

The copyright © of this thesis belongs to its rightful author and/or other copyright owner. Copies can be accessed and downloaded for non-commercial or learning purposes without any charge and permission. The thesis cannot be reproduced or quoted as a whole without the permission from its rightful owner. No alteration or changes in format is allowed without permission from its rightful owner.



**HYBRID MODEL OF POST-PROCESSING TECHNIQUES FOR  
ARABIC OPTICAL CHARACTER RECOGNITION**



**IMAD QASIM HABEEB**

**UUM**  
**Universiti Utara Malaysia**

**DOCTOR OF PHILOSOPHY  
UNIVERSITI UTARA MALAYSIA**

**2016**



Awang Had Salleh  
Graduate School  
of Arts And Sciences

Universiti Utara Malaysia

**PERAKUAN KERJA TESIS / DISERTASI**  
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa  
(We, the undersigned, certify that)

**IMAD QASIM HABEEB**

calon untuk Ijazah \_\_\_\_\_ PhD  
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:  
(has presented his/her thesis / dissertation of the following title):

**"HYBRID MODEL OF POST-PROCESSING TECHNIQUES FOR ARABIC  
OPTICAL CHARACTER RECOGNITION"**

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.  
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : 28 Julai 2016.

*That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:  
July 28, 2016.*

Pengerusi Viva:  
(Chairman for VIVA)

**Assoc. Prof. Dr. Haslina Mohd**

Tandatangan  
(Signature)

Pemeriksa Luar:  
(External Examiner)

**Assoc. Prof. Dr. Shahnorbanun Sahran**

Tandatangan  
(Signature)

Pemeriksa Dalam:  
(Internal Examiner)

**Assoc. Prof. Dr. Faudziah Ahmad**

Tandatangan  
(Signature)

Nama Penyelia/Penyelia-penyelia:  
(Name of Supervisor/Supervisors)

**Dr. Shahrul Azmi Mohd Yusof**

Tandatangan  
(Signature)

Nama Penyelia/Penyelia-penyelia:  
(Name of Supervisor/Supervisors)

**Assoc. Prof. Dr. Yuhanis Yusof**

Tandatangan  
(Signature)

Tarikh:

(Date) July 28, 2016

## **Permission to Use**

In presenting this thesis in fulfillment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for the scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part should be addressed to:



Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

## Abstrak

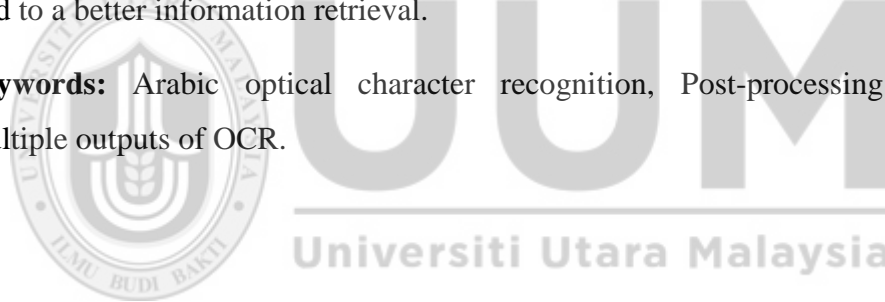
Pengecaman aksara optik (OCR) digunakan untuk mengeluarkan teks yang terkandung di dalam sesuatu imej. Salah satu fasa dalam OCR ialah prapemprosesan dan ianya membetulkan kesalahan teks yang terasil dari OCR. Kaedah berbilang output dalam OCR mengandungi tiga proses iaitu: pembezaan, penjajaran dan pengundian. Teknik pembezaan yang sedia ada mengalami kehilangan ciri-ciri penting kerana ia menggunakan N-versi imej sebagai input. Dalam pada itu, teknik penjajaran yang terdapat dalam kajian adalah berdasarkan penghampiran manakala proses pengundian adalah tidak peka kepada konteks. Kekangan-kekangan ini mengakibatkan kadar ralat yang tinggi dalam OCR. Kajian ini telah mencadangkan tiga teknik pembezaan, penjajaran dan pengundian yang ditambahbaik untuk mengatasi kekurangan yang telah dikenalpasti;. Kesemua teknik ini kemudiannya digabungkan dalam satu model hibrid yang boleh mengecam aksara optik dalam Bahasa Arab. Setiap teknik yang dicadangkan telah dibandingkan dengan tiga teknik berkaitan yang sedia ada secara berasingan. Ukuran prestasi yang digunakan adalah kadar ralat perkataan (WER), kadar ralat aksara (CER) dan kadar ralat bukan perkataan (NWER). Keputusan eksperimen menunjukkan pengurangan relatif kadar ralat dalam semua ukuran untuk teknik-teknik yang telah dinilai. Secara yang serupa, model hibrid juga telah memperolehi nilai WER, CER dan NWER yang lebih rendah iaitu sebanyak 30.35%, 52.42% dan 47.86% apabila dibandingkan dengan tiga model relevan yang sedia ada. Kajian ini menyumbang kepada domain OCR kerana model hibrid yang dicadangkan bagi teknik pasca pemprosesan boleh membantu pengecaman teks Bahasa Arab secara automatik. Oleh itu, ia akan menjurus kepada capaian maklumat yang lebih baik.

**Kata Kunci:** Pengecaman aksara optic Bahasa Arab, teknik pasca pemprosesan, OCR berbilang ouput.

## Abstract

Optical character recognition (OCR) is used to extract text contained in an image. One of the stages in OCR is the post-processing and it corrects the errors of OCR output text. The OCR multiple outputs approach consists of three processes: differentiation, alignment, and voting. Existing differentiation techniques suffer from the loss of important features as it uses N-versions of input images. On the other hand, alignment techniques in the literatures are based on approximation while the voting process is not context-aware. These drawbacks lead to a high error rate in OCR. This research proposed three improved techniques of differentiation, alignment, and voting to overcome the identified drawbacks. These techniques were later combined into a hybrid model that can recognize the optical characters in the Arabic language. Each of the proposed technique was separately evaluated against three other relevant existing techniques. The performance measurements used in this study were Word Error Rate (WER), Character Error Rate (CER), and Non-word Error Rate (NWER). Experimental results showed a relative decrease in error rate on all measurements for the evaluated techniques. Similarly, the hybrid model also obtained lower WER, CER, and NWER by 30.35%, 52.42%, and 47.86% respectively when compared to the three relevant existing models. This study contributes to the OCR domain as the proposed hybrid model of post-processing techniques could facilitate the automatic recognition of Arabic text. Hence, it will lead to a better information retrieval.

**Keywords:** Arabic optical character recognition, Post-processing techniques, Multiple outputs of OCR.



## **Acknowledgement**

Each part of this study is guided, inspired, and supported by many people. Firstly, I would like to thank all the members of my family especially my mother for their unconditional support. My goal would not be achieved without them. The most important support and guidance were from my research supervisors Dr. Shahrul Azmi Mohd Yusof and Assoc. Prof. Dr. Yuhanis Binti Yusof. Thank you very much for your great help and support. It is an honor for me to do a research under your supervisions. I would like to thank all the academic and technical staff in Utara Universiti Malaysia for their help in the study process and providing all the excellent facilities. Finally, I would like to thank the Ministry of Higher Education and Scientific Research in Iraq for financial sponsorship.



## Table of Contents

Permission to Use.....	i
Abstrak .....	ii
Abstract .....	iii
Acknowledgement.....	iv
Table of Contents .....	v
List of Tables.....	ix
List of Figures .....	x
Glossary of Term.....	xii
List of Abbreviations.....	xiii
<b>CHAPTER ONE INTRODUCTION .....</b>	<b>1</b>
1.0 Background .....	1
1.1 Problem Statement .....	8
1.2 Research Questions .....	11
1.3 Research Objectives .....	11
1.4 Significance of the Research .....	12
1.5 Scope of the Research .....	13
1.6 Organization of the Research .....	14
<b>CHAPTER TWO LITERATURE REVIEW .....</b>	<b>16</b>
2.0 Introduction .....	16
2.1 Arabic OCR.....	16
2.1.1 Overview of the Arabic Language .....	17
2.1.2 Arabic OCR Limitations .....	17
2.1.3 Characteristics of the Arabic Language .....	18
2.2 OCR Post-Processing Stage (PPS).....	22
2.2.1 OCR PPS Error .....	22
2.2.2 Functions of OCR PPS Techniques .....	24
2.2.3 Categories of the OCR PPS Correction.....	25
2.3 OCR PPS Techniques .....	25
2.3.1 Multiple Outputs OCR (MO) .....	25
2.3.1.1 Differentiation Process.....	26
2.3.1.2 Alignment Process .....	29
2.3.1.3 Voting Process .....	32



2.3.2 N-grams Language Model.....	34
2.3.2.1 N-grams Language Model Functions.....	34
2.3.2.2 N-grams Language Model for Arabic .....	37
2.3.3 Levenshtein Distance .....	38
2.3.4 Rules-Based Technique.....	40
2.3.5 Noisy Channel Model .....	42
2.3.6 N-gram Distance .....	43
2.3.7 Lexicon.....	45
2.4 Comparison of OCR Post-processing Techniques .....	46
2.5 Hybrid Techniques of OCR PPS .....	48
2.6 Summary .....	51
<b>CHAPTER THREE RESEARCH METHODOLOGY .....</b>	<b>52</b>
3.0 Introduction .....	52
3.1 Research Phases .....	52
3.2 Theoretical Study .....	53
3.3 Design Phase .....	54
3.3.1 Differentiation Technique .....	54
3.3.2 Alignment Technique.....	58
3.3.3 Voting Technique.....	61
3.3.4 Hybrid Model.....	64
3.4 Development Phase.....	65
3.5 Evaluation .....	66
3.5.1 Data Collection.....	67
3.5.1.1 Testing Dataset.....	67
3.5.1.2 Training Dataset.....	68
3.5.2 Experimental Design.....	69
3.5.2.1 Differentiation Technique Evaluation.....	69
3.5.2.2 Alignment Technique Evaluation .....	70
3.5.2.3 Voting Technique Evaluation .....	72
3.5.2.4 Hybrid Model Evaluation.....	73
3.5.3 Measurements .....	74
3.5.4 Statistical Test .....	75
3.6 Summary .....	76

<b>CHAPTER FOUR PROPOSED DIFFERENTIATION TECHNIQUE.....</b>	<b>78</b>
4.0 Introduction .....	78
4.1 Differentiation Technique (EDT) Concept .....	78
4.2 EDT Algorithm .....	84
4.3 Experimental Results .....	86
4.3.1 Word Error Rate (WER) .....	86
4.3.2 Character Error Rate (CER) .....	89
4.3.3 Non-Word Error Rate (NWER) .....	91
4.3.4 Results Discussion .....	94
4.4 Summary .....	95
<b>CHAPTER FIVE PROPOSED ALIGNMENT TECHNIQUE .....</b>	<b>96</b>
5.0 Introduction .....	96
5.1 Alignment Technique (AWS) Concept .....	96
5.2 AWS Algorithm .....	100
5.3 AWS Contributions .....	102
5.4 Experimental Results .....	103
5.4.1 Word Error Rate (WER) .....	103
5.4.2 Character Error Rate (CER) .....	106
5.4.3 Non-Word Error Rate (NWER) .....	108
5.4.4 Results Discussion .....	111
5.5 Summary .....	112
<b>CHAPTER SIX PROPOSED VOTING TECHNIQUE .....</b>	<b>113</b>
6.0 Introduction .....	113
6.1 Voting Technique (VCI) Concept .....	113
6.2 VCI Algorithm .....	115
6.3 VCI Contributions .....	117
6.4 Experimental Results .....	119
6.4.1 Word Error Rate (WER) .....	119
6.4.2 Character Error Rate (CER) .....	122
6.4.3 Non-Word Error Rate (NWER) .....	124
6.4.4 Results Discussion .....	126
6.5 Summary .....	127
<b>CHAPTER SEVEN PROPOSED HYBRID MODEL .....</b>	<b>129</b>

7.0 Introduction .....	129
7.1 Interaction in the Hybrid Model (HMNL) .....	129
7.2 Arabic Challenges .....	132
7.2.1 N-gram Language Model Challenges .....	132
7.2.2 Diacritics .....	140
7.3 Experimental Results .....	141
7.3.1 Word Error Rate (WER) .....	141
7.3.2 Character Error Rate (CER) .....	144
7.3.3 Non-Word Error Rate (NWER) .....	147
7.3.4 Results Discussion .....	149
7.4 Summary .....	150
<b>CHAPTER EIGHT CONCLUSION .....</b>	<b>151</b>
8.0 Introduction .....	151
8.1 Achievement .....	151
8.2 Research Contributions .....	152
8.2 Research Limitations .....	154
8.3 Future Work .....	154
8.3 Summary .....	155
<b>REFERENCES .....</b>	<b>157</b>

## List of Tables

Table 1.1 Some characteristics of Arabic language .....	3
Table 2.1 Shapes of some diacritics in Arabic .....	20
Table 2.2 Differentiation techniques in multiple outputs of OCR.....	26
Table 2.3 Voting techniques in multiple outputs of OCR.....	32
Table 2.4 Limitations of the OCR post-processing techniques. ....	47
Table 2.5 Some techniques used in the OCR post-processing stage.....	49
Table 3.1 Major variables resulted from ANOVA.....	76
Table 4.1 Experimental results of the EDT evaluation using the WER metric. ....	86
Table 4.2 Experimental results of the EDT evaluation using the CER metric. ....	89
Table 4.3 Experimental results of the EDT evaluation using the NWER metric .....	92
Table 5.1 Comparison between AWS technique and other existing techniques.....	102
Table 5.2 Experimental results of the AWS evaluation using the WER metric. ....	103
Table 5.3 Experimental results of the AWS evaluation using the CER metric. ....	106
Table 5.4 Experimental results of the AWS evaluation using the NWER metric. ....	109
Table 6.1 Voting process example. ....	114
Table 6.2 Comparison between VCI technique and other existing techniques.....	118
Table 6.3 Experimental results of the VCI evaluation using the WER metric .....	119
Table 6.4 Experimental results of the VCI evaluation using the CER metric. ....	122
Table 6.5 Experimental results of the VCI evaluation using the NWER metric. ....	124
Table 7.1 Special tokens in the classification stage .....	136
Table 7.2 Example of how to store sentences in Unigram table.....	137
Table 7.3 Example of how to store sentences in Bigram table .....	138
Table 7.4 Example of how to store sentences in Trigram table. ....	138
Table 7.5 Type and size of columns of tables in N-gram language model.....	138
Table 7.6 Comparison between three Arabic corpora.....	139
Table 7.7 Experimental results of the HMNL evaluation using the WER metric. ....	142
Table 7.8 Experimental results of the HMNL evaluation using the CER metric. ....	144
Table 7.9 Experimental results of the HMNL evaluation using the NWER metric. ..	147

## List of Figures

Figure 1.1. The input and output of OCR system. ....	1
Figure 1.2. Categories of OCR systems. ....	2
Figure 1.3. Stages of OCR system with output of each stage. ....	5
Figure 1.4. Multiple outputs of OCR. ....	6
Figure 1.5. Alignment process. ....	7
Figure 1.6. The scope of this research. ....	14
Figure 2.1. Connectivity in Arabic writing ....	19
Figure 2.2. Overlapping in Arabic writing ....	19
Figure 2.3. Diacritics in Arabic writing ....	21
Figure 2.4. Multiple Thresholds technique. ....	28
Figure 2.5. Alignment process. ....	30
Figure 2.6. Simple example on alignment process. ....	31
Figure 2.7. Levenshtein distance example. ....	39
Figure 2.8. Noisy channel model. ....	43
Figure 2.9. Bigram distance example. ....	44
Figure 3.1. Research phases. ....	53
Figure 3.2. Flowchart of Multiple Thresholds technique. ....	55
Figure 3.3. Flowchart of the proposed differentiation technique (EDT). ....	57
Figure 3.4. Flowchart of the existing alignment technique. ....	58
Figure 3.5. Simple example of character alignment algorithm. ....	58
Figure 3.6. Flowchart of the proposed alignment technique (AWS). ....	60
Figure 3.7. Flowchart of the existing voting technique. ....	62
Figure 3.8. Flowchart of the proposed voting technique (VCI) ....	63
Figure 3.9. Whole evaluation process. ....	66
Figure 3.10. Sample image selected from the testing dataset. ....	68
Figure 3.11. Experiments used to evaluate the proposed differentiation technique. .	69
Figure 3.12. Experiments used to evaluate the proposed alignment technique. ....	71
Figure 3.13. Experiments used to evaluate the proposed voting technique. ....	72
Figure 3.14. Experiments used to evaluate the proposed hybrid model. ....	73
Figure 4.1. Differentiation function and its implementation. ....	79
Figure 4.2. Simple example on differentiation cycle for a primary starting pixel. ....	81
Figure 4.3. Simple example of proposed differentiation technique. ....	83

Figure 4.4. Clustered column graph for the WER values listed in Table 4.1. ....	87
Figure 4.5. ANOVA-test results for the WER values. ....	88
Figure 4.6. Clustered column graph for the CER values listed in Table 4.2. ....	89
Figure 4.7. ANOVA-test results for the CER values ....	91
Figure 4.8. Clustered column graph for the NWER values listed in Table 4.3. ....	92
Figure 4.9. ANOVA-test results for the NWER values ....	93
Figure 5.1. Loss of words' locations in MO of OCR.....	97
Figure 5.2. Extraction of words' images in the existing techniques ....	98
Figure 5.3. Extraction of words' images in the proposed technique.....	98
Figure 5.4. Clustered column graph for the WER values listed in Table 5.2 ....	104
Figure 5.5. ANOVA-test results for the WER values ....	105
Figure 5.6. Clustered column graph for the CER values listed in Table 5.3 ....	106
Figure 5.7. ANOVA-test results for the CER values ....	108
Figure 5.8. Clustered column graph for the NWER values listed in Table 5.4 ....	109
Figure 5.9. ANOVA-test results for the NWER values ....	110
Figure 6.1. Clustered column graph for the WER values listed in Table 6.3 ....	120
Figure 6.2. ANOVA-test results for the WER values ....	121
Figure 6.3. Clustered column graph for the CER values listed in Table 6.4 ....	122
Figure 6.4. ANOVA-test results for the CER values ....	123
Figure 6.5. Clustered column graph for the NWER values listed in Table 6.5 ....	125
Figure 6.6. ANOVA-test results for the NWER values ....	126
Figure 7.1. The interaction in the proposed hybrid model.....	130
Figure 7.2. Extract Arabic text from Wikipedia database.....	134
Figure 7.3. Database structure of N-gram language model ....	136
Figure 7.4. Clustered column graph for the WER values listed in Table 7.7 ....	142
Figure 7.5. ANOVA-test results for the WER values ....	143
Figure 7.6. Clustered column graph for the CER values listed in Table 7.8 ....	145
Figure 7.7. ANOVA-test results for the CER values ....	146
Figure 7.8. Clustered column graph for the NWER values listed in Table 7.9 ....	147
Figure 7.9. ANOVA-test results for the NWER values ....	148

## Glossary of Term

**Symbol:** represents the smallest meaningful unit in a writing system, such as character, number, comma, signs, etc.

**Token:** a sequential group of symbols not containing any spaces. It consists of a small number of symbols.

**String:** a sequential group of symbols. It can consist of a large number of symbols including spaces.

**Word:** a token exists in the specific language.

**Cursive Token:** a token has a group of characters joined together.

**Non-word error:** occurs when the word produced from the OCR process does not exist in the language resource.

**Real word error:** occurs when the word produced from the OCR process exists in the language resource, but it does not match with the source text.

