



# **A Fusion-based Extraction of Key Phrases From Abstracts of Scientific Articles**

**By**

**Amjed Kameel Tawfiq Ayoub**

**Supervisor**

**Dr. Muath Refat Alzghool**

**Assistant Professor**

**Submitted in Partial Fulfillment of the Requirements for the Master's Degree in  
Computer Science**

**Faculty of Graduate Studies at AL-Balqa' Applied University  
Salt-Jordan**

**22, May, 2016**

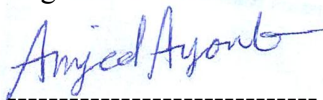
# Declaration of Authorship / Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and in the preparation of the thesis itself has been acknowledged.

I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

  
-----


# Committee Decision

This Thesis is “A Fusion-based Extraction of Key Phrases from Abstracts of Scientific Articles” was successfully defended and approved on: 22<sup>nd</sup> of May 2016.

## Examination Committee

## Signature

Dr. Muath Alzghool, (Supervisor)  
Assistant Professor of Computer Science.



Dr. Hasan Rashaideh  
Assistant Professor of Computer Science.



Dr. shihadeh Alrainy  
Associate Professor of Software Engineering.



Dr. Mohammad GH. I. Al Zamil, (External Examiner)  
Associate Professor of Computer Science,  
Yarmouk University.



# Dedication

I lovingly dedicate this thesis to my parents who supported and encouraged me in every step of the way.

A. Amjed

# Acknowledgment

First and foremost, I would like to express my appreciation to my Professor supervisor Dr. Muath Alzghool for his support, encouragement, and patience throughout the years of obtaining my master's degree and especially throughout the preparation period of my thesis. His technical support, advice, and expertise have contributed essentially to the completion of this thesis.

I thank my colleagues at Al Balqa' Applied University for sharing experiences and knowledge during the time of study.

I would also like to acknowledge the many friends especially Alazrai Dina, students, and professors who assisted and supported my thesis research in one way or another.

# Contents

DECLARATION OF AUTHORSHIP / ORIGINALITY .....	II
COMMITTEE DECISION .....	III
DEDICATION .....	IV
ACKNOWLEDGMENT .....	V
CONTENTS .....	VI
LIST OF TABLES .....	IX
LIST OF FIGURES .....	X
LIST OF ACRONYMS .....	XII
ABSTRACT .....	XIII
<b>1 INTRODUCTION .....</b>	<b>1</b>
<b>1.1 OVERVIEW .....</b>	<b>1</b>
<b>1.2 RESEARCH QUESTIONS .....</b>	<b>2</b>
<b>1.3 SIGNIFICANCE OF THE STUDY .....</b>	<b>3</b>
<b>1.4 CONTRIBUTIONS .....</b>	<b>4</b>
<b>1.5 THESIS STRUCTURE .....</b>	<b>6</b>
<b>2 BACKGROUND CONCEPTS AND RELATED WORK .....</b>	<b>8</b>
<b>2.1 INTRODUCTION .....</b>	<b>8</b>
<b>2.2 OVERVIEW .....</b>	<b>11</b>
<b>2.3 SIMPLE STATISTICS APPROACHES .....</b>	<b>11</b>
<b>2.4 LINGUISTIC APPROACHES .....</b>	<b>12</b>
<b>2.5 MACHINE LEARNING APPROACHES .....</b>	<b>14</b>
<b>2.6 MIXED APPROACHES .....</b>	<b>15</b>
<b>2.7 FUSION-BASED APPROACHES .....</b>	<b>16</b>
<b>3 KEYWORD EXTRACTION TECHNIQUES AND COLLECTION DESCRIPTION .....</b>	<b>18</b>
<b>3.1 KEYWORD EXTRACTION TECHNIQUES .....</b>	<b>18</b>
<b>3.2 COLLECTION .....</b>	<b>19</b>
<b>3.3 EVALUATION METHODS AND PERFORMANCE MEASURES .....</b>	<b>21</b>
<b>4 RAPID AUTOMATIC KEYWORD EXTRACTION .....</b>	<b>23</b>
<b>4.1 INTRODUCTION .....</b>	<b>23</b>
<b>4.2 STOPLIST GENERATION .....</b>	<b>24</b>
<b>4.3 CANDIDATE KEYWORDS .....</b>	<b>26</b>
<b>4.4 KEYWORDS SCORES .....</b>	<b>28</b>
<b>4.5 ADJOINING KEYWORDS .....</b>	<b>31</b>

4.6 EXTRACTED KEYWORDS.....	32
4.7 EVALUATING EFFICIENCY.....	32
5 TEXT RANK MODEL.....	34
5.1 INTRODUCTION .....	34
5.2 GOOGLE PAGE RANK ALGORITHM .....	34
5.3 TEXT RANK MODEL .....	38
5.3.1 Text as a Graph.....	40
5.3.1.1 Bidirectional Graph.....	42
5.3.1.2 Forward Graph.....	44
5.3.1.3 Backward Graph.....	45
5.3.2 Applying Page Rank Algorithm.....	45
5.3.3 Post-Processing Steps to Extract Keywords and Keyphrases.....	46
5.3.4 Weighted Graph .....	46
5.4 EVALUATING EFFICIENCY.....	48
6 N-GRAMS MODEL .....	49
6.1 INTRODUCTION .....	49
6.2 EXTRACTING KEYWORDS .....	49
6.3 CALCULATE TERM FREQUENCY.....	56
6.4 EVALUATING EFFICIENCY.....	58
7 FUSION-BASED TECHNIQUE.....	60
7.1 INTRODUCTION .....	60
7.2 FUSION-BASED TECHNIQUE.....	62
7.2.1 All Keywords Fusion Technique (Base method).....	62
7.2.2 Majority Voting Technique.....	65
7.2.3 Borda Voting Technique.....	67
7.2.4 CombMNZ Technique .....	70
7.2.5 WCombMNZ Technique.....	73
7.2.6 Condorcet Technique.....	76
7.3 EXPERIMENTAL RESULTS .....	79
8 CONCLUSION .....	82
8.1 INTRODUCTION .....	82
8.2 SUMMARY OF THE THESIS .....	82
8.3 ANSWERING RESEARCH QUESTIONS .....	83
8.4 FUTURE WORK .....	85
REFERENCES .....	86
APPENDICES .....	89
APPENDIX A.....	89

APPENDIX B.....	100
الملخص باللغة العربية .....	107



# List of Tables

Table 2.1 Penn Treebank Tagset .....	9
Table 4.1 The 25 most frequent words in the Inspec training set listed in descending order by term frequency .....	25
Table 4.2 The word co-occurrence graph for content words in the sample abstract .....	29
Table 4.3 Word scores calculated from the word co-occurrence graph .....	30
Table 4.4 Candidate keywords and their calculated scores .....	30
Table 4.5 Comparison of keywords extracted by RAKE to manually assigned keywords for the sample abstract.....	32
Table 4.6 RAKE results for recall, precision, and F-measure.....	33
Table 5.1 Results of PageRank Scores of the above Example for 10 Iterations .....	38
Table 5.2 TextRank results for recall, precision, and F-measure .....	48
Table 6.1 Unigram Keywords .....	52
Table 6.2 Bigram Keywords .....	52
Table 6.3 Unigram Keywords after Stopword removal.....	53
Table 6.4 Bigram Keywords after Stopword removal.....	53
Table 6.5 Unigram Keywords after tag filter .....	54
Table 6.6 Bigram Keywords after tag filter .....	54
Table 6.7 Final Unigram Keywords .....	55
Table 6.8 Final Bigram Keywords .....	55
Table 6.9 Final Quadgram Keywords.....	56
Table 6.10 Keywords Frequency (TF).....	56
Table 6.11 N-grams results for recall, precision, and F-measure.....	59
Table 7.1 Extracted keywords by RAKE, TextRank, and N-grams .....	62
Table 7.2 The extracted keywords by All Keywords Fusion technique .....	63
Table 7.3 The extracted keywords by Majority Voting technique .....	65
Table 7.4 Keywords with their scores .....	68
Table 7.5 The extracted keywords by Borda Voting technique .....	68
Table 7.6 Keyword's scores * DF.....	71
Table 7.7 The extracted keywords by CombMNZ technique .....	71
Table 7.8 keywords scores after multiplying by the techniques weight .....	74
Table 7.9 The extracted keywords by WCombMNZ technique .....	74
Table 7.10 The extracted keywords by Condorcet technique .....	77
Table 7.11 Experimental results for the six Fusion-based techniques.....	79
Table 7.12 Number of retrieved and relevant retrieved keywords for the six Fusion-based techniques....	80

# List of Figures

Figure 1.1 General structure of the keyword extraction process.....	2
Figure 3.1 Example of an abstract and its controlled, uncontrolled terms from Hulth 2003 dataset .....	200
Figure 4.1 General structure of RAKE method.....	244
Figure 4.2 A sample abstract from the Inspec test set and its manually assigned keywords.....	27
Figure 4.3 Candidate keywords parsed from the sample abstract.....	277
Figure 4.4 Number of Relevant, Retrieved, RelRet document by RAKE and the results of recall, precision, and F-measure .....	33
Figure 5.1 Example of PageRank Algorithm.....	36
Figure 5.2 General structure of the TextRank model .....	40
Figure 5.3 Example of an English Abstract, Title and its Manual Keywords and Keyphrases for the document in Hulth Dataset .....	41
Figure 5.4 A Tagged English Abstract for the document in Hulth Dataset .....	41
Figure 5.5 The Selected Vertices from document in Hulth Dataset.....	42
Figure 5.6 Bidirectional Graph for document in Hulth Dataset with Window Size=2.....	43
Figure 5.7 Example on Weighted PageRank Scores .....	47
Figure 5.8 Number of Relevant, Retrieved, RelRet document by TextRank.....	48
Figure 6.1 General structure of keyword extraction using N-grams model.....	5050
Figure 6.2 Example of a text document consists of an abstract and title.....	511
Figure 6.3 Tagged Text document .....	511
Figure 6.4 Keyphrases Text File .....	588
Figure 6.5 Number of Relevant, Retrieved, RelRet document by N-grams .....	599
Figure 7.1 General structure of the fusion-based techniques .....	611
Figure 7.2 Comparison between All Keywords Fusion technique and the three other techniques .....	644
Figure 7.3 Comparison between All Keywords Fusion technique and the three other techniques based on recall, precision, and F-measure.....	655
Figure 7.4 Comparison between Majority Voting and the three other techniques based on recall, precision, and F-measure.....	666
Figure 7.5 Comparison between Majority Voting technique and the three other techniques .....	677
Figure 7.6 Example shows the process of Borda count method .....	677
Figure 7.7 Comparison between Borda Voting and the three other techniques based on recall, precision, and F-measure.....	699
Figure 7.8 Comparison between Borda Voting technique and the three other techniques .....	7070
Figure 7.9 Comparison between CombMNZ technique and the three other techniques based on recall, precision, and F-measure .....	722
Figure 7.10 Comparison between CombMNZ technique and the three other techniques .....	733
Figure 7.11 Comparison between WCombMNZ technique and the three other techniques based on recall, precision, and F-measure .....	755
Figure 7.12 Comparison between WCombMNZ technique and the three other techniques.....	766
Figure 7.13 Comparison between Condorcet technique and the three other techniques based on recall, precision, and F-measure .....	788
Figure 7.14 Comparison between Condorcet technique and the three other techniques .....	788
Figure 7.15 Comparison between the six fusion-based techniques based on recall, precision, and F-measure.....	80

Figure 7.16 Comparison of the retrieved and relevant retrieved keywords between the six fusion-based techniques ..... 811

# List of Acronyms

<b>Acronym</b>	<b>Definition</b>
NLP	Natural Language Processing
RAKE	Rapid Automatic Keyword Extraction
IR	Information Retrieval
POS	Part Of Speech
TF	Term Frequency
TF-IEF	Term Frequency – Inverse Document Frequency
EDR	Electronic Dictionary Resources
PW	Position Weight
TF-ITF	Term Frequency – Inverse Term Frequency
PW-IPW	Position Weight – Inverse Position Weight
KEA	Keyword Extraction Algorithm
DCRFs	Dynamite conditional Random Fields
SVM	Support Vector Machines
TF-ISF	Term Frequency – Inverse Sentence Frequency
CL-SR	Cross-Language Speech Retrieval
CLEF	Cross-Language Evaluation Forum
ASR	Automatic Speech Recognition
TREC	Text REtrieval Conference



## **Abstract**

### **A Fusion-based Extraction of Key Phrases From Abstracts of Scientific Articles**

**By**

**Amjed Kameel Tawfiq Ayoub**

**Supervisor**

**Dr. Muath Refat Al Zghool**

**Assistant Professor**

The main objective of this research is to extract keywords from documents using a fusion-based technique, which is based on three other existing techniques that use “Hulth 2003” as a dataset for training and testing.

Keywords can be defined as the words that represent the content of the whole text, which can help users identify what the article is about without having to read it entirely. Filtering documents using “keywords” can be regarded as short summaries, which may save time while searching. Keywords extraction is an important technology in many areas of information technology and document processing such as document tagging, text categorization and summarization, and most importantly, the use in information retrieval. The main aim of any keywords extraction system is to automatically identify the most informative and important words that best describe the document. The simplest possible approach is perhaps to use a frequency criterion to select the important keywords in a document. However, this method was generally found to lead to poor results. Consequently, methods like RAKE, TextRank, and N-grams were explored.

As mentioned above, this research will introduce a fusion-based technique. The three original techniques produce their results as lists of extracted keywords. This new technique

integrates and calculates the results of those three techniques to produce its' own results of more accurate keywords than the keywords that are produced individually.

After implementing the four techniques above, the results were evaluated using precision, recall, and F-measure. The dataset that was used is Hulth 2003, which consists of 2000 scientific abstracts; 1000 of which are used as training data and 500 of which are used as testing data.

The maximum recall that has been achieved on this testing set using RAKE is 0.4996, precision is 0.4061, and F-measure is 0.43. Regarding TextRank, its maximum achieved recall is 0.5044, precision is 0.4081, and F-measure is 0.4324. The last original technique, N-grams, obtained its highest recall of 0.4832, precision of 0.4048, and F-measure of 0.4233. Finally, upon implementing several fusion-based techniques. The highest recall is 0.5117, its precision is 0.4139 and finally, its F-measure is 0.4387.

# Chapter 1

## Introduction

### 1.1 Overview

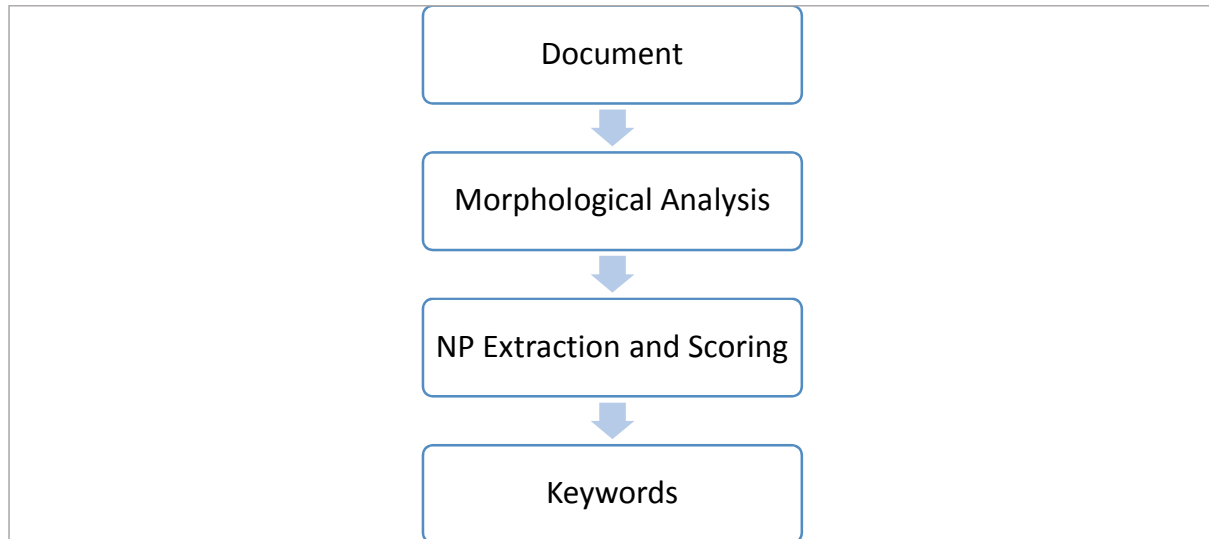
Natural Language Processing (NLP) is the ability of a computer program to understand human speech as it is spoken. It is a technique where computers become more humanized by reducing the gap between humans and computers. NLP is used to understand natural language text or speech. Applications of NLP include a number of fields of studies, such as machine translation, natural language text processing and summarization.

Information Retrieval (IR) is an area where applications of NLP can be viewed to extract information that is required from a huge database. There are many information technology fields where IR is applied to get results faster. One of the NLP's applications is Keywords Extraction System which is the topic of this research.

Keywords play an important role in extracting the correct information based on user requirements. Everyday, thousands of books, papers are published which makes it very difficult to go through all the text material. Instead, there is a need of good information extraction or summarization methods which provide the actual contents of a given document. As such effective keywords are important. Since keyword is the smallest unit which express meaning of entire document (Bracewell & REN, 2005), keywords and keyphrases provide a simple way of describing a document giving the reader a general idea about its content.

Keywords may be useful in various applications such as retrieval engines, browsing interfaces, and text mining. A list of extracted keywords and keyphrases of a document may serve a summary to help readers in searching for relevant information. Manually selecting keywords and keyphrases is not randomly done. In addition, automatic keywords and keyphrases extraction is not an easy task. Therefore, it needs to automate the process of selecting important keywords due

to its performance in managing information (Sarkar, Nasipuri, & Ghose, 2010). Figure 1.1 illustrates the general structure of the keyword extraction process.



*Figure 1.1 General structure of the keyword extraction process*

Early approaches to automatically extracting keywords focus on evaluating corpus-oriented methods. Later on, keyword extraction researches apply methods to select keywords based on individual documents. Corpus-oriented methods typically operate only on single words. This further restricts the accuracy of selecting keywords because single words are often used in multiple and different contexts. In some contexts, single words may be important but not necessarily in others. To avoid such problems, this research is concerned with methods of keyword extraction that operate on individual documents. Such document-oriented methods will extract the same keywords from a document regardless of the current state of a corpus.

## **1.2 Research Questions**

Keywords provide important information about the content of the document. They can help users search through information more efficiently and indicate whether the document is relevant to the user's search or not. Keywords can also be used in different language processing tasks such



as text categorization and IR. However, most documents, and especially the spoken ones, do not provide keywords. Additionally, there is a need to automatically generate keywords for the large amount of written or spoken documents available now.

Many keywords extraction algorithms have been implemented for the English text, and many different methods have been used over the years, and new solutions are continuously being proposed to solve the keywords extraction problem. This research is focused on the fusion-based technique, because we found that combining the outputs of multiple systems has also been tried with some improvements. In this thesis, we have proposed several fusion-based techniques to combine the results of different sources of keywords or results of different keywords extraction methods which will be discussed later.

All fusion-based techniques depend on keyword lists achieved from three existing techniques as an input to be used in extracting more accurate keyword lists. These three existing techniques are the Rapid Automatic Keyword Extraction (RAKE), TextRank, and N-grams.

The research questions of this study are the following:

- Is the fusion-based keyword extraction techniques suitable to improve the results of the extracted keywords?
- Does the keyword position feature help to improve the performance of the fusion-based keywords and keyphrases extraction system?
- Do the keyword frequency feature and document frequency feature help to improve the performance of the fusion-based keywords and keyphrases extraction system?
- Does the keyword weighting feature help to improve the performance of the fusion-based keywords and keyphrases extraction system?

### **1.3 Significance of the Study**

Automatic keywords extraction has a main task of identifying a small set of words, key phrases, or keywords from a document. Those keywords describe the meaning of the document. The process of identifying keywords should be done systematically and with minimal or no human intervention, depending on the model. The goal of automatic keyword extraction is to apply the

power and speed of computation to the problems of assigning keywords without the high costs and drawbacks associated with human indexers.

Keyword search is usable and powerful in enabling efficient scanning of large document collections. Keywords of any document provide important information about the content which represents the meaning of the document. Keywords can also be used to enrich the presentation of search results. They are also applied to improve the functionality of Information Retrieval systems (IR). Furthermore, Any IR system that lists documents related to a primary document's keywords and supports keyword's hyperlinks between documents, enables users to quickly access related materials (Jones & Paynter, 2002).

The manual extraction of keywords is slow, expensive, and full of mistakes. Therefore, most algorithms and systems that help users perform automatic keyword extraction have been proposed.

## **1.4 Contributions**

The thesis contributions are summarized as follows:

We implemented RAKE, TextRank, N-grams approaches for a collection of 1500 abstracts (abstracts and their titles) for journal papers of Computer Science and Information Technology topics regarding the linguistic and the word position feature in the document (abstract or title). And we compared the generated automatic keywords with a dataset of keywords that were found manually.

We designed a fusion- based keywords and keyphrases extraction system, which uses the results of the abovementioned approaches in order to improve the accuracy of the extracted keyword list and suggest more meaningful and expressive generated keywords and keyphrases for documents written in English (abstracts and their titles) than the other three lists.

The first technique we present is RAKE. RAKE was founded by Stuart R., Dave E., Nick C. and Wendy C., 2010. It is implemented based on keywords containing multiple words more than on standard punctuations or stopwords. This technique uses stopwords and word delimiters

to partition the documents into candidate keywords which will be given scores for each based on some calculations. Finally, extracted keywords will be produced based on these scores (Rose, Engle, Carmer, & Cowley, 2010).

The second technique is TextRank. This technique was established by Mihalcea R. and Tarau P., 2004. TextRank that has been used in this thesis is implemented by M. Alhadidi (Alhadidi, 2013) based on the co-occurrence links between words. It makes use of voting or weighing words in order to extract keywords. The technique is first implemented by constructing a graph which reflects relationships between different vertices (words). These vertices are extracted from the given texts and then using ranking algorithm, the words are given their scores. Finally, the words with the highest scores in the document will be chosen as keywords after an iterative algorithm is used to compute the ranking value of each vertex of the graph (Mihalcea & Tarau, 2004).

The third approach, Hulth-2003 N-grams approach, was introduced by Anette Hulth 2003. The implementation of the technique was done by A. Alklifat (Alklifat, 2014). All unigrams, bigrams, and trigrams up until five-grams were extracted. After that, a stop list is used, where all terms that are starting or ending with a stopword are removed from all keyword sequences. The technique then selects filtering N-grams keywords based on specific rules. Then the keywords are sorted based on their frequency. Finally, the technique generates the keyphrases text file (Hulth A. , 2003).

We designed and implemented a fusion-based keywords and keyphrases extraction system using the results of the three previous approaches. The proposed system in this research is different from the original systems in terms of:

- Considering the keyword position in the original keyword lists.
- Using keyword frequency in the document.
- Using document frequency for the keywords.
- Considering weighting the keywords.

An evaluation system has been constructed to compare the generated automatic keywords and keyphrases with the manual ones in that specific dataset. The best fusion-based technique was the WCombMNZ that achieved F-measure value of 0.4387.

## 1.5 Thesis Structure

This thesis is divided into eight chapters as follows:

- Chapter 1 is the introduction chapter. It starts with an overview of keywords extraction. Then, it states the research questions. After that, it emphasizes the significance of the study. And lastly, the chapter presents the research contribution with a brief introduction to the three original techniques, RAKE, TextRank, N-grams, as well as a brief introduction to the fusion-based technique.
- Chapter 2: This chapter starts with a brief introduction to the English language and the POS tag. Then, the chapter presents a detailed overview of the keywords and keyphrases extraction approaches which are simple statistics, linguistics, machine learning, mixed, and fusion-based approaches.
- Chapter 3: This chapter explains the keyword extraction techniques which are RAKE, TextRank, and N-grams and their implementation process. Then, it describes the used dataset collection. Finally, the evaluation methods are discussed.
- Chapter 4: This chapter presents the Rapid Automatic Keyword Extraction method that is extremely efficient. It shows the RAKE's generation of the stoplist, and how it deals with the texts to pick up the candidate keywords, and how it scores them to extract the keywords. Finally, the evaluating efficiency is explained.
- Chapter 5: This chapter shows the TextRank model that simulates the same idea of the PageRank algorithm using words instead of web pages. Additionally, the text is represented

as a graph (Forward, Backward, Bidirectional or weighted) on the collected dataset after a detailed explanation of the PageRank algorithm and a clear example are added. Finally the evaluating efficiency is explained.

- Chapter 6: This chapter shows the N-grams model. It discusses the model's architecture and its phases. The chapter also explains the basic steps of the model implementation. Finally, the evaluating efficiency is discussed.
- Chapter 7: This chapter presents several fusion-based techniques that have been implemented in order to reach high results. It explains the structure of the techniques and the basic steps of each technique implementation. Finally, the experimental results are explained.
- Chapter 8: This chapter presents the conclusions of this study. A summary of the thesis is introduced. The research questions are answered and the contributions of this study are discussed. Finally future works on the current system are presented and suggested.

