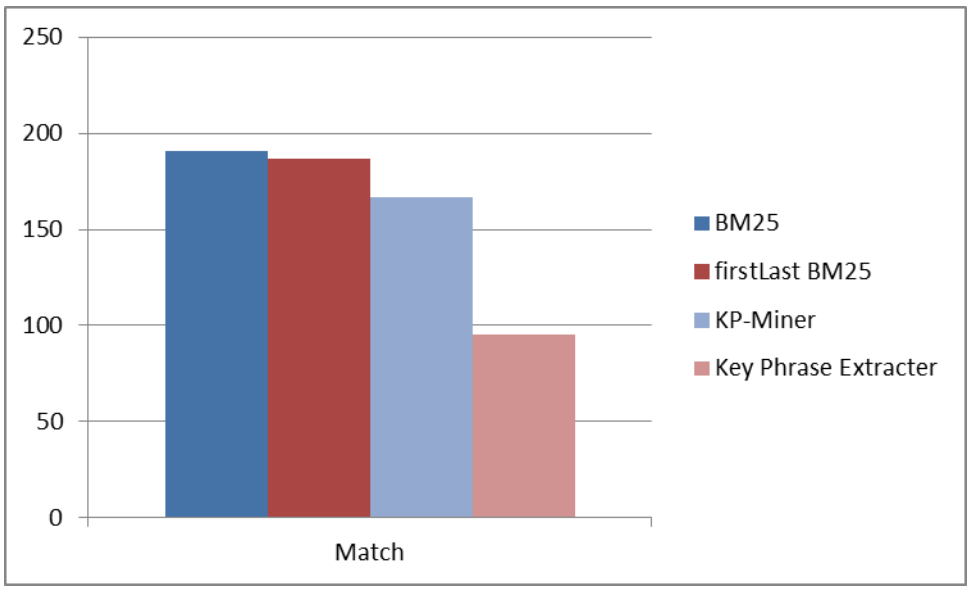


Figure 18- Exact matching results for comparative analysis 4B

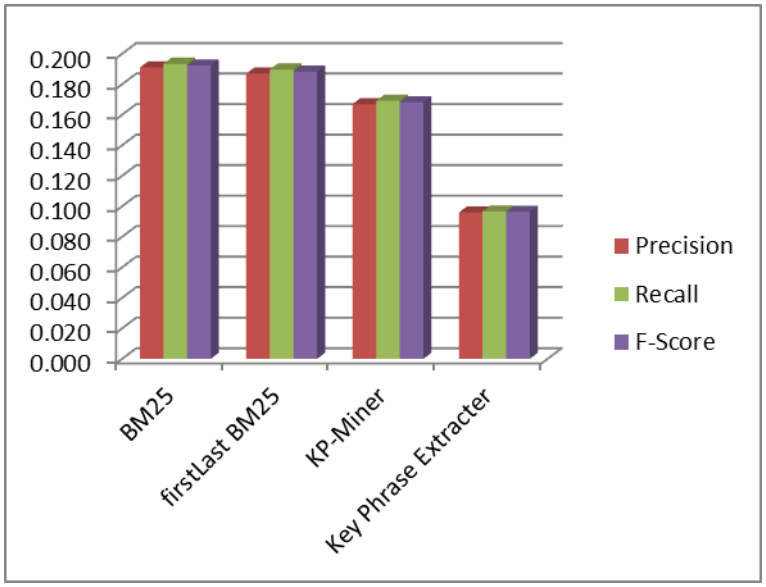


Figure 19- Precision, Recall, and F-Score Results for KEA vs. others on Arabic (4B)

5.3 Discussions

After analyzing the results from these experiments, we observed that KEA does not depend heavily on term weighting to extract keyphrases. Even though, there was an enhancement observed after using BM25, this enhancement was not significant enough to claim that one variation clearly outperformed the default TF-IDF. For the third comparative analysis, which studied the impact of term weighting and occurrence of the candidate phrases, no major enhancements were noticed. Hence, our hypotheses regarding the weight of the occurrence can be disregarded.