**Wrong-word:** also known as an incorrect word. It refers to either non-word error or real word error.

**Document Image:** represents any image containing a text.

**Model:** a symbolic representation of concepts. It can be a schematic model or mathematical.

**Lexicon:** a list of words that belongs to a specific language. It does not contain any information to describe the words.

## List of Abbreviations

<b>OCR</b>	Optical character recognition
<b>HR</b>	Handwriting recognition
<b>MO</b>	Multiple outputs of OCR
<b>LD</b>	Levenshtein distance
<b>PCA</b>	ProbCons alignment
<b>SWA</b>	Smith–Waterman alignment
<b>LDB</b>	Levenshtein distance with backtrack
<b>WER</b>	Word error rate
<b>CER</b>	Character error rate
<b>NWER</b>	Non-word error rate
<b>EDT</b>	Enhanced differentiation technique
<b>ASW</b>	Alignment by using words separation
<b>VCI</b>	Voting by using context information of sentences
<b>CGLL</b>	Candidates' list generation by using N-gram language model and LD
<b>HMNL</b>	A hybrid of MO, N-gram language model, and Levenshtein distance.
<b>MOUMT</b>	Multiple outputs using multiple threshold values
<b>MOUMS</b>	Multiple outputs using multiple scanning
<b>MOUMO</b>	Multiple outputs using multiple OCR systems
<b>PPS</b>	Post-processing stage

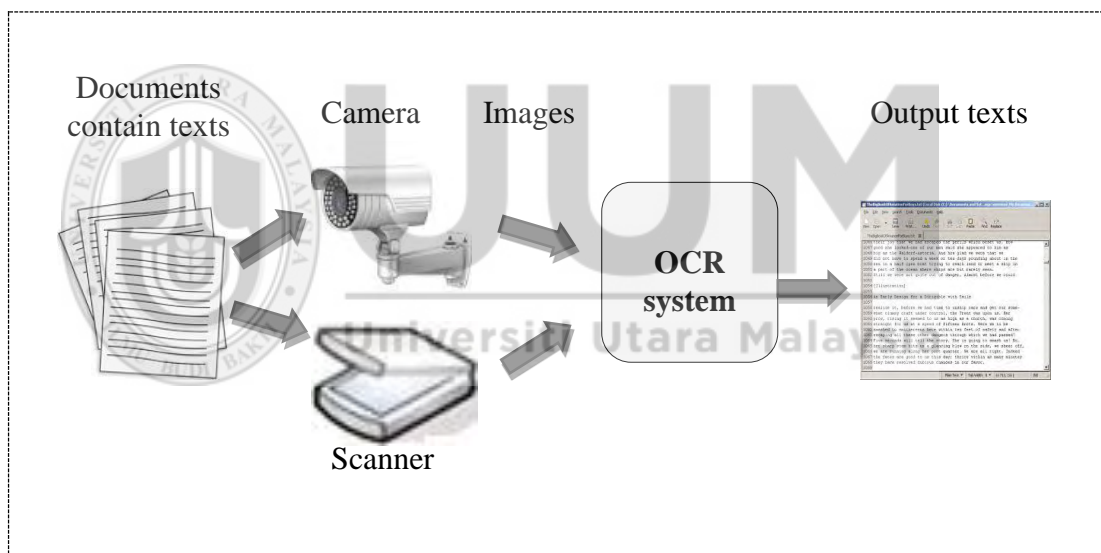


# CHAPTER ONE

## INTRODUCTION

### 1.0 Background

An optical character recognition, commonly referred to as OCR, is used to extract and recognize texts within images (Bassil & Alwani, 2012c). Several commercial OCR systems are currently available for various purposes, such as mail sorting systems, plate number recognition systems (Singh, Bacchuwar, & Bhasin, 2012). Figure 1.1 shows the input and output of an OCR system.



*Figure 1.1. The input and output of an OCR system*

There are four categories of OCR systems (El-Mahallawy, 2008). The first category is based on the type of input to these systems: offline or online. The second category depends on the mode of writing: handwritten or machine printed. The third category depends on the connectivity of a text: isolated symbols or cursive words. The last category depends on font restrictions: single font or Omni-font (Al-Badr &

Mahmoud, 1995; El-Mahallawy, 2008). Figure 1.2 shows the categories of OCR systems.

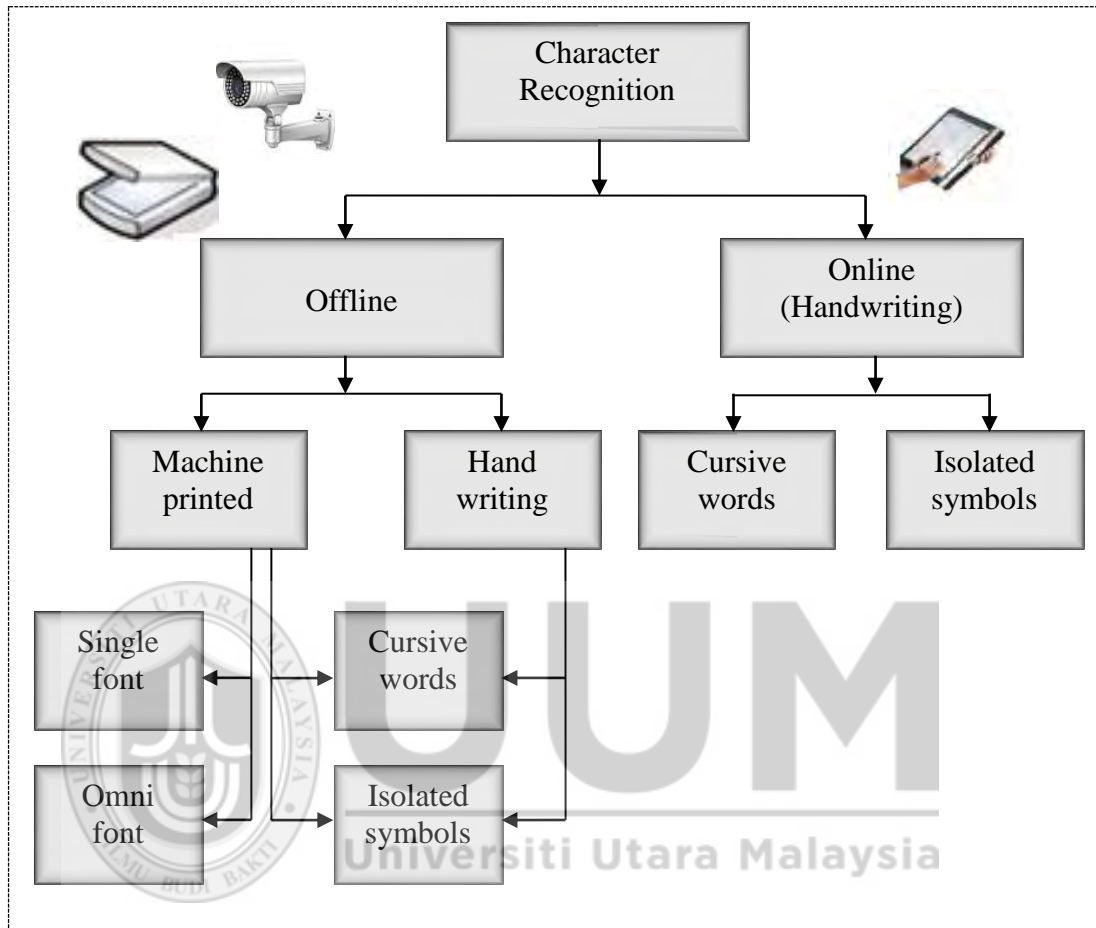


Figure 1.2. Categories of OCR systems, Source: El-Mahallawy (2008).

The offline type of OCR system receives an image as input from a file, a camera, or a digital scanner. It manipulates the image after it is completely captured (El-Mahallawy, 2008). On the other hand, the online type of OCR system receives its input data from devices, such as a tablet, in real time. It displays each separated character or cursive word after it is drawn (AL-Shatnawi, AL-Salaimeh, AL-Zawaideh, & Omar, 2011). The recognition of text in an image by online OCR is better than an offline system. The reason is that the OCR engine in real time can

detect a lot of information, such as the direction of writing, starting points and stopping points of text symbols, etc (Bassil & Alwani, 2012c; El-Mahallawy, 2008).

The accuracy of OCR systems is still considered an open problem in the following cases. The first case is for the cursive written-based languages like Arabic, Persian, Kurdish and Urdu (Al-Masoudi & Al-Obeidi, 2015; Al-Zaydi & Salam, 2015; Bassil & Alwani, 2012c). The second case is for low scanning resolution image (Ma & Agam, 2012, 2013). The last case is when the image contains noise (Herceg, Huyck, Johnson, Van Guilder, & Kundu, 2005; Lund, Ringger, & Walker, 2014). Arabic has an OCR error rate greater than Latin character-based languages. This is due to the unique characteristics of this language (Abulnaja & Batawi, 2012; AL-Shatnawi et al., 2011; El-Mahallawy, 2008). Some of the characteristics of the Arabic language are shown in Table 1.1.

Table 1.1

*Some characteristics of the Arabic language*

Characteristics	Example in Arabic
Cursive Arabic writing	إياك نعبد و إياك نستعين
Three groups of dots	المشرق , المغرب
Hamza ( ء )	أصبرهم , أرزقهم
Madda ( ~ )	دأبة , السماء
Diacritics	بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
Overlapping	إلا , بجبل

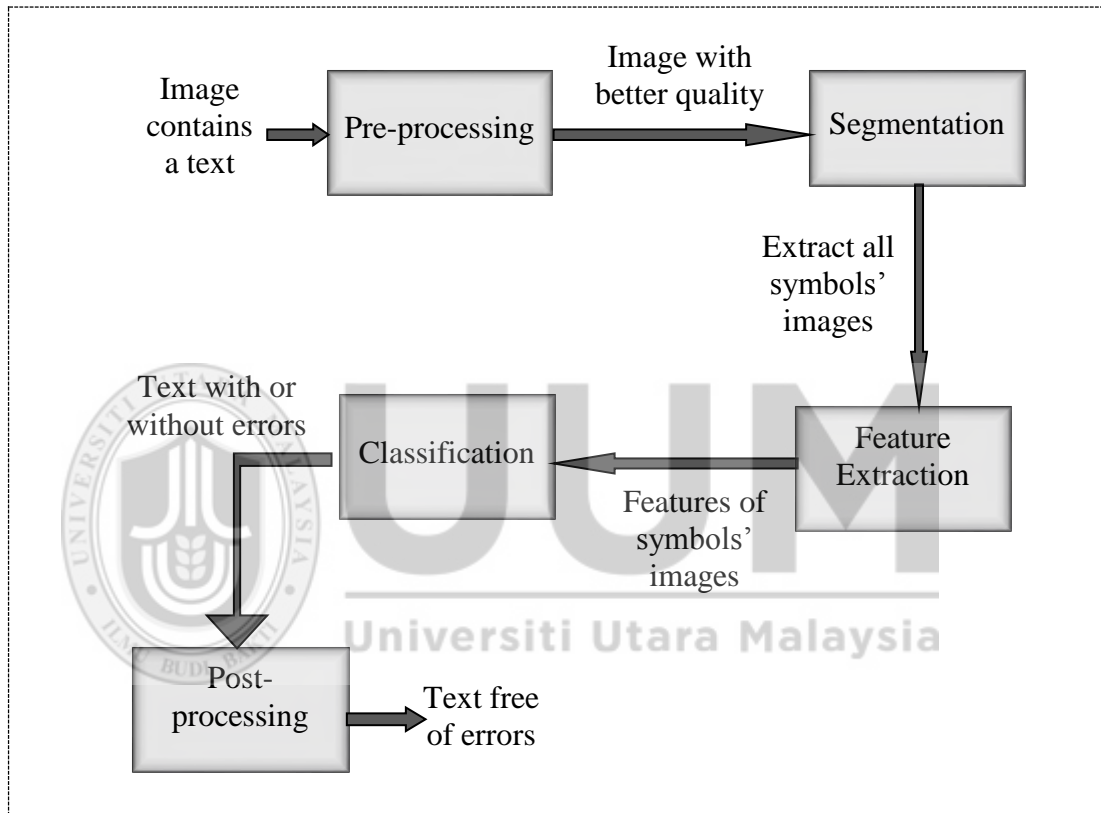
Source: El-Mahallawy (2008).

The first property in this table shows that Arabic is a cursive language, and its direction is written from right to left. Furthermore, the shape of its letters varies depending on their position in the word. The second property shows that some of the Arabic letters have dots. The third property shows that some characters may have a *Hamza* "ء". The fourth property shows that some characters in some situations have a *Madda* "~". The fifth property is diacritics, which consist of signs located above or below the characters of a word. The last property shows that there is a vertical overlap between the letters (AL-Shatnawi et al., 2011; El-Mahallawy, 2008).

On the other hand, the error rates of cursive written-based languages, low scanning resolution images, and noise images are varied from one to another. For examples, cursive written-based languages have OCR error rate ranges from 38.83% to 47.88% (Al-Masoudi & Al-Obeidi, 2015), while low scanning resolution images have OCR error rate ranges from 21.5% (Mai, Huynh, & Doan, 2014) to 35% (Ma & Agam, 2013). Lastly, the error rate of OCR for noise images can even reach up to 100% (Lund, Kennard, & Ringger, 2013b). The values of the OCR error rates mentioned above are approximate because they depend on three factors: size and type of testing dataset (El-Mahallawy, 2008), value of scanning resolution of the images (Ma & Agam, 2013) and types of noises in the image (Ahmad, Mahmoud, & Fink, 2016; Lund & Ringger, 2009).

Although the OCR error rate exceeds 1% for the cases mentioned above but by assuming it is equal to this value, it means two errors in forty words, assuming each word contains five letters. Therefore, in a normal book with 100,000 words, which is equal to 500,000 characters, it leads to 5,000 corrections, which is difficult to process manually. Another example, if the character error rate of the OCR is 10%, it means

that twenty are errors in forty words, and 50,000 errors in a normal book containing 100,000 words (Barnes, 2011). The OCR system usually consists of five stages: preprocessing, segmentation, feature extraction, classification, and post-processing (El-Mahallawy, 2008). Figure 1.3 shows the stages of the OCR system with the output of each stage.



*Figure 1.3. Stages of the OCR system, Source: El-Mahallawy (2008).*

The goal of the preprocessing stage is to improve the quality of the original image to make it more suitable for the operations of the OCR systems. In the segmentation stage, all the symbols' images of a text are extracted and isolated from the original image. The symbols are units of a writing system for a specific language, such as characters, numbers, signs, etc. The feature extraction stage is a process of identifying useful information from the symbols' images. In the classification stage, the extracted features of unknown images of symbols are compared with predefined

stored samples in order to identify their type. Lastly, the post-processing stage will check and correct the resulting text from the classification stage of the OCR to make sure it is free from errors (AL-Shatnawi et al., 2011).

Figure 1.3 shows OCR stages when there is a single OCR output. However, in recent years, post-processing stage was improved by including multiple outputs of OCR (Al Azawi, 2015; Lund, 2014; Volk, Furrer, & Sennrich, 2011). The idea of multiple outputs is to look for differences between several outputs of OCR and then choose the best among them. Figure 1.4 shows multiple outputs model based on the work of Lund (2014).

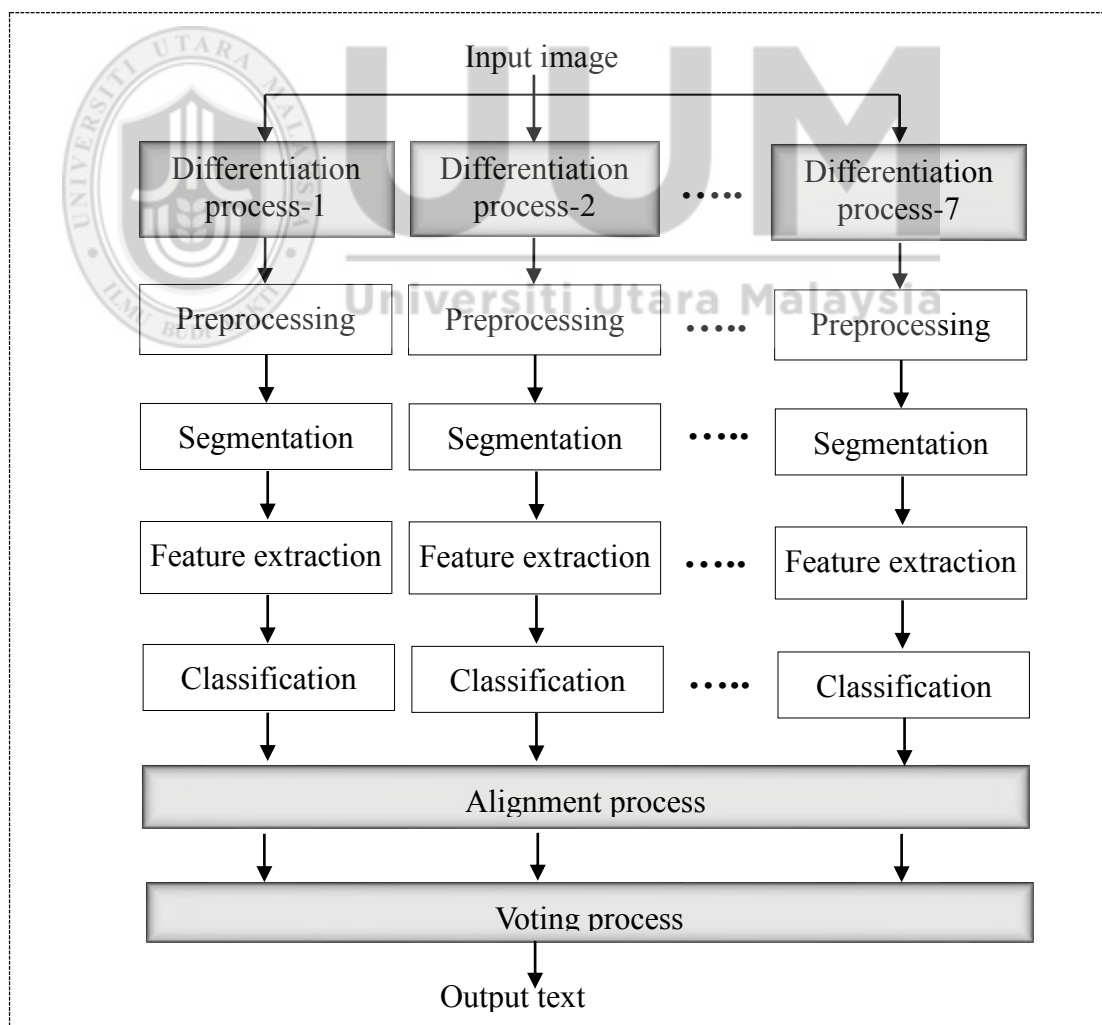


Figure 1.4. Multiple outputs of OCR

Figure 1.4 shows that differentiation process is used to produce 7 versions of the input image. These versions are similar but not identical. The goal of the differentiation process is to generate differences between the versions entering the following OCR stages to produce different outputs. The use of multiple outputs (MO) leads to an alignment problem of the resulting texts (Cai, 2013; Lund et al., 2013b; Lund, Walker, & Ringger, 2011). Figure 1.5 displays an alignment process between the resulting texts of three different OCR engines.