# Chapter 2

## Background Concepts and Related Work

### 2.1 Introduction

English is one of the most spoken languages in the world. It is natively spoken by 45 countries; around 335 million native speakers, and 350 million second language speakers. English is considered as the unofficial international language of the world. English has the largest vocabulary of any language; the second edition of Oxford English Dictionary identifies around 171, 476 words, 47, 156 obsolete terms, and an additional 9, 500 derived words. Further, the total number of distinct words is about 228,132. Adding the technical and scientific vocabularies, this number grows to approach approximately 1.5 million words.

Considering the importance of the English language internationally, authors and writers of books, articles, novels, and researches tend to publish their work in English in order to gain and expand their audience. For this purpose, our research's aim is to extract keywords from articles that are written in English.

In the English language many words are used in different ways. This means that a word can function as several different Part Of Speech (POS). In other words, all words in the English language are divided into eight different categories. Each category has a different role/function in the sentence.

The traditional POS, excluding the two articles (a/an, the). These were:

1. **Nouns:** Nouns are naming words, which are used to refer to people, objects, or things. A noun functions as a subject or object of a verb and can be modified by an adjective.

2. **Pronouns:** Pronouns take the place of a noun or refer back to a noun, so pronouns are words that are used instead of a noun.
3. **Adjectives:** Adjectives give a description and more detailed information about a noun or a pronoun, just like describing a person or a thing.
4. **Verbs:** A verb is used to show an action, and gives information about actions.
5. **Prepositions:** Prepositions usually come before the noun or the pronoun to show where, when, how and why, in order to connect them to other words in the sentence, so they show a relationship between words and phrases.
6. **Conjunctions:** Conjunction is a word that joins parts of a sentence together.
7. **Adverbs:** Adverbs are words that describe more information about verbs, adjectives, adverbs, or the entire sentence.
8. **Interjections:** Interjections are used to express emotions or surprise in a form of a short sound, word, or a phrase.

For more clarification, some of the algorithms in this research operate based on the English POS tag. And the used methods had been working on Penn Treebank POS tagset which is shown in Table 2.1.

*Table 2.1 Penn Treebank Tagset*

POS Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	there is
IN	preposition subordinating conjunction	in, of, like
JJ	adjective	green

<b>POS Tag</b>	<b>Description</b>	<b>Example</b>
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest
LS	list marker	l
MD	modal	could, will
NN	noun, singular or mass	table
NNS	noun plural	Tables
NNP	proper noun, singular	John
NNPS	proper noun, plural	Vikings
PDT	predeterminer	both the boys
POS	possessive ending	friend's
PRP	personal pronoun	I, he, it
PRP\$	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
TO	To	to go, to him
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes
WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP\$	possessive wh-pronoun	whose
WRB	wh-adverb	where, when

This chapter details the history of keywords and keyphrases extraction systems and the previous works done in this field. Brief overview is presented. Most approaches in keywords extraction are viewed. Keywords and keyphrases extraction systems on English language are mentioned.



## 2.2 Overview

Manual keywords extraction process is very slow, exhausted, expensive, and prone for mistakes as mentioned above. For this reason, many automatic algorithms and systems for keywords extraction have been proposed. The methods are divided into five categories (Oelze, 2009):

- Simple statistics approaches
- Linguistics approaches
- Machine learning approaches
- Mixed approaches
- Fusion-based approaches

## 2.3 Simple Statistics Approaches

These methods are simple because they are concerned with the term frequency more than the document frequency and the position of the keyword. Furthermore, they have limited requirements and they do not require the training data. Words statistics can be used to identify the keywords in the document. Additionally, these methods are easy to use and they produce good results in general (Oelze, 2009).

Salton and his partners have used techniques that were based on the word frequency feature. They have proposed the **TF\*IDF** method. They concluded that a high-term frequencies in individual documents leads to high recall performance. On the other hand, low-term frequencies in the whole dataset were useful for high precision (Salton, Yang, & Yu, 1975).

Matsuo and Ishizuka based their work on a single document without using a corpus. They extracted then counted the frequent terms, after that the co-occurrences terms are counted. If a term appears frequently with a particular subset of terms, it is likely to be meaningful and considered as a keyword (Matsuo & Ishizuka, 2004).

Jiao and his partners also applied their algorithm for a single document without using a corpus but in Chinese language. They combined N-grams and word co-occurrence statistical analysis to carry out a Chinese keyword extraction experiment. Then, they extracted candidate keywords with bi-gram model which is used later to generate a set of co-occurrences between these words and frequent words. According to the analysis result, keywords were chosen from bi-grams and their experimental results were satisfying (Jiao, Liu, & bo Jia, 2007).

Mahgoub and his partners proposed a text mining technique for automatically extracting association rules on a collections of textual documents. The technique was called Extracting Association Rules from Text (EART). They have used keyword features to discover association rules amongst keywords labeling the documents. In their work, the EART system ignored the order in which the words occur. Instead they focused on the words and their statistical distributions in documents. They combined XML technology with Information Retrieval scheme (**TF-IDF**) and used Data Mining technique for association rules discovery (Mahgoub, Rösner, Ismail, & Torkey, 2008).

Wartena, Brussee and Slakhorst have based on the word co-occurrence in the document. In other words they concerned with the relations between words. The alternative relevance measures are computed by defining co-occurrence distributions for words and comparing these distributions with the document and the corpus distribution. For two corpora of abstracts with manually assigned keywords, they compared manually extracted keywords with different automatically extracted ones. Their results showed that using word co-occurrence information can improve precision and recall over **TF** and **IDF** (Wartena, Brussee, & Slakhorst, 2010).

## 2.4 Linguistic Approaches

Linguistic approaches use the linguistic features of the words. They are focused on linguistic features such as POS, syntactic structures, and semantic qualities. All of these features tend to add value, or functioning sometimes as filters for bad keywords (Oelze, 2009).

Hulth used different methods of integrating linguistics into keyword extraction. Moreover, she selected the terms as keywords based on three features; document frequency, collection

frequency, relative position of its first occurrence in a document and the term's POS tag. A better result is obtained as measured by keywords previously assigned by professional indexers when she did not only rely on statistics such as term frequency and N-grams, but when she also added linguistic features to the representation such as syntactic features (Hulth A. , 2003).

Lexical resources such as WordNet and the Electronic Dictionary Resources (EDR) have been used in several NPL tasks. WordNet has been used far more often than the EDR. Plas and his partners have used both resources on the same task. The task was automatic assignment of keywords to multi-party dialogue episodes. They showed that the use of lexical resources in such a task improved the performances than the use of a purely statistically based method (Plas, Pallotta, Rajman, & Ghorbel, 2004).

Xinghua and Wu proposed utilized linguistic features to represent the importance of the word position in a document, by using a Position Weight (**PW**) algorithm for keyword extraction. They measured the relationship between a topical term and its co-occurrence terms by using three methods: Term Frequency Inverse Term Frequency (**TF-ITF**), Position Weight Inverse Position Weight (**PW-IPW**), and Chi-Square (**Chi2**). The co-occurrence terms that have the highest degree of correlation and exceeded a co-occurrence frequency threshold were combined together with the original topical term to form a final keyword (Hu & Wu, 2006).

Ercan and Cicekli (Ercan & Cicekli, 2007) have used a lexical chain that holed a set of semantically related words of a text and they suggested that a lexical chain represented the semantic content of a part of the text. Although lexical chains have been extremely used in text summarization, their usage for keyword extraction problem has not been fully investigated. In their work, a keyword extraction technique that used lexical chains was described, and encouraging results were obtained.

Zhao and his friends (Zhao, Yang, & Ma, 2010) proposed a keyword extraction method named "Tag-Based Keyword Extraction" to extract keywords based on tags. They found that their method can be compared to the existing keyword extraction methods. They used linguistic features and/or statistical features of texts to accomplish keywords extraction with other valuable information.

## 2.5 Machine Learning Approaches

Keyword extraction can be considered as supervised learning. The machine learning mechanism works as follows. First a set of training documents is provided to the system, each of which has a range of human-chosen keywords as well. Then the gained knowledge is applied on new test documents to find keywords (Oelze, 2009).

Suzuki and his partners (Suzuki, Fukumoto, & Sekiguchi, 1998) proposed a method for keyword extraction of radio news. Two procedures were in their work: term weighting and keyword extraction. In term weighting, a feature vector of each domain was calculated using an encyclopedia and newspaper articles. On the other hand, in keyword extraction, keywords were extracted using feature vectors and result of domain identification. The results of experiments demonstrated the applicability of the method.

Witten and his partners (Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999) have described Keyword Extraction Algorithm (KEA) as an algorithm for automatically extracting keyphrases from text. KEA identified candidate keyphrases using lexical methods, and then calculated feature values for each candidate, then they used a machine-learning algorithm to predict which candidates are good keyphrases. They used a large test corpus to evaluate KEA's effectiveness in terms of how many author-assigned keyphrases are correctly identified.

Mihalcea and Tarau have introduced TextRank. TextRank is a graph-based ranking model for text processing. They showed how this model can be successfully used in natural language applications. They proposed two innovative unsupervised methods for keyword and sentence extraction, and showed that the results they achieved can be matched with previously published results (Mihalcea & Tarau, 2004).

Tiwari and his partners (Tiwari, Zhang, & Solorio, 2010) described a framework to extract precise information about co-expression relationship among genes, from published literature using a supervised machine learning approach. They used a graphical model, Dynamic Conditional Random Fields (DCRFs), for training their classifier. Their approach was based on semantic

analysis of text to classify the predicates describing co-expression relationship rather than detecting the presence of keywords.

## 2.6 Mixed Approaches

Keyword extraction can be mainly applied by combining the methods mentioned above or use some heuristic knowledge in the task of keyword extraction, such as the position, length, layout feature of the words, html tags around of the words, etc. (Oelze, 2009).

The overview of the related works of automatic keyword extraction has an advantage of being faster and less expensive than human intervention. However, currently existing solutions for automatic keyword extraction require either training examples or domain specific knowledge.

Keyphrases are important for document summarization, clustering, and topic search. Only a small amount of documents have manually assigned keyphrases which require an intensive human interaction. Therefore, it is favorable to automate the keyphrases extraction process. Frank and his partners (Frank, Paynter, Witten, Gutwin, & al., 1999) have shown that a simple procedure for keyphrases extraction based on the naive Bayes scheme performed comparably to the state of the art. It explained how this procedure's performance can be helpful by automatically modifying the extraction process to the specific document collection on a large collection of technical reports in computer science. They have shown that the quality of the extracted keyphrases improved significantly when domain-specific information is exploited.

Zhang and his partners proposed using both "global context information" and "local context information" for extracting keywords from documents. They also proposed methods for performing the tasks on the basis of Support Vector Machines (SVM). The proposed method has been applied to document classification, a typical text mining processing. When keyword extraction method is used, the accuracy of document classification can be dramatically improved (Zhang, Xu, Tang, & Li, 2006).

Dias and Malheiros used an algorithm for automatic keywords extraction for the Portuguese Language. In their work they focused on the extraction of keywords for theses on

several fields of knowledge. The KEA was used, together with a stemming technique specific to Portuguese and a manually created list of stopwords. Their results that were obtained were good enough for practical use.

Al-Hashemi studied a technique to produce a summary of an original text (Al-Hashemi, 2010). His model is divided into four steps. The preprocess stages converted the unstructured text into structured. In the first step, the system removed the stop words and assigning the POS tag for each word in the text and store the result in a table. The second step was to extract the important keyphrases in the text by implementing an algorithm through ranking the candidate words. The system used the extracted keywords/keyphrases to select the important sentence. They have used similarity measurements and features such as the existence of the keywords/keyphrases in it, the relation between the sentence and the title. The system used Term frequency, inverse document frequency and words existence in the document title to distinguish keywords. The third step of their proposed system was to extract the sentences with the highest rank. The fourth step was the filtering step which reduced the amount of the candidate sentences in the summary in order to produce a qualitative summary using **KF-IDF** measurement.

Gupta and Lehal proposed an automatic keywords extraction system for Punjabi language (Gupta & Lehal, 2011). The system included various phases like removing stop words, Identification of Punjabi nouns and noun stemming, Calculation of Term Frequency and Inverse Sentence Frequency (**TF-ISF**). The extracted keywords were helpful in automatic indexing, text summarization, information retrieval, classification, clustering, topic detection, tracking and web searches.

## **2.7 Fusion-Based Approaches**

M. Alzghool proposed five novel data fusion-based techniques which is concerned with speech information retrieval. The first fusion-based technique works by combining different models' results with their appropriate weights. The second technique uses cluster-based fusion technique. The third technique combines highly-varied retrieval results. The fourth technique is based on a heuristic derivation of each retrieval weight. The fifth fusion-based technique is based

on the probability theory. Alzghool's system achieved the best results in the Cross-Language Speech Retrieval (CL-SR) task at Cross-Language Evaluation Forum (CLEF) in 2005 and 2007 (Alzghool, 2009).

Fox and Shaw proposed six methods for combining five systems runs in 1993 based on the similarity values of documents to each query for each of the runs (Shaw & Fox, 1994) (Fox & Shaw, 1993). The best improvement was achieved by using CombSUM. CombSUM is the summation of the set of similarity values of the documents for each system's run.

D. He and J.-W. Ahn (He & Ahn, 2006) explored data fusion techniques for integrating the manually-generated meta-data information with the Automatic Speech Recognition (ASR) transcripts. They have explored a weighted CombMNZ model with different weight ratios and multiple iterations. Their initial results suggest that a simple unweighted combination method is useful in written retrieval environments.

R. M. Terol and his partners (Terol, Martinez-Barco, & Palomar, 2007) used two stages: the first stage is increasing the weight of some topic terms by applying a set of rules based on the representation of the topics by means of logic forms. Those logic forms are based on the analysis of syntactic dependencies in the topic descriptions and in the automatically-generated portion of the collection; the second stage is applying IR system, which uses the Okapi similarity measure to the collection, to produce overlapping passages.

M. Montague and J. A. Aslam presented a new algorithm for improving retrieval results by combining document ranking functions: Condorcet-fuse. Beginning with one of the two major classes of voting procedures from Social Choice Theory, the Condorcet procedure, they apply a graph-theoretic analysis that yields a sorting-based algorithm that is both efficient and effective. The algorithm performs successfully on Text REtrieval Conference (TREC) data. Condorcet-fuse significantly outperforms Borda-fuse, the analogous representative from the other major class of voting algorithms. The experimental results for Condorcet-fuse show that it usually outperforms standard metasearch algorithm (Montague & Aslam, 2002).

In 1972, Fisher and Elchesen showed that document retrieval results were improved by combining the results of two Boolean searches: one over the title words of the documents, and one over manually-assigned index terms (H. L. Fisher, 1972).

# Chapter 3

## Keyword Extraction Techniques and Collection Description

This chapter briefly describes the keyword extraction techniques that are implemented. These include the RAKE, TextRank, and N-grams techniques. Details of the document collections used in the experiments are then presented. The document collections is a part of Hulth 2003 collected by A. Hulth 2003 (Hulth A. , 2003). In addition, the description of the performance measures will be presented.

### 3.1 Keyword Extraction Techniques

#### **Rapid Automatic Keyword Extraction (RAKE)**

RAKE is an unsupervised, domain-independent and language-independent method used for extracting keywords from individual documents. RAKE presents results on a Hulth 2003 dataset of technical abstracts. The results showing that RAKE is computationally efficient in achieving higher precision and comparable recall scores. Next, RAKE uses a novel method for generating stoplists. Stoplists are used to configure RAKE for specific domains and corpora.

#### **TextRank for Keyword Extraction**

The expected end result for this application is a set of words or phrases that represent a given natural language text. The units to be ranked are sequences of one or more lexical units extracted from text, which represent the vertices that are added to the text graph. Any relation that can be determined between two lexical units is a potentially useful connection (edge) that can be added between two such vertices. Authors used a co-occurrence relation, controlled by the distance



between word occurrences; two vertices are connected if their corresponding lexical units co-occur within a window of maximum words, where can be set anywhere from 2 to 10 words. Co-occurrence links express relations between syntactic elements, and similar to the semantic links found useful for the task of word sense disambiguation (Mihalcea & Tarau, 2004), they represent cohesion indicators for a given text.

The vertices added to the graph can be restricted by syntactic filters, which select only lexical units of a certain part of speech. One, for instance, can consider only nouns and verbs in addition to the graph, and consequently draw potential edges depending on relations that can be established between nouns and verbs. Authors experimented with various syntactic filters, including: all open class words, nouns and verbs only, etc., with best results observed for nouns and adjectives only (Y., 2012).

### **N-grams approach**

In a first set of runs, the terms were defined in a manner where all unigrams, bigrams, and trigrams were extracted. Thereafter a stoplist was used, where all terms beginning or ending with a stopword were removed. Finally all remaining tokens were stemmed. In this paper, this manner of selecting terms is referred to as the N-gram approach (Hulth A. , 2003). The overview of the related works reveals that the automatic keyword extraction is faster and less expensive than human intervention.

## **3.2 Collection**

The dataset “Hulth 2003” consists of 2000 abstracts for journal papers from Computer Science and Information Technology. The abstracts are divided into a training set with 1000 abstracts, a validation set with 500 abstracts, and a testing set with 500 abstracts.

The abstracts are from the years 1998 to 2002, from journal papers, and from the disciplines Computers and Control, and Information Technology. Each abstract contains the title and the abstract.

Note, however, that the title is sometimes longer than one line. The abstract has two sets of keywords assigned by a professional indexer: a set of controlled terms (.contr), such as terms restricted to the Inspec Thesaurus, contains the controlled manually assigned keywords, separated with semicolon; and a set of uncontrolled terms (.uncontr) that contains the uncontrolled manually assigned keywords, separated with semicolon. The uncontrolled terms are the keywords that used in Hulth's experiments. Figure 3.1 shows an example which consists of an abstract and its controlled, uncontrolled terms.

<p><b>.Abstr:</b> the first line is the title and the lines is abstract</p>	<div> <div>Waiting for the wave to crest [wavelength services]</div> <div> Wavelength services have been hyped ad nauseam for years. But despite their quick turn-up time and impressive margins, such services have yet to live up to the industry's expectations. The reasons for this lukewarm reception are many, not the least of which is the confusion that still surrounds the technology, but most industry observers are still convinced that wavelength services with ultimately flourish </div> </div> <div> <div><b>Contr:</b> keywords, separated with semicolon.</div> <div>optical fibre networks; telecommunication</div> </div> <div> <div><b>.Uncontr:</b> keywords, separated with semicolon.</div> <div>wavelength services; fiber optic networks; Looking Glass Networks; PointEast Research</div> </div>
---	---

Figure 3.1 Example of an abstract and its controlled, uncontrolled terms from Hulth 2003 dataset

Both the controlled terms and the uncontrolled terms may or may not be presented in the abstracts. However, the manually assigned keywords were generated using the entire documents. For the experiments described here, only the uncontrolled terms were considered, as they are presented in the abstracts (76.2% as opposed to 18.1%) (Hulth A. , 2003).

English words in the collected dataset documents (abstracts and their titles) were tagged based on the Penn Treebank tagset which is shown in Table 2.1. The new tagged dataset is used in the implementation of the TextRank and N-grams approaches.

### 3.3 Evaluation Methods and Performance Measures

The results are evaluated using precision, recall, and F-measure. The maximum recall that can be achieved on this research is less than 100%, since many manually built keywords and keyphrases are not existed in the documents (abstracts and their titles). Precision and recall are the basic measures used in evaluating search strategies (Mihalcea & Tarau, 2004).

Recall, as in Equation 3.1, is the ratio of the number of relevant keywords or keyphrases retrieved to the total number of relevant keywords in the file. The recall ratio is usually expressed as a percentage (Creighton., 2013).

$$\text{Recall} = R/N \quad (3.1)$$

Where  $R$  is the number of relevant keywords and keyphrases retrieved, and  $N$  is the number of relevant keywords in the collection.

Precision, as in Equation 3.2, is the ratio of the number of relevant keywords retrieved to the total number of irrelevant and relevant keywords retrieved. Similar to the recall ratio, the precision ratio is also usually expressed as a percentage (Creighton., 2013). Recall and precision are inversely related. In other words, when recall increases, precision decreases, and vice versa. High recall means that an algorithm returned most of the relevant keywords, while high precision means that an algorithm returned substantially more relevant keywords than irrelevant.

$$\text{Precision} = R/D \quad (3.2)$$

$R$  is the number of relevant keywords and keyphrases retrieved and  $D$  is the number of keywords and keyphrases retrieved.

F-measure, as in Equation 3.3, is a measure that combines precision and recall. It also called balanced F-score.

F-measure can be calculated as follows (Manning & Schutze, 2000):

$$F = 2 * \left( \frac{Recall * Precision}{Recall + Precision} \right) \quad (3.3)$$

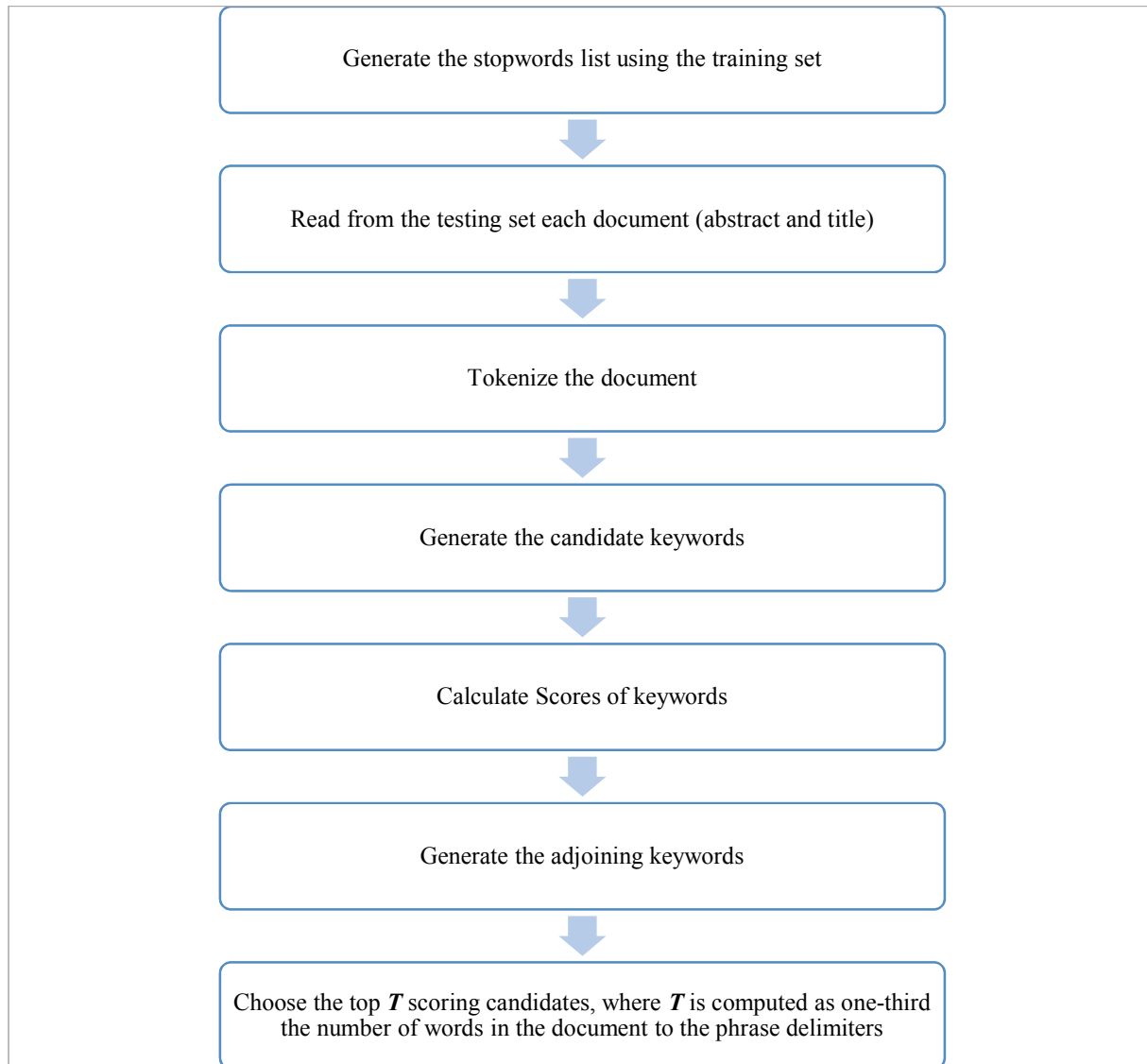
This measure can be considered as a measure of a test's accuracy. Since it considers both the precision and the recall of the test. Recall, Precision and F-measure are often used in IR field for measuring search, document classification, and query classification performance.

# Chapter 4

## Rapid Automatic Keyword Extraction

### 4.1 Introduction

This chapter deals with Rapid Automatic Keyword Extraction (RAKE) which was founded by Stuart R., Dave E., Nick C. and Wendy C., 2010 (Rose, Engle, Carmer, & Cowley, 2010). RAKE is an efficient keyword extraction method that operates on individual documents to enable application to dynamic collections, is easily applied to new domains, and operates well on multiple types of documents, particularly those that do not follow specific grammar conventions. RAKE is based on the observation that keywords frequently contain multiple words but rarely contain standard punctuation or stop words. Perl language was used in the technique implementation. The input parameters for RAKE consists of a list of stopwords (or stoplist), a set of phrase delimiters, and a set of word delimiters. RAKE uses stopwords and phrase delimiters to partition the document text into candidate keywords, which are sequences of content words as they occur in the text. Co-occurrences of words within these candidate keywords are meaningful and allow us to identify word co-occurrence. Word associations are measured in a manner that automatically adapts to the style and content of the text, enabling adaptive and fine-grained measurement of word co-occurrences that will be used to score candidate keywords. Finally the evaluation methods were performed and explained. Figure 4.1 shows the general structure of the method.