In the final comparative analysis where we compared KEA results against two keyphrase extraction algorithms, we saw that our variations outperformed both KP-Miner as well as Kumar's Keyphrase Extractor in English and Arabic. Our variations outperformed Kumar's Extractor because our variations depended on KEA. Such results are partially due to KEA being a machine learning algorithm uses training documents to build the extraction models. On the other hand, Keyphrase Extractor uses only statistical information to extract the keyphrases. These results are in agreement with what we stated in section 2.2.2.

We noticed also that the results did not change dramatically when testing for different languages. Hence, our KEA variations can be used on other languages, given that the code of the algorithm supports Unicode and the datasets are available in Unicode. Overall, the results obtained from Arabic evaluations were better than their English counterparts. We believe this due to the fact we are extracting more keyphrases that led to enhance the precision, recall, and F-measure. Hence, we recommend in the future to test

the keyphrase extraction algorithm to extract higher number than four which we used in the English tests.

Since different readers interpret texts differently, we cannot claim that all the keyphrases that failed to exactly match the manual keyphrases are wrong. This is why the average precision, recall, and F-measure did not exceed 20%.

When reviewing the obtained results, we found several keyphrases that are very close to the manual keyphrase but do not match it completely. We propose to use the partial matching in the future to give more insights about the obtained results.

Additionally, we measured the performance of KEA after our modifications. For the English dataset, the performance results showed that the system was able to create the training model and extract 400 keyphrases from 100 documents in approximately 45 seconds. For the Arabic dataset, the system was trained on 300 documents and was able to extract 1000 keyphrases from 100 test documents in around 2 minutes. This was expected as the dataset is larger and each Arabic character is represented UTF8 format with 16 bit long while each English character is represented by 8 bits.

5.4 Summary

In this chapter, we discussed the results obtained from our comparative analysis. We noticed that when testing on English, Porter stemmer performed well. However, Khoja stemmer did not perform the same in Arabic. We saw the exact matching with Khoja was lowered. We saw in the second group of evaluations conducted to select the top term weighting scheme that our suggestions for using BM25 for the TF-IDF enhanced

insignificantly the results of exact matched measure. Also, the suggested BM25-LOGG scored well in these empirical evaluations. For the last occurrence and first-last occurrence comparative analyses, the default configuration topped other variations. This is due to the noise generated by the references and appendices sections and the nature of our datasets where important keyphrases are on the beginning of the documents. In the final set of experiments where we compared our KEA variations with KP-Miner and Kumar's Extractor algorithms, our variations topped both of these algorithms on the English and Arabic datasets.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this research work, we conducted a survey of the current automatic keyphrase extraction approaches. We also prepared and presented an Arabic dataset for automatic keyphrase extraction. We have extracted the dataset from Arabic Wikipedia and King Abdullah Initiative for Arabic Content. We have extracted manually the keyphrases of the documents of the dataset. The dataset contains 400 documents along with their correspondence 400 keyphrases files. The dataset will be publicly available for future automatic keyphrase extraction research on Arabic language.

As part of this thesis, we analyzed the effects of using stemmers and not using them when extracting keyphrases. In addition, we studied some measures used in selecting keyphrases and proposed new ones. We proposed utilizing BM25 as a term weighting measure instead of TF-IDF and the combination of first and last occurrences instead of the first occurrence measure. Then we implemented these measures as part of KEA and conducted an empirical study to evaluate top enhancement for KEA. The outperformed variations, like KEA with BM25, were tested against existing two algorithms found in the research i.e. Keyphrase Extractor and KP- Miner.

Our results showed miner enhancements for our variations over the original KEA but these enhancements were insignificant. Finally, our variations outperformed two automatic keyphrase extraction algorithms i.e., Keyphrase Extractor and KP-Miner in both Arabic and English.