Image	<div>Arabic language is a complex</div>																											
Output of OCR1	Arab'c langu ge is comaie x																											
Output of OCR2	Anbic language u a camplc																											
Output of OCR3	Arabic language is a complex																											
Alignment of three OCR outputs																												
A	r	a	b	'	c		l	a	n	g	u		g	e		i	s			c	o	m	a	i	e	x		
A	n		b	i	c			l	a	n	g	u	a	g	e		u	a			c	a	m	p	l	c		
A	r	a	b	i	c			l	a	n	g	u	c	g	e		i	s	a			c	o	m	p	l	e	x

Figure 1.5. Alignment process

Figure 1.5 shows a different number of characters resulting from three different OCR engines. This causes vertical words overlapping between the OCR resulting texts. Therefore, an alignment process is required to align each character with corresponding in other OCR outputs. After the alignment process, a voting process is used to select the best word between the MO of the OCR.

In addition to the multiple outputs of OCR, there are other techniques and models used in the OCR post-processing stage, such as Levenshtein distance (Daðason,

2012; Magdy & Darwish, 2008; Naseem, 2004) and N-gram language model (Al-Masoudi & Al-Obeidi, 2015; Bassil & Alwani, 2012c; Kanoun, Alimi, & Lecourtier, 2011). The Levenshtein distance is used to measure the difference between two strings (Daðason, 2012). In the OCR post-processing stage, the Levenshtein distance is used to generate a candidates' list, and to select the best between them for each wrong word in the OCR output text (Naseem, 2004).

The N-gram language model is used to provide the probability for a sequence of words. The probability depends on the frequency of words or frequency of sentences in a large corpus (Daniel Jurafsky & Martin, 2009). Detection of a wrong word occurs if the language model does not provide the probability of the sentence. A correction of this error is based on the high probability of other sentences. A large corpus is needed in order to build an accurate language model (Daðason, 2012; Daniel Jurafsky & Martin, 2009).

To sum up, existing techniques in OCR post-processing still require improvement. This is due to the high error rate, and for Arabic OCR, it can reach up to 47.88% (Al-Masoudi & Al-Obeidi, 2015). If post-processing techniques can be improved, then the load on other OCR stages may be reduced, and the total accuracy of the OCR can be increased (Alex, Grover, Klein, & Tobin, 2012; Bassil & Alwani, 2012c; El-Mahallawy, 2008; Goswami & Sharma, 2013; Lund et al., 2013b; Lund & Ringger, 2011; Ma & Agam, 2013; Saber, Ahmed, Elsis, & Hadhoud, 2016).

## **1.1 Problem Statement**

OCR accuracy is still considered an open problem for the Arabic language, even if the images are noise-free and have high scanning resolutions (Akila et al., 2015; Al-



Masoudi & Al-Obeidi, 2015; Shafii, 2014). The multiple outputs technique used in OCR post-processing has shown to be able to reduce the error rate as compared to the using a single output (Batawi & Abulnaja, 2012; Lund et al., 2013b; Volk et al., 2011). This technique incorporates three processes; differentiation, alignment, and voting. Nevertheless, these processes suffer from several drawbacks and they are discussed in the following paragraphs.

In the differentiation process, Lund (2014) produced seven OCR outputs using seven threshold values. The utilization of several threshold values leads to the loss of some important features in the characters' images. This is because the pixels, which are above the threshold value, will be identified as white. Hence, resulting the increment of a number of wrong words in the OCR outputs (Al-Zaydi & Salam, 2015). On the other hand, the usage of different classifiers to generate several OCR outputs as demonstrated by Kittler, Hatef, Duin, and Matas (1998) may reduce the performance of the best classifier. This is similar to the technique of employing different OCR software (Al Azawi, 2015; Lund, 2014; Lund et al., 2013b; Volk et al., 2011). However, Al-Zaydi and Salam (2015) reported that combining different OCR software is considered a complex and manual process because it requires handling each OCR software manually. Another technique is based on scanning the input image for three times as presented by Al-Zaydi and Salam (2015). However, scanning image three times is considered a boring process. Furthermore, the resulting difference from scanning image three times does not greatly reduce OCR error rate. Therefore, an enhanced differentiation technique is required that does not require combining different OCR software or different classifiers. Furthermore, it should improve the technique of Lund (2014) by reducing the effect of losing important features from the characters' images in order to decrease OCR error rate.

Existing techniques for aligning the multiple outputs OCR are based on approximation (Al-Zaydi & Salam, 2015; Cai, 2013; Lund, 2014; Lund et al., 2013b; Lund et al., 2011; Pervez et al., 2014). This means that the resulting texts of the MO may contain errors. In detail, these alignment techniques require high computer resources due to the executing character alignment algorithm between each pair of OCR outputs (Al-Zaydi & Salam, 2015; Lund, 2014; Lund et al., 2013b; Lund et al., 2011; Pervez et al., 2014; Volk et al., 2011). For example, to align only two OCR outputs that contain 5000 character each, it requires creating two matrixes of size together ( $2 * (5000 \text{ rows} * 5000 \text{ columns})$ ), which is equal to the 50,000,000 cells reserved in the main memory of a computer (Do, Mahabhashyam, Brudno, & Batzoglu, 2005; Lund, 2014). Furthermore, the executing character alignment algorithm requires more computer resources for aligning three OCR outputs or more. Therefore, an alignment technique is required to make the alignment process is exact and to prevent executing any character alignment algorithm.

In the voting process, past researchers, such as Volk et al. (2011), Lund et al. (2014), Lund (2014), and Al-Zaydi and Salam (2015) focused on lexicon-based voting. The technique is not reliable as it is not context-aware. For example, by assuming the sentence “*a good \_\_\_\_\_ can't sleep*” has three OCR outputs: “cop”, “cap”, and “cup” to complete it. Hence, it is difficult to choose an appropriate word for the sentence because all these words are found in the lexicon. Therefore, an enhanced voting technique based on context information of sentence is required.

Based on the mentioned weakness, improvements on the post-processing techniques are needed to increase the accuracy of OCR. This can be achieved by combining the strengths of existing techniques of the OCR post-processing.

## 1.2 Research Questions

In order to improve the accuracy of the OCR for the Arabic language, several questions need to be addressed:

- i. What are the strengths and weakness of the existing OCR post-processing techniques?
- ii. How to design an enhanced differentiation technique to produce better OCR outputs than what is obtained from the existing techniques?
- iii. How to design an alignment technique that prevents overlapping words between the multiple outputs of the OCR?
- iv. How to design an enhanced voting technique that selects the best word from multiple outputs of OCR?
- v. How to integrate the proposed differentiation, alignment and voting techniques into a model to reduce the error rate for the Arabic OCR?

## 1.3 Research Objectives

The goal of this study is to develop a hybrid model of the OCR post-processing techniques in order to improve the accuracy of the Arabic OCR. The specific objectives are:

- i. To design an enhanced differentiation technique that produces higher OCR accuracy.
- ii. To design an alignment technique that prevents word overlapping between the multiple outputs of the OCR.

- iii. To design an enhanced voting technique based on N-gram language model and Levenshtein algorithm that selects context-aware words from the multiple outputs of OCR.
- iv. To develop and evaluate a hybrid model of the OCR post-processing techniques in terms of word error rate, character error rate, and non-word error rate.

#### **1.4 Significance of the Research**

As a direct effect of this study, a new hybrid model of OCR post-processing techniques for the Arabic language has been developed. This model is designed by combining the strength of the MO of the OCR, N-gram language model, and Levenshtein distance. This combination improves the accuracy of OCR for the Arabic language. Furthermore, this model can be used by other cursive languages that use Arabic characters in writing, such as Persian, Kurdish, and Urdu. In addition to that, this model can be also used by Latin-based languages directly, or with some modifications to improve the OCR accuracy for noise and low scanning resolution images.

The significance of this research as well includes three important techniques. The first is the enhanced differentiation technique which is used to produce better OCR outputs than what are obtained from the existing techniques. The second is the novel alignment technique which is used to prevent words overlapping between the MO of the OCR. The last is the enhanced voting technique which is used to select the best words from the multiple outputs of OCR. On the other hand, a comparison study of the weakness and strengths of the OCR post-processing techniques is also presented.

The contributions of this study will lead to increase the OCR knowledge for the researchers, and can be used for more improvement in this topic.

As a practical effect of this study, the Arabic language is spoken by over 330 million people in 22 Arab countries (Farghaly & Shaalan, 2009). Therefore, developing OCR systems for this language can serve hundreds of millions of people in the world. Furthermore, improving the accuracy of OCR, allows programs of mobile and other applications to recognize Arabic text accurately.

### **1.5 Scope of the Research**

This study followed mainly the Ph.D. research produced by Lund (2014) in treating the OCR engines as black boxes. Furthermore, it also followed most related work in OCR post-processing error correction in which no attempt is made to directly modify the techniques of other OCR stages (Al-Masoudi & Al-Obeidi, 2015; Al-Zaydi & Salam, 2015; Bassil & Alwani, 2012c; Dağason, 2012; Ramanan, Ramanan, & Charles, 2014; Volk et al., 2011). In other words, this study has only focused on improving the most important OCR post-processing techniques as shown in Figure 1.6. In addition to that, this study has only focused on offline recognition and printed images. Furthermore, the testing dataset of the hybrid model of the OCR post-processing techniques has included only noise-free images that have standard scanning resolution. Lastly, from cursive written-based languages, the study has only tested the Arabic language.

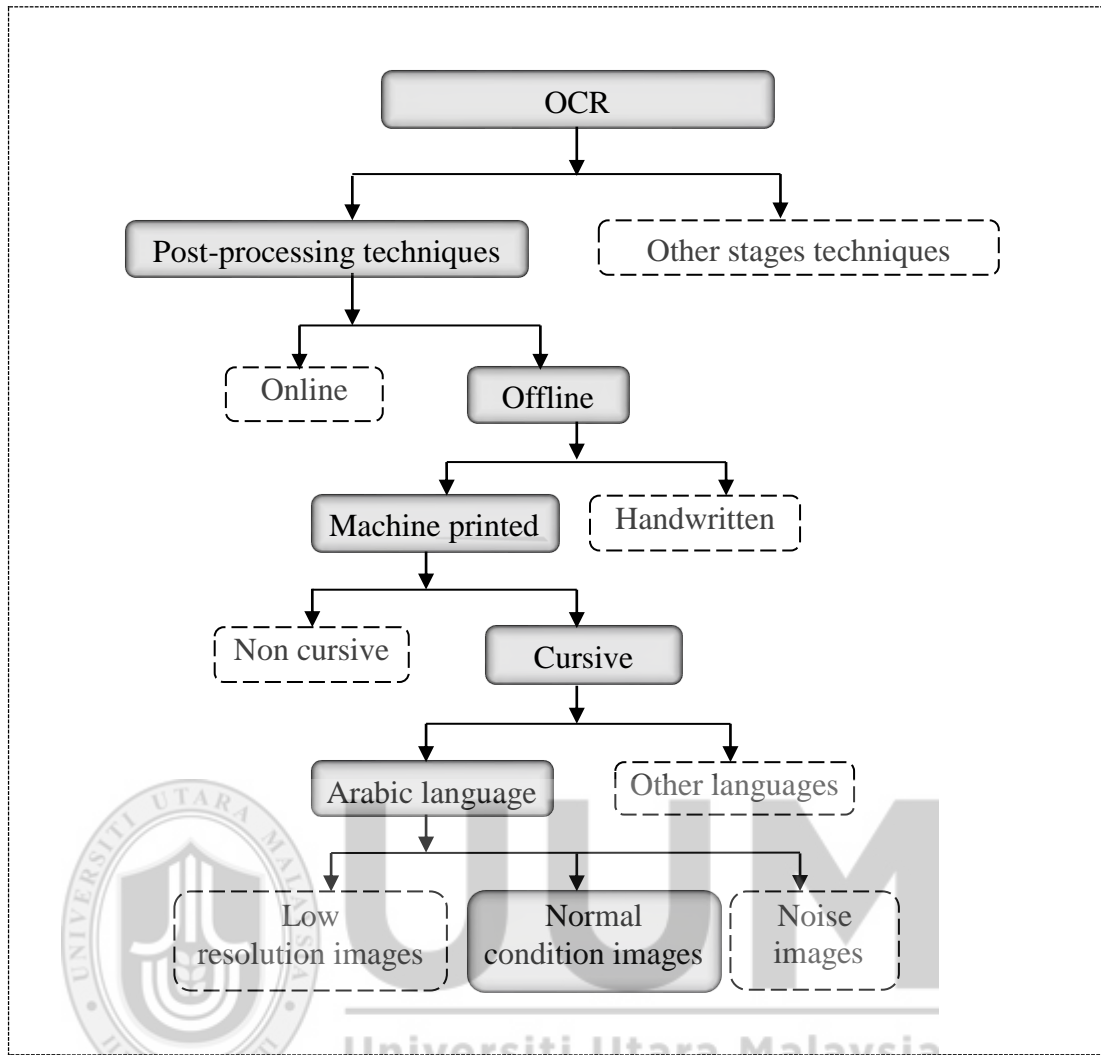


Figure 1.6. Scope of this research

## 1.6 Organization of the Research

This research contains eight chapters. Chapter 1 includes the necessary information for understanding the concepts that are used in the later chapters. Chapter 2 discusses the literature review with a description of the different aspects relating to the research area. Chapter 3 presents the methodology's steps that were used in the research. Chapters 4, 5, 6, and 7 explained the designing and evaluating the differentiation technique, alignment technique, voting technique and a hybrid model of OCR post-processing techniques respectively. Finally, Chapter 8 includes the research

summary, contributions, research limitations, and recommendations for future research. At the end of the study, the references used in the research.



## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.0 Introduction**

Chapter 2 describes the background and previous work of this study. Furthermore, it provides a critical review of existing work related to this research. This can identify the limitations in the current work and reveal where this study fits in to fill the identified gap. The focus of the discussion in this chapter revolves around the identified problems and objectives that were explained in Chapter 1. This chapter is organized into seven sections. An overview of the Arabic language and challenges of its characters, when applied to the OCR systems, are given in Section 2.1. The OCR post processing stage and its errors are explained in Section 2.2. The existing individual techniques of the OCR post-processing stage are discussed in Section 2.3. A comparison between the limitations of the existing individual techniques of the OCR post-processing stage is presented in Section 2.4. Existing hybrid techniques of the OCR post-processing stage are described in Section 2.5. Finally, a summary of the content of Chapter 2 is shown in Section 2.6.

#### **2.1 Arabic OCR**

This section contains three parts. The first one gives an overview of the Arabic language, while the second and third parts discuss the common challenges of Arabic when applied to the OCR systems. The common challenges of Arabic OCR can be classified into two types: limitations and characteristics. The limitations represent the reasons that make the process of the evolution of the OCR for the Arabic language slow compared to English. The characteristics represent the properties that



distinguish Arabic from the Latin letter-based languages (El-Mahallawy, 2008; Saber et al., 2016).

### **2.1.1 Overview of the Arabic Language**

The Arabic language has been classified into three categories: classical Arabic, standard Arabic, and dialectal Arabic. The first is the language used in the Quran. The standard Arabic is derived from the classical Arabic. It is often used in universities, schools, government, printed publications, TV, and the Internet. Dialectal Arabic is the spoken language used by the Arab people. It is different from one region to another and is usually unwritten. Sometimes the differences are enough to be considered by linguists as distinct languages (Badawi, 1996; Farghaly & Shaalan, 2009).

There are hundreds of languages for use in communication among people in the world. Latin characters are adopted in writing systems for greater than 89 languages; while Arabic characters are used by more than 24 different languages. Examples of the languages that use Arabic script in writing are Persian, Urdu, Kurdish, etc. Most printed text for Latin-based languages uses non-cursive scripts, while all Arabic-based languages use cursive scripts in printed text. Non-cursive scripts mean that each symbol has a separate shape without overlapping letters (Sattar, 2009).

### **2.1.2 Arabic OCR Limitations**

The companies that develop OCR systems may lack the incentives and motivations to support languages with non-Latin alphabets. They do not have knowledge about the characteristics of these languages (Daðason, 2012). The Arabic language has not

been given sufficient attention compared to the attention that has been given to the English language (El-Mahallawy, 2008; Khorsheed, 2002; Saber et al., 2016).

As stated by (Al-Badr and Mahmoud (1995); Bassil and Alwani (2012c); El-Mahallawy (2008); Saber et al. (2016)), the limitations of the Arabic language when used in OCR systems include:

- i. Lack of publications in Arabic as compared to English such as articles in journals, conferences, books, and theses. These publications will support the Arabic OCR system.
- ii. Lack of available resources and tools for Arabic characters compared to English, such as standard testing dataset, dictionaries, supporting staff, programming tools, and a web corpus. These resources will support practical techniques that are used in improving the Arabic OCR system.
- iii. Researchers in Arabic OCR started later than English.
- iv. The techniques developed for Latin languages are difficult to apply to Arabic without modification. The reason is that the special characteristics of Arabic characters are incompatible with Latin characters. These characteristics are explained in the next section.

### **2.1.3 Characteristics of the Arabic Language**

In this section, this research has been shown why OCR error rate of Arabic is greater than OCR error rate of Latin-based languages. As stated by (Akila et al., 2015; M Attia, Rashwan, and Khallaaf (2002); El-Mahallawy (2008); Khorsheed (2002)), the characteristics of the Arabic language when used in OCR systems are:

- i. **Cursive writing:** the Arabic language uses a writing system from right to left. Furthermore, Arabic text is characterized by the neighboring character connectivity. However, some characters may or may not be connected with its neighbors. It depends on the position of the character in the word. The segmentation stage within the OCR system for Latin is easier than Arabic. The characters of Latin can be extracted simply due to their separation from each other. Figure 2.1 displays examples of the connectivity in Arabic writing.

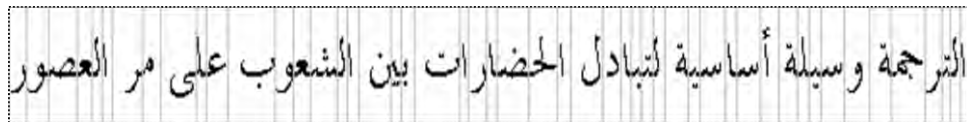


Figure 2.1. Connectivity in Arabic writing, Source: El-Mahallawy (2008).

- ii. **Overlapping:** Arabic contains vertical overlaps between some characters. Extraction of these characters as a rectangular box in the OCR segmentation stage is more difficult than the same operation in Latin. The reason is that each rectangular box of any single character may contain parts of another letter. Vertical overlaps confuse the recognition process. Figure 2.2 displays examples of the overlapping between some characters in Arabic writing.



Figure 2.2. Overlapping in Arabic writing, Source: El-Mahallawy (2008).

- iii. **Diacritics:** Arabic font contains diacritics, which are signs located above or below the letters. Table 2.1 below displays the shapes of some diacritics in Arabic.

Table 2.1

*Shapes of some diacritics in Arabic*

◌َ	◌ِ	◌ِ	◌ُ	◌ُ	◌ِ
◌َ	◌ِ	◌ِ	◌ُ	◌ُ	◌ِ
◌َ	◌ِ	◌ِ	◌ُ	◌ُ	◌ِ
اَ	اِ	وِ	وُ	آ	ؤ
لاَ	لاِ	لاِ	لُ	لُ	ؤ
سَ	سِ	سِ	سُ	سُ	لأ
سَ	سِ	سِ	سُ	سُ	م

The presence or absence of these signs is based on the writer of the text. They help a reader to understand the pronunciation of a word. Some of the diacritics represented short vowels in Arabic. This is because no characters represent them. If they are written or not written, the word is still the same word. However, if the word contains the wrong diacritics, it is considered as an error. Diacritics cause confusion in OCR systems, especially when the images contain noise. They make predicting any correct character more difficult than English. Furthermore, if the OCR engines completely ignore the diacritics, they may be misinterpreted as dots by the OCR systems. On the other hand, if diacritics are considered by the OCR systems, then the word

that includes diacritics and the word that does not contain them will become different, although they represent the same word. Figure 2.3 displays examples of the diacritics in Arabic writing.