*Figure 4.1 General structure of RAKE method*

## 4.2 Stoplist generation

Stoplists are widely used in IR and text analysis applications. In addition to the English stopwords list, RAKE generates another stoplist based on evaluating the use of term frequency as a metric for automatically selecting words for the stoplist. The algorithm is based on that the words adjacent to, and not within, keywords are less likely to be meaningful and therefore are good

choices for stop words. By identifying each abstract in the training set, the words occurring adjacent to words in the abstract's uncontrolled keyword list. The frequency of each word occurring adjacent to a keyword was accumulated across the abstracts. Words that occurred more frequently within keywords than adjacent to them were excluded from the stoplist. Table 4.1 shows the 25 most frequent words in the training set generated by RAKE listed in a descending order based on the term frequency. The rest of the stopwords are included in Appendix A.

*Table 4.1 The 25 most frequent words in the Inspec training set listed in descending order by term frequency*

<b>Word</b>	<b>Term frequency</b>	<b>Document frequency</b>	<b>Adjacency frequency</b>	<b>Keyword frequency</b>
the	8611	978	320	3
of	5546	939	440	68
and	3644	911	251	23
a	3600	893	179	3
to	3000	879	77	10
in	2657	837	196	8
is	1974	757	84	0
for	1912	767	234	9
that	1129	590	35	0
with	1065	577	85	3
are	1049	576	38	1
this	964	581	22	0
on	919	550	65	8
an	856	501	45	0
we	822	388	12	0
by	773	475	32	0

<b>Word</b>	<b>Term frequency</b>	<b>Document frequency</b>	<b>Adjacency frequency</b>	<b>Keyword frequency</b>
as	743	435	22	0
be	595	395	1	0
it	560	369	2	13
system	507	255	7	202
can	452	319	19	0
based	451	293	27	15
from	447	309	13	0
using	428	282	50	0
control	409	166	237	0

### 4.3 Candidate Keywords

For more explanation, an example is applied which is a title and an abstract taken from the testing set of dataset Hulth-2003. The document name is (1939.ABSTR). The sample abstract from the Inspec test set and its manually assigned keyword is shown in Figure 4.2.



<b>Manually assigned keywords:</b>	Compatibility of systems of linear constraints over the set of natural numbers.
linear constraints	
set of natural numbers	
linear Diophantine equations	Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given.
strict inequations	
nonstrict inequations	
upper bounds	These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types
minimal generating sets	

Figure 4.2 A sample abstract from the Inspec test set and its manually assigned keywords

RAKE begins the extraction process by parsing the text into a set of candidate keywords. First, the document text is split into words based on the word delimiters, then the text is split again into sequences of words at phrase delimiters and stop word positions. Words within a sequence are assigned the same position in the text and together are considered as a candidate keyword. Figure 4.3 shows the candidate keywords in the order that they are parsed from the sample abstract.

*Compatibility – systems – linear constraints – set – natural numbers – criteria – compatibility – system – linear Diophantine equations – strict inequations – nonstrict inequations – upper bounds – components – minimal set – solutions – algorithms – minimal generating sets – solutions – systems – criteria – corresponding algorithms – constructing – minimal supporting set – solving – systems – systems*

Figure 4.3 Candidate keywords parsed from the sample abstract

## 4.4 Keywords Scores

After the candidate keyword is ready, a graph of word co-occurrences is built which is shown in Table 4.2 below. A score is calculated for each candidate keyword which is the sum of its member word scores. Several metrics have been evaluated for calculating word scores based on the degree and frequency of word vertices in the graph:

1. Word frequency ( $freq(w)$ ),
2. Word degree ( $deg(w)$ ), and
3. Ratio of degree to frequency ( $deg(w) / freq(w)$ ).

Table 4.2 The word co-occurrence graph for content words in the sample abstract

	algorithm	bounds	compatibility	components	constraints	constructing	corresponding	criteria	Diophantine	equations	generating	inequations	linear	minimal	natural	nonstrict	numbers	Set	sets	solving	strict	supporting	system	systems	upper
algorithm	2						1																		
bounds		1																							1
compatibility			2																						
components				1																					
constraints					1								1												
constructing						1																			
corresponding	1						1																		
criteria								2																	
Diophantine									1	1			1												
equations									1	1			1												
generating											1			1					1						
inequations												2				1					1				
linear					1				1	1			2												
minimal											1			3				2	1			1			
natural															1		1								
nonstrict												1				1									
numbers																	1								
set														2				3				1			
sets											1			1					1						
solving																				1					
strict												1									1				
supporting														1				1				1			
system																							1		
systems																								4	
upper		1																							1

In summary,  $\text{freq}(w)$  indicates the number of words that occur frequently but not the number of words which they co-occur. The  $\text{deg}(w)$  indicates the number of words that occur often and in longer candidate keywords. Further, the ratio of degree to frequency ( $\text{deg}(w) / \text{freq}(w)$ )

indicates the number of words that occur in longer candidate. These metric scores for each of the content words in the sample abstract are listed in the Table 4.3.

*Table 4.3 Word scores calculated from the word co-occurrence graph*

	algorithm	bounds	compatibility	components	constraints	constructing	corresponding	criteria	Diophantine	equations	generating	inequations	linear	minimal	natural	nonstrict	numbers	Set	sets	solving	strict	supporting	system	systems	upper
deg(w)	3	2	2	1	2	1	2	2	3	3	3	4	5	8	2	2	2	6	3	1	2	3	1	4	2
freq(w)	2	1	2	1	1	1	2	2	1	1	1	2	2	3	1	1	1	3	1	1	1	1	1	4	1
deg(w)/ freq(w)	1.5	2	1	1	2	1	2	1	3	3	3	2	2.5	2.7	2	2	2	2	3	1	2	3	1	1	2

The score for each candidate keyword is computed as the sum of its ratio scores. Which are listed in Table 4.4.

*Table 4.4 Candidate keywords and their calculated scores*

Candidate Keywords	Scores
minimal generating sets	8.7
linear Diophantine equations	8.5
minimal supporting set	7.7
minimal set	4.7
linear constraints	4.5
natural numbers	4
strict inequations	4
nonstrict inequations	4
upper bounds	4

Candidate Keywords	Scores
corresponding algorithms	3.5
set	2
algorithms	1.5
compatibility	1
systems	1
criteria	1
system	1
component	1
constructing	1
solving	1

## 4.5 Adjoining Keywords

As discussed above, RAKE splits candidate keywords by stop words, so the extracted keywords do not contain interior stop words. To solve this problem, RAKE looks for pairs of keywords that adjoin one another more than once in the same document and in the same order. A new candidate keyword is then created as a combination of the previous keywords and their interior stop words. The score of the new keyword is the sum of its member keyword scores. It should be noted that relatively few of these linked keywords are extracted, which adds to their significance. Because adjoining keywords must occur twice in the same order within the document, their extraction is more common on texts that are longer than short abstracts.

## 4.6 Extracted Keywords

After candidate keywords are scored, some of them must be selected to be extracted. After candidate keywords are scored, some of them must be selected to be the extracted keywords, for this the top  $T$  scoring candidates are selected as keywords for the document, which  $T$  is computed as one-third the number of words in the graph. This sample abstract contains 28 content words, resulting in  $T=9$  keywords. Table 4.5 below lists the keywords extracted by RAKE compared to the sample abstract's manually assigned keywords. There are manually assigned keywords not extracted by RAKE because approximately 70% of the manually assigned keywords did not exist in the abstracts.

*Table 4.5 Comparison of keywords extracted by RAKE to manually assigned keywords for the sample abstract*

Extracted by RAKE	Manually assigned
Minimal generating sets	Minimal generating sets
linear Diophantine equations	linear Diophantine equations
minimal supporting set	
minimal set	
linear constraints	linear constraints
natural numbers	
strict inequations	strict inequations
nonstrict inequations	nonstrict inequations
upper bounds	upper bounds
	Set of natural numbers

## 4.7 Evaluating efficiency

The experiments results were performed based on Recall, Precision, and F-measure as shown in the Table 4.6, and it achieved 0.43 of F-measure and has been tested on Hulth 2003 dataset.

Table 4.6 RAKE results for recall, precision, and F-measure

	Recall	Precision	F-measure
<b>RAKE</b>	0.4996	0.4061	0.4300

The right side of Figure 4.4 shows a comparison of relevant keywords with retrieved keywords and with relevant retrieved keywords by RAKE. The left side of Figure 4.4 shows RAKE results for recall, precision, and F-measure illustrated in a chart.

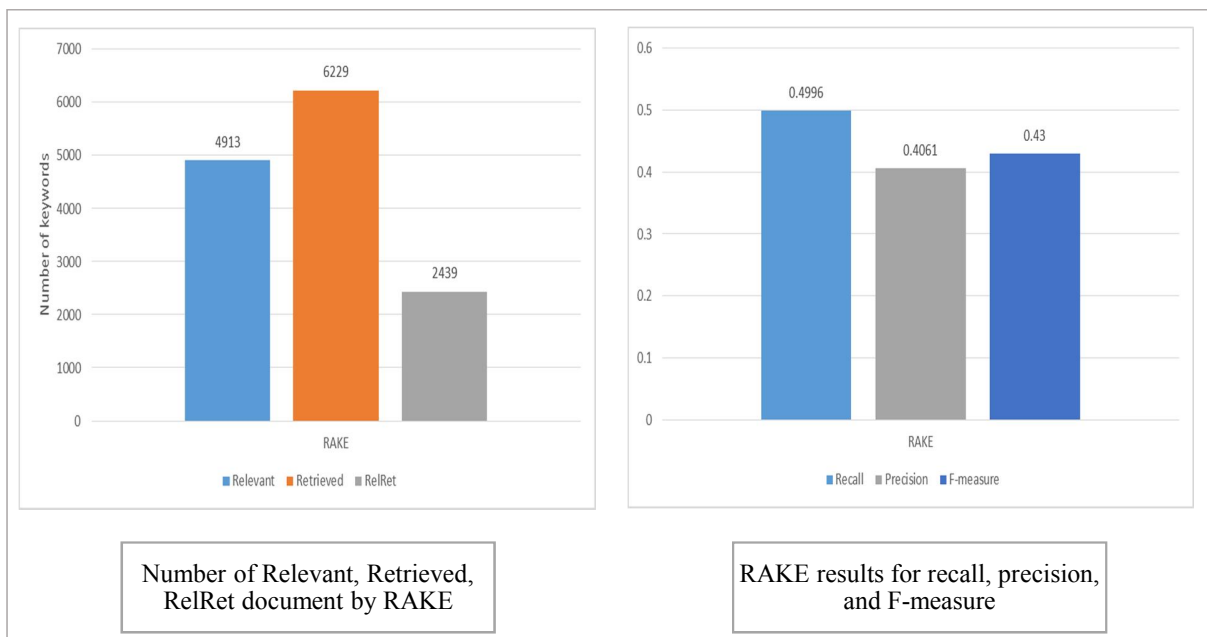


Figure 4.4 Number of Relevant, Retrieved, RelRet document by RAKE and the results of recall, precision, and F-measure

# Chapter 5

## Text Rank Model

### 5.1 Introduction

Keywords and keyphrases are defined as a sequence of one or more words extracted from a document that provide important information about the content of the document to describe its meaning. Two methods can be used in order to extract keywords from documents; unsupervised and supervised. The unsupervised method assigns each word a score and ranks the words according to their scores in the document. The score is computed based on a combination of statistical and linguistic feature, including term frequency, word position, signature, lexical chains and more. On the other hand, the supervised keywords extraction approach treats the task as a two class classification problem at the word level. Each word is represented by a vector of features, which can be adopted as the key of extraction. Features can be defined from linguistics, such as term significance or term location in a document (Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999).

This chapter starts with an explanation to keywords and keyphrases extraction system on English text. Next, a detailed explanation of PageRank algorithm and a clear example are clarified. TextRank model that simulates the idea of PageRank, using words instead of web pages, where text is implemented as a graph (Forward, Backward, Bidirectional or weighted) is also explained. Finally the evaluation methods were performed and discussed.

### 5.2 Google Page Rank Algorithm

Google is a search engine owned by Google. It is the largest search engine on the web. The company's mission is to make information accessible globally. Google receives over 200 million



queries each day through its various services. The heart of Google's searching software is PageRank. PageRank is a system for ranking web pages developed by Larry Page and Sergey Brin to decide how much a certain web page is important compared to other web pages based on the authority measure. A common measure of authority is the in-degree pages which points to a specific page (Brin & Page, 1998).

A PageRank algorithm implemented on a graph that is constructed from web pages as vertices is a way of deciding the importance of a vertex (a web page) within this graph. By taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information. The basic idea in a graph-based ranking model is voting. When one vertex (web page) links to another one, it is basically creating a vote for that other vertex. The higher the number of votes for a vertex, the higher the importance of that vertex. Each vertex will have a score that is determined based on the votes that are cast for it and on the scores of the vertices casting these votes (Mihalcea & Tarau, 2004).

Let  $G = (V, E)$  refer to the constructed graph. Where  $V$  is the set of vertices in the graph that represents the web pages, and  $E$  is the set of the edges between vertices. An edge is constructed in the graph between two vertices when one vertex points to the other vertex. For a given vertex  $V$ , let  $In(V)$  represent the set of vertices that points to the vertex  $(V)$ , and  $Out(V)$  represent the set of vertices that  $(V)$  points to. According to Brin and Page Equation 5.1 (Brin & Page, 1998), the score of a given vertex  $(V_i)$  is computed as follows:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad (5.1)$$

Where  $d$  refers to the damping factor, its value can be set between 0 and 1. The damping factor has the role of integrating the model into the probability of jumping from a given vertex to another random vertex in the graph. The factor  $d$  is usually set to 0.85 (Brin & Page, 1998).

The figure below shows an example to explain the idea of PageRank algorithm. The graph in this example consists of a subset of pages (page  $A$ , page  $B$ , page  $C$  and page  $D$ ) and their links as follows:

- Page  $A$  points to pages  $B$ ,  $C$  and  $D$
- Page  $B$  points to page  $D$

- Page **D** points to page **B** and **C**

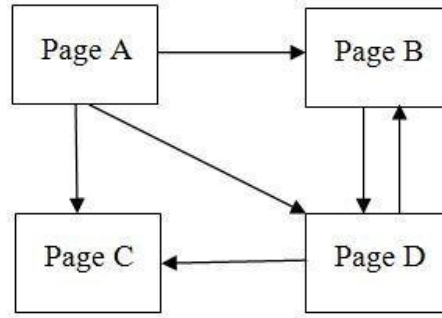


Figure 5.1 Example of PageRank Algorithm

Initially, the graph  $G = (V, E)$  is constructed as shown in the figure above, where  $V = \{A, B, C, D\}$  and  $E = \{(A, B), (A, C), (A, D), (B, D), (D, B), (D, C)\}$ . An initial PageRank value is assigned for each vertex in the graph by 1.

$$(S(A) = 1, S(B) = 1, S(C) = 1 \text{ and } S(D) = 1)$$

Then, Equation 5.1 is applied on each vertex for n number of iterations until convergence is achieved. Convergence is achieved when a value called an error rate for each vertex in the graph approaches or falls below a given value. The error rate is computed from two successive iterations, which can be defined as the difference between the score of the vertex at iteration  $K+1$ , and the score computed at iteration  $k$ , as shown in Equation 5.2 (Mihalcea & Tarau, 2004).

$$\text{Error Rate} = S^{k+1}(V) - S^k(V) \quad (5.2)$$

The PageRank scores for all pages in Figure 5.1 are computed as follows:

- For Iteration 0 (Initial Iteration):
  - $S(A)=1$

- $S(B)=1$
  - $S(C)=1$
  - $S(D)=1$
- For Iteration 1:
  - The score for page  $A$ :
    - $S(A) = (1 - 0.85) + 0.85 * 0$
    - $S(A) = 0.15$
  - The score for page  $B$ :
    - $S(B) = 0.15 + 0.85(\frac{S(A)}{OUT(A)} + \frac{S(D)}{OUT(D)})$
    - $S(B) = 0.15 + 0.85(\frac{1}{3} + \frac{1}{2})$
    - $S(B) = 0.8583$
  - The score for page  $C$ :
    - $S(C) = 0.15 + 0.85(\frac{S(A)}{OUT(A)} + \frac{S(D)}{OUT(D)})$
    - $S(C) = 0.15 + 0.85(\frac{1}{3} + \frac{1}{2})$
    - $S(C) = 0.8583$
  - The score for page  $D$ :
    - $S(D) = 0.15 + 0.85(\frac{S(A)}{OUT(A)} + \frac{S(B)}{OUT(B)})$
    - $S(D) = 0.15 + 0.85(\frac{1}{3} + \frac{1}{1})$
    - $S(D) = 1.2833$

These scores are calculated after running a small Perl program that has been designed in this research to calculate the PageRank scores for this example. Iterations will be continued until convergence is achieved. Table 5.1 shows various stages of iterations and their results. PageRank scores are stable after iteration 18 for each page.

Table 5.1 Results of PageRank Scores of the above Example for 10 Iterations

<i>Iteration No.</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
0	1	1	1	1
1	0.15	0.858333333	0.858333333	1.283333333
2	0.15	0.737916667	0.737916667	0.922083333
3	0.15	0.584385417	0.584385417	0.819729167
4	0.15	0.540884896	0.540884896	0.689227604
5	0.15	0.485421732	0.485421732	0.652252161
6	0.15	0.469707169	0.469707169	0.605108472
7	0.15	0.449671101	0.449671101	0.591751093
8	0.15	0.443994215	0.443994215	0.574720436
9	0.15	0.436756185	0.436756185	0.569895082

The algorithm is applied for  $n$  iterations until convergence is achieved. Convergence is achieved when the error rate for any vertex in the graph falls below a given threshold value. A threshold value of 0.0001 is used here (Mihalcea & Tarau, 2004). Note that the PageRank score for page *A* is constant at any iteration number. Its error rate value after any iteration number is 0. Since there are no pages pointing to *A*, in-degree for *A* is 0.

During the final stage, pages are sorted according to their PageRank scores in a descending order to show the importance of each page as follows:

- *Page (D)* = 0.557539231
- *Page (C)* = 0.429456273
- *Page (B)* = 0.429456273
- *Page (A)* = 0.15

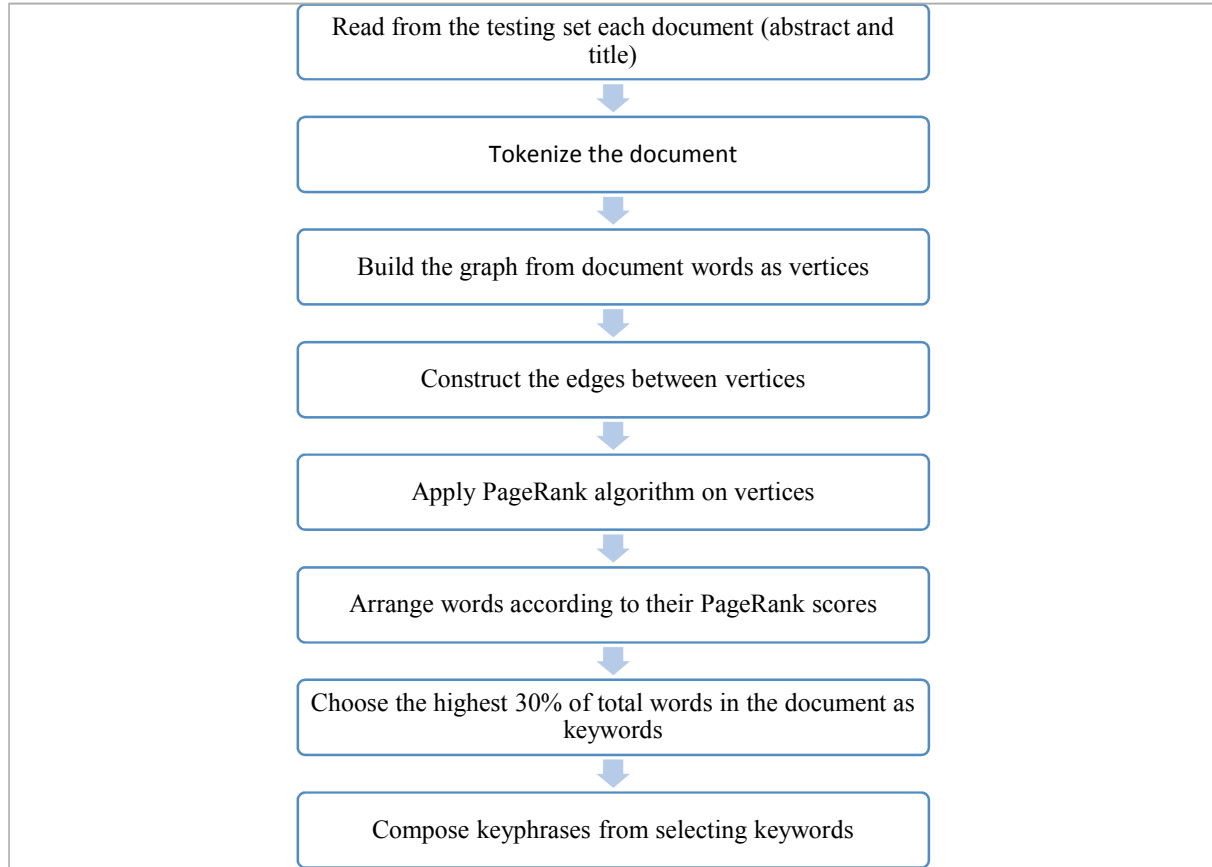
### 5.3 Text Rank Model

TextRank, as discussed by Mihalcea and Tarau (2004), constructs a network graph using candidate keywords as nodes, and co-occurrence to draw edges between them. TextRank then runs the PageRank algorithm upon the graph to rank each keyword's importance.

The TextRank algorithm uses of the Hulth (2003) dataset. The specific implementation of the algorithm performs the following steps:

1. Sentence boundaries are detected and each sentence is separated,
2. Each sentence is POS tagged using the Stanford POS tagger (using tags from the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993),
3. Vertices are chosen from the text based on their POS tag (currently only nouns and adjectives),
4. Edges are drawn between vertices that fall within a co-occurrence window of size  $n$ :
  - Edges can be bidirectional, forward directional, or backward directional,
  - If edge weighting is used, then the frequency count of that co-occurrence relation is used as the weight value.
5. PageRank algorithm is then run upon the constructed graph using initial values of 1 for each vertex until convergence within a threshold occurs:
  - A threshold value of 0.0001 is used,
  - A damping factor  $d = 0.85$  is used.
6. Vertices are sorted by their PageRank scores in a descending order and those  $k$  tokens are chosen as keywords:
  - Each token is expanded into a set of keyphrases by searching for each occurrence of the token in the original text, and for each occurrence collecting all adjacent words that are eligible tokens; and concatenating them into a phrase,
  - A keyphrase count value of  $k = \text{floor} \left( \frac{\text{total tokens}}{3} \right)$  is used, where Total Tokens is the total number of tokens in each document.

Figure 5.2 shows the general structure of the TextRank model.



*Figure 5.2 General structure of the TextRank model*

### 5.3.1 Text as a Graph

The main step of TextRank model is to construct a graph from the text, then apply the PageRank algorithm to the constructed graph. Figure 5.3 shows a document taken as an example from their dataset. The document name is (300.ABSTR) from Hulth (2003) dataset. First, every token in the document is tokenized and tagged as shown in Figure 5.4.

<b>Manually assigned keywords:</b>	The plot thins: thin-client computer systems and academic libraries.
Academic libraries	
Thin-client	
Computer systems	The few libraries that have tried thin client architectures have noted a number of compelling reasons to do so. For starters, thin client devices are far less expensive than most PCs. More importantly, thin client computing devices are believed to be far less expensive to manage and support than traditional PC.

Figure 5.3 Example of an English Abstract, Title and its Manual Keywords and Keyphrases for the document in Hulth Dataset

The_DT plot_NN thins_VBZ :_: thin-client_JJ computer_NN systems_NNS and_CC academic_JJ libraries_NNS ._.
The_DT few_JJ libraries_NNS that_WDT have_VBP tried_VBN thin_JJ client_NN architectures_NNS have_VBP noted_VBN a_DT number_NN of_IN compelling_JJ reasons_NNS to_TO do_VB so_RB ._.
For_IN starters_NNS ,_, thin_JJ client_NN devices_NNS are_VBP far_RB less_RBR expensive_JJ than_IN most_JJS PCs_NNS ._.
More_RBR importantly_RB ,_, thin_JJ client_NN computing_NN devices_NNS are_VBP believed_VBN to_TO be_VB far_RB less_RBR expensive_JJ to_TO manage_VB and_CC support_VB than_IN traditional_JJ PCs_NNS ._.