6.2 Future Work

We suggest working on below ideas as future implementation:

- Proposing the Arabic dataset to the community for more enrichment and future utilization.
- Incorporating more stemmers and stop-words to evaluate and measure the performance.
- Increasing the manual keyphrases in the English dataset making it possible to run the experiment on the top ten candidates instead of the top four candidates.
- Investigating the incorporation of partial matching as an additional measure to evaluate keyphrase extraction algorithms.
- Investigating or developing comprehensive vocabulary files in SKOS format that can cover general topics in English and in Arabic.
- Presenting the outperformed algorithm, which is KEA with BM25 as weighting measures, to Saudi Aramco Management to be adopted and incorporated in the current Content Management Systems to handle Arabic and English content.

References

[1] X. Wan and J. Xiao, "CollabRank : Towards a Collaborative Approach to Single-Document Keyphrase Extraction," no. August, pp. 969–976, 2008.

[2] O. Medelyan and I. Witten, "Domain • independent automatic keyphrase indexing with small training sets," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 7, pp. 1026–1040, 2008.

[3] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, and B. Wang, "Automatic Keyword Extraction from Documents Using Conditional Random Fields," vol. 3, 2008.

[4] R. Rodrigues and K. Asnani, "Concept Based Search Using LSI and Automatic Keyphrase Extraction," vol. I, no. 1, pp. 43–53, 2011.

[5] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank, "Improving browsing in digital libraries with keyphrase indexes," *Decis. Support Syst.*, vol. 27, no. 1–2, pp. 81–104, Nov. 1999.

[6] J. Goldstein and V. Mittal, "Multi-document summarization by sentence extraction," *Proc. 2000 NAACL-ANLPWorkshop Autom. Summ.*, vol. 4, pp. 40–48, 2000.

[7] A. M. Azmi and S. Al-Thanyyan, "A text summarizer for Arabic," *Comput. Speech Lang.*, vol. 26, no. 4, pp. 260–273, Aug. 2012.

[8] T. El-Shishtawy and F. El-Ghannam, "Keyphrase based Arabic summarizer (KPAS)," *Informatics Syst. (INFOS), 2012 8th Int. Conf. on. IEEE*, pp. 7–14, 2012.

[9] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," *Proc. ACL Work. Intell. scalable text Summ.*, vol. 17, no. 1, pp. 10–17, 1997.

[10] D. Lawrie, W. B. Croft, and A. Rosenberg, "Finding topic words for hierarchical summarization," *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '01*, pp. 349–357, 2001.

[11] E. D'Avanzo and B. Magnini, "A Keyphrase-based Approach to Summarization: the LAKE System at DUC-2005," *Proc. DUC*, 2005.

[12] Y. Zhang, N. Zincir-heywood, and E. Milios, “Term-Based Clustering and Summarization of Web Page Collections,” pp. 60–74, 2004.

[13] K. Hammouda, D. Matute, and M. Kamel, “Corephrase: Keyphrase extraction for document clustering,” *Mach. Learn. Data Min. Pattern Recognit.*, pp. 265–274, 2005.

[14] P. Tonella and F. Ricca, “Using keyword extraction for web site clustering,” *Web Site Evol. 2003. Theme Archit. Proceedings. Fifth IEEE Int. Work.*, pp. 41–48, 2003.

[15] S. Lee and H.-J. Kim, “News Keyword Extraction for Topic Tracking,” *2008 Fourth Int. Conf. Networked Comput. Adv. Inf. Manag.*, pp. 554–559, Sep. 2008.

[16] D. Zhu and A. L. Porter, “Automated extraction and visualization of information for technological intelligence and forecasting,” *Technol. Forecast. Soc. Change*, vol. 69, no. 5, pp. 495–506, Jun. 2002.

[17] T. Ong and H. Chen, “Updateable PAT-Tree approach to Chinese key phrase extraction using mutual information: A linguistic foundation for knowledge management,” in *The second Asian Digital Library Conference*, 1999, pp. 63–84.