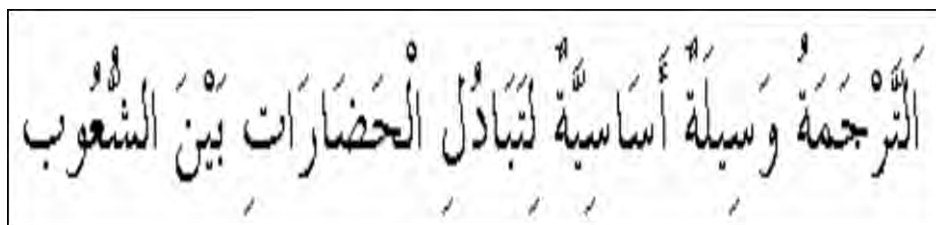


Figure 2.3. Diacritics in Arabic writing, Source: El-Mahallawy (2008).

The problem of diacritics is not only in the OCR engine itself, it is also in the techniques that are used in correcting the OCR errors. These techniques include edit distance technique, N-gram language model, Lexicon, etc. For example, all eight words in the following group “زرع”, “زُرِعَ”, “زَرَعَ”, “زُرِعَ”, “زُرِعَ”, “زُرِعَ”, “زُرِعَ”, “زُرِعَ” are treated as different by these techniques, although they are considered as a single word in Arabic.

- iv. **Morphology:** Arabic has a complex and rich morphology (M. E. Attia, 2000; Habash & Roth, 2011). Morphology is the study of how valid words can be generated from a root and pattern (Daniel Jurafsky & Martin, 2009). A root is a word that does not have a prefix or a suffix character. For example, the word “cat” is a root, while the word “cats” is not. A root in Arabic is a sequence of three, four, and five characters (M. E. Attia, 2000). The pattern is a template of consonants and vowels. Generating valid words in Arabic is much harder than in English. The average number of valid words resulted from the similarity among Arabic words is 26.5, for English is 3.0, and for French is 3.5 (Shaalán, Samih, Attia, Pecina, & van Genabith, 2012). Since

Arabic morphology is much richer, there is no corpus containing all the Arabic words with their forms (Shaalán et al., 2012; Watson, 2007).

## **2.2 OCR Post-processing Stage (PPS)**

As mentioned previously, in post-processing stage (PPS), the resulting text of the OCR is checked to make sure it is valid. Furthermore, it corrects any mistakes in the OCR output text. There is a great interest in the development of the post-processing stage because most OCR systems produce errors (Ramanan et al., 2014).

### **2.2.1 OCR PPS Error**

The OCR process usually produces two kinds of errors: non-word errors and real word errors. The non-word error occurs when the word produced from the OCR process does not exist in the language resource, such as the word "*foed*". The real word error occurs when the word produced from the OCR process exists in the language resource, but it does not match with the source text, such as the word "*too*" in the sentence "*I want too eat*" (Bassil & Alwani, 2012c; Kukich, 1992).

Correction of real word errors is harder than correcting non-word errors because real word errors are difficult to detect. The OCR correction techniques of non-word errors are unsuitable for real word errors. The reason is that, the detection and correction of real word errors are based on the context of the sentence, while detection and correction of non-word errors are not. Context-sensitive spelling correction is the name of any correction based on the information context of sentences in a corpus. As stated by (Dağason (2012); Kukich (1992); Naseem (2004)), real word errors range from 15 to 40% of total errors. These errors are a problem in the OCR applications where auto correction is required.

Errors of the OCR post-processing stage are different from human errors. OCR errors can result from a similarity between printed characters, the presence of noise, low-resolution images, etc (Barnes, 2011; Naseem, 2004). Human errors occur when a person knows the valid spelling of the word, but he makes mistakes in writing the word. For example, the writer replaces an intended letter by another one of which its key on the keyboard is a neighbor to the key of the intended letter. Another reason for human errors is when the person forgets the valid spelling of the word. For example, if the writer writes "*recieve*" instead of "*receive*" (Damerau, 1964; Naseem, 2004).

The scope and type of OCR errors make spelling checker programs rarely used as effective tools for them (Barnes, 2011; Naseem, 2004). These programs are usually designed for text processing with an error rate not exceeding 3% (Bassil & Alwani, 2012a; Kukich, 1992). The OCR error rate often exceeds this rate in case of noise, low-resolution images or cursive written typed languages (Akila et al., 2015; Saber et al., 2016). Spelling programs can identify a non-word, but their correction is based mainly on human input. Therefore, for a user who uses spelling checker programs in text processing, it is enough for him/her to determine the error and choice of the best word from the suggestion list. The reason is that the number of errors is low in text processing (Daniel Jurafsky & Martin, 2009; Taghva & Stofsky, 2001). On the other hand, the correction techniques of the OCR errors use an automatic way in correction and should reach the accuracy of 100%, and this rate has not been achieved yet (Barnes, 2011; Naseem, 2004; Saber et al., 2016).

### **2.2.2 Functions of OCR PPS Techniques**

Most techniques of the OCR post-processing stage have three functions: error detection, generation of candidates and error correction. The first function is to find all the wrong words in the output OCR text. The second function is to generate candidate words from the language resources for each wrong word. The third function is to correct all the wrong words by selecting the best suggestion for each of the words (Naseem & Hussain, 2007).

The error correction can be either manual or automatic. The first type gives the proofreader the ability to do the manual correction. In the second type, the correction process will decide the best correction word to use, and the wrong word is automatically replaced by the chosen candidate word. Automatic correction is used in most natural language processing applications, such as OCR and speech processing (Ramanan et al., 2014).

In the auto correction of OCR errors, the ranking process of the candidate words should be decided according to their importance in the sentences (Naseem & Hussain, 2007). When the auto correction is implemented, the incorrect word is automatically replaced by the first ranking word in an ordered candidates' list (Naseem, 2004). The choice of the appropriate technique for ranking is very important. The reason is that it may replace the incorrect word with another word that can be found in the language resource but it is unsuitable for the sentence, resulting in the desired goal of correction being unachieved (Dađason, 2012).



### **2.2.3 Categories of the OCR PPS Correction**

In general, the detection and correction of errors can be classified into three categories: proofreading-based correction, isolated word-based correction, context-based correction. Proofreading-based correction requires manually reading and correcting the text produced by the OCR process. This is inefficient as it is time-consuming, especially when the number of words is in the thousands (Lee & Chen, 1996).

In the second category, the process of correction is based only on the wrong word itself. The wrong word is processed in isolation without giving any attention to the context. This category cannot correct real word errors. In the last category, the process of correction is based on the wrong word and the context of the sentence. Contextual information is used for ranking the candidate words. The third category is used to correct real word errors and non-word errors (Islam & Inkpen, 2009).

## **2.3 OCR PPS Techniques**

This section discusses the existing individual techniques used in the OCR post-processing stage.

### **2.3.1 Multiple Outputs OCR (MO)**

As mentioned in Chapter 1, Lund and Ringger (2011) mentioned that the idea of the multiple outputs is to look for differences between MO of the OCR in order to choose the best from among them. According to Al Azawi (2015) and Volk et al. (2011), using multiple outputs of OCR is better than using single OCR output.

Multiple outputs consist of three processes: differentiation, alignment, and voting. The following sub-sections describe the existing techniques used in these processes.

### 2.3.1.1 Differentiation Process

As mentioned in Chapter 1, differentiation process is used to produce N-versions of input images. These versions are similar but not identical. They are not identical in order to produce different OCR outputs. Table 2.2 shows existing techniques used in the differentiation process.

Table 2.2

*Differentiation techniques in multiple outputs of OCR*

Author	Multiple Thresholds	Multiple OCR software	Multiple Classifiers	Multiple Scanning
Al Azawi (2015)		√		
Al-Zaydi and Salam (2015)				√
Lund (2014)	√			
Lund et al. (2013b)	√			
Batawi and Abulnaja (2012)		√		
Volk et al. (2011)		√		
Lund and Ringger (2011)		√		
Lund and Ringger (2009)		√		
Kittler et al. (1998)			√	
Lopresti and Zhou (1997)				√

From Table 2.2, Lopresti and Zhou (1997) and Al-Zaydi and Salam (2015) used the Multiple Scanning technique in order to produce various OCR outputs. According to the experimental results of Al-Zaydi and Salam (2015), scanning the image for multiple times is considered a time-consuming, and a better differentiation technique is required to increase the number of correct words in the OCR outputs.

Kittler et al. (1998) used Multiple Classifiers technique to generate four OCR outputs. However, according to Lund (2014), using multiple classifiers can reduce the performance of the best classifier. In other words, different classifiers produce different OCR accuracy. Hence, the accuracy of the final OCR output may be reduced because the output of the best classifier will be combined with outputs of other low-accuracy classifiers.

The differentiation technique used by Volk et al. (2011), Lund et al. (2013b), Lund (2014), and Al Azawi (2015) are based on combining different OCR software. However, according to Al-Zaydi and Salam (2015), combining different OCR software is considered a difficult process because it requires handling each OCR software output manually. On the other hand, Lund (2014) tested two differentiation techniques: combining different OCR software and using multiple thresholds. His experimental results showed that OCR accuracy of multiple thresholds is similar to the OCR accuracy obtained from combining different OCR software.

This research adopts the Multiple Thresholds technique as proposed by Lund (2014). The reasons are: first, it does not require scanning image multiple times as stated by Al-Zaydi and Salam (2015). Second, it does not require combining different classifiers that can reduce the performance of the best classifier as mentioned by

Lund (2014). Lastly, it does not require combining multiple OCR software that is considered as a manual process by Al-Zaydi and Salam (2015). The differentiation technique proposed by Lund (2014) uses 7 threshold values and produces 7 versions of the input image. Figure 2.4 shows the 7 versions of the input image produced using this technique.

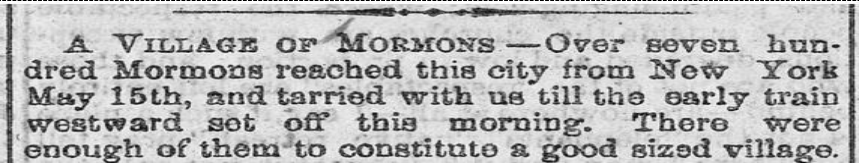

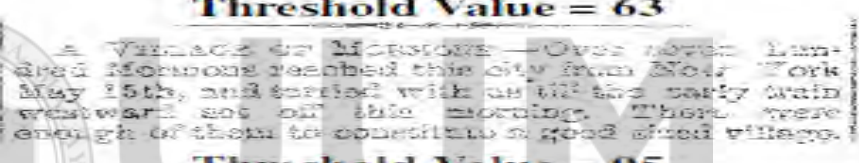
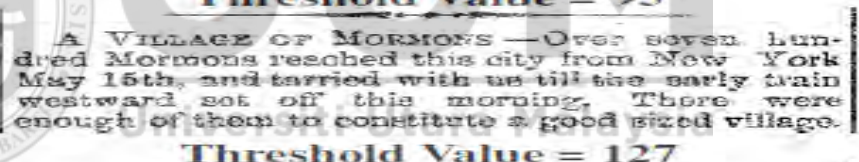
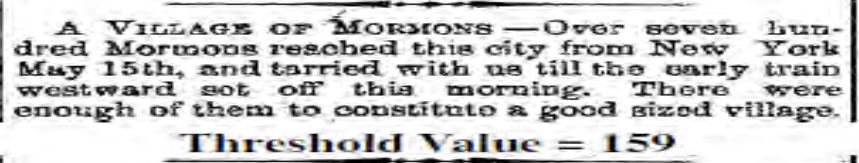
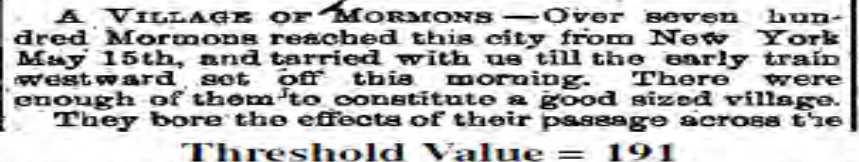

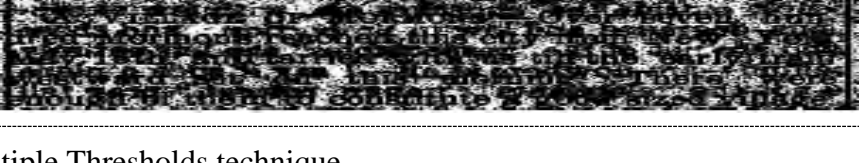
Input image	
Version 1 using threshold value of 31	
Version 2 using threshold value of 63	
Version 3 using threshold value of 95	
Version 4 using threshold value of 127	
Version 5 using threshold value of 159	
Version 6 using threshold value of 191	
Version 7 using threshold value of 223	

Figure 2.4. Multiple Thresholds technique

Source: Lund (2014).

From Figure 2.4, it can be seen that some important features of the characters' images are lost especially for threshold values of 31, 63, 191, and 223. This is because the pixels values above the identified threshold are transformed as white and the ones that are larger than the threshold become black. For example, when using threshold value 31, the image will become close to the white. For threshold value 63, the image will become more intense. Likewise, for threshold value 191, the image will become more intense. For threshold value 223, the image will become close to the black. Thus, these threshold values result in losing some important features of the characters' images. This causes increasing number of wrong words in the OCR outputs. Therefore, this research enhanced Multiple Thresholds technique so that the number of wrong words in the OCR outputs can be reduced.

#### **2.3.1.2 Alignment Process**

Figure 2.5 shows an alignment process used by researchers, such as Lund et al. (2013b), Lund (2014), Al Azawi and Breuel (2014), and Al-Zaydi and Salam (2015). This figure shows that N-versions of the input image, which result from a differentiation process, passed to the OCR engines, which later transform them into N-outputs of text. These N-outputs of text have a different number of characters in each OCR output due to the insertion, deletion, and substitution of characters. This causes vertical words overlapping between the OCR resulting texts. Therefore, the alignment process is required to align each character with corresponds in other OCR outputs, which later leads to parallel words in OCR outputs.

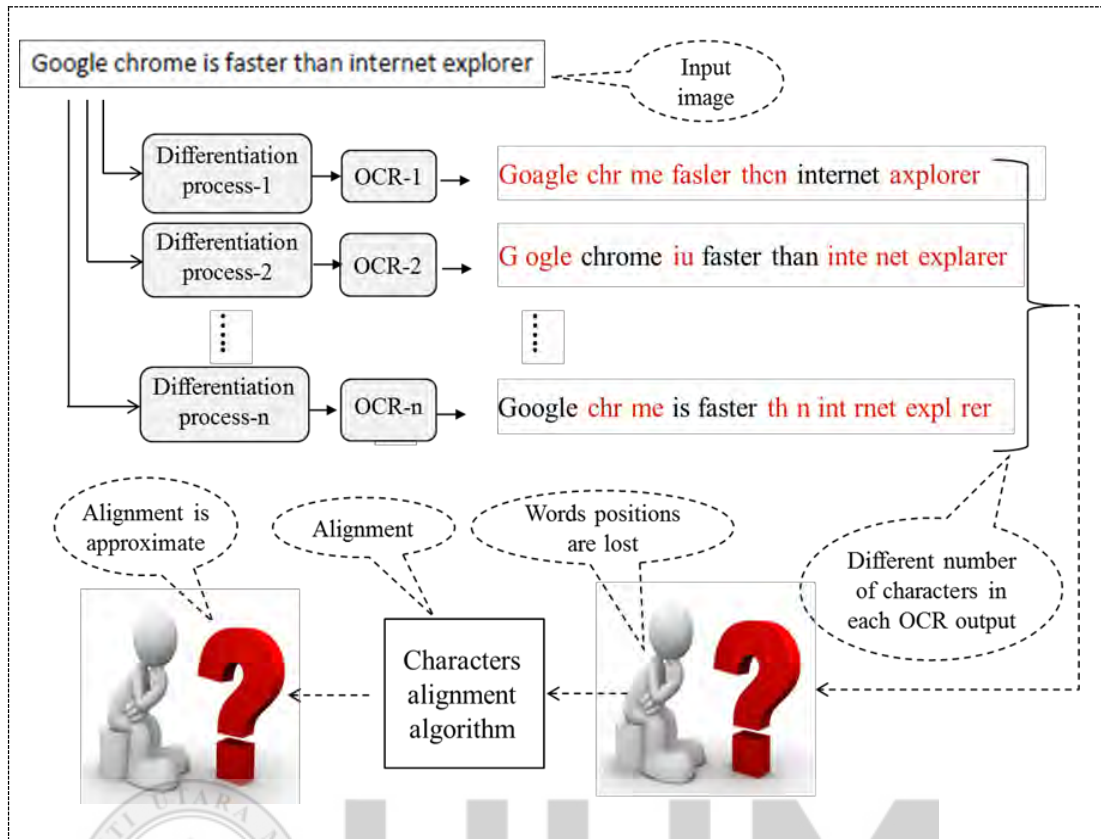


Figure 2.5. Alignment process.

Figure 2.5 also shows that the alignment process is performed by executing a character alignment algorithm. Examples of character alignment algorithms are Progressive algorithm (Lund, 2014), Smith–Waterman algorithm (Al-Zaydi & Salam, 2015), ProbCons algorithm (Pervez et al., 2014). However, alignment process for three OCR outputs or more after executing a character alignment algorithm becomes approximate as stated by Notredame (2002) and Lund (2014). Figure 2.6 shows a simple example that explains why the alignment process is approximate for three OCR outputs or more.

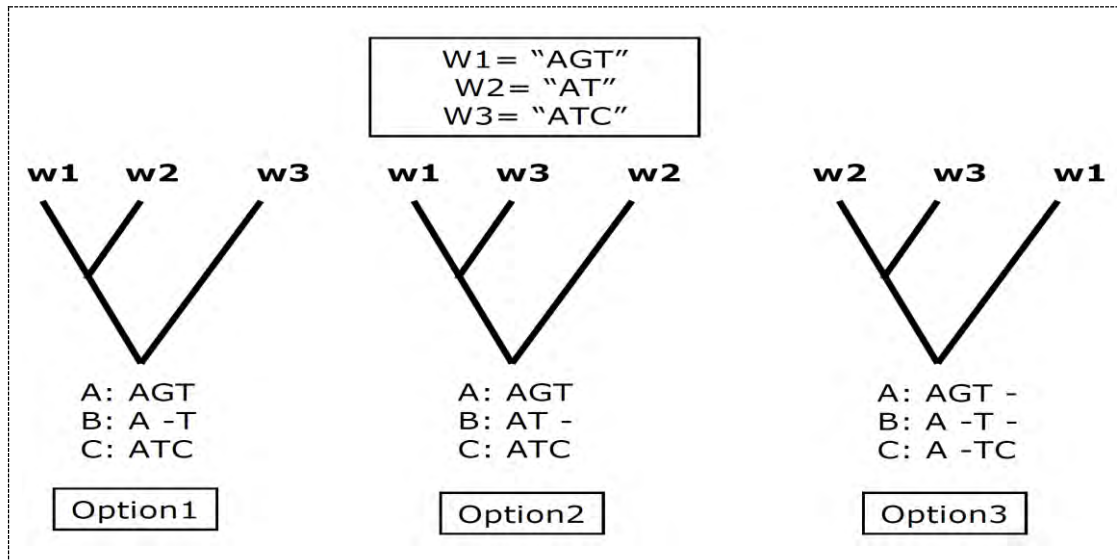


Figure 2.6. Simple example of an alignment process

Figure 2.6 shows alignment process for only three words. This research has been referred to them as  $w1$ ,  $w2$ , and  $w3$  respectively. The characters of these words are “AGT”, “AT”, “ATC” respectively. After that, the outputs of aligning  $w1$ ,  $w2$ , and  $w3$  are A, B, and C respectively. Figure 2.6 also shows that three options have been resulted from aligning  $w1$ ,  $w2$ , and  $w3$ . The first option occurs when  $w1$  is aligned with  $w2$  then with  $w3$ . The second option occurs when  $w1$  is aligned with  $w3$  then with  $w2$ . The last option occurs when  $w2$  is aligned with  $w3$  then with  $w1$ . From Figure 2.6, it can be clearly seen that the positions of the characters in these options are different. Therefore, it is difficult to identify the best alignment between them. The example in Figure 2.6 uses only three words with few characters in each one of them. The problem of alignment becomes hard when each OCR output contains a large number of characters (Lopresti & Zhou, 1997; Lund et al., 2014).