Figure 5.4 A Tagged English Abstract for the document in Hulth Dataset

Then, nouns of types (NN, NNS, NNP and NNPS) and adjectives of types (JJ, JJR and JJS) as shown in Table 2.1 in chapter 2, are selected to be the vertices to construct a graph of the above document. Co-occurrence relation is used to connect between the selected vertices, controlled by the distance between word occurrences.





window size. Figure 5.6 shows the bidirectional graph built for the English document in Hulth dataset with a window size=2.

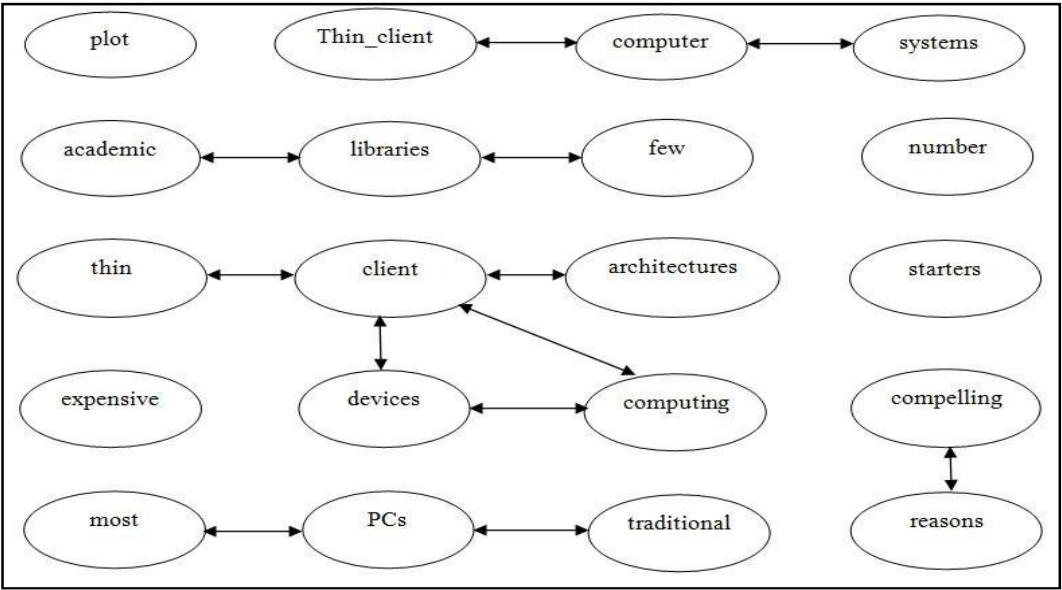
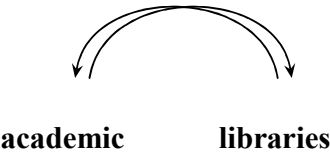


Figure 5.6 Bidirectional Graph for document in Hulth Dataset with Window Size=2

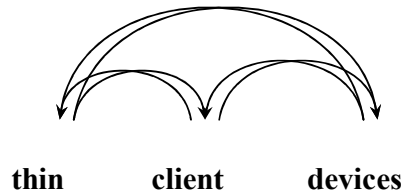
When the used window is set to 2, the highlighted tokens (vertices of the constructed graph) in yellow in Figure 5.5 are tested to find if there is any existed vertices within a window size=2 in the original document. Then, there will be a bidirectional relation between these vertices according to their locations in the original document. For example, the words (academic) and (libraries) are both adjacent and selected as vertices in the constructed graph, therefore, there will be a bidirectional relation between them as follows:

- (academic) will point to (libraries)
- (libraries) will point to (academic) as shown below:



If the window size is set to 3, the highlighted tokens (vertices of the constructed graph) in yellow in Figure 5.5 are tested again to find if there is any existed vertices within a window size=3 in the original document. Then, there will be a bidirectional relation between these vertices according to their locations in the original document. For example, the words (thin), (client) and (devices) are adjacent and selected as vertices in the constructed graph, so there will be a bidirectional relation between them.

- (thin) will point to (client)
- (client) will point to (thin)
- (thin) will point to (devices)
- (devices) will point to (thin)
- (client) will point to (devices)
- (devices) will point to (client) as shown below:



### 5.3.1.2 Forward Graph

Edges in a forward graph are directional. Every vertex points to its successors according to the used window size and its position in the text. If the window size is set to 2, the highlighted tokens (vertices of the constructed graph) in yellow in Figure 5.5 are tested to find if there is any existed vertices within a window size=2 in the original document. Then, there will be a forward relation between these vertices according to their locations in the original document.

If the used window is set to 3, the highlighted tokens (vertices of the constructed graph) in yellow in Figure 5.5 are tested again to find if there is any existed vertices within a window size=3 in the original document. Then, there will be a forward relation between these vertices according to their locations in the original document. For example, the words (thin), (client) and (devices) are adjacent and selected as vertices in the constructed graph, so there will be a forward relation

between them; (thin) will point to (client), (thin) will point to (devices), and (client) will point to (devices).

### **5.3.1.3 Backward Graph**

Contrary to what was stated in forward graph, in backward graph, every vertex is pointing to its ancestors. However, backward graph is also based on the used window size and the vertex position in the text. If the window size is set to 2, the highlighted tokens (vertices of the constructed graph) in yellow in Figure 5.5 are tested to find if there is any existed vertices within a window size=2 in the original document. Then, there will be a backward relation between these vertices according to their locations in the original document. For example, the words (academic) and (libraries) are both adjacent and selected as vertices in the constructed graph, so (libraries) will point to (academic).

If the used window is set to 3, the highlighted tokens (vertices of the constructed graph) in yellow in Figure 5.5 are tested again to find if there is any existed vertices within a window size=3 in the original document. Then, there will be a backward relation between these vertices according to their locations in the original document. For example, the words (thin), (client) and (devices) are adjacent and selected as vertices in the constructed graph, so there will be a backward relation between them; (client) will point to (thin), (devices) will point to (thin), and (devices) will point to (client).

### **5.3.2 Applying Page Rank Algorithm**

All lexical units that pass the syntactic filter are added to the graph, and edges are drawn between those lexical units that co-occur within a window of words. After the graph is constructed (Bidirectional, Forward or Backward). The score associated with each vertex is set to an initial value of 1. The PageRank Equation 5.1 described in Section 5.2 is run on the graph vertices for several iterations until it converges usually for 20-30 iterations, at an initial value of 0.0001 (Mihalcea & Tarau, 2004). Convergence is achieved when the error rate for any vertex in the graph falls below that threshold value. This error rate is approximated with the difference between two successive scores for graph vertices.

Once the final score is obtained for each vertex in the graph in each document, vertices are sorted in a descending order according to their PageRank scores. The top 30% of the total tokens existed in each document are retained for post-processing steps for keywords and keyphrases extraction in a PageRank list.

### **5.3.3 Post-Processing Steps to Extract Keywords and Keyphrases**

After applying the above syntactic filters that select only lexical units of a certain part of speech, assigning a PageRank score for each vertex in the constructed graph, arranging vertices in a descending order according to their PageRank scores, and choosing the top 30% of the total number of words in the document as keywords, the post-processing stage to extract keywords and keyphrases is detailed as follows:

1. Scan the original document token by token,
2. Add all single tokens found in the text and returned by the PageRank algorithm to a candidate list of keyphrases with the corresponding PageRank score,
3. Add all successive tokens found in the original text as a keyphrase to the keyphrases list with the corresponding PageRank score. This corresponding PageRank score is equal to the summation of tokens PageRank scores constituting that keyphrase,
4. Sort all the returned keywords and keyphrases based on their PageRank scores.

To summarize the above steps, all tokens in the original document are scanned. If there are existed two or more successive keywords, then a keyphrase will be composed of these successive keywords. Otherwise, a keyword of a single word is selected.

### **5.3.4 Weighted Graph**

Vertices are connected by links (edges). These links may be strong or poor depending on several factors. For example, words may occur more than once in the text. Or there exist some words have greater PageRank scores pointing to other words. These links between these words are stronger, which means a high PageRank score may be obtained.

The strength of the connection between two vertices  $V_i$  and  $V_j$  is expressed as a weight  $W_{ij}$  for such relations. This weight can be added to the link between two vertices to raise the PageRank score (Mihalcea & Tarau, 2004). Therefore, a new formula of PageRank equation that takes the algorithm is formed to compute ranks by taking into account the weight as shown in Equation 5.3 below.

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ij}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j) \quad (5.3)$$

To illustrate Equation 5.3, suppose the links between two words are as shown in Figure 5.7. The relation in **A** can be considered stronger than the relation in **B**, Edge weight can be taken into account as a measure to compute the PageRank score associated with each vertex in the graph.

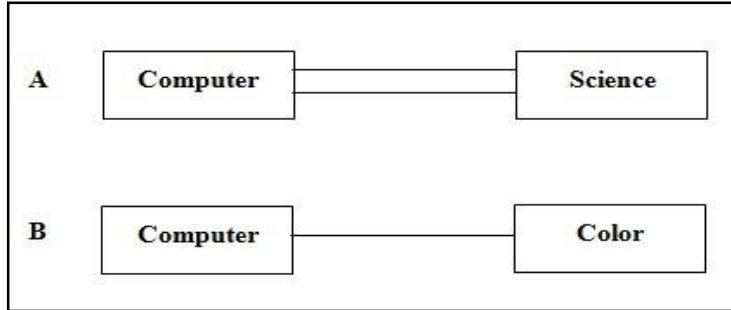


Figure 5.7 Example on Weighted PageRank Scores

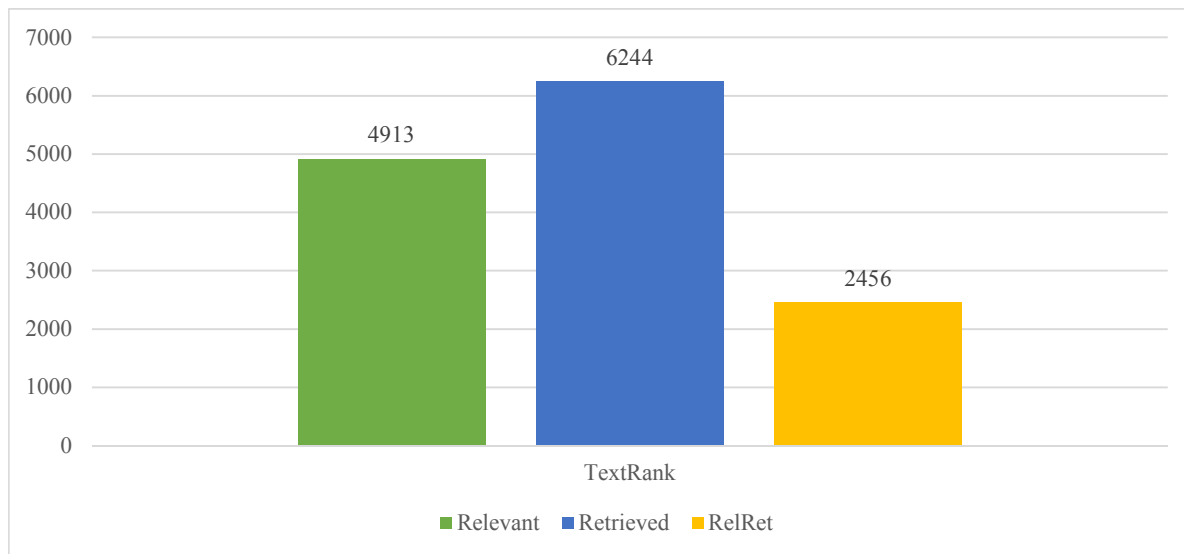
## 5.4 Evaluating efficiency

The TextRank model was tested on the Hulth 2003 dataset, and the results are evaluated using precision, recall, and F-measure, and it achieved 0.4324 of F-measure as shown in the Table 5.2.

*Table 5.2 TextRank results for recall, precision, and F-measure*

	Recall	Precision	F-measure
<b>TextRank</b>	0.5044	0.4081	0.4324

Figure 5.8 shows number of the relevant keywords, retrieved keywords, and the number of relevant retrieved keywords by TextRank.



*Figure 5.8 Number of Relevant, Retrieved, RelRet document by TextRank*

# Chapter 6

## N-Grams Model

### 6.1 Introduction

Based on adding linguistic knowledge to the representation, rather than relying only on statistics, Hulth (Hulth A. , 2003) used different methods of incorporating linguistics into keywords extraction.

Automatic keywords extraction should apply the power and speed of computation to the problems of access and discoverability. Adding value to information retrieval without the significant costs and drawbacks associated with human indexers (Hulth A. , 2003).

### 6.2 Extracting keywords

Keyword extraction is very important as it helps a lot to understand and to know what subject is being talked about. This section details the implementation of keywords and keyphrases extraction system as shown in Figure 6.1.

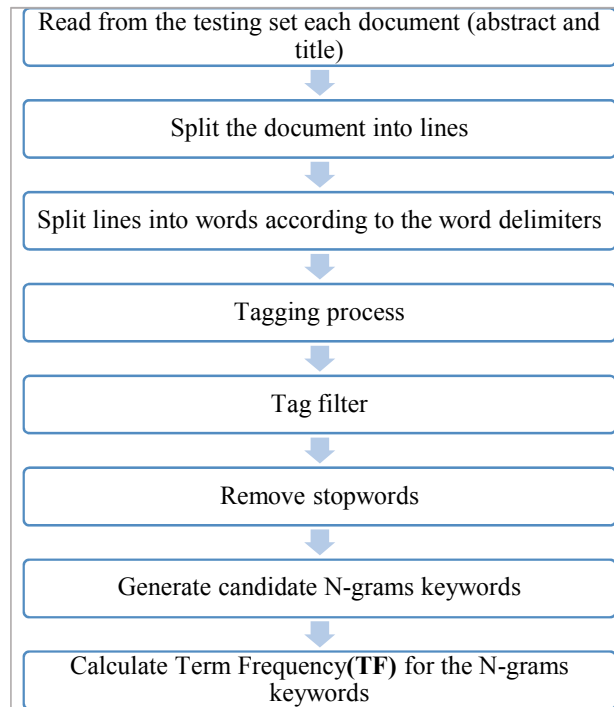


Figure 6.1 General structure of keyword extraction using N-grams model

In this model, after reading the text document the first step is to tag each word using the Penn Treebank POS tagset. Figure 6.2 shows a text document which is an example of the text before the tagging process and every token in the document is tagged as shown in Figure 6.3. The second step is to generate the candidate N-grams keywords, this process is based on the following three stages: extract keywords for all the text documents, stopwords removal, and the selection of a tag filter use two sets of tagged keywords. Third step is to calculate term frequency (**TF**) for each final candidate N-grams keywords. Finally sorting keywords based on their *TF* in an ascending order, in order to select the final keywords.



Analyzing the benefits of 300 mm conveyor-based AMHS.

While the need for automation in 300 mm fabs is not debated, the form and performance of such automation is still in question. Software simulation that compares conveyor-based continuous flow transport technology to conventional car-based wafer-lot delivery has detailed delivery time and throughput advantages to the former.

Figure 6.2 Example of a text document consists of an abstract and title

Analyzing \_VBG the \_DT benefits \_NNS of \_IN 300 \_CD mm \_NN conveyor-based \_JJ AMHS \_NN.

While \_IN the \_DT need \_NN for \_IN automation \_NN in \_IN 300 \_CD mm \_NN fabs \_NNS is \_VBZ not \_RB debated \_VBN, the \_DT form \_NN and \_CC performance \_NN of \_IN such \_JJ automation \_NN is \_VBZ still \_RB in \_IN question \_NN. Software \_NNP simulation \_NN that \_WDT compares \_VBZ conveyor-based \_JJ continuous \_JJ flow \_NN transport \_NN technology \_NN to \_TO conventional \_JJ car-based \_JJ wafer-lot \_NN delivery \_NN has \_VBZ detailed \_VBN delivery \_NN time \_NN and \_CC throughput \_NN advantages \_NNS to \_TO the \_DT former \_JJ

Figure 6.3 Tagged Text document

After the document is tagged, it is ready to be moved into the second step to generate the candidate N-grams keywords based on the three stages which are clearly described below:

**Stage1.** Extract keywords which is done as follows:

Assume that number of keywords extracted is: ( $K$ ), and number of words in the document is: ( $W$ ). In this model five sequences (Unigram, Bigram, Trigram, Quadgram and 5-gram) were selected. A different number of keywords from keyword sequences in the experiments were taken.

A. Unigram (1-gram sequence):  $K(W_n)$  one word. For example, some of the Unigram words from Figure 6.3 are as shown in Table 6.1 below.

Table 6.1 Unigram Keywords

<b>W<sub>n</sub>, One word</b>	<b>W<sub>n</sub>, One word</b>	<b>W<sub>n</sub>, One word</b>	<b>W<sub>n</sub>, One word</b>	<b>W<sub>n</sub>, One word</b>
Analyzing_VBG	for_IN	performance_NN	conveyor-based_JJ	delivery_NN
the_DT	automation_NN	of_IN	continuous_JJ	time_NN
benefits_NNS	in_IN	automation_NN	flow_NN	and_CC
of_IN	mm_NN	is_VBZ	transport_NN	throughput_NN
300_CD	fabs_NNS	still_RB	technology_NN	advantages_NNS

B. Bigram (2-gram sequence):  $K(W_n, W_{n+1})$  two words. As shown below in Table 6.2.

Table 6.2 Bigram Keywords

<b>W<sub>n</sub> - W<sub>n+1</sub>, Two words</b>	<b>W<sub>n</sub> - W<sub>n+1</sub>, Two words</b>	<b>W<sub>n</sub> - W<sub>n+1</sub>, Two words</b>
Analyzing_VBG the_DT	not_RB debated_VBN	conveyor-based_JJ continuous_JJ
the_DT benefits_NNS	debated_VBN ,_	continuous_JJ flow_NN
benefits_NNS of_IN	,_ the_DT	flow_NN transport_NN
of_IN 300_CD	the_DT form_NN	transport_NN technology_NN
300_CD mm_NN	form_NN and_CC	technology_NN to_TO
mm_NN conveyor-based_JJ	and_CC performance_NN	to_TO conventional_JJ

C. Trigram (3-gram sequence):  $K(W_n, W_{n+1}, W_{n+2})$  three words. It will be as shown above in the Bigram model but for three words

D. Quadgram (4-gram sequence):  $K(W_n, W_{n+1}, W_{n+2}, W_{n+3})$  four words.

E. 5-gram (5-gram sequence):  $K(W_n, W_{n+1}, W_{n+2}, W_{n+3}, W_{n+4})$  Five words.

**Stage2.** Remove the stopwords, the stopwords are included in Appendix B. The candidate N-grams keywords are selected as follows:

- A. Remove  $K(W_n)$  from Unigram (1-gram sequence), where  $W_n$  is a stopword. For example, some of the Unigram keywords after stopwords removal from Table 6.1 are shown in Table 6.3.

Table 6.3 Unigram Keywords after Stopword removal

$W_n$ , One words	$W_n$ , One words	$W_n$ , One words	$W_n$ , One words
Analyzing_VBG	debated_VBN	continuous_JJ	delivery_NN
benefits_NNS	form_NN	flow_NN	time_NN
300_CD	performance_NN	transport_NN	throughput_NN
mm_NN	automation_NN	technology_NN	advantages_NNS

- B. Remove  $K(W_n, W_{n+1})$  from Bigram (2-gram sequence), where also  $W_n$  or  $W_{n+1}$  is a stopword. For example, some of the Bigram keywords after stopwords removal from Table 6.2 are shown in Table 6.4 below.

Table 6.4 Bigram Keywords after Stopword removal

$W_n, W_{n+1}$ , Two words	$W_n, W_{n+1}$ , Two words	$W_n, W_{n+1}$ , Two words
300_CD mm_NN	compares_VBZ conveyor-based_JJ	car-based_JJ wafer-lot_NN
mm_NN conveyor-based_JJ	conveyor-based_JJ continuous_JJ	wafer-lot_NN delivery_NN
conveyor-based_JJ AMHS_NN	continuous_JJ flow_NN	detailed_VBN delivery_NN
300_CD mm_NN	flow_NN transport_NN	delivery_NN time_NN
mm_NN fabs_NNS	transport_NN technology_NN	throughput_NN advantages_NNS
Software_NNP simulation_NN	conventional_JJ car-based_JJ	

- C. Remove  $K(W_n, W_{n+1}, W_{n+2})$  from Trigram (3-gram sequence), where  $W_n$ ,  $W_{n+1}$  or  $W_{n+2}$  is a stopword, the same process as in the Unigram and Bigram models.
- D. Remove  $K(W_n, W_{n+1}, W_{n+2}, W_{n+3})$  from Quadgram (4-gram sequence), here  $W_n$ ,  $W_{n+1}$ ,  $W_{n+2}$  or  $W_{n+3}$  is a stopword.
- E. Remove  $K(W_n, W_{n+1}, W_{n+2}, W_{n+3}, W_{n+4})$  from 5-gram (5-gram sequence), where  $W_n$ ,  $W_{n+1}$ ,  $W_{n+2}$ ,  $W_{n+3}$  or  $W_{n+4}$  is a stopword.

**Stage3.** Tag filtering based on some rules as shown below:

A. Use two sets **A1** and **A2** because there are no keywords containing other tags. The last word in any N-grams must be from set **A2** because there are no keywords ending other tags.

**A1** = {NN, NNS, NNP, NNPS, JJ, JJR, JJS}.

**A2** = {NN, NNS, NNP, NNPS}.

B. In case of Unigram sequence  $K(W_n)$ , the sequence is selected if  $W_n$  belongs to **A2** and must be three character or more. For example, the Unigram keywords after tag filter are shown in Table 6.5.

Table 6.5 Unigram Keywords after tag filter

<b>W<sub>n</sub>, One words</b>	<b>W<sub>n</sub>, One words</b>	<b>W<sub>n</sub>, One words</b>	<b>W<sub>n</sub>, One words</b>
benefits_NNS	automation_NN	technology_NN	advantages_NNS
AMHS_NN	question_NN	wafer-lot_NN	
automation_NN	Software_NNP	delivery_NN	
fabs_NNS	simulation_NN	delivery_NN	
form_NN	flow_NN	time_NN	
performance_NN	transport_NN	throughput_NN	

C. In case of Bigram sequence  $K(W_n, W_{n+1})$ , the sequence is selected if  $W_{n+1}$  belongs to **A2**,  $W_n$  belongs to **A1** and  $W_n, W_{n+1}$  must be three character or as shown in Table 6.6.

Table 6.6 Bigram Keywords after tag filter

<b>W<sub>n</sub>, W<sub>n+1</sub>, Two words</b>	<b>W<sub>n</sub>, W<sub>n+1</sub>, Two words</b>	<b>W<sub>n</sub>, W<sub>n+1</sub>, Two words</b>
conveyor-based_JJ AMHS_NN	transport_NN technology_NN	throughput_NN advantages_NNS
Software_NNP simulation_NN	car-based_JJ wafer-lot_NN	
continuous_JJ flow_NN	wafer-lot_NN delivery_NN	
flow_NN transport_NN	delivery_NN time_NN	

- D. In Trigram sequence  $K(W_n, W_{n+1}, W_{n+2})$ , the sequence is selected if  $W_{n+2}$  belongs to **A2**,  $W_n, W_{n+1}$  belongs to **A1** and  $W_n, W_{n+1}, W_{n+2}$  must be three character or more.
- E. In case of Quadgram sequence  $K(W_n, W_{n+1}, W_{n+2}, W_{n+3})$ , the sequence is selected if  $W_{n+3}$  belongs to **A2**,  $W_n, W_{n+1}, W_{n+2}$  belongs to **A1** and  $W_n, W_{n+1}, W_{n+2}, W_{n+3}$  must be three character or more.
- F. In case of 5-gram sequence  $K(W_n, W_{n+1}, W_{n+2}, W_{n+3}, W_{n+4})$  the sequence is selected if  $W_{n+4}$  belongs to **A2**,  $W_n, W_{n+1}, W_{n+2}, W_{n+3}$  belongs to **A1** and  $W_n, W_{n+1}, W_{n+2}, W_{n+3}, W_{n+4}$  must be three character or more. And in this example all 5-gram keywords are deleted because All Keywords isn't belongs to **A2**.
- G. Delete all keywords Unigram-keywords if exist in Bigram, Trigram, Quadgram and 5-gram keywords. For example, the Unigram Keywords after deleting them are shown in Table 6.7.

Table 6.7 Final Unigram Keywords

<b>W<sub>n</sub>, One words</b>	<b>W<sub>n</sub>, One words</b>
benefits_NNS	performance_NN
fabs_NNS	question_NN
form_NN	

- H. Also, delete all keywords Bigram-keywords if exist in other N-grams keywords. For example, the Bigram Keywords after deleting them are shown in Table 6.8.

Table 6.8 Final Bigram Keywords

<b>W<sub>n</sub>, W<sub>n+1</sub>, Two words</b>	<b>W<sub>n</sub>, W<sub>n+1</sub>, Two words</b>
conveyor-based_JJ AMHS_NN	delivery_NN time_NN
Software_NNP simulation_NN	throughput_NN advantages_NNS

- I. Delete all keywords Trigram-keywords if exist in the other N-grams, and in this example all keywords are deleted.

- J. Finally, delete all keywords Quadgram keywords if exist in 5-gram keywords. For example, the Quadgram Keywords after deleting them are shown in Table 6.9.