[18] S. Kim and M. Kan, “Re-examining automatic keyphrase extraction approaches in scientific articles,” *Proc. Work. Multiword Expressions Identification, Interpret. Disambiguation Appl.*, pp. 9–16, 2009.

[19] S. Khan, I. Fatima, R. Irfan, and K. Latif, “A Refined Methodology for Automatic Keyphrase Assignment to Digital Documents.,” *JDIM*, vol. 9, no. 2, pp. 55–63, 2011.

[20] T. Nguyen and M. Kan, “Keyphrase extraction in scientific publications,” *Asian Digit. Libr. Look. Back 10 Years Forg. New Front.*, pp. 317–326, 2007.

[21] S. R. El-Beltagy and A. Rafea, “KP-Miner: A keyphrase extraction system for English and Arabic documents,” *Inf. Syst.*, vol. 34, no. 1, pp. 132–144, Mar. 2009.

[22] T. El-Shishtawy and A. Al-Sammak, “Arabic Keyphrase Extraction using Linguistic knowledge and Machine Learning Techniques,” *arXiv Prepr. arXiv1203.4605*, pp. 1–8, 2012.

[23] P. Lopez and L. Romary, “HUMB : Automatic Key Term Extraction from Scientific Articles in GROBID,” no. July, pp. 248–251, 2010.

[24] T. Nguyen and M. Luong, “WINGNUS: Keyphrase extraction utilizing document logical structure,” *Proc. 5th Int. Work. Semant. Eval.*, pp. 166–169, 2010.

[25] W. You, D. Fontaine, and J.-P. Barthes, “Automatic Keyphrase Extraction with a Refined Candidate Set,” *2009 IEEE/WIC/ACM Int. Jt. Conf. Web Intell. Intell. Agent Technol.*, pp. 576–579, 2009.

[26] K. Sarkar, M. Nasipuri, and S. Ghose, “A New Approach to Keyphrase Extraction Using Neural Networks,” vol. 7, no. 2, 2010.

[27] N. Kumar and K. Srinathan, “Automatic keyphrase extraction from scientific documents using N-gram filtration technique,” *Proceeding eighth ACM Symp. Doc. Eng. - DocEng ’08*, p. 199, 2008.

[28] O. Medelyan and I. H. Witten, “Thesaurus based automatic keyphrase indexing,” *Proc. 6th ACM/IEEE-CS Jt. Conf. Digit. Libr. - JCDL ’06*, p. 296, 2006.

[29] A. Bellaachia and M. Al-dhelaan, “Learning from Twitter Hashtags : Leveraging Proximate Tags to Enhance Graph-based Keyphrase Extraction,” pp. 348–357, 2012.

[30] A. Balagopalan, L. L. Balasubramanian, V. Balasubramanian, N. Chandrasekharan, and A. Damodar, “Automatic keyphrase extraction and segmentation of video lectures,” *2012 IEEE Int. Conf. Technol. Enhanc. Educ.*, pp. 1–10, Jan. 2012.

[31] I. Witten, G. Paynter, and E. Frank, “KEA: Practical automatic keyphrase extraction,” *Proc. fourth ACM Conf. Digit. Libr.*, pp. 254–255, 1999.

[32] F. Liu, F. Liu, and Y. Liu, “A Supervised Framework for Keyword Extraction,” vol. 19, no. 3, pp. 538–548, 2011.

[33] J. Wang, L. Li, and F. Ren, “An improved method of keywords extraction based on short technology text,” *Nat. Lang. Process. Knowl. Eng.*, pp. 1–6, 2010.

[34] B. Hong and D. Zhen, “An Extended Keyword Extraction Method,” *Phys. Procedia*, vol. 24, pp. 1120–1127, Jan. 2012.

[35] L. Z. Liao and H. Y. Huang, “Microblog Keyphrase Extraction Based on Similarity Features,” *Proc. 2013 Int. Conf. Adv. Comput. Sci. Electron. Inf.*, pp. 284–287, 2013.