If the alignment process is approximate for three OCR outputs or more, then the resulting texts of the MO after the alignment process may contain errors. Furthermore, the existing alignment techniques, such as of Volk et al. (2011), Pervez

et al. (2014), and Lund et al. (2011), requires executing a character alignment algorithm between each pair of OCR outputs. However, Lund (2014) and Lopresti and Zhou (1997) mentioned that if execution number of character alignment algorithm is increased, then the probability of errors in the alignment process will be increased. Therefore, this research designed a novel alignment technique to make the alignment process is exact and to prevent executing any character alignment algorithm. The detail of the novel alignment technique was described in Chapter 3 and explained in detail in Chapter 5.

### 2.3.1.3 Voting Process

As mentioned in Chapter 1, the voting process is used to select the best word among the multiple outputs of the OCR. Table 2.3 shows existing techniques used in the voting process.

Table 2.3

*Voting techniques in multiple outputs of OCR*

Author	Majority	Majority & Lexicon
Al Azawi (2015)	√	
Al-Zaydi and Salam (2015)		√
Lund (2014)		√
Lund et al. (2013b)		√
Batawi and Abulnaja (2012)	√	
Volk et al. (2011)		√
Lund and Ringger (2011)		√



Author	Majority	Majority & Lexicon
Lund and Ringger (2009)		√
Kittler et al. (1998)	√	
Lopresti and Zhou (1997)	√	

Table 2.3 shows that some researchers use the Majority technique in voting process while others use a combination of Majority & Lexicon. However, both voting techniques are not reliable for auto correction because they do not give any attention to the context of a sentence around an incorrect word. Two examples are presented next to illustrate why context information of words before and after an incorrect word is important in the voting process.

In the first example, assuming that the sentence “*a good \_\_\_\_\_ can't sleep*” has three OCR outputs: “cop”, “cap”, and “cap”. The Majority technique will select the word “cap” rather than “cop” because the first-word “cap” outnumbers “cop”. This is incorrect because “cop” is suitable to complete the sentence, while “cap” is inappropriate. On the other hand, the Majority & Lexicon technique will also select “cap” even if it is unsuitable for the sentence. The reason is that both “cap”, and “cop” are found in the lexicon, and “cap” is more frequent than “cop”.

For the second example, if there are three candidate words “cop”, “cup”, and “cap” that resulted from a lexicon having the same distance to the wrong word “cep”, then, it is difficult to select the appropriate word from among them for a sentence. This is because “cop”, “cup”, and “cap” are found in a lexicon and no one of them has the

majority. Therefore, an enhanced voting technique based on context information of sentence is required to handle more cases of OCR outputs.

Since the Majority technique does not work well in several cases of OCR outputs as mentioned by Lund (2014), this technique was not chosen. On the other hand, since the Majority & Lexicon technique is commonly used by many researchers as shown in Table 2.3, this technique was chosen to be enhanced for the voting process. The design of the enhanced voting technique is based on context information of a sentence around an incorrect word. This has been described in Chapter 3 and explained in detail in Chapter 6.

### **2.3.2 N-gram Language Model**

The N-gram language model is a statistical model that provide the probability for a sequence of words (Daniel Jurafsky & Martin, 2009). A probability is dependent on the frequency of the words or frequency of the sentences in a large corpus. The language model can be used to detect and correct real word errors. Potential real word errors happen if the probability of a sentence provided by the language model is almost zero. A correction of this error is based on the high probability of other sentences. The most important point in this approach is that it does not require a confusion set or predefined rules. However, a large corpus is needed in order to build an accurate language model. Equation 2.7 is used to estimate the N-gram probability (Daðason, 2012; Daniel Jurafsky & Martin, 2009).

#### **2.3.2.1 N-gram Language Model Functions**

The N-gram language model has two functions: (1) predicting the next word from the previous (N-1) words and (2) predicting the probability of a whole sentence (Daniel

Jurafsky & Martin, 2009). Both functions of the N-gram language model suffer from a limitation when used in the OCR post-processing stage (Daniel Jurafsky & Martin, 2009; Raaid & Rafid, 2015). The following paragraphs explain the limitation of each function.

The first function of the N-gram language model has a limitation. It only uses previous ( $N - 1$ ) words to predict the next word in a sentence and it ignores the impact of the next ( $M + 1$ ) words in predicting the best candidate for the wrong word. To illustrate more, the sentence “*The student weat from home to school to study*” is taken as an example. Assuming that the word “*went*” was misspelled as “*weat*”, the list of sentences with a high frequency that results from the *Google3*-gram database is:



Sentence1= “*The student sweet*”

Sentence2= “*The student week*”

Sentence3= “*The student wants*”

Sentence4= “*The student went*”

With the Language model, the word “*weat*” will be replaced by the word “*sweet*” for two reasons. The first reason is the frequency of the sentence “*The student sweet*” is more than the three other sentences in the *Google3*-gram database, and secondly it uses the previous ( $N - 1$ ) words only to predict the next word. Therefore, it will ignore the impact of the next ( $M + 1$ ) words (“*from home to school to study*”) in the sentence that comes after the non-word “*weat*”.

The previous example shows the importance of taking into consideration previous ( $N-1$ ) words and the next ( $M+1$ ) words in predicting the best candidate for the wrong word in OCR systems. However, the first function of a language model can be used effectively for some word prediction applications, such as when typing in a search engine; but, it has limitations when used for OCR systems (Daniel Jurafsky & Martin, 2009).

The second function of the N-gram language model also has a limitation in the OCR system. It uses the probability of the whole sentence only in predicting the valid sentence. For illustration, the sentence “*I want Endish food*” is used as an example. Assuming that the word “*English*” was misspelled as “*Endish*”, the list of sentences with a high frequency that results from the *Google4*-gram database is:

Sentence1= “*I want **Chinese** food*”

Sentence2= “*I want **Indian** food*”

Sentence3= “*I want **fresh** food*”

Sentence4= “*I want **English** food*”

With the Language model, the word “*Endish*” will be replaced by the word “*Chinese*” because the frequency of the sentence “*I want Chinese food*” is more than the three other sentences in the *Google4*-gram database. This example shows the importance of using another way to work together with the probability of a language model to improve the accuracy of the output OCR text (Daniel Jurafsky & Martin, 2009).

In summary, the N-gram language models have two limitations when used alone in the OCR post-processing stage. The first function of the Language models ignores the impact of the next (M+1) words in predicting the best candidate for the wrong word. Therefore, if previous words are wrong, the language model will not give any probability. In the second function of the Language models, if any sentence  $S$  has a probability greater than the intended sentence, the language model will select  $S$  because its frequency is more than the other sentences. This research has been used the N-gram language model in generating candidates list for incorrect word as explained in Chapter 3.

#### **2.3.2.2 N-gram Language Models for Arabic**

There is a fact that a large N-gram gives more accuracy than a short N-gram. However, seeing a short N-gram in a large corpus is easier than seeing a large N-gram in the same corpus (Dan Jurafsky, Martin, Kehler, Vander Linden, & Ward, 2000). For example, a 5-gram gives more accuracy than a 2-gram; but the chance of seeing a 5-gram is less than seeing a 2-gram in the same corpus. Arabic has a lack of available web corpus as compared to English (AbdelRaouf, Higgins, Pridmore, & Khalil, 2010). As a result, it is hard to build large Arabic N-gram language models. Accuracy may also be affected due to the difficulty in seeing a large N-gram.

In addition to previous limitations on the Arabic language, there is another reason that makes creating language models more difficult than the English language. The similarity among Arabic words is high. As mentioned previously, the average number of valid forms that can result from the input word for Arabic is 26.5. For English, it is 3.0 and for French is 3.5 (Shaan et al., 2012). The number of valid forms is calculated using addition, substitution, deletion and transposition of the

characters among the language resources. The high similarity of Arabic valid forms needs a larger corpus to create a strong statistical language model that can give an accurate probability (Shaalán et al., 2012; Zribi & Ahmed, 2003).

Lastly, the N-gram language model suffers from diacritics when used in Arabic language (Muaz, 2011). In summary, the Arabic N-gram language model has three common challenges: a lack of available web corpora, similarity among the words and diacritics. The solutions for previous Arabic challenges have been explained in Chapter 7.

### 2.3.3 Levenshtein Distance

The term of “edit distance” is used for calculating the difference between two strings, where every insertion, deletion, transposition or substitution of a single character is considered as a single edit (Navarro, 2001). For example, "*stdy*" is a non-word in English, because it does not exist in this language. It requires one insertion to become "*study*" that is considered a correct word in English (Daðason, 2012). In the OCR post-processing stage, Levenshtein distance is used to measure the edit distance between a wrong word and all words in the language resource. Therefore, any word in the language resource, which has a single or multiple edit distances, will be considered as a candidate for correction. However, any candidate word having the least numbers of editing operations will be considered as the best candidate (Andoni & Krauthgamer, 2012). Figure 2.7 shows an example of how to calculate the edit distance between two words: "*Here*" and "*erefmere*" by using the Levenshtein distance (Daðason, 2012; Naseem, 2004).

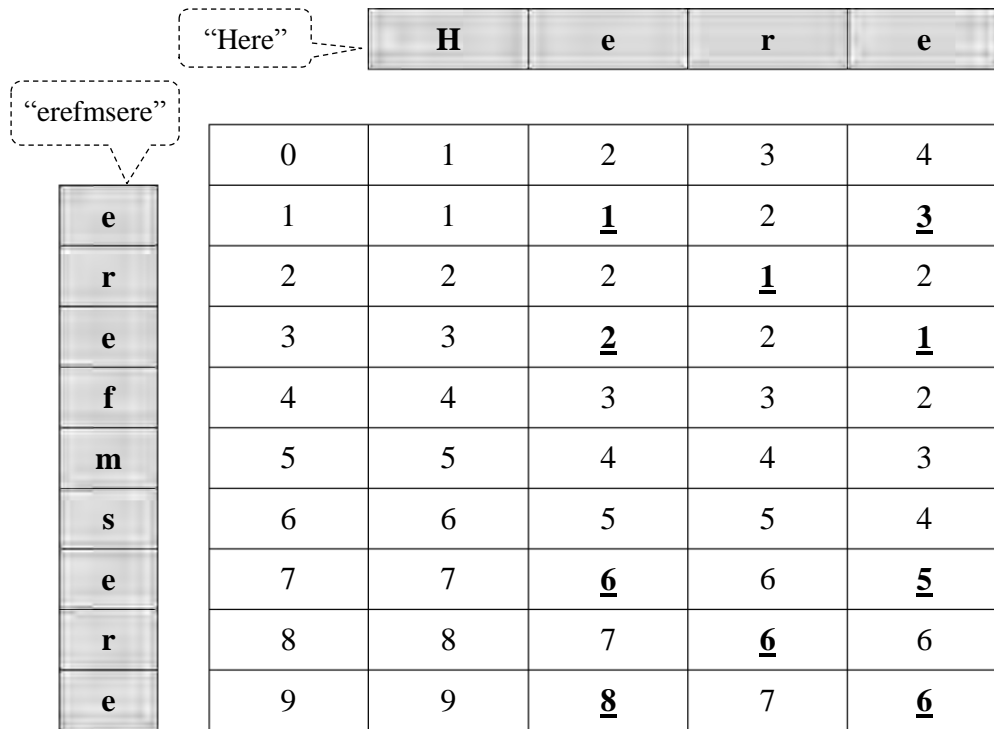


Figure 2.7. Levenshtein distance example

From Figure 2.7, it can be seen that there is a need for a matrix to measure the edit distance between the two tokens. The matrix can be filled from the upper-left to the lower-right corner. Each jump horizontally or vertically corresponds to an insert or a delete, respectively. The cost is normally set to 1 for each of the operations: insertion, deletion, and substitution. The diagonal jump can cost either one if the two characters in the row and column do not match or 0 if they do. After filling all cells in the matrix, the number in the lower-right corner is the Levenshtein distance between both tokens.

The Levenshtein distance has three limitations when used in the OCR post-processing stage (Naseem, 2004). The first limitation has been mentioned previously, it stated that many candidate words that are chosen by the Levenshtein distance have the same edit distance for the incorrect word. Therefore, it is difficult to select the best candidate from them. The second limitation is that the Levenshtein distance

belongs to the category of isolated word-based correction (Daniel Jurafsky & Martin, 2009; Naseem, 2004).

The third limitation of the Levenshtein distance, when used in the OCR post-processing stage, is that it needs to be calculated millions of times for each wrong word (Dařason, 2012). This calculation is not efficient for two reasons: it reduces the accuracy because of a huge number of candidates, which may be reaching thousands, and it is very time-consuming to process (Bard, 2007). In order to make the Levenshtein distance more efficient in the post-processing stage of OCR, a specific set of words needs to be chosen, thus improving the processing speed (Daniel Jurafsky & Martin, 2009).

#### **2.3.4 Rules-Based Technique**

The rules-based technique is used to detect and correct non-word errors and real word errors (Dařason, 2012). For example, if the word "*wear*" is followed directly by the word "*are*" then it should be replaced by the word "*where*". Another example, the word "*wear*" should be changed to the word "*where*" if the word "*going*" appeared within three words of "*wear*". The rules-based technique can be either a disambiguation technique or syntactic technique (Dařason, 2012).

The disambiguation technique is a common task in natural language processing applications in determining the meaning of a sentence. For example, the word "*orange*" can refer to the word "*fruit*" or to the word "*color*". Therefore, it can create a rule that the word "*orange*" should refer to the word "*fruit*" if it is followed by the word "*juice*". This rule is generated from the context of the sentence "*I drink orange juice every day*" (Islam & Inkpen, 2009).



The disambiguation technique needs confusion sets when being applied in any research field (Islam & Inkpen, 2009). In the OCR post-processing stage, the resulting OCR words that are confused with other words will be grouped in confusion sets. For example, the following two sets, [*where, wear*] and [*there, their*]. Therefore, if any word that belongs to these confusion sets appears in an OCR text, then the technique will evaluate these words to see whether or not it fits the context. Otherwise, if the word does not fit the sentence, it is flagged as an error, and the correction is performed using the words in the same confusion set (Daðason, 2012; Golding & Roth, 1999).

The confusion set technique has two limitations (Naseem, 2004). The first is that it can never be sure whether or not the sets of the error patterns are enough to represent all errors. The second problem is that even if all sets of all error patterns are defined, modeling of thousands of sets requires a lot of work to be accomplished (Daðason, 2012; Naseem, 2004).

In the syntactic technique, all the rules of a specific language are defined and recorded to be used later in detecting text errors (Naseem, 2004). The syntactic technique can perform word by word checking or whole sentence checking. In word by word checking, the OCR application builds a candidates' list of all the words that can be used as the next suggestion for any correct word in an image. The candidates' list is built using the syntactic rules. If a word that resulted from the OCR process is not one among the candidates of the expected words, then it can be considered as an error (Daðason, 2012).

In whole sentence checking, the OCR application builds a candidates' list based on the whole sentence to check whether it is true or not. The words in the sentence are subjected to the syntactic rules. If any of the rules can be applied to the whole sentence, then it is correct. Otherwise, if no rule can be applied, then it is considered a potential error. For example, in the structure of a sentence, the transitive verb needs a subject followed by an object, and both should be nouns (Islam & Inkpen, 2009; Naseem, 2004).

Collecting all the rules manually is an infeasible and time-consuming process. The use of syntactic rules to check the structure of the sentences may not give the best solution for many cases (Naseem, 2004). The reason is that the sentences can have many words as suggestions. Furthermore, the task of programming large-scale rules is clearly infeasible. In addition to that, it needs a matured knowledge of all the syntactic rules of the language to be included in the OCR systems (Dağason, 2012; El-Mahallawy, 2008; Govindan & Shivaprasad, 1990; Magdy & Darwish, 2008; Pratt, 1991). In summary, it is difficult to make rules for all errors. Modeling of thousands of rules requires a lot of work, and it is a heavy computation and slow (Kai, 2010).

### **2.3.5 Noisy Channel Model**

A noisy channel model is a probabilistic error model used to select the intended word from several words. It can be trained for one or different languages. The idea for this comes from the process of sending a message through a noisy channel, which causes errors in the message. Therefore, if the behavior of the noisy channel can be modeled, then it can know the actual intended word from several words using this

model (Daniel Jurafsky & Martin, 2009; Shannon & Weaver, 2002). Figure 2.8 shows a simple diagram of a noisy channel model.

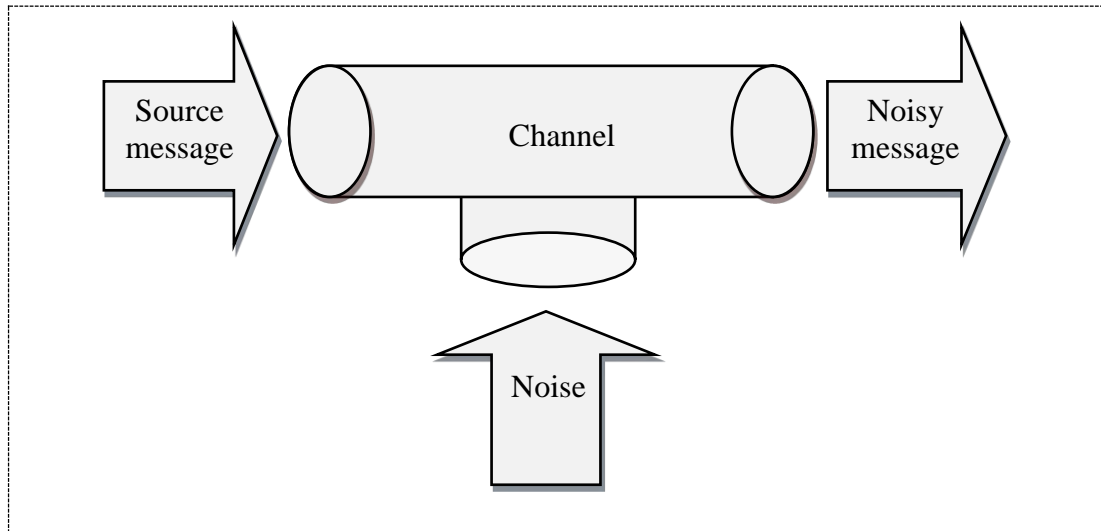


Figure 2.8. Noisy channel model

In the OCR post-processing stage, if a wrong word is given, then the goal is to find which valid word from the language resource corresponds to it. The shortcoming of the noisy channel model is that it requires confusion set. Furthermore, the best correction provided by this model might not really be the best solution. For example, it might suggest "acres" as the best correction of "acress" although the best solution is "actress" because it does not consider the context information of the sentence (Naseem, 2004).