Table 6.9 Final Quadgram Keywords

$W_n, W_{n+1}, W_{n+2}, W_{n+3}$ , Four words
conveyor-based_JJ continuous_JJ flow_NN transport_NN
continuous_JJ flow_NN transport_NN technology_NN
conventional_JJ car-based_JJ wafer-lot_NN delivery_NN

## 6.3 Calculate Term Frequency

This section represent the third step of the N-grams model, which concerned with calculate term frequency (*TF*) for each final candidate N-grams keywords. All frequencies (*TF*) will be calculated as follows:

a. Keywords Unigram =  $TF(W_n)$ . (6.1)

b. Keywords Bigram =  $TF(W_n + W_{n+1})$ . (6.2)

c. Keywords Trigram =  $TF(W_n + W_{n+1} + W_{n+2})$ . (6.3)

d. Keywords Quadgram =  $TF(W_n + W_{n+1} + W_{n+2} + W_{n+3})$ . (6.4)

e. Keywords 5-gram =  $TF(W_n + W_{n+1} + W_{n+2} + W_{n+3} + W_{n+4})$ . (6.5)

Then Keywords are sorted based on term frequency (*TF*) in an ascending order, as shown in Table 6.10.

Table 6.10 Keywords Frequency (*TF*).

Keywords	$TF = \sum_{n=1}^5 TF(W_n)$
conventional_JJ car-based_JJ wafer-lot_NN delivery_NN	5 = 1+1+1+2

conveyor-based_JJ continuous_JJ flow_NN transport_NN	5 = 2+1+1+1
continuous_JJ flow_NN transport_NN technology_NN	4 = 1+1+1+1
conveyor-based_JJ AMHS_NN	3 = 2+1
delivery_NN time_NN	3 = 2+1
throughput_NN advantages_NNS	2 = 1+1
Software_NNP simulation_NN	2 = 1+1
benefits_NNS	1 = 1
fabs_NNS	1 = 1
form_NN	1 = 1
performance_NN	1 = 1
question_NN	1 = 1
Total <i>TF</i> of all Keywords	29

After the above process is done, and in order to select keywords, the number of keywords selected from each sequences is calculated as follows:

1. Calculate the Weight (***Wt***) of each sequence. This weight is calculated as shown below:

$$\bullet \text{ } Wt \text{ of Unigram } (Wtu) = \frac{\sum \text{ unigrams Keywords in dataset}}{\sum \text{ } N\text{-Grams Keywords in dataset}} \quad (6.6)$$

$$\bullet \text{ } Wt \text{ of Bigram } (Wtb) = \frac{\sum \text{ Bigram Keywords in dataset}}{\sum \text{ } N\text{-Grams Keywords in dataset}} \quad (6.7)$$

$$\bullet \text{ } Wt \text{ of Trigram } (Wtt) = \frac{\sum \text{ Trigram Keywords in dataset}}{\sum \text{ } N\text{-Grams Keywords in dataset}} \quad (6.8)$$

$$\bullet \text{ } Wt \text{ of Quadrigram } (Wtq) = \frac{\sum \text{ Quadrigram Keywords in dataset}}{\sum \text{ } N\text{-Grams Keywords in dataset}} \quad (6.9)$$

$$\bullet \text{ } Wt \text{ of 5 - gram } (Wt5) = \frac{\sum \text{ 5-gram Keywords in dataset}}{\sum \text{ } N\text{-Grams Keywords in dataset}} \quad (6.10)$$

2. Calculate the Number keywords to be selected from all sequences (Unigram, Trigram, Quadgram, 5-gram) as follows:

$$\bullet \text{ Keywords from Unigram } (NKI) = \sum \text{ unigrams Keywords} * Wtu. \quad (6.11)$$

- Keywords from Bigram ( $NK2$ ) =  $\sum unigrams\ Keywords * Wtb.$   
(6.12)

- Keywords from Trigram ( $NK3$ ) =  $\sum unigrams\ Keywords * Wtt.$   
(6.13)

- Keywords from Quadgram ( $NK4$ ) =  $\sum unigrams\ Keywords * Wtq.$   
(6.14)

- Keywords from 5-gram ( $NK5$ ) =  $\sum unigrams\ Keywords * Wt5.$   
(6.15)

Finally the extracted keyphrases is listed in a text file. The keyphrases are shown in Figure 6.4.

```
conventional car-based wafer-lot delivery
conveyor-based continuous flow transport
continuous flow transport technology
conveyor-based amhs
throughput advantages
delivery time
software simulation
```

*Figure 6.4 Keyphrases Text File*

## 6.4 Evaluating efficiency

The N-grams model was tested on the Hulth 2003 dataset, and the results are evaluated using precision, recall, and F-measure, and it achieved 0.4233 of F-measure as shown in Table 6.11.



Table 6.11 N-grams results for recall, precision, and F-measure

	Recall	Precision	F-measure
<b>N-grams</b>	0.4832	0.4048	0.4233

Figure 6.5 shows number of the relevant keywords, retrieved keywords, and the number of relevant retrieved keywords by N-grams.

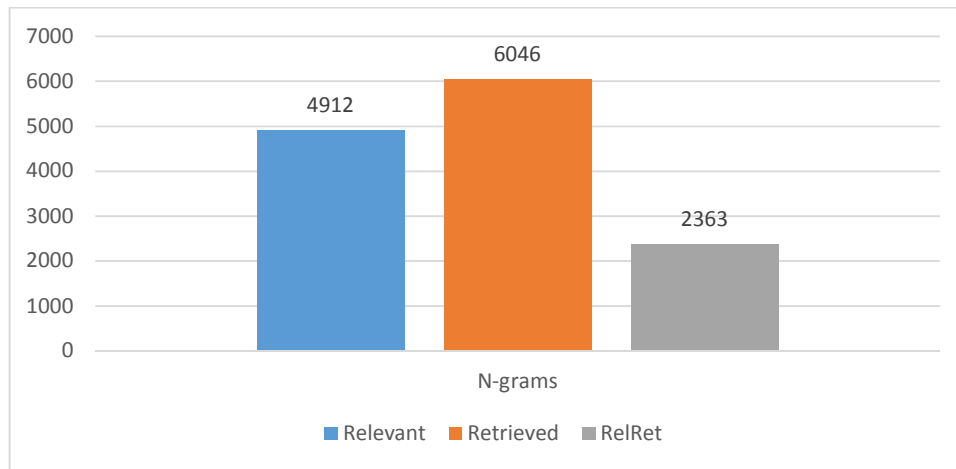


Figure 6.5 Number of Relevant, Retrieved, RelRet document by N-grams

# Chapter 7

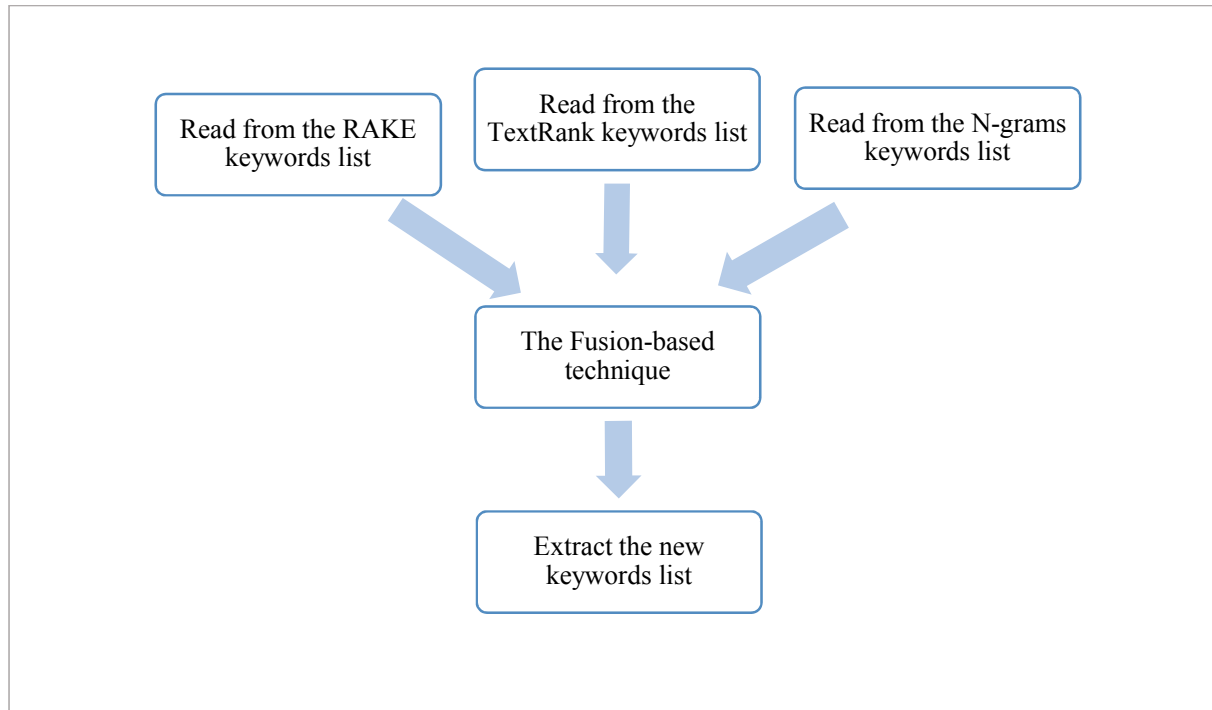
## Fusion-based Technique

### 7.1 Introduction

This chapter uses several fusion-based techniques. Fusion-based techniques extract keywords by combining the final results that are achieved from the three other techniques; RAKE, TextRank, and N-grams. Combining different knowledge sources has proved successful in improving the performance of individual sources on several NLP tasks (some of them are closely related to or involved in term extraction), such as context-sensitive spelling correction, POS tagging, parsing, text classification and filtering, etc. Through this research, few different fusion-based techniques were experimented in order to produce more accurate keywords which will lead to higher results than other techniques mentioned above.

The following sections explain the techniques that are used. The techniques are most likely to work based on voting procedures. The first technique is very simple. The technique is called All Keyword Fusion (Base method). It is based on term frequency. In order to achieve better results, the second technique was used. It is called Majority Voting. The second technique is based on document frequency. The third technique's strategy is different from the previous two techniques as it is concerned with the term weight instead of document frequency and term frequency according to Borda count voting procedure (Montague & Aslam, 2002). This technique is called Borda Voting. When the third technique was implemented, some candidate keywords have the same weight which created a problem in extracting an accurate keyword list. The fourth technique, which is called CombMNZ, is based on a combination of term weight and document frequency. In the fifth technique we tried to enhance the CombMNZ technique. We presented the WCombMNZ that give each technique RAKE, TextRank, and N-grams weight. Finally, to solve the problem that was mentioned in the third technique, we used the Condorcet voting algorithm (Montague & Aslam, 2002) in the sixth technique. This technique is called Condorcet.

Figure 7.1 below shows the general structure of the fusion-based techniques.



*Figure 7.1 General structure of the fusion-based technique*

For more clarification, we used an example from the results that had been extracted before using the three techniques, which are RAKE, TextRank, and N-grams which refers to an abstract from Hulth 2003 dataset. The abstract named (2047.ABSTR). Table 7.1 shows keywords extracted by the three techniques mentioned above exactly as sorted in each technique list.

Table 7.1 Extracted keywords by RAKE, TextRank, and N-grams

Keywords extracted by RAKE	Keywords extracted by TextRank	Keywords extracted by N-grams
delivering management science/operations research	management science/operations research	traditional pert/cpm algorithm
traditional pert/cpm algorithm	traditional pert/cpm algorithm	spreadsheet environment
generalized pert/cpm implementation	pert/cpm implementation	project network
paper describes	spreadsheet environment	management science/operations research
critical path	critical path	generalized pert/cpm implementation
project network	project network	critical path
spreadsheet environment		
implementation		
spreadsheet		

## 7.2 Fusion-Based Technique

We implemented fusion-based techniques in order to generate more accurate keyword lists to automatically extract keywords to achieve higher results than other techniques. The techniques have been implemented in Perl programming language. The fusion-based techniques work by combining the results that consist of the keywords lists from the other three techniques. In this research, as mentioned above, several techniques have been implemented and discussed in the following sections.

### 7.2.1 All Keywords Fusion Technique (Base method)

As mentioned in the introduction, this technique is simple; it is only concerned with keyword frequency. First, it reads the three previous techniques' keywords lists (RAKE,

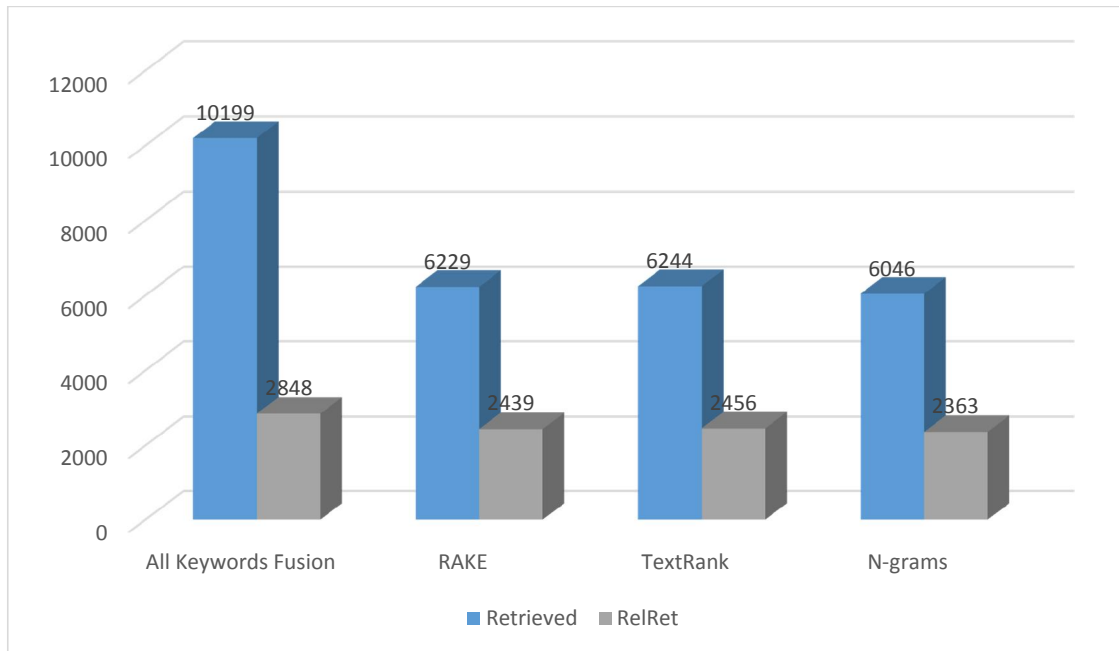
TextRank, and N-grams) and based on the keyword frequency, it extracts all keywords that occurred in the three abovementioned lists and creates a new list accordingly. Table 7.2 shows the extracted keywords by All Keywords Fusion technique for the example above.

*Table 7.2 The extracted keywords by All Keywords Fusion technique*

<b>Keywords extracted by All Keywords Fusion</b>
critical path
spreadsheet environment
traditional pert/cpm algorithm
project network
generalized pert/cpm implementation
management science/operations research
delivering management science/operations research
paper describes
implementation
spreadsheet
pert/cpm implementation

The results are evaluated using precision, recall, and F-measure. This technique achieved a recall value of 0.5825, a precision value of 0.292, and F-measure value of 0.3746.

Figure 7.2 shows a comparison between All Keywords Fusion technique and the three other techniques based on number of the relevant keywords, retrieved keywords, and the number of relevant retrieved keywords.



*Figure 7.2 Comparison between All Keywords Fusion technique and the three other techniques*

We noticed from Figure 7.2 that after combining the three lists, the number of retrieved keywords is significantly higher than the numbers of the other three techniques. This is because we retrieved all keywords in the three lists, which led to a higher recall value and a lower precision value that led to a lower F-measure value.

Figure 7.3 shows comparison between All Keywords Fusion and the three other techniques based on recall, precision, and F-measure.

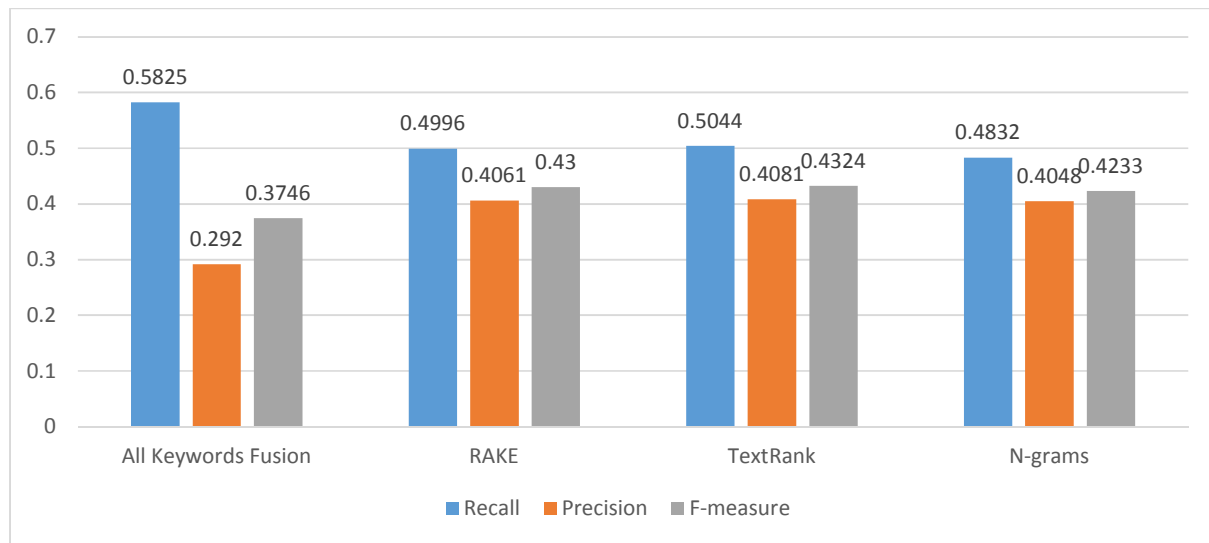


Figure 7.3 Comparison between All Keywords Fusion technique and the three other techniques based on recall, precision, and F-measure

## 7.2.2 Majority Voting Technique

Similar to the All Keywords Fusion, Majority Voting technique also reads the results from the first three techniques, RAKE, TextRank, and N-grams. However, in contrast to All Keywords Fusion technique, this technique is implemented based on the document frequency for each keyword. After reading the keyword results from the first three techniques, Majority Voting technique creates a new keyword list only if the document frequency of the keyword is two or more. Table 7.3 shows the extracted keywords by Majority Voting approach with their document frequency.

Table 7.3 The extracted keywords by Majority Voting technique

Keywords extracted by Majority Voting	DF
critical path	3
spreadsheet environment	3
traditional pert/cpm algorithm	3
project network	2
generalized pert/cpm implementation	2
management science/operations research	2

The results are evaluated using precision, recall, and F-measure. This technique achieved significantly better results. The technique achieved a recall value of 0.5048, a precision value of 0.4181, and F-measure value of 0.4382. Figure 7.4 shows a comparison between Majority Voting and the three other techniques based on recall, precision, and F-measure.

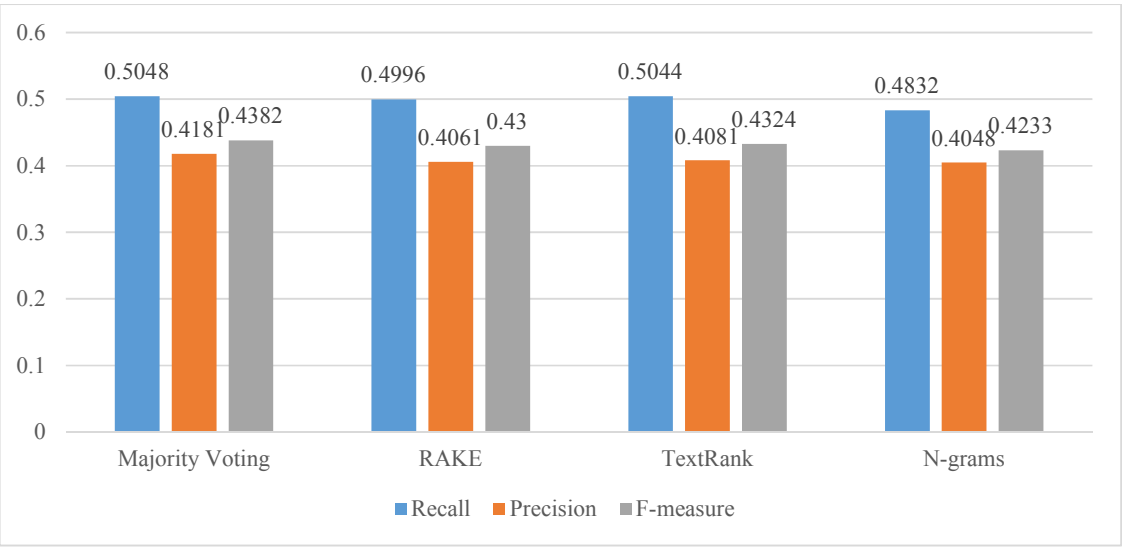


Figure 7.4 Comparison between Majority Voting and the three other techniques based on recall, precision, and F-measure

Figure 7.4 shows that F-measure has improved from the All Keywords Fusion technique, because it retrieved a smaller number of keywords. Consequently, we achieved a lower recall value and a higher precision value than the All Keywords Fusion technique.

Figure 7.5 compares Majority Voting techniques with the other three techniques based on the number of the relevant keywords, retrieved keywords, and the number of relevant retrieved keywords.



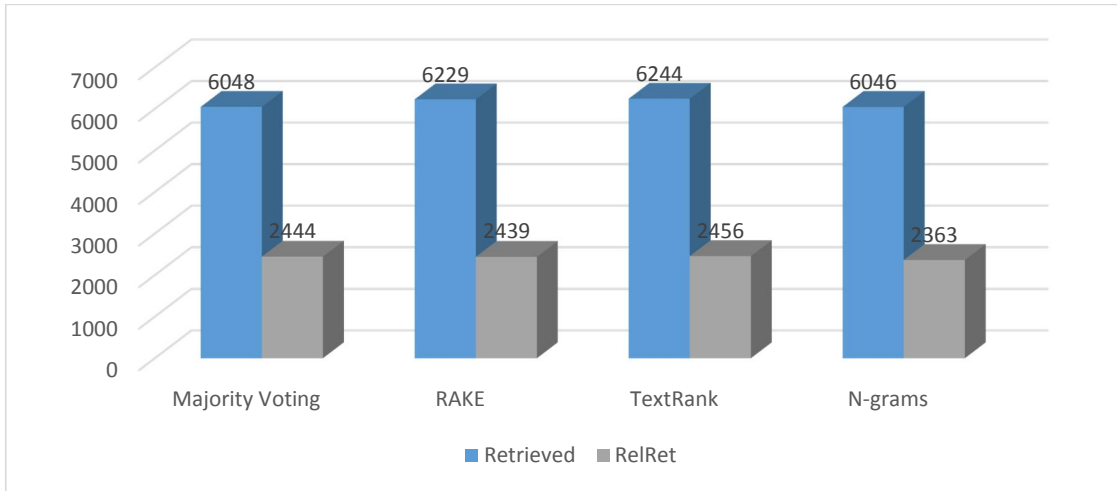


Figure 7.5 Comparison between Majority Voting technique and the three other techniques

### 7.2.3 Borda Voting Technique

This technique takes the term weight in consideration based on the Borda count voting procedure. In the Borda count voting procedure, for each candidate keyword, the top candidate keyword receives  $n$  points (if there are  $n$  candidates in the list), the second candidate keyword receives  $n-1$  points, and so on. The candidate keyword with the most points wins and is included in the new candidate keyword list (Montague & Aslam, 2002). Then, the highest  $N$  keywords are extracted as the new list, where  $N$  is the average number of keywords retrieved from the three lists. The general idea is to award points to candidates based on preference schedule, then declare the winner to be the candidate with the most points. Figure 7.6 shows a simple example of how the Borda method works.

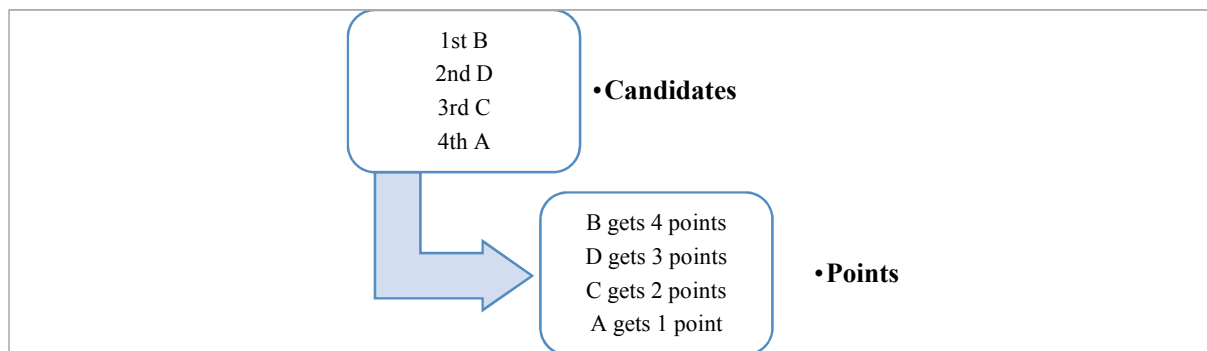


Figure 7.6 Example shows the process of Borda count method

For our example, Table 7.4 shows the keywords for each list with the assigned weight and the new list with the scores after the summation process.