[36] M. Danilevsky, C. Wang, and N. Desai, “Kert: Automatic extraction and ranking of topical keyphrases from content-representative document titles,” *CoRR*, 2013.

[37] J. Ziv and A. Lempel, “Compression of individual sequences via variable-rate coding,” *Inf. Theory, IEEE Trans.*, no. 24.5, pp. 530–536, 1978.

[38] K. Frantzi and S. Ananiadou, “Extracting nested collocations,” *Proc. 16th Conf. Comput. Linguist.*, vol. 1, pp. 41–46, 1996.

[39] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[40] Q. DONG, Z., AND DONG, “HowNet,” 1988. [Online]. Available: http://www.keenage.com.

[41] S. Kim, O. Medelyan, M. Kan, and T. Baldwin, “Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles,” *Proc. 5th Int. Work. Semant. Eval.*, pp. 21–26, 2010.

[42] I. Fatima, A. Khattak, Y. Lee, and S. Lee, “Automatic Documents Annotation by Keyphrase Extraction in Digital Libraries Using Taxonomy,” *Futur. Inf. Technol.*, 2011.

[43] Z. Li, D. Zhou, Y.-F. Juan, and J. Han, “Keyword extraction for social snippets,” *Proc. 19th Int. Conf. World wide web - WWW ’10*, p. 1143, 2010.

[44] A. Mishra and G. Singh, “Improving keyphrase extraction by using document topic information,” in *2011 IEEE International Conference on Granular Computing*, 2011, pp. 463–467.

[45] N. Pudota, A. Dattolo, A. Baruzzo, and C. Tasso, “A New Domain Independent Keyphrase Extraction System,” *Digit. Libr.*, 2010.

[46] A. Miles and S. Bechhofer, “SKOS simple knowledge organization system reference,” 2009.

[47] E. Frank, “Description of KEA.” [Online]. Available: http://www.nzdl.org/Kea/description.html. [Accessed: 10-Oct-2012].

[48] R. Irfan, S. Khan, I. A. Khan, and M. A. Ali, “KeaKAT: An Online Automatic Keyphrase Assignment Tool,” *2012 10th Int. Conf. Front. Inf. Technol.*, pp. 30–34, Dec. 2012.

[49] Alyona Medelyan, “Maui Indexer.” [Online]. Available: http://maui-indexer.appspot.com/. [Accessed: 05-Apr-2013].

[50] and I. Indian Institute of Technology Kanpur, Food and Agriculture Organization, “Agrotagger.” [Online]. Available: http://agropedialabs.iitk.ac.in/Tagger/. [Accessed: 04-Apr-2013].

[51] Sakhr, “Sakhr Keywords Extractor.” [Online]. Available: http://aramedia.com/keywordextraction.htm. [Accessed: 01-Jul-2012].

[52] K. Sudo, S. Sekine, R. Grishman, and N. York, “Cross-lingual Information Extraction System Evaluation,” *Proc. 20th Int. Conf. Comput. Linguist.*, 2004.

[53] M. Paukkeri, I. T. Nieminen, and P. Matti, “A Language-Independent Approach to Keyphrase Extraction and Evaluation,” *Proc. COLING*, pp. 83–86, 2008.

[54] DBI Technologies, “Extractor.” [Online]. Available: http://www.dbi-tech.com/productPage_Extractor.aspx. [Accessed: 01-Aug-2012].

[55] Basis Technology, “BasisTech.” [Online]. Available: http://www.basistech.com/products/. [Accessed: 01-Aug-2012].

[56] Smart Logic, “Classification and Text Mining Server.” [Online]. Available: http://www.smartlogic.com/home/products/semaphore-modules/classification-and-text-mining-server. [Accessed: 02-Aug-2012].

[57] Yahoo, “Yahoo Search API, Term Extraction.” [Online]. Available: http://developer.yahoo.com/search/content/V1/termExtraction.html. [Accessed: 02-Aug-2012].

[58] Wordsfinder LTD, “WordsFinder Keyword Extractor API.” [Online]. Available: http://www.wordsfinder.com/api_key.php. [Accessed: 02-Aug-2012].