### 2.3.6 N-gram Distance

It is used to measure the similarity between two sequences of strings (Naseem, 2004). The value of the first character in terms of the "N-gram" can be 1, 2, 3, 4... n. The term itself represents a sequence of  $N$  neighboring symbols in a token. An n-gram is called unigram when  $N=1$ , bigram when  $N=2$  and trigram when  $N=3$  and so on. The accuracy of the similarity increases when the value of  $N$  is high and vice

versa. Figure 2.9 shows an example of how to calculate the similarity between two tokens "*went*" and "*want*" using bigrams distance (Bassil & Alwani, 2012a; Naseem, 2004).

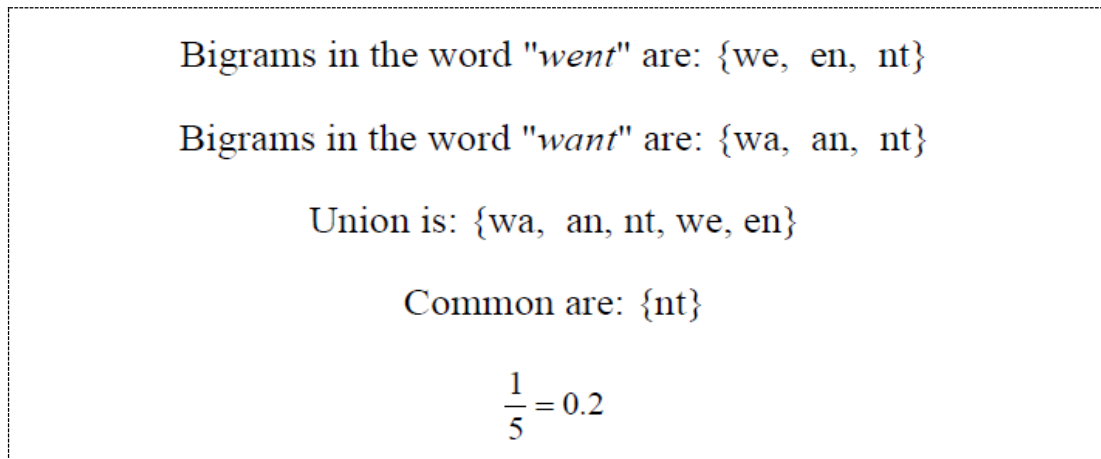


Figure 2.9. Bigram distance example, Source: Bassil and Alwani (2012c).

Figure 2.9 shows that the similarity is measured by dividing the number of shared N-grams of the two sequences by the total number of N-gram in the same two sequences. The n-gram distance does not show good accuracy on short strings. For example, the trigram distance will not give any similarity if two strings have the length three, and there are one or two different symbols between them. To solve this problem, different values of N are defined for different lengths of strings. As an example, for tokens of length two or less, unigrams can be used, other tokens of length three or more, bigrams can be used and so on (Naseem, 2004).

In summary, the n-gram distance belongs to the category of isolated word-based correction. It cannot rely on auto correction. The reason is that it does not give any regard for the context of the sentence around the incorrect word (Naseem, 2004). Furthermore, many candidate words can result from the n-gram distance having the same similarity, so it is difficult to select the intended word from among them.

### 2.3.7 Lexicon

A lexicon is considered as an index of words that belong to a specific language. It does not contain any information to describe its words. On the other hand, the dictionary is a lexicon with additional information (Barnes, 2011). A lexicon is one of the key elements in many applications, such as natural language processing, optical character recognition and translation software (Kenter, Erjavec, & Fišer, 2012). A lexicon is used for two functions: in the detection of non-word errors and in finding a suggestion list for these errors (Daðason, 2012).

For the English language, there are many lexicons that are independent and can be used in any of these applications. Likewise, Arabic has many lexicons but most of them were not designed to be integrated within these applications (AbdelRaouf et al., 2010). On the other hand, Arabic lexicons that can be integrated into these applications contain fewer numbers of words. This is because words are inserted into the lexicon manually by humans and not automatically (Bassil & Alwani, 2012c). Furthermore, because of their small size, they will not be effective in detecting the wrong words. It is possible to make the right word as wrong because it does not exist in the lexicon (Daðason, 2012).

The size of the lexicon depends on the language used. It may contain millions of words to be effective. Otherwise, if the size of a lexicon is small, then the process of the OCR will flag many right words as non-word errors. Furthermore, it cannot give candidate words to correct non-word errors (Barnes, 2011). The two most used techniques for finding a list of candidates for incorrect words from a lexicon are the Levenshtein distance (Daðason, 2012; Magdy & Darwish, 2008; Naseem, 2004) and the N-gram distance (Bassil & Alwani, 2012a). Some candidate words that result by

using the two previous techniques have the same similarity (Naseem, 2004). Therefore, it should be another way, in addition to lexicons, to determine the appropriate word for the sentence (Kanoun et al., 2011).

In summary, a lexicon-based correction cannot be relied on in auto correction for four reasons. The first, candidate words are arranged without any regard for the context of the sentence around the incorrect word. The second, most candidate words have the same similarity, so it is difficult to select the intended word. The third, it cannot detect real word errors (Bassil & Alwani, 2012c). The last, millions of words in the lexicon need to be tested for similarity.

#### **2.4 Comparison of OCR Post-processing Techniques**

This section presents a comparison (Table 2.4) on limitations of the OCR post-processing techniques. From Table 2.4, it can be seen that these techniques have several limitations. However, the multiple outputs, N-gram language model, and rules-based are the best techniques as they do not belong to a category of isolated word-based correction. The isolated word-based correction cannot correct real word errors. In other words, the multiple outputs of OCR, N-gram-based language model, and rules-based technique have the ability to correct both non-word and real word errors.

In detail, the N-gram-based language model performs the same work of the rules-based technique in selecting the intended word from the candidates' list. However, the N-gram language model does not require predefined rules. As mentioned by Kai (2010), the modeling large number of rules requires a large computational cost. Hence, indicating that the N-gram language model is better than the rule-based

technique. On the other hand, both the rule-based technique and N-gram language model require a large corpus. The rule-based technique uses a large corpus to create confusion sets while the N-gram language model uses a large corpus to build its database.

Table 2.4

*Limitations of the OCR post-processing techniques*

Existing techniques	Limitations	
	Isolated word-based correction	Context-based correction
Multiple outputs OCR		Differentiation problem, alignment problem, and voting problem (Lund, 2014).
N-gram language model		Requires large corpus, and it suffers from missing N-grams Al-Zaydi and Salam (2015)
Rules-based techniques		Requires large corpus, and modeling thousands of rules require a lot of work and it is a heavy computation and slow (Kai, 2010).
Levenshtein distance	Many candidates will have the same edit distance for incorrect word (Bassil & Alwani, 2012a).	
Noisy channel model	It is difficult to represent all OCR errors (Daniel Jurafsky & Martin, 2009).	
N-gram distance	Many candidates will have the same edit distance for incorrect word (Naseem, 2004).	
Lexicon-based correction	Many candidates will have the same edit distance for incorrect word (Daðason, 2012).	

Based on the previous discussion, the N-gram language model and multiple outputs of OCR are the best techniques to be used in the OCR post-processing. This opinion

has also been supported by researchers, such as Islam and Inkpen (2009), Kanoun et al. (2011), Abulnaja and Batawi (2012), Bassil and Alwani (2012b), Al Azawi (2015), and Al-Zaydi and Salam (2015).

## **2.5 Hybrid Techniques of the OCR PPS**

In computer science, the term “*Hybrid*” is a combination of different techniques, which are separated from each other naturally. The reason is to generate something new, which has the ability to take advantages of these different techniques (Alobaedy, 2015). These techniques are combined together in three ways: sequence, parallel, or mixed.

In the sequence way, the output of any technique is passed to the input of the next technique and so on. In a parallel way, all the techniques are performed at the same time, and their outputs are combined in a single output. The last one uses both sequence and parallel ways together in its work (Alobaedy, 2015; Boyell & Ruston, 1963). In OCR post-processing stage, several researchers used either individual or hybrid techniques to improve the accuracy of OCR systems. Table 2.5 shows the existing techniques that have been performed by several researchers followed by a discussion on them.



Table 2.5

*Some techniques used in the OCR post-processing stage*

<b>Author</b>	<b>Techniques</b>	<b>Target Errors</b>
Strohmaier, Ringlsetter, Schulz, and Mihov (2003)	Hybrid: Levenshtein distance & lexicon	Non-word errors
Magdy and Darwish (2008)	Hybrid: Character-based noisy model & character-based N-grams language model	Non-word errors
Lund and Ringger (2009)	Hybrid: Multiple outputs OCR & lexicon	Non-word errors & real word errors
Habash and Roth (2011)	Hybrid: Morphological feature model & character based N-gram models	Non-word errors
Volk et al. (2011)	Hybrid: Multiple outputs OCR & lexicon	Non-word errors & real word errors
Lund and Ringger (2011)	Hybrid: Multiple outputs OCR & lexicon	Non-word errors & real word errors
Barnes (2011)	Hybrid: Lexicon & unigram language model	Non-word errors
Bassil and Alwani (2012c)	Single: 5-grams language model	Non-word errors & real word errors
Alex et al. (2012)	Hybrid: Lexicon and rules-based	Non-word errors
Aljarrah et al. (2012)	Single: Lexicon	Non-word errors

Author	Techniques	Target Errors
Vu Hoang and Aw (2012)	Hybrid: Combining unigrams, bigrams, and trigrams language models	Non-word errors & real word errors
Lund et al. (2013b)	Hybrid: Multiple outputs OCR & lexicon	Non-word errors & real word errors
Springmann et al. (2014)	Single: Lexicon	Non-word errors
Ramanan et al. (2014)	Hybrid: Lexicon and rules-based	Non-word errors
Lund (2014)	Hybrid: Multiple outputs OCR & lexicon	Non-word errors & real word errors
Abdulkader and Casey (2015)	Single: Character-based noisy channel model	Non-word errors
Raaid and Rafid (2015)	Single: N-gram language model	Non-word errors & real word errors
Silfverberg and Rueter (2015)	Single: N-gram language model	Non-word errors & real word errors
Al-Zaydi and Salam (2015)	Hybrid: Multiple outputs OCR & lexicon	Non-word errors & real word errors
Al Azawi (2015)	Hybrid: Multiple outputs OCR & Language model	Non-word errors & real word errors
Al-Masoudi and Al-Obeidi (2015)	Hybrid: Combining unigrams, bigrams, and trigrams language models	Non-word errors & real word errors

There are five points can be obtained from Table 2.5. Firstly, the most widely used techniques in the OCR post-processing stage are lexicon, N-gram language model and multiple outputs OCR. Secondly, most researchers did not combine the best techniques of the OCR post-processing stage together, which are the multiple outputs OCR and N-gram language model. Thirdly, there is a trend to use multiple outputs OCR and N-gram language model in recent years. Fourthly, most solutions in the post-processing stage have used hybrid techniques. Lastly, some techniques target only non-word errors, even if they are hybrid techniques and vice versa. If some techniques target only non-word errors, then the accuracy of the OCR output will be reduced because a large number of errors, which are real word errors, cannot be handled by these techniques.

## **2.6 Summary**

The writing system of the Arabic language is different from that of Latin-based languages. This difference has caused high error rates in the OCR output text for this language, especially when the texts are worn out or their colors have changed. Since the OCR system consists of five stages, then improving any stage will contribute to the error rate reduction of the OCR system. The techniques of the OCR post-processing stage suffer from different limitations. Therefore, there is a chance to improve them or to benefit from the strengths of combining them in a hybrid model.

## **CHAPTER THREE**

### **RESEARCH METHODOLOGY**

#### **3.0 Introduction**

In the existing techniques of OCR post-processing, the experimental approach has been the main guide in developing any new technique (Al-Zaydi & Salam, 2015; Alex et al., 2012; Aljarrah et al., 2012). Therefore, this study has followed the same procedures to improve the existing OCR post-processing techniques for the Arabic language.

This chapter is organized as follows. Section 3.1 explains the research phases. This is followed by Section 3.2 that presents the theoretical study of this research. Section 3.3 discusses the design of the hybrid model, while the development of the hybrid model is presented in Section 3.4. The evaluation process is explained in Section 3.5. Finally, the summary of Chapter 3 is presented in Section 3.6.

#### **3.1 Research phases**

This research consisted of four phases (Figure 3.1): a theoretical study, design, development, and evaluation. The following sections describe the details of each phase.

Phases	Activities	Outcomes
Theoretical study	Review previous and current literature	<ul style="list-style-type: none"> <li>• Problem formulation</li> <li>• Comparison study</li> <li>• Summary of literature</li> </ul>
Design	<ul style="list-style-type: none"> <li>• Differentiation tech. design</li> <li>• Alignment tech. design</li> <li>• Voting tech. design</li> <li>• Hybrid model of the OCR post-processing techniques design</li> </ul>	The design flowcharts of the proposed techniques and hybrid model
Development	Develop the hybrid model and proposed techniques.	The developed hybrid model and proposed techniques
Evaluation	<ul style="list-style-type: none"> <li>• Differentiation tech. evaluation</li> <li>• Alignment tech. evaluation</li> <li>• Voting tech. evaluation</li> <li>• Hybrid model evaluation</li> </ul>	<ul style="list-style-type: none"> <li>• Differentiation tech. (1<sup>st</sup> obj.)</li> <li>• Alignment tech. (2<sup>nd</sup> obj.)</li> <li>• Voting tech. (3<sup>rd</sup> obj.)</li> <li>• Hybrid model (4<sup>th</sup> obj.)</li> </ul>

Figure 3.1. Research phases

### 3.2 Theoretical Study

The initial step in this study is the theoretical study. In this step, three directions were used in analyzing the research problem. The first direction was studying a state of the art in OCR for the Arabic language. The second direction was studying the characteristics of the Arabic language as compared to English. The last direction was studying a state of the art of techniques used in the OCR post-processing stage for cursive and non-cursive languages. This information was extracted from the literature obtained in different types of publications, such as conference proceedings, technical reports, books, theses, and journals, with a focus on recent publications.

The purpose of studying the OCR regarding the Arabic language was to identify the problem statement, research questions, research objectives, research significance and research scope. The purpose of studying the common characteristics presented by the Arabic language as compared to English regarding OCR was to identify their effects on OCR process. The goal from the identification was to understand which group was related to the techniques of the OCR post-processing stage and which group was related to other OCR stages. Lastly, the purpose of studying the existing techniques of the OCR error correction was to understand the limitations of each one of them. If these techniques and their limitations are understood, then there is a chance to develop them.

### **3.3 Design Phase**

In this phase, the designs of the proposed techniques were constructed. These are designs for differentiation technique, alignment technique, voting technique, and the hybrid model. The following sub-sections present the design steps of the proposed techniques and the hybrid model.

#### **3.3.1 Differentiation Technique**

This research enhanced Multiple Thresholds technique of Lund (2014). The steps undertaken to enhance the technique are:

Step1: Obtain the description of Multiple Thresholds technique (Lund, 2014).

Step2: Draw a flowchart based on the description of Step1. Figure 3.2 shows the flowchart of Multiple Thresholds technique.

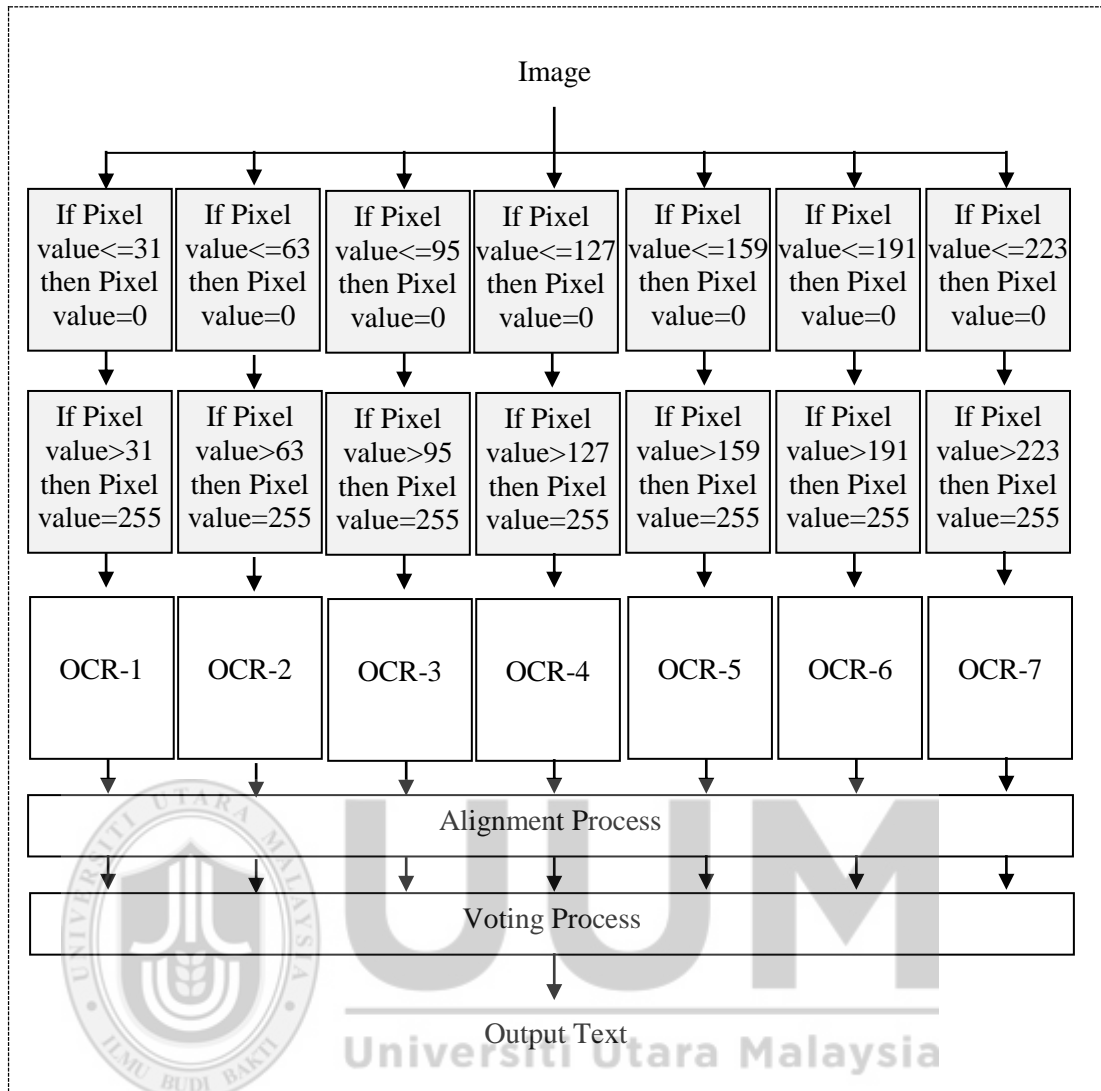


Figure 3.2. Flowchart of Multiple Thresholds technique

Step3: Develop an algorithm for Multiple Thresholds technique based on the flowchart of Figure 3.2.

Step4: Run the algorithm of Multiple Thresholds technique using Arabic images for each threshold value (31, 63, 95, 127, 159, 191, and 223) separately.

Step5: For each threshold value mentioned in Step 4, the OCR error rate was measured.

Step6: Based on the results of the OCR error rate, this research selected the best

three threshold values, which are (127, 159, and 191), to be used in the proposed technique.

Step7: Round the threshold values of 127, 159, and 191 to 130, 160, and 190 respectively, to make the difference between them are same.

Step8: It was found that Multiple Thresholds technique has a problem, which is a loss of important features from characters images because this technique changed some gray pixels values to white (255) as shown in Figure 3.2.