*Table 7.4 Keywords with their scores*

<b>Keyword</b>	<b>RAKE Score</b>	<b>TextRank Score</b>	<b>N-grams Score</b>	<b>Final Score</b>
critical path	5	2	1	8
spreadsheet environment	3	3	5	11
traditional pert/cpm algorithm	8	5	6	19
project network	4	1	4	9
generalized pert/cpm implementation	7	0	2	9
management science/operations research	0	6	3	9
delivering management science/operations research	9	0	0	9
paper describes	6	0	0	6
implementation	2	0	0	2
spreadsheet	1	0	0	1
pert/cpm implementation	0	4	0	4

During the process of the technique implementation, we faced an issue that limited the technique from accurately extracting keywords; some candidate keywords have the same weight which leads to an inaccurate keyword list. Table 7.5 shows the extracted keywords by Borda Voting technique with their corresponding scores.

*Table 7.5 The extracted keywords by Borda Voting technique*

<b>Keywords extracted by Borda Voting</b>	<b>Score</b>
traditional pert/cpm algorithm	19
spreadsheet environment	11
project network	9
management science/operations research	9
delivering management science/operations research	9
generalized pert/cpm implementation	9

The results are evaluated using precision, recall, and F-measure. This technique achieved slightly worse results than the previous two fusion-based techniques. This technique achieved a recall value of 0.5012, a precision value of 0.4054, and F-measure value of 0.4294. Figure 7.7 shows a comparison between Borda Voting and the three other techniques based on recall, precision, and F-measure.

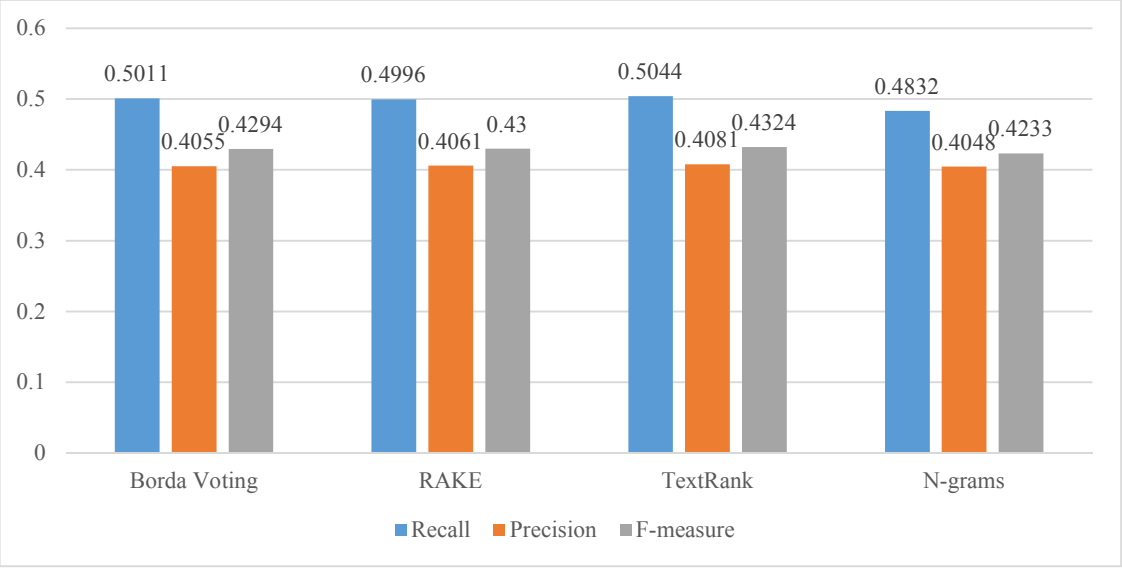


Figure 7.7 Comparison between Borda Voting and the three other techniques based on recall, precision, and F-measure

According to Figure 7.7, we noticed that F-measure is slightly lower than the Majority Voting technique, because some keywords have higher weights based on their position in the list, but they do not exist in other lists. This means that these words may not be important but because they achieved high weights, they were considered as keywords. For example, the keyword “delivering management science/operations research” has a weight of ‘9’ but it occurred in only one list. To solve this problem, the following technique was implemented.

Figure 7.8 compares Borda Voting techniques with the other three techniques based on the number of the retrieved keywords, and the number of relevant retrieved keywords.

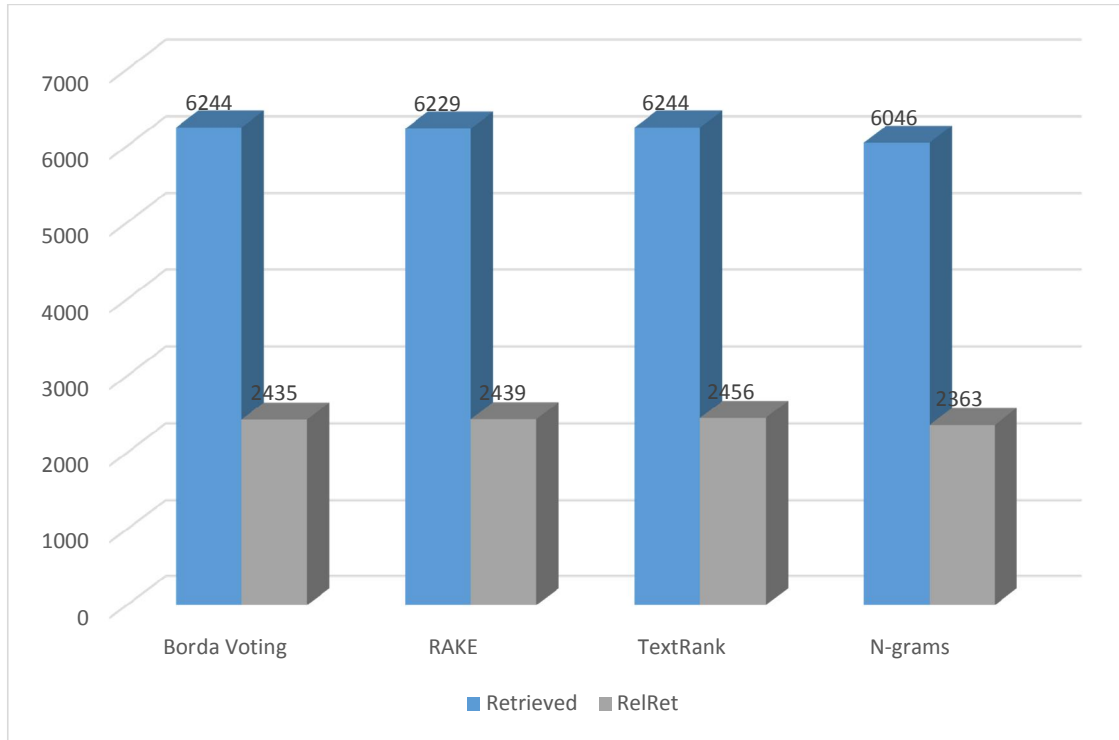


Figure 7.8 Comparison between Borda Voting technique and the three other techniques

### 7.2.4 CombMNZ Technique

CombMNZ is implemented by combining both the list frequency and the keyword weight which was used in Borda technique. This technique works upon the CombMNZ algorithm (Montague & Aslam, 2002). In CombMNZ algorithm, it sums up the weight of the retrieved keywords, then the sum is multiplied by the document frequency. Finally, the keywords are ranked in an ascending order based on their final scores. As in Borda Voting technique, the highest  $N$  keyword scores are extracted as the new list, where  $N$  is the average number of keywords retrieved from the three lists. Table 7.6 shows the keywords' scores and the document frequency for each keyword, and the new scores.

Table 7.6 Keyword's scores \* DF

Keyword	Score	DF	Score * DF
critical path	8	3	24
spreadsheet environment	11	3	33
traditional pert/cpm algorithm	19	3	57
project network	9	3	27
generalized pert/cpm implementation	9	2	18
management science/operations research	9	2	18
delivering management science/operations research	9	1	9
paper describes	6	1	6
implementation	2	1	2
spreadsheet	1	1	1
pert/cpm implementation	4	1	4

After extracting the highest  $N$  keywords, Table 7.7 shows the extracted keywords by CombMNZ technique with their new scores.

Table 7.7 The extracted keywords by CombMNZ technique

Keywords extracted by CombMNZ technique	Score
traditional pert/cpm algorithm	57
spreadsheet environment	33
project network	27
critical path	24
management science/operations research	18
generalized pert/cpm implementation	18

The results are evaluated using precision, recall, and F-measure. CombMNZ technique produced relatively better results compared to the first three techniques. This technique achieved a recall value of 0.5095, a precision value of 0.4128, and F-measure value of 0.4372. Based on our example, we noticed that the keyword “delivering management science/operations research” with the corresponding weight of ‘9’ does not exist in the extracted keyword list by CombMNZ because when multiplying the keyword score by the document frequency, the less fortunate keywords that

didn't appear in the list of the previous technique is now more fortunate to appear in the list of this technique. Instead, the keyword “critical path” existed in the list of this technique even though its weight was ‘8’. This happened because the keyword “critical path” has already existed in the three original techniques’ keyword lists. This keyword has a document frequency of ‘3’. Therefore, after multiplying its score weight by its document frequency, the keyword’s new weight is now ‘24’. Figure 7.9 shows a comparison between CombMNZ technique and the three other techniques based on recall, precision, and F-measure.

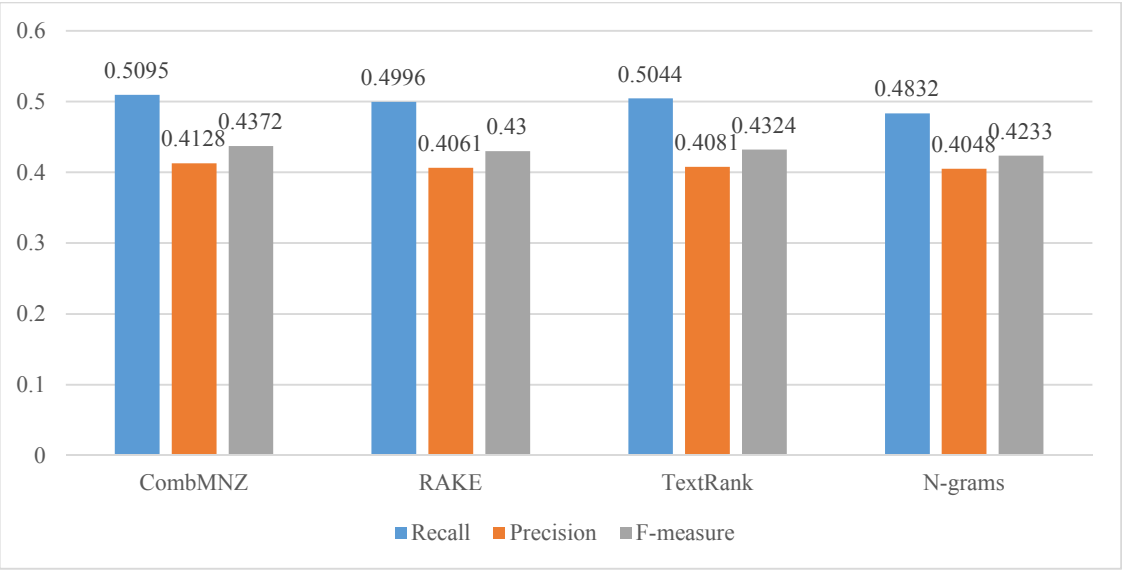


Figure 7.9 Comparison between CombMNZ technique and the three other techniques based on recall, precision, and F-measure

Consequently, CombMNZ technique achieved a larger number of relevant retrieved keywords than the number of relevant retrieved keywords by Borda Voting technique. As a result, this technique achieved a higher recall value and a higher precision value which led to a higher F-measure value. Figure 7.10 compares CombMNZ techniques with the other three techniques based on the number of the retrieved keywords, and the number of relevant retrieved keywords.

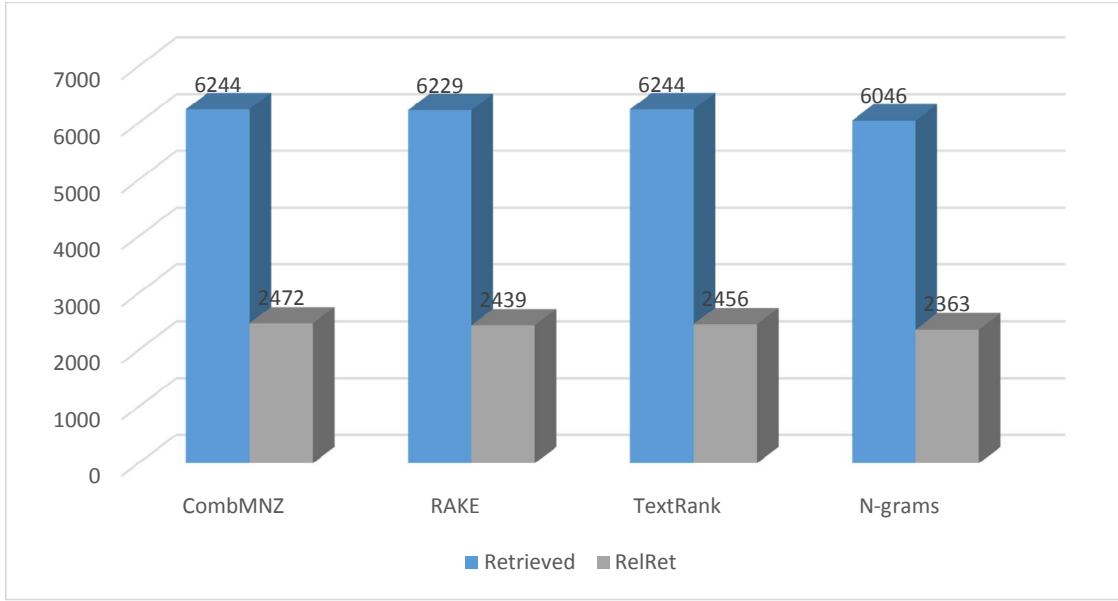


Figure 7.10 Comparison between CombMNZ technique and the three other techniques

### 7.2.5 WCombMNZ Technique

WCombMNZ is implemented similarly to the CombMNZ technique where the document frequency and term weight are considered in extracting the keyword list. Additionally, a new weight is assigned to each technique, RAKE, TextRank, and N-grams according to their performance in extracting keywords, where the sum of the weights of the three techniques is ‘1’ as follows:

$$\bullet \quad W_{(RAKE)} + W_{(TextRank)} + W_{(N-grams)} = 1 \quad (7.1)$$

As mentioned above, the weights were assigned to each technique based on their performance. For more clarification and after doing many calculations, RAKE achieved F-measure value of 0.43 so, we assigned a weight of 0.25. TextRank achieved F-measure value of 0.4324. Therefore, we assigned a weight of 0.55. Lastly, N-grams achieved F-measure value 0.4233. As a result, we assigned a weight of 0.2. Then, the original weight of the keyword is multiplied by the corresponding new assigned weight for each technique. The keywords are then sorted according to their new weights in an ascending order. Finally, the highest  $N$  keyword scores are extracted as the new list, where  $N$  is the average number of keywords retrieved from the three lists. Table 7.8

shows the keywords' scores, the document frequency for each keyword, and the new scores after multiplying the weight by the document frequency and the technique weight.

*Table 7.8 keywords scores after multiplying by the techniques weight*

<b>Keyword</b>	<b>Score</b>	<b>DF</b>	<b>Score * DF * <math>W_{(technique)}</math></b>
critical path	8	3	2.35
spreadsheet environment	11	3	3.5
traditional pert/cpm algorithm	19	3	5.85
project network	9	3	2.35
generalized pert/cpm implementation	9	2	1.9
management science/operations research	9	2	4.05
delivering management science/operations research	9	1	1.8
paper describes	6	1	1.2
implementation	2	1	0.4
spreadsheet	1	1	0.2
pert/cpm implementation	4	1	2.2

After extracting the highest  $N$  keywords, Table 7.9 shows the extracted keywords by WCombMNZ technique with their new scores.

*Table 7.9 The extracted keywords by WCombMNZ technique*

<b>Keywords extracted by WCombMNZ technique</b>	<b>Score</b>
traditional pert/cpm algorithm	5.85
management science/operations research	4.05
spreadsheet environment	3.5
project network	2.35
critical path	2.35
pert/cpm implementation	2.2

Table 7.9 shows that the problem of having keywords with the same weight values still exists. In our example these keywords are “project network” and “critical path”. We will tackle this problem in the following technique. The results of the WCombMNZ are evaluated using precision, recall, and F-measure. WCombMNZ technique produced relatively better results compared to the first



four techniques. This technique achieved a recall value of 0.5117, a precision value of 0.4139, and F-measure value of 0.4387. Figure 7.11 shows a comparison between WCombMNZ technique and the three other techniques based on recall, precision, and F-measure.

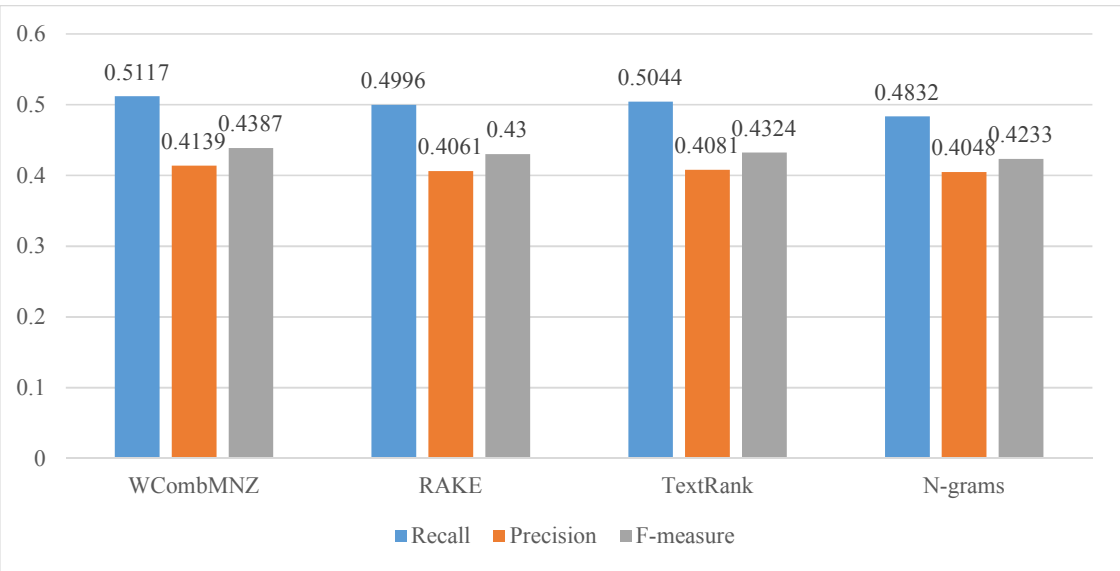


Figure 7.11 Comparison between WCombMNZ technique and the three other techniques based on recall, precision, and F-measure

WCombMNZ retrieved a larger number of relevant keywords than the number of relevant keywords of the previous fusion-based techniques which led to a higher value of F-measure. Figure 7.12 compares WCombMNZ techniques with the other three techniques based on the number of the retrieved keywords, and the number of relevant retrieved keywords.

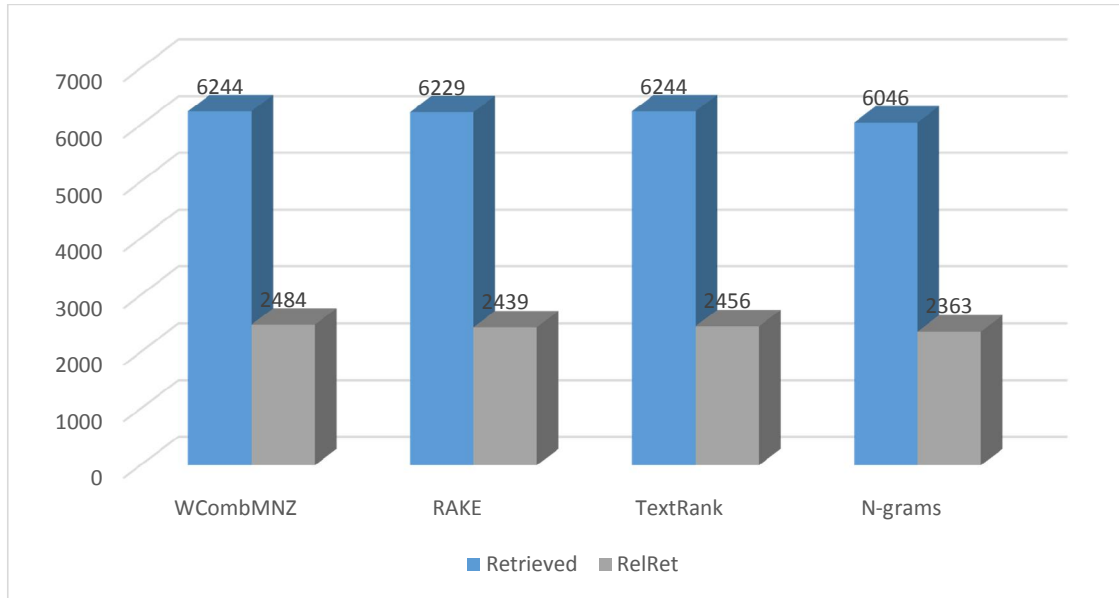


Figure 7.12 Comparison between WCombMNZ technique and the three other techniques

### 7.2.6 Condorcet Technique

Condorcet is similar to the five previous fusion-based techniques because it reads the extracted keyword lists from the three techniques (RAKE, TextRank, and N-grams). In this technique, we are concerned with solving the problem mentioned in Borda Voting technique which is that some keywords have the same weight. Condorcet voting algorithm was used to solve this problem.

Condorcet voting algorithm specifies that the winner is the candidate keyword that beats or ties with every other candidate keyword in a pair-wise comparison. The algorithm assumes that any candidate keyword that can beat all other candidates in a head-to-head contest should win and be extracted to the new list (Montague & Aslam, 2002). In other words, the candidate keyword that has the highest positions in its original lists is assigned with a higher priority than other candidate keywords. In our example, we have two keywords that have the same weight as mentioned in the previous technique; “project network” and “critical path”. Based on Table 7.1, the keyword “critical path” occurred in the RAKE list and in the TextRank list before the keyword “project network”. However, in N-grams list the keyword “project network” occurred before the keyword “critical path”. Because the keyword “critical path” occurred twice before the keyword

“project network”, it is assigned with a higher priority. Table 7.10 shows the extracted keywords by Condorcet technique.

*Table 7.10 The extracted keywords by Condorcet technique*

<b>Keywords extracted by Condorcet technique</b>
traditional pert/cpm algorithm
management science/operations research
spreadsheet environment
critical path
project network
pert/cpm implementation

In contrast to the keyword extracted list by WCombMNZ technique, the keyword “critical path” appears before the keyword “project network” in the keyword extracted list by Condorcet technique as shown in Table 7.10.

Even though a different strategy was implemented in the Condorcet technique, the technique failed to generate better results than the first five fusion-based techniques. The results are evaluated using precision, recall, and F-measure. This technique achieved a recall value of 0.5017, a precision value of 0.384, and F-measure value of 0.4187. Figure 7.13 shows a comparison between Condorcet technique and the three other techniques based on recall, precision, and F-measure.

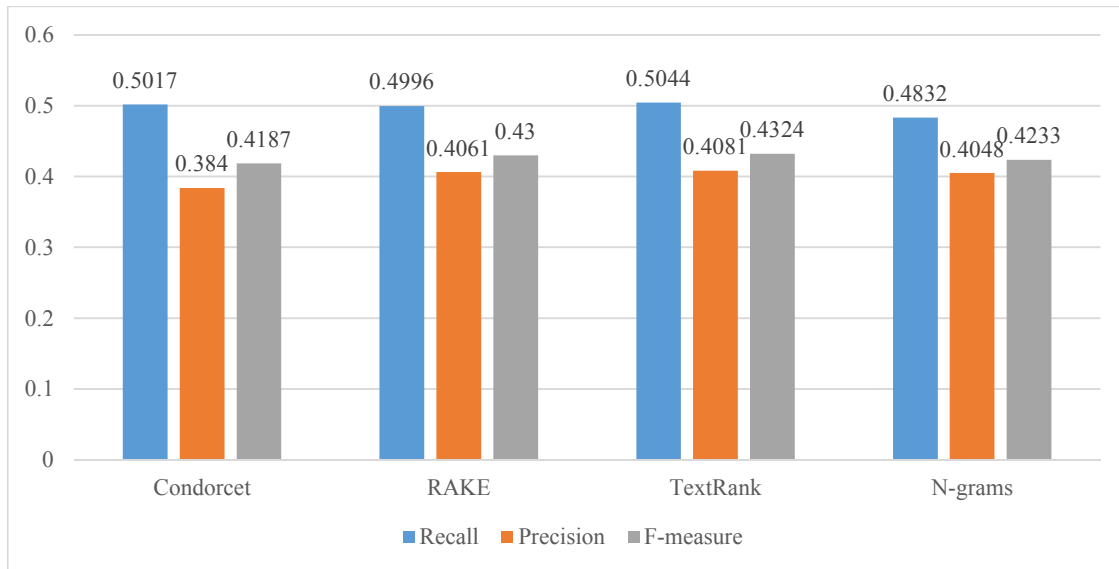


Figure 7.13 Comparison between Condorcet technique and the three other techniques based on recall, precision, and F-measure

Figure 7.14 compares Condorcet techniques with the other three techniques based on the number of the relevant keywords, retrieved keywords, and the number of relevant retrieved keywords.