[59] Alchemy Inc, “Alchemy API.” [Online]. Available: http://www.alchemyapi.com/api/keyword/. [Accessed: 02-Aug-2012].

[60] S. A. Mann, William C. and Thompson, “Rhetorical Structure Theory: Toward a Functional Theory of text Organization,” vol. 3, pp. 234–281, 1988.

[61] M. W. B. M. Browne, *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.

[62] S. Robertson and K. Jones, “Relevance weighting of search terms,” *J. Am. Soc. Inf. Sci.*, vol. 27, no. 3, pp. 129–146, 1976.

[63] G. Mathias and C. Largeron, “BM25t : a BM25 extension for focused information retrieval,” pp. 1–25, 2011.

[64] M. Karkali, “Keeping Keywords Fresh : A BM25 Variation for Personalized Keyword Extraction Categories and Subject Descriptors,” in *Proceedings of the 2nd Temporal Web Analytics Workshop*, 2012, pp. 17–24.

[65] W. B. CROFT and D. J. HARPER, “Using Probabilistic Models of Document Retrieval Without Relevance Information,” *J. Doc.*, vol. 35, no. 4, pp. 285–295, Dec. 1979.

[66] E. Chisholm and T. Kolda, “New term weighting formulas for the vector space method in information retrieval,” *Comput. Sci. Math. Div. Oak Ridge Natl. Lab.*, 1999.

[67] S. N. Kim, “Dataset for Automatic Keyphrase Extraction from Scientific Articles.” [Online]. Available: https://github.com/snkim/AutomaticKeyphraseExtraction. [Accessed: 08-Aug-2012].

[68] Wikipedia, “Arabic Wikipedia.” [Online]. Available: http://ar.wikipedia.org/wiki. [Accessed: 01-Jul-2012].

[69] King Abdullah Initiative for Arabic Content, “King Abdullah Initiative for Arabic Content.” [Online]. Available: http://www.econtent.org.sa. [Accessed: 07-Oct-2012].

[70] O. Shaaban, “Automatic Diacritics Restoration for Arabic Text,” King Fahd University of Petroleum and Minerals, 2013.

[71] V. Tymchenko, “BzReader, an application to browse Wikipedia compressed dumps offline.” [Online]. Available: http://code.google.com/p/bzreader/. [Accessed: 01-Jul-2012].

[72] The Wikimedia Foundation Inc, “Database dump progress.” [Online]. Available: http://download.wikimedia.org/backup-index.html. [Accessed: 01-Jul-2012].

[73] Ayaspell-dic Project, “Arabic Stopwords.” [Online]. Available: http://arabicstopwords.sourceforge.net. [Accessed: 13-Sep-2012].

[74] 1ste Keuze B.V., “Rank.nl Arabic Stopwords.” [Online]. Available: http://www.ranks.nl/stopwords/arabic.html. [Accessed: 13-Sep-2012].

[75] S. Khoja and R. Garside, “Stemming arabic text,” *Lancaster, UK, Comput. Dep. Lancaster Univ.*, 1999.

[76] M. F. Porter, “An algorithm for suffix stripping, Program,” vol. 14, no. 3, p. 130−137, 1980.

[77] J. B. Lovins, “Development of a Stemming Algorithm,” *Mech. Transl. Comput. Linguist.*, vol. 11, pp. 22–31, 1968.

[78] Agricultural Information Management Standards (AIMS), “AGROVOC Agricultural Thesaurus.” [Online]. Available: http://aims.fao.org. [Accessed: 11-Nov-2012].

[79] UNESCO, “UNESCO nomenclature for fields of science and technology (UNESCO6).” [Online]. Available: http://skos.um.es/unesco6. [Accessed: 11-Nov-2012].

[80] O. Medelyan, “Human-competitive automatic topic indexing, PhD thesis,” University of Waikato, New Zealand, 2009.