Step9: Based on the problem in Step 8, an improvement was made to change some gray pixels values to black (0) if they met certain conditions as shown in Figure 3.3 that represents the flowchart of the proposed differentiation technique. The proposed technique has been referred to as EDT by this research, which means enhanced differentiation technique.

**Enhancement made:**

From Figure 3.2, it can be seen that each pixel value greater than threshold value “x” will be changed to 255. Hence, some important features of characters’ images are lost. The effect of losing some features from the characters’ images is that the number of wrong words in the OCR outputs are increased (Al-Zaydi & Salam, 2015). In contrast, Figure 3.3 of the proposed technique shows that some gray pixels values will be changed to the black if they located in the path of black pixels. This is because two important reasons. The first, this research focused on gray pixels in the path of black pixels because white pixels do not represent any information about images of characters while black pixels give strong evidence that they may represent information about them. The second, this research finds the path of black pixels by



focusing only on gray pixels that must be located beside black pixels while other gray pixels that do not satisfy this condition will be ignored. The algorithm, examples, and contributions of the proposed differentiation technique are explained in Chapter 4.

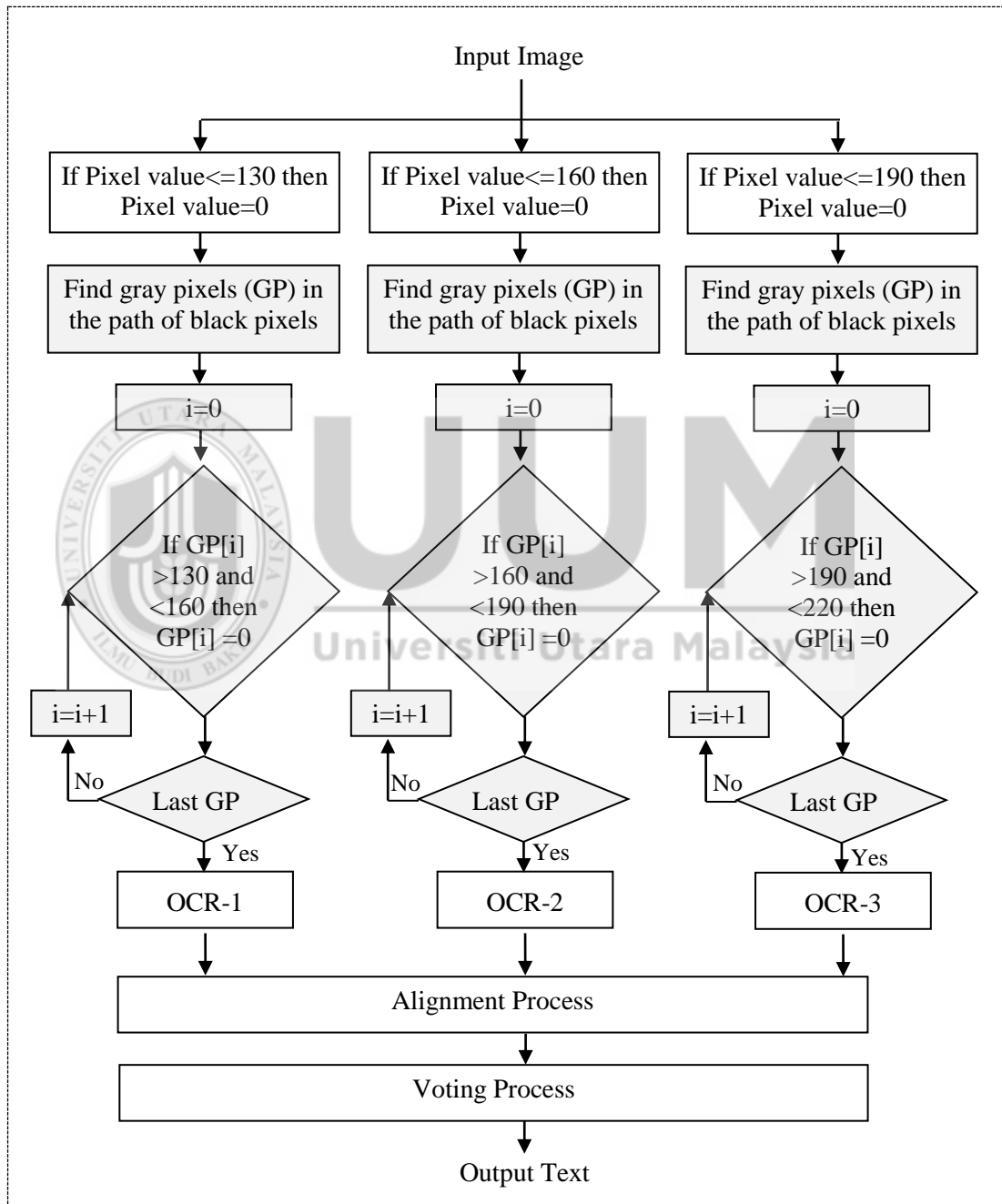


Figure 3.3. Flowchart of the proposed differentiation technique (EDT)

### 3.3.2 Alignment Technique

The steps undertaken to design the alignment technique are:

Step1: Obtain the description of the existing alignment technique from Lund (2014).

Step2: Draw a flowchart based on the description of Step1. Figure 3.4 shows the flowchart of the existing alignment technique.

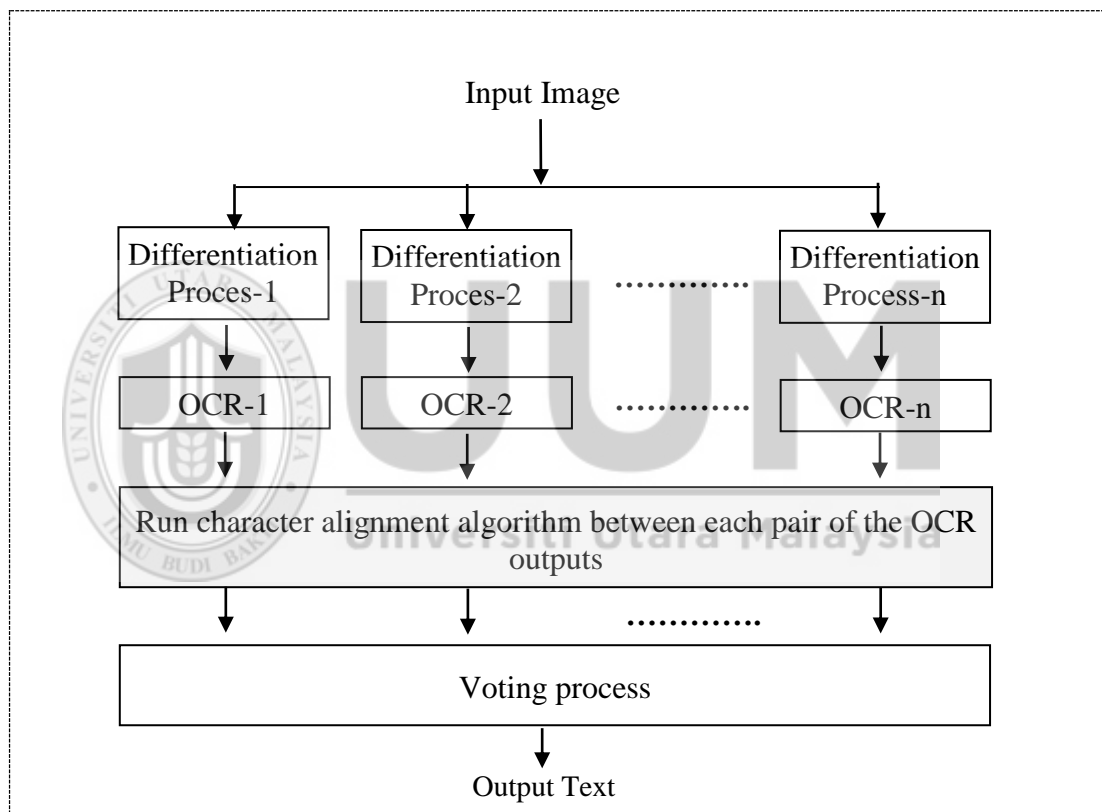


Figure 3.4. Flowchart of the existing alignment technique.

Step3: Develop an algorithm based on the flowchart of Figure 3.4, and run it using Arabic images.

Step4: The N-outputs of text resulted from OCR engines were analyzed, and it was found that these N-outputs of text have a different number of characters due

to the inserting, deleting, and substituting of characters.

Step5: Run character alignment algorithm of Lund (2014) between each pair of OCR outputs to align each character in the OCR output with corresponds in other. Figure 3.5 shows a simple example of character alignment algorithm.

Input image	To be or not to be movie																																																																								
Output1 of OCR	Tc be not t bc mov	Different number of characters in each OCR output																																																																							
Output2 of OCR	To bc or not to be movie																																																																								
Output3 of OCR	Ta ba or nat be movie																																																																								
Run character alignment algorithm	<table><tr><td>T</td><td>c</td><td></td><td>b</td><td>e</td><td></td><td></td><td></td><td>n</td><td>o</td><td>t</td><td></td><td>t</td><td></td><td>b</td><td>c</td><td></td><td>m</td><td>o</td><td>v</td><td></td></tr><tr><td>T</td><td>o</td><td></td><td>b</td><td>c</td><td></td><td></td><td>o</td><td>r</td><td></td><td>n</td><td>o</td><td>t</td><td></td><td>t</td><td>o</td><td></td><td>b</td><td>e</td><td></td><td>m</td><td>o</td><td>v</td><td>i</td><td>e</td></tr><tr><td>T</td><td>a</td><td></td><td>b</td><td>a</td><td></td><td></td><td>o</td><td>r</td><td></td><td>n</td><td>a</td><td>t</td><td></td><td></td><td></td><td></td><td>b</td><td>e</td><td></td><td>m</td><td>o</td><td>v</td><td>i</td><td>e</td></tr></table>		T	c		b	e				n	o	t		t		b	c		m	o	v		T	o		b	c			o	r		n	o	t		t	o		b	e		m	o	v	i	e	T	a		b	a			o	r		n	a	t					b	e		m	o	v	i	e
T	c		b	e				n	o	t		t		b	c		m	o	v																																																						
T	o		b	c			o	r		n	o	t		t	o		b	e		m	o	v	i	e																																																	
T	a		b	a			o	r		n	a	t					b	e		m	o	v	i	e																																																	

Figure 3.5. Simple example of character alignment algorithm

Step6: The results of Step 5 were analyzed. It was found that N-outputs of OCR text have some errors in the alignment. This showed that running character alignment algorithm is unsuitable to be used and new alignment technique is required.

Step7: Since the main goal of alignment process is to align each word in the OCR output with corresponds in other OCR outputs (Lund, 2014). Therefore, an alignment technique based on words rather than characters was proposed. Figure 3.6 shows the flowchart of the proposed alignment technique. The proposed technique has been referred to as AWS by this research, which

means alignment by using words separation.

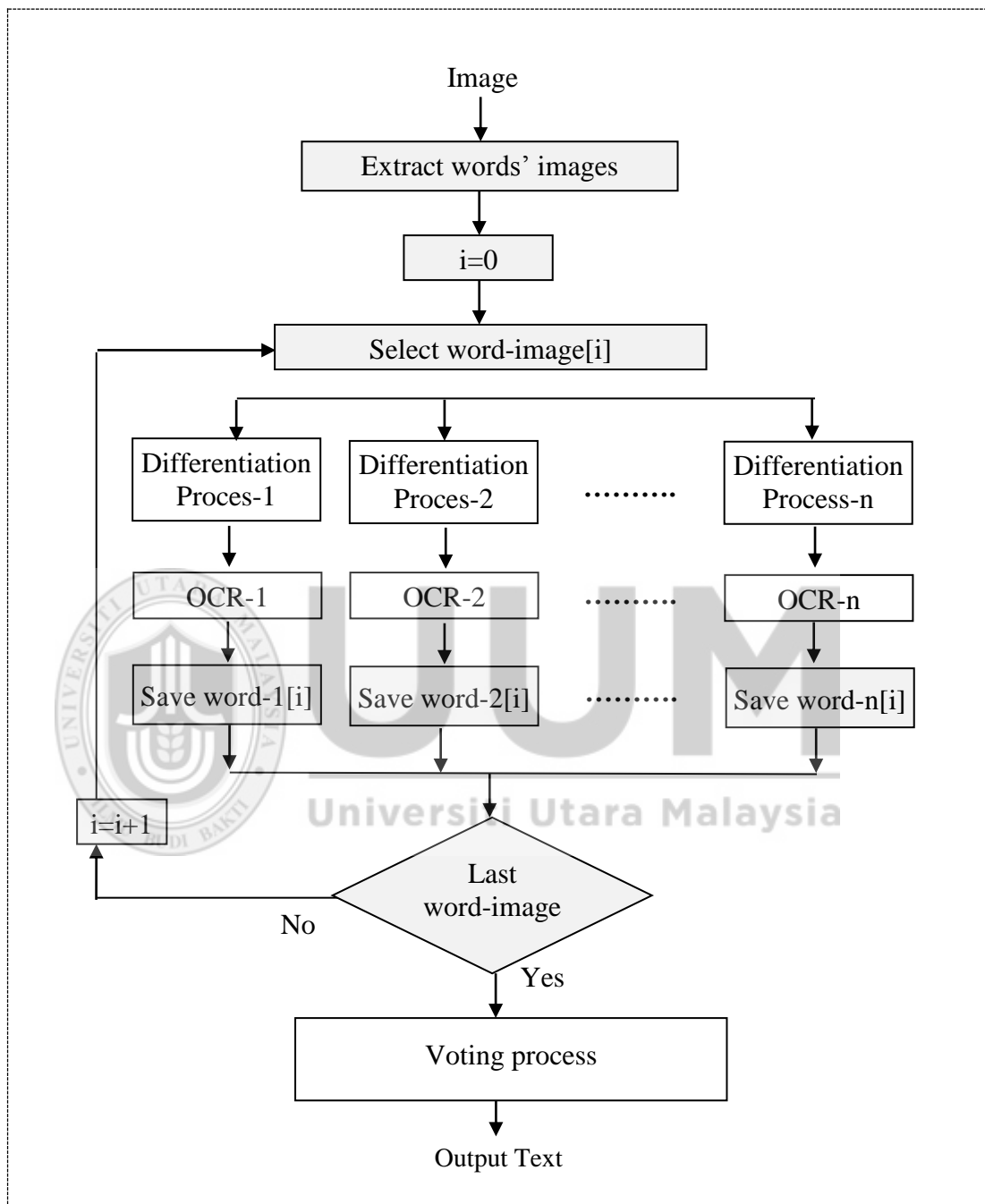


Figure 3.6. Flowchart of the proposed alignment technique (AWS)

#### Enhancement made:

There is a main difference between Figure 3.4 of the existing technique of Lund (2014) and Figure 3.6 of the proposed technique. The difference is that in Figure 3.4

a character alignment algorithm is required to execute between each pair of OCR outputs. As mentioned in Section 2.3.1.2, if the character alignment algorithm is executed between each pair of OCR outputs, then alignment process will become approximate (Al-Zaydi & Salam, 2015; Al Azawi & Breuel, 2014; Lund, 2014; Lund, Kennard, & Ringger, 2013a; Lund et al., 2011).

From Figure 3.6, it can be seen that the goal of the alignment process, which is preparing OCR outputs to the voting process by aligning each word in the OCR output with corresponding in other OCR outputs has been achieved. This is because words' locations are saved before sending any image to OCR engines. Therefore, deleting, misrecognizing, and inserting of characters will not change the locations of words in each OCR output. Furthermore, resulting texts of OCR multiple outputs are already aligned according to the words of the input image. The algorithm, examples, and contributions of the proposed alignment technique are explained in Chapter 5.

### **3.3.3 Voting Technique**

This research enhanced existing voting technique of Al-Zaydi and Salam (2015). The steps undertaken to enhance this technique are:

Step1: Obtain the flowchart (Figure 3.7) of the voting technique from Al-Zaydi and Salam (2015).

Step2: Develop an algorithm based on the flowchart of Figure 3.7, and run it using three different OCR outputs.

Step3: The results of Step 2 were analyzed, and it was found that final OCR output text has some errors due to the voting process. This is because existing voting

technique does not give any attention to the context of a word in a sentence.

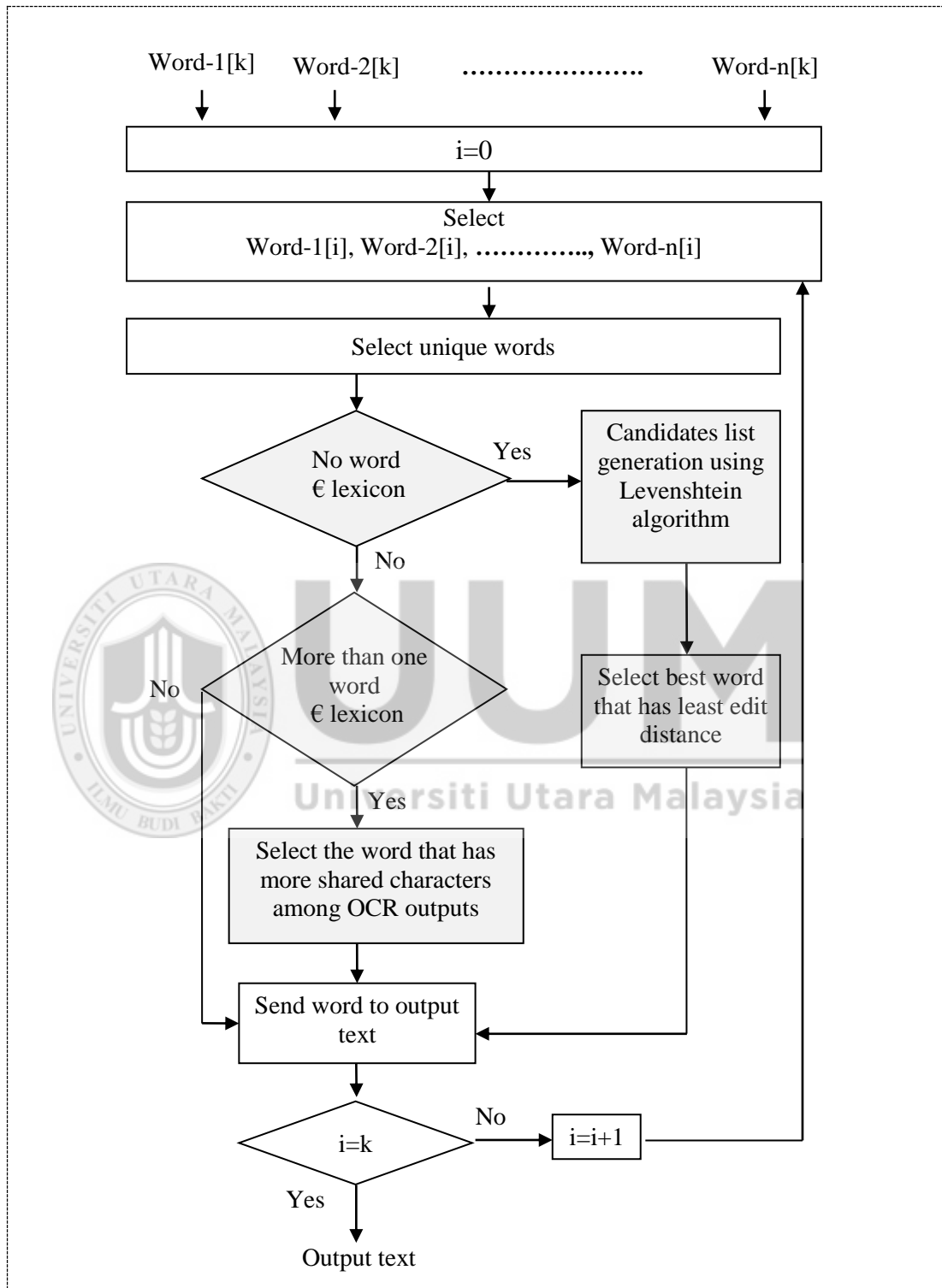


Figure 3.7. Flowchart of existing voting technique

Step4: Based on results of Step 3, the voting technique (Figure 3.8) was enhanced based on context information of a sentence around an incorrect word. The proposed technique has been referred to as VCI by this research, which means voting by using context information of sentences.