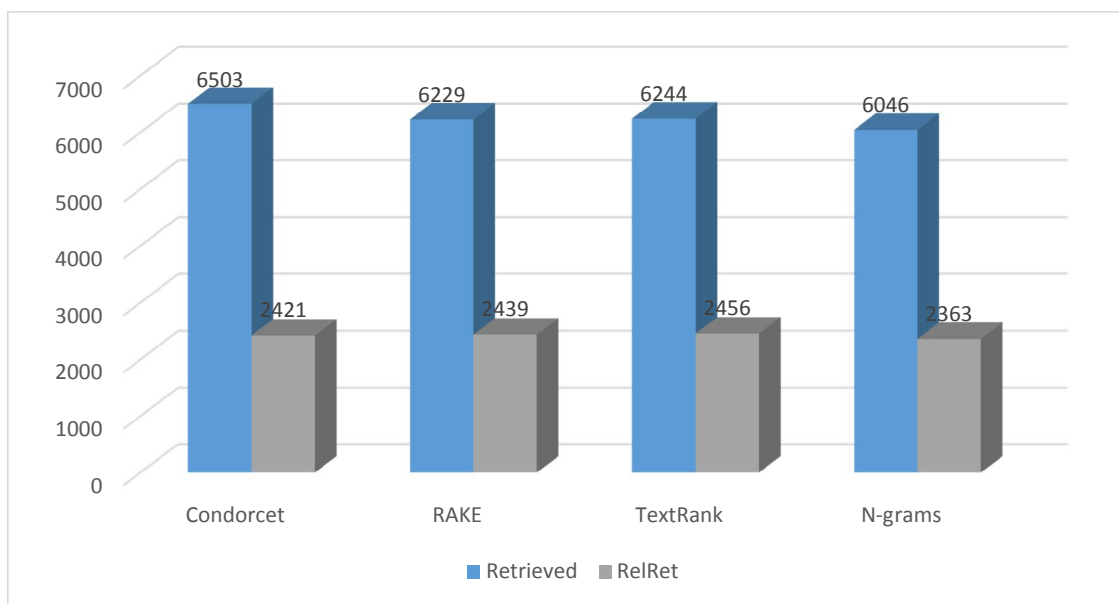


Figure 7.14 Comparison between Condorcet technique and the three other techniques

The Condorcet technique retrieved a smaller number of relevant keywords than the number of relevant keywords retrieved by the other fusion-based techniques. Additionally, the Condorcet technique has a larger number of retrieved keywords which led to achieving a lower precision value than the other fusion-based techniques. All of these changes have led to a lower F-measure value in the Condorcet technique.

## 7.3 Experimental Results

Table 7.11 shows the experimental results related to the techniques discussed in this chapter.

*Table 7.11 Experimental results for the six Fusion-based techniques*

<b>Fusion-Based Technique</b>	<b>Recall</b>	<b>Precision</b>	<b>F-measure</b>
All Keywords Fusion	0.5825	0.292	0.3746
Majority Voting	0.5048	0.4181	0.4382
Borda Voting	0.5012	0.4054	0.4294
CombMNZ	0.5095	0.4128	0.4372
WCombMNZ	0.5117	0.4139	0.4387
Condorcet	0.5017	0.384	0.4187

In Table 7.11, we noticed that all recall values are close to each other except for the recall value of the All Keywords Fusion technique. This is because All Keywords Fusion technique retrieved a large number of keywords. In return, F-measure values are also close to each other in all fusion based techniques except for the F-measure value of the All Keyword Fusion which achieved the lowest F-measure value. In conclusion, WCombMNZ proved to be the best technique

compared to the other five fusion-based techniques which achieved the highest F-measure value. Figure 7.15 shows a comparison between the six fusion-based techniques.

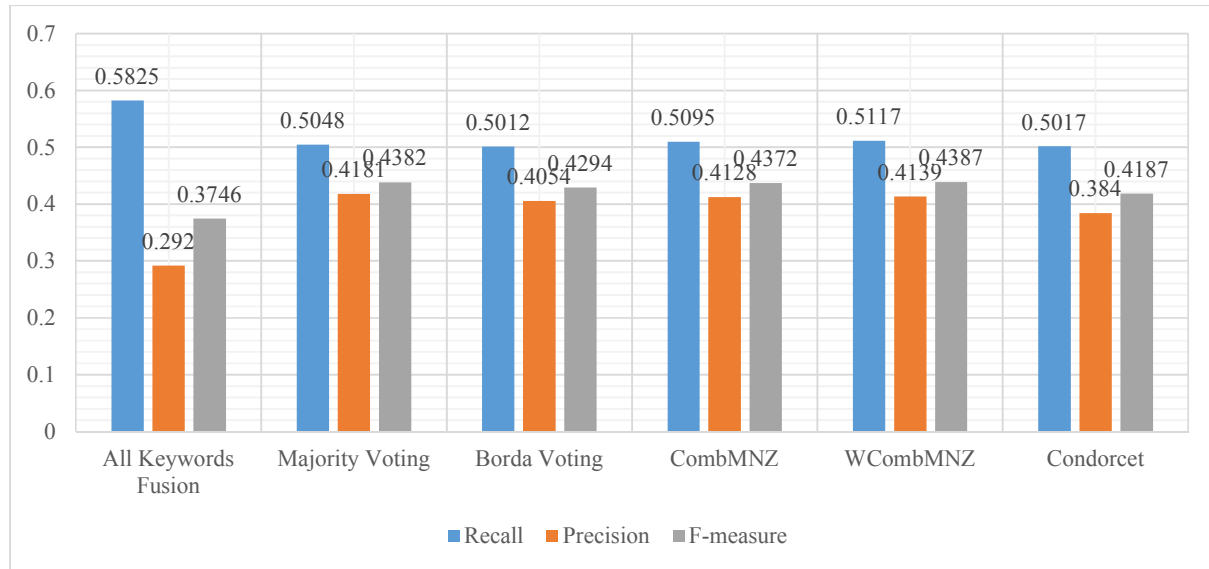


Figure 7.15 Comparison between the six fusion-based techniques based on recall, precision, and F-measure

Table 7.12 shows the number of retrieved and relevant retrieved keywords for the six fusion-based techniques.

Table 7.12 Number of retrieved and relevant retrieved keywords for the six Fusion-based techniques

Fusion-Based Technique	Retrieved keywords	RelRet keywords
All Keywords Fusion	10199	2848
Majority Voting	6048	2444
Borda Voting	6244	2435
CombMNZ	6244	2472
WCombMNZ	6244	2484
Condorcet	6503	2421

Based on Table 7.12, All Keyword Fusion technique achieved the highest number of retrieved keywords. This explains why All Keyword Fusion technique scored the lowest F-measure value as mentioned above. In addition, Borda Voting, CombMNZ, and WCombMNZ achieved the exact number of retrieved keywords. Regarding the number of relevant retrieved keywords, All Keywords Fusion technique achieved the highest number, but because of the large number of retrieved keywords, All Keywords Fusion technique achieved a low precision value. The second highest number of relevant retrieved keywords was achieved by the WCombMNZ technique and because of the usual number of retrieved keywords, it achieved a high precision value. This is the reason why WCombMNZ is considered to be the best fusion-based technique. Figure 7.16 shows a comparison of the retrieved and relevant retrieved keywords between the six fusion-based techniques.

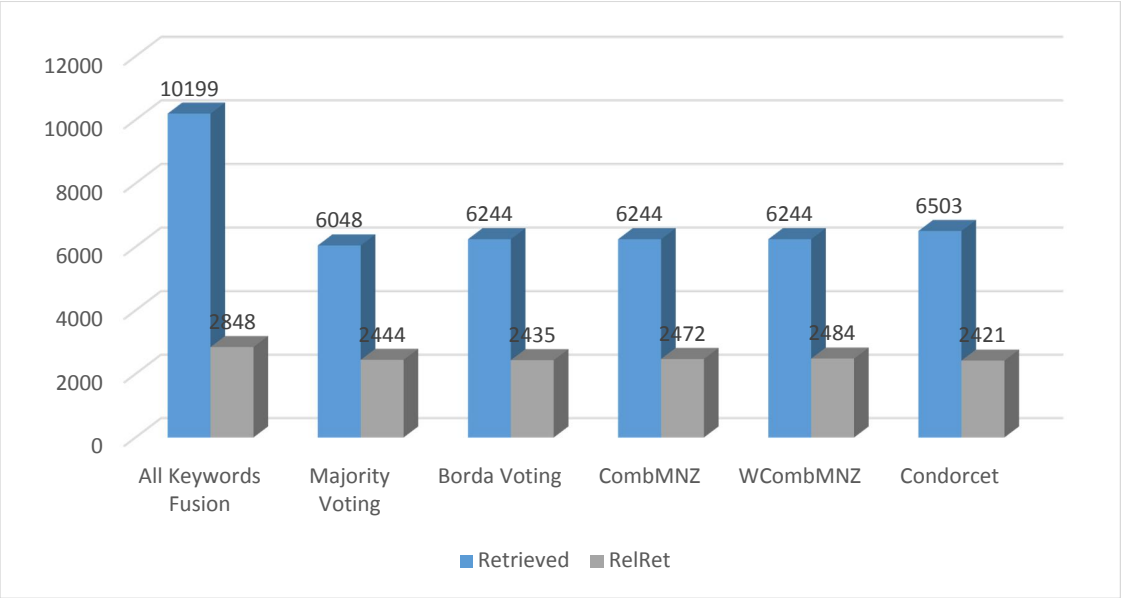


Figure 7.16 Comparison of the retrieved and relevant retrieved keywords between the six fusion-based techniques

# Chapter 8

## Conclusion

### 8.1 Introduction

This chapter presents the conclusion to this research. The summary of the thesis is introduced. The research questions are discussed. And finally, future works of the current system are presented.

### 8.2 Summary of the Thesis

The main objective of this research is to extract keywords from abstracts using a fusion-based technique from three other original techniques, which all shared the same dataset “Hulth 2003”, as training and testing dataset. The first technique is RAKE (Rose, Engle, Carmer, & Cowley, 2010). RAKE is based on keywords usually containing multiple words, but rarely standard punctuations or stopwords. This technique uses stopwords and word delimiters to partition the documents into candidate keywords which will be given scores for each, based on some calculations. Lastly, extracted keywords will be produced based on their scores.

The second technique is TextRank. This technique is based on the co-occurrence links between words. It makes use of voting or recommendation between words to extract keywords. The technique is first implemented by constructing a graph which reflects relationship between different vertices (words), which will be extracted from the given texts, and then using PageRank algorithm to calculate score of words. Finally, highest score of words in the document will be chosen as keywords after an iterative algorithm is used to compute the ranking value of each vertex of the graph (Mihalcea & Tarau, 2004).



The third technique Hulth-2003 N-grams approach. All unigrams, bigrams, trigrams, quad-gram, and 5-gram were extracted. Thereafter a stop list was used, where all terms beginning or ending with a stopword were removed. Finally all remaining tokens were stemmed using Porter's stemmer. And the keywords were extracted (Hulth A. , 2003).

This research discussed the fusion-based technique. The techniques above produce their results as lists of extracted keywords. This new technique integrates and calculates the results of those three techniques to produce its' own results of more accurate keywords than the keywords that are produced individually.

In order to achieve the highest results possible, we implemented six different fusion-based techniques. Each one of the six techniques used a different strategy. The first technique was All Keywords Fusion which is based on term frequency, the second was Majority Voting, which is based on document frequency, the third was Borda Voting that is based on Borda count voting procedure by assigning weights for each candidate keyword, the fourth technique is based on combining term weight and document frequency using CombMNZ algorithm, which named CombMNZ, the fifth fusion-based technique was WCombMNZ which is similar to the CombMNZ but with the difference of the weight for each technique from the original RAKE, TextRank, and N-grams that was assigned. Finally, the sixth technique was named Condorcet technique, it used the Condorcet voting algorithm. After implementing the six fusion-based techniques, it was concluded that the technique that generated the highest results was the WCombMNZ technique compared to the other five techniques. The technique achieved a recall value of 0.5117, a precision value of 0.4139, and F-measure value of 0.4387.

### **8.3 Answering Research Questions**

This research has proved that it is possible to design a keywords and keyphrases extraction system using RAKE, TextRank, N-grams, and Fusion-based techniques on English texts.

The research questions and their brief answers are as follows:

- Is the fusion-based keyword extraction techniques suitable to improve the results of the extracted keywords?

The experimental results have shown that it is possible to build a fusion-based keyword extraction system based on combining the results of RAKE, TextRank, and N-grams techniques that operate on individual documents, and enhance the results value of recall, precision, and F-measure when using the weight feature for each keyword list, and when a weight is assigned for the three original techniques above.

- Does the keyword position feature help to improve the performance of the fusion-based keywords and keyphrases extraction system?

The experimental results have proven that the keyword position is an important factor. When assigning more weights to keywords that exist in a high list position, the results will also significantly improved.

- Do the keyword frequency feature and document frequency feature help to improve the performance of the fusion-based keywords and keyphrases extraction system?

The experimental results have proven that the keyword frequency in each list and document frequency for each keyword are both important factors. When multiplying each keyword frequency by its document frequency, the results will also be significantly improved.

- Does the keyword weighting feature help to improve the performance of the fusion-based keywords and keyphrases extraction system?

The experimental results have shown that the weight feature is very important and useful, when assigning weight for each keyword in the list, the results will be improved. Also, when assigning each technique from the original techniques RAKE, TextRank, and N-grams according to their F-measure values, the results will be improved.

## 8.4 Future Work

This research can be further developed and improved in a number of directions as follows:

- Improving the system to be applied to more documents by increasing dataset size.
- Applying different weighting methods to the fusion-based techniques that could be useful for improving results.
- Applying the keywords and keyphrases extraction using RAKE, TextRank, and N-grams to documents that have larger lengths than just abstracts.
- Applying the keywords and keyphrases extraction using RAKE, TextRank, and N-grams to multi documents of the same topic.

# References

- Al-Hashemi, R. (2010). Text Summarization Extraction System (TSES) Using Extracted Keywords. *Int. Arab J. e-Technol.*, 1(4), 164-168.
- Alhadidi, M. (2013). *Keywords Extraction Using Page Rank Algorithm for Arabic Text*. AL-Balqa' Applied University.
- Alklifat, A. (2014). *Investigating the Performance of Applying Information Retrieval Models on Contextual Suggestion Track*. Al-Balqa' Applied University.
- Alzghool, M. (2009). Investigating different models for cross-language information retrieval from automatic speech transcripts.
- Bracewell, D. B., Ren, F., & Kuriowa, S. (2005). Multilingual single document keyword extraction for information retrieval. In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on* (pp. 517-522). IEEE.
- Brin, S., Page, L., & Motwani, R. (1998). The page rank citation ranking: bringing order to the web [EB/OL].
- Creighton. (2013). Retrieved from [http://www.creighton.edu/fileadmin/user/hsl/docs/ref/searching\\_\\_recall\\_precision.pdf](http://www.creighton.edu/fileadmin/user/hsl/docs/ref/searching__recall_precision.pdf)
- Ercan, G., & Cicekli, I. (2007). Using lexical chains for keyword extraction. *Information Processing & Management*, 43(6), 1705-1714.
- Fisher, H. L., & Elchesen, D. R. (1972). General: Effectiveness of Combining Title Words and Index Terms in Machine Retrieval Searches.
- Fox, E. A., & Shaw, J. A. (1994). Combination of multiple searches. *NIST SPECIAL PUBLICATION SP*, 243-243.
- Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999, July). Domain-specific keyphrase extraction. In *IJCAI* (Vol. 99, pp. 668-673).
- Fukumoto, F., Sekiguchi, Y., & Suzuki, Y. (1998, August). Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 373-374). ACM.
- Gupta, V., & Lehal, G. S. (2011). Automatic keywords extraction for Punjabi language. *International Journal of Computer Science Issues*, 8(5), 327-331.
- He, D., & Ahn, J. W. (2005). *Pitt at CLEF05: data fusion for spoken document retrieval* (pp. 773-782). Springer Berlin Heidelberg.

- Hu, X., & Wu, B. (2006, December). Automatic keyword extraction using linguistic features. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on* (pp. 19-23). IEEE.
- Hulth, A. (2003, July). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 216-223). Association for Computational Linguistics.
- Hulth, A. (2003). TextRank: Bringing order into texts. In *Proceedings of the 2003 conference on empirical methods in natural language processing*.
- Insight, M. (2013). Retrieved from [http://mathinsight.org/undirected\\_graph\\_definition](http://mathinsight.org/undirected_graph_definition)
- Jiao, H., Liu, Q., & Jia, H. B. (2007, December). Chinese keyword extraction based on N-gram and word co-occurrence. In *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on* (pp. 152-155). IEEE.
- Jones, S., & Paynter, G. W. (2002). Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications. *Journal of the American Society for Information Science and Technology*, 53(8), 653-677.
- Mahgoub, H., Rösner, D., Ismail, N., & Torkey, F. (2008). A text mining technique using association rules extraction. *International journal of computational intelligence*, 4(1), 21-28.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing* (Vol. 999). Cambridge: MIT press.
- Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2), 313-330.
- Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01), 157-169.
- Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.
- Montague, M., & Aslam, J. A. (2002, November). Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 538-548). ACM.
- Oelze, I. (2009). Automatic Keyword Extraction for Database Search.
- Pallotta, V., Van Der Plas, L., Rajman, M., & Ghorbel, H. (2004). Automatic keyword extraction from spoken text. a comparison of two lexical resources: the EDR and WordNet. *arXiv preprint cs/0410062*.

- Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining*, 1-20.
- Salton, G., Yang, C. S., & Yu, C. T. (1975). A theory of term importance in automatic text analysis. *Journal of the American society for Information Science*, 26(1), 33-44.
- Sarkar, K., Nasipuri, M., & Ghose, S. (2010). A new approach to keyphrase extraction using neural networks. *arXiv preprint arXiv:1004.3274*.
- Shaw, J. A., & Fox, E. A. (1994). Combination of Multiple Searches. *Proceedings of the third Text REtrieval Conference*.
- Terol, R. M., Martinez-Barco, P., & Palomar, M. (2006). Applying logic forms and statistical methods to CL-SR performance. In *Evaluation of Multilingual and Multi-modal Information Retrieval* (pp. 766-769). Springer Berlin Heidelberg.
- Tiwari, R., Zhang, C., & Solorio, T. (2010, August). A supervised machine learning approach of extracting coexpression relationship among genes from literature. In *Information Reuse and Integration (IRI), 2010 IEEE International Conference on* (pp. 98-103). IEEE.
- Wartena, C., Brussee, R., & Slakhorst, W. (2010, August). Keyword extraction using word co-occurrence. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on* (pp. 54-58). IEEE.
- Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999, August). KEA: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries* (pp. 254-255). ACM.
- Wei, Y. (2012). An iterative approach to keywords extraction. In *Advances in Swarm Intelligence* (pp. 93-99). Springer Berlin Heidelberg.
- Zhang, K., Xu, H., Tang, J., & Li, J. (2006). Keyword extraction using support vector machine. In *Advances in Web-Age Information Management* (pp. 85-96). Springer Berlin Heidelberg.
- Zhao, L., Yang, L., & Ma, X. (2010, December). Using tag to help keyword extraction. In *Computer and Information Application (ICCIA), 2010 International Conference on* (pp. 95-98). IEEE.

# Appendices

## Appendix A

### RAKE Stopwords

<b>Word</b>	<b>Term Frequency</b>	<b>Document Frequency</b>	<b>Adjacency Frequency</b>	<b>Keyword Frequency</b>
<b>the</b>	<b>8611</b>	<b>978</b>	<b>320</b>	<b>3</b>
<b>of</b>	<b>5546</b>	<b>939</b>	<b>440</b>	<b>68</b>
<b>and</b>	<b>3644</b>	<b>911</b>	<b>251</b>	<b>23</b>
<b>a</b>	<b>3600</b>	<b>893</b>	<b>179</b>	<b>3</b>
<b>to</b>	<b>3000</b>	<b>879</b>	<b>77</b>	<b>10</b>
<b>in</b>	<b>2657</b>	<b>837</b>	<b>196</b>	<b>8</b>
<b>is</b>	<b>1974</b>	<b>757</b>	<b>84</b>	<b>0</b>
<b>for</b>	<b>1912</b>	<b>767</b>	<b>234</b>	<b>9</b>
<b>that</b>	<b>1129</b>	<b>590</b>	<b>35</b>	<b>0</b>
<b>with</b>	<b>1065</b>	<b>577</b>	<b>85</b>	<b>3</b>
<b>are</b>	<b>1049</b>	<b>576</b>	<b>38</b>	<b>1</b>
<b>this</b>	<b>964</b>	<b>581</b>	<b>22</b>	<b>0</b>
<b>on</b>	<b>919</b>	<b>550</b>	<b>65</b>	<b>8</b>
<b>an</b>	<b>856</b>	<b>501</b>	<b>45</b>	<b>0</b>
<b>we</b>	<b>822</b>	<b>388</b>	<b>12</b>	<b>0</b>
<b>by</b>	<b>773</b>	<b>475</b>	<b>32</b>	<b>0</b>
<b>as</b>	<b>743</b>	<b>435</b>	<b>22</b>	<b>0</b>
<b>be</b>	<b>595</b>	<b>395</b>	<b>1</b>	<b>0</b>
<b>can</b>	<b>452</b>	<b>319</b>	<b>19</b>	<b>0</b>
<b>based</b>	<b>451</b>	<b>293</b>	<b>27</b>	<b>15</b>
<b>from</b>	<b>447</b>	<b>309</b>	<b>13</b>	<b>0</b>
<b>using</b>	<b>428</b>	<b>282</b>	<b>50</b>	<b>0</b>
<b>which</b>	<b>402</b>	<b>280</b>	<b>10</b>	<b>0</b>

<b>or</b>	<b>315</b>	<b>218</b>	<b>6</b>	<b>0</b>
<b>have</b>	<b>301</b>	<b>219</b>	<b>8</b>	<b>0</b>
<b>has</b>	<b>297</b>	<b>225</b>	<b>20</b>	<b>0</b>
<b>at</b>	<b>296</b>	<b>216</b>	<b>10</b>	<b>0</b>
<b>new</b>	<b>294</b>	<b>197</b>	<b>17</b>	<b>4</b>
<b>two</b>	<b>287</b>	<b>205</b>	<b>6</b>	<b>5</b>
<b>used</b>	<b>262</b>	<b>204</b>	<b>5</b>	<b>0</b>
<b>was</b>	<b>254</b>	<b>125</b>	<b>8</b>	<b>0</b>
<b>these</b>	<b>252</b>	<b>200</b>	<b>5</b>	<b>0</b>
<b>also</b>	<b>251</b>	<b>219</b>	<b>3</b>	<b>0</b>
<b>such</b>	<b>249</b>	<b>198</b>	<b>4</b>	<b>0</b>
<b>its</b>	<b>222</b>	<b>169</b>	<b>3</b>	<b>0</b>
<b>one</b>	<b>218</b>	<b>175</b>	<b>5</b>	<b>2</b>
<b>not</b>	<b>217</b>	<b>181</b>	<b>1</b>	<b>0</b>
<b>proposed</b>	<b>213</b>	<b>156</b>	<b>3</b>	<b>0</b>
<b>more</b>	<b>211</b>	<b>171</b>	<b>3</b>	<b>1</b>
<b>their</b>	<b>206</b>	<b>160</b>	<b>6</b>	<b>0</b>
<b>use</b>	<b>199</b>	<b>159</b>	<b>7</b>	<b>4</b>
<b>our</b>	<b>185</b>	<b>130</b>	<b>4</b>	<b>0</b>
<b>between</b>	<b>183</b>	<b>138</b>	<b>9</b>	<b>0</b>
<b>some</b>	<b>174</b>	<b>145</b>	<b>5</b>	<b>0</b>
<b>when</b>	<b>172</b>	<b>139</b>	<b>6</b>	<b>0</b>
<b>into</b>	<b>159</b>	<b>139</b>	<b>5</b>	<b>0</b>
<b>than</b>	<b>157</b>	<b>133</b>	<b>4</b>	<b>0</b>
<b>were</b>	<b>149</b>	<b>87</b>	<b>13</b>	<b>0</b>
<b>different</b>	<b>148</b>	<b>119</b>	<b>6</b>	<b>0</b>
<b>how</b>	<b>144</b>	<b>109</b>	<b>3</b>	<b>0</b>
<b>three</b>	<b>142</b>	<b>102</b>	<b>4</b>	<b>0</b>
<b>other</b>	<b>142</b>	<b>124</b>	<b>6</b>	<b>0</b>
<b>over</b>	<b>127</b>	<b>105</b>	<b>8</b>	<b>3</b>
<b>given</b>	<b>113</b>	<b>93</b>	<b>2</b>	<b>0</b>