[81] “KP-Miner,” *Samhaa R.El-Beltagy and Ahmed Rafea*. [Online]. Available: http://www.claes.sci.eg/coe_wm/kpminer. [Accessed: 04-Jul-2012].

[82] V. Rijsbergen, *Information Retrieval*, Second Edi. Butterworths, 1979.

[83] K. Leininger, “Interindexer consistency in PsycINFO,” *J. Librariansh. Inf. Sci.*, vol. 32, no. 1, pp. 4–8, Mar. 2000.

Appendix A

Evaluations of the Usage of Vocabulary Files

In this section we discuss the evaluations that we conducted on KEA using different vocabulary files. The purpose is to study the effect of the vocabulary files on the obtained results from KEA. In our first evaluation, we tested KEA with three different settings i.e. without any vocabulary files, with the AGROVOC, and with UNESCO6. We used our English dataset which contains 244 documents. The results are shown in Table 15. KEA with no vocabulary files scored 60 exact matches where KEA with AGROVOC scored only 14 exact keyphrases. This is a decrease in the quality of returned keyphrases by 76.7%. In the other hand, KEA with scored the lowest exact matches with only 10 which represents a decrease by 83.3% from the default behavior. The precision, recall, and F-score followed the same results. Our English dataset contains scientific papers from different fields related to computer science. Since both vocabulary files focus on other fields, the results from using these files shown to be poor.

Table 15 - Results on the SM2010 English dataset

Measure	Default - No Vocabulary	AGROVOC	UNESCO6
Exact Match	60	14	10
Precision	0.150	0.035	0.025
Recall	0.156	0.037	0.020
F-Score	0.153	0.036	0.022

In our second evolution, we elected to use a specific dataset and relevant to one of the vocabulary files we have. The selected dataset is the FAO-780 dataset [80]. This dataset contains 780 documents and it is provided by the Food and Agriculture Organization of the United Nations. It contains several surveys, facts, and other agriculture topics covering different geographical regions around the world. The documents in this dataset ranged from 4 to 339 pages. Number of words ranged from 1,338 in the smallest document to reach to 73,510 words in the largest file. The size of these documents ranged from 9 Kilo Byte to 551 Kilo Byte. The manual keyphrases varied from 3 to 14.

For our evaluation purposes, we selected a subset contains 175 documents – 75 training and 100 test. The results are displayed in Table 16. Since the FAO780 dataset is relevant to the AGROVOC dataset, the exact matches feature showed an improvement by approximately 148% over the default KEA settings without a vocabulary usage. In this evaluation and with the usage of UNESCO6 as a vocabulary database, the results decreased by 73.3% from the default behavior without KEA.

Table 16 - Results for the evaluation on FAO-780 dataset

Measure	Default - No Vocabulary	AGROVOC	unesco6
Exact Match	60	149	16
Precision	0.156	0.187	0.022
Recall	0.304	0.284	0.173
F-Score	0.153	0.249	0.029

These evaluations imply the correlation between the coverage of the vocabulary file and the tested dataset.

Appendix B

Helper Tools

During the course of the thesis preparation, we used several tools and software that helped us in accomplishing our work. This section mentions some of the used tools.

- Eclipse: we used Eclipse as our main development environment. The version used is the "Java EE IDE" available at www.eclipse.org
- Mendeley: we used this tool to organize our references and sync them online. It has a Microsoft Office plugin which came very handy when adding and searching for our references. In Mendeley, we can different resources like journal papers, books, web pages …etc. It has the capability to search automatically for the full information about journal and conference papers. Mendeley can be downloaded from www.mendeley.com
- Microsoft Excel: we used this software with our experiment results. The calculation for exact matching, precision, recall, and F-score were done in Excel. Also, we used this software to create the graphical representation of the results.
- Total Assistant: we used this tool to provide us with statistical information about our Arabic dataset with regard to number lines, words, and characters. It is available at www.surefiresoftware.com/totalassistant/
- BzReader: we used this tool to search for Wikipedia articles. It allows offline browsing of the Wikipedia dump files. The tool can be downloaded from http://code.google.com/p/bzreader/