<b>present</b>	<b>112</b>	<b>92</b>	<b>2</b>	<b>0</b>
<b>may</b>	<b>112</b>	<b>98</b>	<b>1</b>	<b>0</b>
<b>developed</b>	<b>112</b>	<b>99</b>	<b>5</b>	<b>0</b>
<b>through</b>	<b>108</b>	<b>94</b>	<b>6</b>	<b>1</b>
<b>only</b>	<b>107</b>	<b>97</b>	<b>3</b>	<b>0</b>
<b>will</b>	<b>106</b>	<b>79</b>	<b>3</b>	<b>0</b>
<b>under</b>	<b>95</b>	<b>77</b>	<b>3</b>	<b>0</b>
<b>various</b>	<b>88</b>	<b>76</b>	<b>2</b>	<b>0</b>
<b>where</b>	<b>87</b>	<b>80</b>	<b>2</b>	<b>0</b>
<b>several</b>	<b>86</b>	<b>76</b>	<b>4</b>	<b>0</b>
<b>efficient</b>	<b>84</b>	<b>71</b>	<b>9</b>	<b>5</b>
<b>applied</b>	<b>83</b>	<b>77</b>	<b>4</b>	<b>2</b>
<b>compared</b>	<b>79</b>	<b>69</b>	<b>2</b>	<b>0</b>
<b>provide</b>	<b>78</b>	<b>69</b>	<b>1</b>	<b>0</b>
<b>what</b>	<b>77</b>	<b>55</b>	<b>2</b>	<b>0</b>
<b>provides</b>	<b>76</b>	<b>70</b>	<b>5</b>	<b>0</b>
<b>any</b>	<b>75</b>	<b>63</b>	<b>1</b>	<b>0</b>
<b>presents</b>	<b>74</b>	<b>71</b>	<b>3</b>	<b>0</b>
<b>about</b>	<b>74</b>	<b>64</b>	<b>4</b>	<b>0</b>
<b>obtained</b>	<b>74</b>	<b>59</b>	<b>2</b>	<b>0</b>
<b>no</b>	<b>73</b>	<b>66</b>	<b>1</b>	<b>0</b>
<b>without</b>	<b>71</b>	<b>63</b>	<b>2</b>	<b>0</b>
<b>while</b>	<b>69</b>	<b>64</b>	<b>3</b>	<b>0</b>
<b>possible</b>	<b>66</b>	<b>56</b>	<b>6</b>	<b>0</b>
<b>same</b>	<b>66</b>	<b>61</b>	<b>1</b>	<b>0</b>
<b>among</b>	<b>66</b>	<b>60</b>	<b>1</b>	<b>0</b>
<b>make</b>	<b>65</b>	<b>62</b>	<b>1</b>	<b>0</b>
<b>designed</b>	<b>64</b>	<b>57</b>	<b>1</b>	<b>0</b>
<b>do</b>	<b>63</b>	<b>59</b>	<b>10</b>	<b>0</b>
<b>called</b>	<b>63</b>	<b>45</b>	<b>4</b>	<b>0</b>
<b>within</b>	<b>62</b>	<b>55</b>	<b>2</b>	<b>0</b>

during	61	50	5	0
propose	60	53	1	0
because	60	55	1	0
including	59	53	4	0
existing	56	48	2	0
improve	54	48	1	0
better	52	49	2	0
describe	50	44	1	0
good	49	42	1	0
after	48	42	3	0
defined	47	32	3	1
does	47	40	3	1
four	47	39	4	0
available	47	40	2	0
help	47	44	3	3
useful	46	45	2	0
uses	45	38	2	0
due	45	42	1	0
achieve	45	41	3	0
reduce	45	43	1	0
allows	43	41	2	0
introduced	42	36	2	0
x	41	19	3	2
novel	39	33	4	0
find	38	35	1	0
include	38	35	1	0
much	38	33	1	0
via	36	33	3	0
obtain	35	33	1	0
therefore	35	33	2	0
resulting	35	30	1	0

<b>determine</b>	<b>35</b>	<b>32</b>	<b>1</b>	<b>0</b>
<b>less</b>	<b>34</b>	<b>31</b>	<b>2</b>	<b>0</b>
<b>could</b>	<b>34</b>	<b>33</b>	<b>1</b>	<b>0</b>
<b>take</b>	<b>33</b>	<b>31</b>	<b>1</b>	<b>0</b>
<b>discuss</b>	<b>33</b>	<b>32</b>	<b>1</b>	<b>0</b>
<b>basic</b>	<b>31</b>	<b>27</b>	<b>1</b>	<b>0</b>
<b>consider</b>	<b>31</b>	<b>28</b>	<b>1</b>	<b>0</b>
<b>purpose</b>	<b>31</b>	<b>30</b>	<b>1</b>	<b>0</b>
<b>original</b>	<b>31</b>	<b>30</b>	<b>1</b>	<b>0</b>
<b>evaluate</b>	<b>31</b>	<b>30</b>	<b>3</b>	<b>0</b>
<b>gives</b>	<b>31</b>	<b>28</b>	<b>1</b>	<b>0</b>
<b>across</b>	<b>30</b>	<b>29</b>	<b>1</b>	<b>0</b>
<b>especially</b>	<b>30</b>	<b>30</b>	<b>1</b>	<b>0</b>
<b>generate</b>	<b>30</b>	<b>26</b>	<b>3</b>	<b>0</b>
<b>tested</b>	<b>29</b>	<b>28</b>	<b>1</b>	<b>0</b>
<b>introduce</b>	<b>29</b>	<b>28</b>	<b>2</b>	<b>0</b>
<b>consists</b>	<b>28</b>	<b>23</b>	<b>1</b>	<b>0</b>
<b>construct</b>	<b>28</b>	<b>25</b>	<b>1</b>	<b>1</b>
<b>needed</b>	<b>28</b>	<b>26</b>	<b>3</b>	<b>0</b>
<b>along</b>	<b>28</b>	<b>27</b>	<b>4</b>	<b>0</b>
<b>although</b>	<b>27</b>	<b>26</b>	<b>2</b>	<b>0</b>
<b>achieved</b>	<b>27</b>	<b>27</b>	<b>1</b>	<b>0</b>
<b>instead</b>	<b>27</b>	<b>26</b>	<b>1</b>	<b>0</b>
<b>look</b>	<b>26</b>	<b>22</b>	<b>1</b>	<b>0</b>
<b>additional</b>	<b>26</b>	<b>24</b>	<b>1</b>	<b>0</b>
<b>every</b>	<b>26</b>	<b>25</b>	<b>1</b>	<b>0</b>
<b>according</b>	<b>26</b>	<b>23</b>	<b>2</b>	<b>0</b>
<b>taken</b>	<b>25</b>	<b>24</b>	<b>1</b>	<b>0</b>
<b>before</b>	<b>25</b>	<b>24</b>	<b>1</b>	<b>0</b>
<b>cannot</b>	<b>24</b>	<b>24</b>	<b>3</b>	<b>0</b>
<b>makes</b>	<b>24</b>	<b>21</b>	<b>1</b>	<b>0</b>

<b>derive</b>	<b>24</b>	<b>23</b>	<b>1</b>	<b>0</b>
<b>your</b>	<b>24</b>	<b>24</b>	<b>1</b>	<b>0</b>
<b>extensions</b>	<b>24</b>	<b>23</b>	<b>1</b>	<b>0</b>
<b>might</b>	<b>24</b>	<b>21</b>	<b>1</b>	<b>0</b>
<b>employed</b>	<b>24</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>discusses</b>	<b>24</b>	<b>22</b>	<b>1</b>	<b>1</b>
<b>includes</b>	<b>24</b>	<b>18</b>	<b>2</b>	<b>0</b>
<b>had</b>	<b>23</b>	<b>19</b>	<b>1</b>	<b>0</b>
<b>five</b>	<b>22</b>	<b>21</b>	<b>1</b>	<b>0</b>
<b>typical</b>	<b>22</b>	<b>16</b>	<b>1</b>	<b>0</b>
<b>upon</b>	<b>22</b>	<b>18</b>	<b>4</b>	<b>0</b>
<b>previously</b>	<b>22</b>	<b>22</b>	<b>1</b>	<b>0</b>
<b>understanding</b>	<b>21</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>considers</b>	<b>21</b>	<b>20</b>	<b>1</b>	<b>0</b>
<b>q</b>	<b>21</b>	<b>12</b>	<b>2</b>	<b>0</b>
<b>implementing</b>	<b>21</b>	<b>21</b>	<b>1</b>	<b>0</b>
<b>produced</b>	<b>20</b>	<b>17</b>	<b>1</b>	<b>0</b>
<b>apply</b>	<b>20</b>	<b>17</b>	<b>1</b>	<b>0</b>
<b>providing</b>	<b>20</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>whose</b>	<b>20</b>	<b>19</b>	<b>2</b>	<b>0</b>
<b>get</b>	<b>19</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>analyze</b>	<b>19</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>last</b>	<b>19</b>	<b>19</b>	<b>3</b>	<b>0</b>
<b>successful</b>	<b>19</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>constructing</b>	<b>19</b>	<b>17</b>	<b>2</b>	<b>0</b>
<b>developing</b>	<b>19</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>y</b>	<b>18</b>	<b>17</b>	<b>1</b>	<b>0</b>
<b>suitable</b>	<b>18</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>taking</b>	<b>18</b>	<b>18</b>	<b>1</b>	<b>0</b>
<b>increased</b>	<b>17</b>	<b>17</b>	<b>1</b>	<b>0</b>
<b>extend</b>	<b>17</b>	<b>16</b>	<b>1</b>	<b>0</b>

<b>having</b>	<b>17</b>	<b>17</b>	<b>2</b>	<b>0</b>
<b>characterize</b>	<b>17</b>	<b>14</b>	<b>1</b>	<b>0</b>
<b>improving</b>	<b>16</b>	<b>14</b>	<b>2</b>	<b>0</b>
<b>allowing</b>	<b>16</b>	<b>16</b>	<b>2</b>	<b>0</b>
<b>towards</b>	<b>16</b>	<b>15</b>	<b>2</b>	<b>0</b>
<b>supports</b>	<b>16</b>	<b>15</b>	<b>1</b>	<b>0</b>
<b>near</b>	<b>16</b>	<b>14</b>	<b>3</b>	<b>0</b>
<b>against</b>	<b>16</b>	<b>14</b>	<b>1</b>	<b>1</b>
<b>creating</b>	<b>15</b>	<b>15</b>	<b>2</b>	<b>0</b>
<b>introduces</b>	<b>15</b>	<b>14</b>	<b>1</b>	<b>0</b>
<b>offer</b>	<b>15</b>	<b>13</b>	<b>1</b>	<b>0</b>
<b>desired</b>	<b>15</b>	<b>14</b>	<b>2</b>	<b>0</b>
<b>create</b>	<b>15</b>	<b>14</b>	<b>1</b>	<b>0</b>
<b>looks</b>	<b>15</b>	<b>14</b>	<b>1</b>	<b>1</b>
<b>define</b>	<b>15</b>	<b>11</b>	<b>2</b>	<b>0</b>
<b>larger</b>	<b>15</b>	<b>15</b>	<b>2</b>	<b>0</b>
<b>explore</b>	<b>15</b>	<b>13</b>	<b>1</b>	<b>0</b>
<b>realistic</b>	<b>15</b>	<b>14</b>	<b>1</b>	<b>0</b>
<b>fact</b>	<b>15</b>	<b>13</b>	<b>1</b>	<b>0</b>
<b>built</b>	<b>15</b>	<b>14</b>	<b>1</b>	<b>0</b>
<b>adopted</b>	<b>15</b>	<b>15</b>	<b>1</b>	<b>0</b>
<b>becomes</b>	<b>15</b>	<b>13</b>	<b>1</b>	<b>0</b>
<b>prior</b>	<b>15</b>	<b>14</b>	<b>1</b>	<b>0</b>
<b>induced</b>	<b>14</b>	<b>14</b>	<b>2</b>	<b>0</b>
<b>popular</b>	<b>14</b>	<b>13</b>	<b>4</b>	<b>2</b>
<b>enables</b>	<b>14</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>applying</b>	<b>14</b>	<b>13</b>	<b>2</b>	<b>0</b>
<b>performs</b>	<b>14</b>	<b>13</b>	<b>1</b>	<b>0</b>
<b>produce</b>	<b>14</b>	<b>12</b>	<b>1</b>	<b>0</b>
<b>simulate</b>	<b>13</b>	<b>13</b>	<b>1</b>	<b>1</b>
<b>avoid</b>	<b>13</b>	<b>11</b>	<b>1</b>	<b>0</b>

<b>serious</b>	<b>13</b>	<b>12</b>	<b>1</b>	<b>0</b>
<b>around</b>	<b>13</b>	<b>12</b>	<b>2</b>	<b>0</b>
<b>supporting</b>	<b>13</b>	<b>11</b>	<b>1</b>	<b>0</b>
<b>move</b>	<b>13</b>	<b>12</b>	<b>1</b>	<b>0</b>
<b>combining</b>	<b>13</b>	<b>12</b>	<b>2</b>	<b>0</b>
<b>sophisticated</b>	<b>13</b>	<b>11</b>	<b>1</b>	<b>0</b>
<b>evaluating</b>	<b>13</b>	<b>13</b>	<b>1</b>	<b>0</b>
<b>powerful</b>	<b>12</b>	<b>12</b>	<b>1</b>	<b>0</b>
<b>play</b>	<b>12</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>identifying</b>	<b>12</b>	<b>11</b>	<b>3</b>	<b>0</b>
<b>developers</b>	<b>12</b>	<b>12</b>	<b>1</b>	<b>0</b>
<b>though</b>	<b>11</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>optimize</b>	<b>11</b>	<b>11</b>	<b>2</b>	<b>0</b>
<b>ensure</b>	<b>11</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>concerns</b>	<b>11</b>	<b>11</b>	<b>1</b>	<b>0</b>
<b>addresses</b>	<b>11</b>	<b>9</b>	<b>2</b>	<b>0</b>
<b>defines</b>	<b>11</b>	<b>11</b>	<b>1</b>	<b>0</b>
<b>extracting</b>	<b>11</b>	<b>10</b>	<b>2</b>	<b>0</b>
<b>helps</b>	<b>11</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>considering</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>specified</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>excellent</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>explain</b>	<b>10</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>formed</b>	<b>10</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>named</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>reach</b>	<b>10</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>compute</b>	<b>10</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>enabling</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>situations</b>	<b>10</b>	<b>10</b>	<b>1</b>	<b>0</b>
<b>attempt</b>	<b>10</b>	<b>10</b>	<b>2</b>	<b>0</b>
<b>phases</b>	<b>10</b>	<b>8</b>	<b>1</b>	<b>0</b>

<b>concerning</b>	<b>9</b>	<b>5</b>	<b>3</b>	<b>0</b>
<b>evolving</b>	<b>9</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>achieving</b>	<b>9</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>auto</b>	<b>9</b>	<b>8</b>	<b>1</b>	<b>2</b>
<b>estimating</b>	<b>9</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>comprehensive</b>	<b>9</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>simulating</b>	<b>9</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>come</b>	<b>9</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>permits</b>	<b>9</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>track</b>	<b>9</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>substantially</b>	<b>9</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>reasonable</b>	<b>9</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>caused</b>	<b>9</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>entering</b>	<b>9</b>	<b>9</b>	<b>1</b>	<b>0</b>
<b>fails</b>	<b>8</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>involves</b>	<b>8</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>scales</b>	<b>8</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>subsequent</b>	<b>8</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>coming</b>	<b>8</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>holds</b>	<b>8</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>commonly</b>	<b>8</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>proper</b>	<b>8</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>predictions</b>	<b>8</b>	<b>8</b>	<b>1</b>	<b>0</b>
<b>exhibit</b>	<b>7</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>former</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>really</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>demonstrates</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>plays</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>studying</b>	<b>7</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>preserve</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>offering</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>

<b>affect</b>	<b>7</b>	<b>6</b>	<b>2</b>	<b>0</b>
<b>assess</b>	<b>7</b>	<b>7</b>	<b>1</b>	<b>0</b>
<b>indicates</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>executive</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>reveal</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>integrating</b>	<b>6</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>appears</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>onto</b>	<b>6</b>	<b>6</b>	<b>2</b>	<b>0</b>
<b>utilizing</b>	<b>6</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>save</b>	<b>6</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>incorporating</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>performing</b>	<b>6</b>	<b>4</b>	<b>2</b>	<b>0</b>
<b>causes</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>quick</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>0</b>
<b>throughout</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>employ</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>arise</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>bring</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>specifying</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>respective</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>investigating</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>assumptions</b>	<b>5</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>evolve</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>combine</b>	<b>5</b>	<b>5</b>	<b>1</b>	<b>0</b>
<b>appeared</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>learn</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>went</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>continues</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>simpler</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>brought</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
<b>responsible</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>



<b>termed</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>0</b>
---------------	----------	----------	----------	----------

# Appendix B

## Stopwords

a	follows	nonetheless	theirs
about	following	noone	them
above	for	nope	themselves
ac	formal	nor	then
according	former	nos	thence
accordingly	formerly	not	thenceforth
across	forth	note	there
actually	forty	noted	thereabout
ad	forward	notes	thereabouts
adj	found	noting	thereafter
af	four	nothing	thereby
after	fra	notwithstanding	therefor
afterwards	frequently	now	therefore
again	from	nowadays	therein
against	front	nowhere	thereof
al	fuer	o	thereon
albeit	further	obtain	thereto
all	furthermore	obtained	thereupon
almost	furthest	obtaining	these
alone	g	obtains	they
along	gave	och	thing
already	general	of	things
als	generally	off	third
also	get	often	thirty
although	gets	og	this
always	getting	ohne	those
am	give	ok	thou
among	given	old	though
amongst	gives	om	thousand

an	giving	on	thousands
and	go	once	three
another	going	onceone	thrice
any	gone	one	through
anybody	good	only	throughout
anyhow	got	onto	thru
anyone	great	or	thus
anything	greater	ot	thy
anyway	h	other	thysself
anywhere	had	others	til
apart	haedly	otherwise	till
apparently	half	ou	time
are	halves	ought	times
aren	hardly	our	tis
arise	has	ours	to
around	hasn	ourselves	together
as	hast	out	too
aside	hath	outside	tot
at	have	over	tou
au	haven	overall	toward
auf	having	owing	towards
aus	he	own	trillion
aux	hence	p	trillions
av	henceforth	par	twenty
avec	her	para	two
away	here	particular	u
b	hereabouts	particularly	ueber
be	hereafter	past	ugh
became	hereby	per	uit
because	herein	perhaps	un
become	hereto	please	unable
becomes	hereupon	plenty	und
becoming	hers	plus	under

been	herself	por	underneath
before	het	possible	unless
beforehand	high	possibly	unlike
began	higher	pour	unlikely
begin	highest	poured	until
beginning	him	pouring	up
begins	himself	pours	upon
behind	hindmost	predominantly	upward
bei	his	previously	us
being	hither	pro	use
below	how	probabilistic	used
beside	however	probably	useful
besides	howsoever	prompt	usefully
best	hundred	promptly	user
better	hundreds	provide	users
between	i	provides	uses
beyond	ie	provided	using
billion	if	providing	usually
both	ihre	q	v
briefly	ii	quite	van
but	im	r	various
by	immediately	rather	vast
c	important	re	ve
came	in	ready	very
can	inasmuch	really	via
cannot	inc	recent	vom
canst	include	recently	von
caption	included	regardless	voor
captions	includes	relatively	vs
certain	including	respectively	w
certainly	indeed	round	want
cf	indoors	s	was
choose	inside	said	wasn

chooses	insomuch	same	way
choosing	instead	sang	ways
chose	into	save	we
chosen	inward	saw	week
clear	is	say	weeks
clearly	isn	second	well
co	it	see	went
come	its	seeing	were
comes	itself	seem	weren
con	j	seemed	what
contrariwise	ja	seeming	whatever
cos	journal	seems	whatsoever
could	journals	seen	when
couldn	just	sees	whence
cu	k	seldom	whenever
d	kai	self	whensoever
da	keep	selves	where
dans	keeping	send	whereabouts
das	kept	sending	whereafter
day	kg	sends	whereas
de	kind	sent	whereat
degli	kinds	ses	whereby
dei	km	seven	wherefore
del	l	seventy	wherefrom
della	la	several	wherein
delle	large	shall	whereinto
dem	largely	shalt	whereof
den	larger	she	whereon
der	largest	short	wheresoever
deren	las	should	whereto
des	last	shouldn	whereunto
di	later	show	whereupon
did	latter	showed	wherever

didn	latterly	showing	wherewith
die	le	shown	whether
different	least	shows	whew
din	les	si	which
do	less	sideways	whichever
does	lest	significant	whichsoever
doesn	let	similar	while
doing	like	similarly	whilst
don	likely	simple	whither
done	little	simply	who
dos	ll	since	whoever
dost	long	sing	whole
double	longer	single	whom
down	los	six	whomever
du	lower	sixty	whomsoever
dual	lowest	sleep	whose
due	ltd	sleeping	whosoever
durch	m	sleeps	why
during	made	slept	wide
e	mainly	slew	widely
each	make	slightly	will
ed	makes	small	wilt
eg	making	smote	with
eight	many	so	within
eighty	may	sobre	without
either	maybe	some	won
el	me	somebody	worse
else	meantime	somehow	worst
elsewhere	meanwhile	someone	would
em	med	something	wouldn
en	might	sometime	wow
end	million	sometimes	x
ended	mine	somewhat	xauthor

ending	miss	somewhere	xcal
ends	mit	soon	xnote
enough	more	spake	xother
es	moreover	spat	xsubj
especially	most	specific	y
et	mostly	speek	ye
etc	mr	speeks	year
even	mrs	spit	yes
ever	ms	spits	yet
every	much	spitting	yipee
everybody	mug	spoke	you
everyone	must	spoken	your
everything	my	sprang	yours
everywhere	myself	sprung	yourself
except	n	staves	yourselves
excepts	na	still	yu
excepted	nach	stop	z
excepting	namely	strongly	za
exception	nas	substantially	ze
exclude	near	successfully	zu
excluded	nearly	such	zum
excludes	necessarily	sui	iii
excluding	necessary	sulla	iv
exclusive	need	sung	vi
f	needs	supposing	<p>
fact	needed	sur	
facts	needing	t	
far	neither	take	
farther	nel	taken	
farthest	nella	takes	
few	never	taking	
ff	nevertheless	te	
fifty	new	ten	

finally	next	tes	
first	nine	than	
five	ninety	that	
foer	no	the	
follow	nobody	thee	
followed	none	their	



## الملخص باللغة العربية



استخراج الجمل المفتاحيه المعتمد على اندماج المقالات العلميه

إعداد

امجد كميل توفيق ايوب

المشرف الرئيسي

د. معاذ رفعت الزغول

كلية الدراسات العليا

جامعة البلقاء التطبيقية

السلط- الأردن

22 أيار, 2016

الكلمات المفتاحية توفر معلومات هامة حول مضمون الوثيقة. ويمكن أن تساعد المستخدمين من خلال البحث عن المزيد من المعلومات بشكل كاف و تبين ما إذا كانت الوثيقة ذات صلة لبحث المستخدم أو لا. ويمكن أيضا أن تستخدم الكلمات المفتاحية في مهام مختلفة مثل تصنيف النص و استرجاع المعلومات. ومع ذلك، فإن معظم الوثائق، لا توفر الكلمات المفتاحية. بالإضافة إلى ذلك، هناك حاجة لإنشاء كلمات مفتاحية لكمية كبيرة من الوثائق المكتوبة أو المنطوقة بشكل تلقائي.

على مر السنين، استخدمت العديد من الأساليب و الخوارزميات لاستخراج الكلمات المفتاحية من النص المكتوب باللغة الإنجليزية. ويركز هذا البحث على التقنية القائمة على الاندماج، لأننا وجدنا أن الجمع بين مخرجات نظم متعددة يؤدي الى بعض التحسينات. في هذا البحث، اقترحنا العديد من التقنيات القائمة على الاندماج في الجمع بين نتائج مصادر مختلفة من الكلمات المفتاحية أو دمج نتائج أساليب مختلفة في استخراج الكلمات المفتاحية و هي ثلاثة أساليب رئيسية، RAKE, TextRank, N-grams وتقوم التقنيات على دمج نتائج هذه الأساليب الثلاث لاستخراج قائمة كلمات مفتاحية أفضل من نتائج هذه الأساليب الثلاثة.

لقد تم بناء ستة تقنيات مختلفة ( All Keywords Fusion, Majority Voting, BordaVoting, ) CombMNZ, WCombMNZ, Condorcet ) تعتمد على دمج نتائج الأساليب الثلاث المذكورة أعلاه، وتختلف عنها بما يلي:

- تأخذ بعين الاعتبار موقع الكلمة في القوائم الاصلية
- استخدام تكرار الكلمة
- إعطاء وزن لكل كلمة اتباعا لخوارزميات معينه

و قد تم تقييم نتائج هذه التقنيات على مجموعة بيانات Hulth 2003 , وهي عبارة عن ملخصات مقالات علمية، و تم استخدام معايير ال Recall, Precision, F-measure في التقييم، ولقد حصلنا على نتائج محسنة، وكانت افضلها تقنية سميت WCombMNZ و حصلت على F-measure بدرجة 0.4387.