- DropBox: we used this utility to sync our files between different machines. It was used to save our source code, datasets, experiment results, references, and even the Thesis itself. This utility is available on different platforms like desktop, web, and mobile. It can be downloaded from www.dropbox.com

Appendix C

Content of Enclosed CD-ROM

During the preparation for this thesis, we generated and used several material including sources code, experiment files, software, resources, and other documents. The provided CD-ROM has the content shown in figure 20 and below is the description of the content:

- The thesis document which contains all the write-up and the description of the conducted work.
- Thesis presentation which is a Microsoft PowerPoint presentation summarizes the thesis documents in a set of slides.
- Resources folder: is a folder contains all the papers referenced in this thesis.
- Implementation folder: is a folder contains the source code of KEA. The main folder contains utilities classes to test KEA for English and Arabic. Also, it has the source code for the analyzer class which is used in the empirical study.
- Analysis folder: contains the results for the exact matched, precision, recall, and F-score for each file in the datasets from all comparative analyses. The results are stored in 6 Excel spreadsheets.
- Datasets folder: contains two sub directories for the English and Arabic datasets. The folder for Arabic dataset is divided into folders: the first contains the dataset with their categories and the second contains the files in a flat format.

- Helper Tools folder: contains the installation files for some of the tools mentioned in Appendix B.

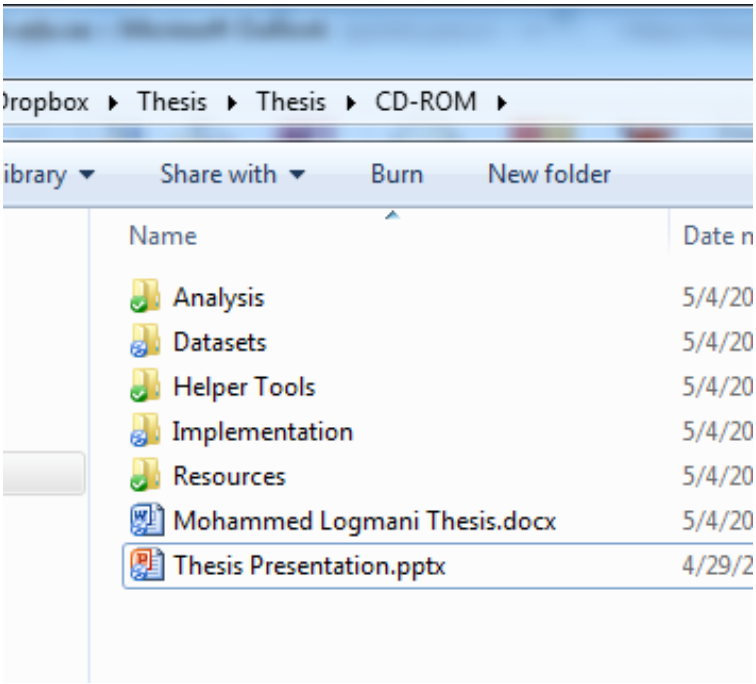


Figure 20- CD-ROM Content

Vitae

Name : Mohammed Suliman Al Logmani

Nationality : Saudi

Date of Birth : 18/03/1979

Email : mohammed.logmani@aramco.com

Address : P.O.Box 11340 Dhahran 31311

Academic Background : Mohammed Al Logmani is a graduate student at King Fahd University of Petroleum and Minerals. He earned his Bachelor's degree with first honor from King Abdulaziz University. His graduate and undergraduate major is Information and Computer Science. He participated in the 2009 International Conference on Information & Knowledge Engineering, IKE 2009. In this conference, he presented his paper titled "Empirical Evaluation of LSI-Based Indexing Techniques of Arabic Gigaword Documents". Mohammed works at Saudi Aramco in the Information Technology admin area. He works as an IT applications consultant and architect in the field of Content Management and Collaboration solutions.