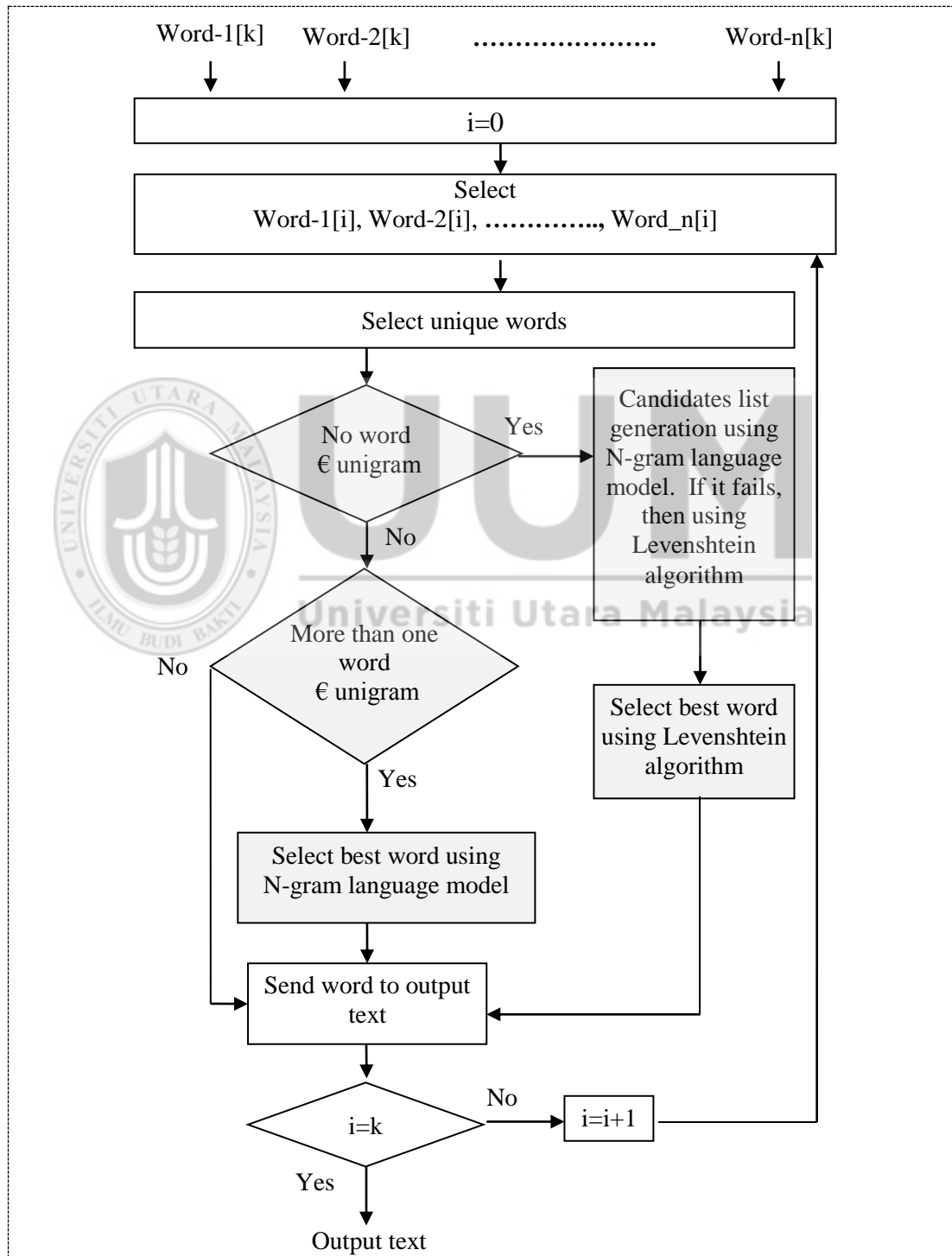


Figure 3.8. Flowchart of the proposed voting technique (VCI)

**Enhancement made:**

There are two major differences between Figure 3.8 of the proposed technique and Figure 3.7 of the existing technique. The first is that the proposed technique checks if any word is correct or not by using unigram while existing technique uses a lexicon. Unigram is better than a lexicon because unigram contains all words of the lexicon, and it contains also additional words representing most frequent words of specific language such as names, new words, etc (Bassil & Alwani, 2012c).

The second is that the proposed technique selects the best word from OCR outputs based on context information provided by N-gram language model while existing technique does not. As mentioned by Naseem (2004), context-based correction is better than isolated word correction. The algorithm, examples, and contributions of the proposed voting technique are explained in Chapter 6.

**3.3.4 Hybrid Model**

In this section, the steps used to produce the hybrid model of the post-processing techniques are briefly described below:

Step1: Extract words'-images from the input image and store them in an array.

Step2: Pass each word-image sequentially to the differentiation process to produce 3-versions of the word-image.

Step3: Pass each word-image of each version to the single OCR engine to turn it into a word.

Step4: The sequence of words resulting from each OCR engine is combined in a



single array so that three arrays will be produced from three OCR outputs.

Step4: The voting process receives three arrays from Step 4, and it will perform a process to select the best among them.

The proposed model has been referred to as HMNL by this research, which means a hybrid of OCR multiple outputs, N-gram language model, and Levenshtein algorithm. The details of the proposed model are explained in Chapter 7.

### **3.4 Development Phase**

This phase developed a software prototype for measuring the error rate of the OCR for the hybrid model and for the existing techniques. This development included choosing a programming language, selecting the OCR engine, a database of n-gram language model and an operating environment. The prototype was developed using the followings:

- Visual studio. Net 2012 technology, using VB. Net language.
- To avoid unnecessary waste of time, cost, and effort of building OCR system from scratch, the Tesseract OCR engine version 3.02 for converting images into text was used. The engine is supported by Google (Patel, Patel, & Patel, 2012), and it is used by many researchers in developing OCR post-processing techniques (Al-Masoudi & Al-Obeidi, 2015; Lund, 2014; Patel et al., 2012). It contains many parameters to control the stages of the pre-processing, segmentation, features extraction, classification, and post-processing.
- Microsoft SQL Server 2008 that was used to build the database of the N-gram language model.

- Microsoft window seven for an operating environment.

### 3.5 Evaluation

Figure 3.9 shows the whole evaluation process that has been done to evaluate differentiation technique, alignment technique, voting technique, and a hybrid model of post-processing techniques. It shows that the evaluation process has been performed separately for each technique. Furthermore, each evaluation process represents one of the objectives of this research. Figure 3.9 also shows that the evaluation process includes collecting the testing dataset and training dataset. On the other hand, a software prototype was developed for testing the proposing techniques and hybrid model. The following sub-sections explain in detail each part of the evaluation process.

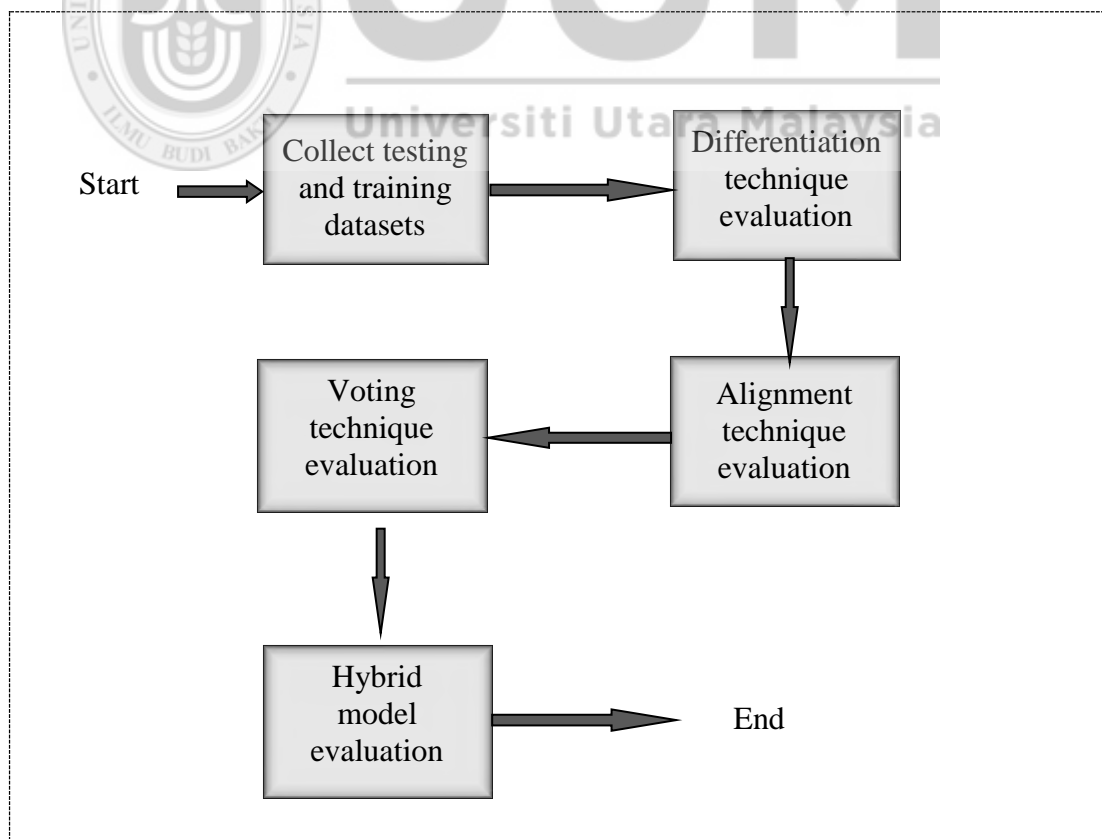


Figure 3.9. Whole evaluation process

### 3.5.1 Date Collection

Two datasets, i.e. testing dataset and training dataset were used for this study. The following sub-sections described them.

#### 3.5.1.1 Testing Dataset

This study followed the same scenario used by (Al-Masoudi and Al-Obeidi (2015); Al-Zaydi and Salam (2015); Batawi and Abulnaja (2012); El-Mahallawy (2008)) to create the testing dataset. The characteristics of the testing dataset were:

- It contained 231896 graphemes in the form of Arabic documents. The grapheme means the smallest meaningful unit of a writing system.
- It was chosen randomly from Arabic websites on the Internet.
- It contained, in addition to the characters of a text, the special symbols, such as commas, brackets, etc.
- It included eight different Arabic fonts. The names of these fonts are Simplified Arabic, Tahoma, Microsoft sans Serif, Courier new, Times New Roman, Arial, Adobe Arabic, and Traditional Arabic.
- For each font, six different sizes ranging from 10 to 20 were included.

The texts in these documents acted as a reference text during the evaluation process. To generate the test images from the reference, the text was first printed on papers. Then, the hardcopy was scanned at 300 dpi with a gray level in a modern scanner to produce the test dataset images. Figure 3.10 shows a sample image selected from the testing dataset.

استطاعت طهران ترسيخها في الواقع الميداني السوري والشرق أوسطي عموماً. ووفق تلك القناعة فإن طهران خلقت واقعا عسكريا من الصعب تجاوزه، وخاصة بعد أن سيطرت على القلب الإستراتيجي لسوريا الممتد من العاصمة دمشق وحتى الساحل، وتشكل هذه المنطقة بالمعنى الإستراتيجي قلب الشرق الأوسط لتموضعه على كتلة جغرافية ذات طبيعة جبلية تشرف على القسم الأكبر من الشرق الأوسط، بحيث يصبح لبنان والجزء الحيوي والمأهول من إسرائيل تحت مراقبتها، وحتى بعض أجزاء الجنوب التركي. هذا إضافة إلى إمكانية عمل كريدور بري يصل العراق عبر ريف حمص الشرقي، الأمر الذي يتيح وجود خريطة متماسكة وصلبة وتتوفر على إمداد لوجستي من طهران حتى "صور" جنوب لبنان. وفي الواقع الجغرافي السوري، تشكل تلك المناطق المشار إليها قلب سوريا، لتحكمها بشبكة خطوط المواصلات بكامل البلاد، ومن ناحية أخرى تشكل الجزء الأكبر المأهول من سوريا، كما تتمتع بطبيعة جغرافية تسهل التحصن بها من قبل قوات عسكرية منظمة، فضلا عن إشرافها على مناطق عسكرية داخلية واسعة، وتحويلها بالمعنى العسكري الكلاسيكي إلى مناطق ساقطة حريبا، بالإضافة إلى امتلاك هذه المناطق لأهم عنصر اقتصادي حياتي وهو المياه. وبالتوازي مع ذلك، أوجدت إيران خريطة مهشمة هي عبارة عن قطع متناثرة في شمال سوريا وفي جنوبها وشرقها، وهي مناطق إما صارت جزرا معزولة لا تشكل مخاطر حقيقية، وقد يجري ضمها بعد أن يصار إلى إنهاكها بسياسة الحصار والتجويع، وإما هي مخترقة من قبل التنظيمات التي تدعمها طهران بالسرا أو تخترقها بطريقة ما. ولعل الانتصار الأهم بالنسبة لإيران في سوريا يتمثل بالاختراق الكامل للنظام السياسي السوري نهائيا، وتحويله إلى مجرد كيان إيراني داخلي، مثل أي كيان

Figure 3.10. Sample image selected from the testing dataset.

### 3.5.1.2 Training Dataset

In order to use an N-gram language model for the Arabic language, a large web corpus is needed to train it. This study has used a Wikipedia database, which is freely available and can be downloaded as one file in the XML format (Habeeb, Yusof, & Ahmad, 2014). The Wikipedia database has been chosen for several reasons (Mohammed Attia, Toral, Tounsi, Monachini, & van Genabith, 2010; Habeeb et al., 2014; Knopp, 2010):

- It is the largest free source of Arabic web corpus.
- It is used for many topics, such as document retrieval, data mining, etc.
- It contains more than 2322000 Arabic articles.
- It is a multiple domains source that contains 25 different categories.
- There is a rapid growth in Arabic articles.

- The database is constantly updated.

### 3.5.2 Experimental design

In the experimental design, the goals of the experiments and how these goals can be achieved are explained (Rardin & Uzsoy, 2001). Four groups of experiments were conducted to achieve four goals. Each goal was related to one of the research objectives as shown in the following subsections.

#### 3.5.2.1 Differentiation Technique Evaluation

Figure 3.11 shows the evaluation process of the proposed differentiation technique.

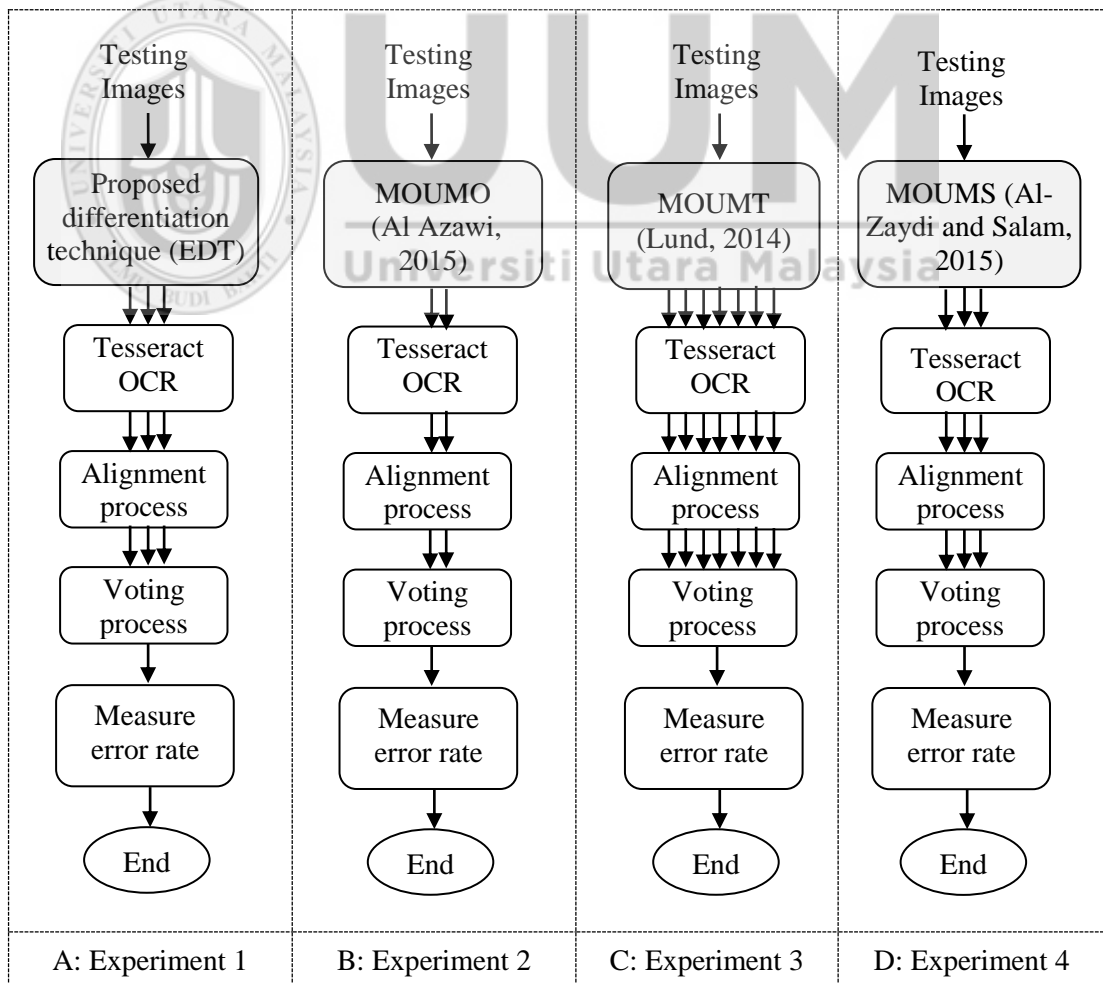


Figure 3.11. Experiments used to evaluate the proposed differentiation technique

Figure 3.11 shows that the evaluation process has been achieved by comparing the output of the OCR using the proposed differentiation technique with the outputs of OCR using three related existing techniques. It also shows that the existing techniques using in the evaluation process are MOUMT technique (Lund, 2014), MOUMO technique (Al Azawi, 2015), and MOUMS technique (Al-Zaydi & Salam, 2015). MOUMT used seven values of threshold, MOUMO used multiple OCR systems, and MOUMS used scanning an image several times.

On the other hand, all experiments in Figure 3.11 used same testing dataset as described in Section 3.5.1.1. Furthermore, they also use same metrics in measuring OCR error rate. These metrics are word error rate, character error rate, and non-word error rate. The explanation on how can measure these metrics are described in Section 3.5.3. The alignment process has been performed in all experiments using the Smith-Waterman technique (Al-Zaydi & Salam, 2015; Lund, 2014) while voting process has been performed using the technique proposed by Al-Zaydi and Salam (2015).

### **3.5.2.2 Alignment Technique Evaluation**

This section explained the evaluation process of the proposed alignment technique to reduce the error rate of the OCR. The evaluation process has been achieved by comparing the output of the OCR using the proposed alignment technique with the outputs of OCR using three related existing techniques. Figure 3.12 shows the experiments used to evaluate the proposed alignment technique.

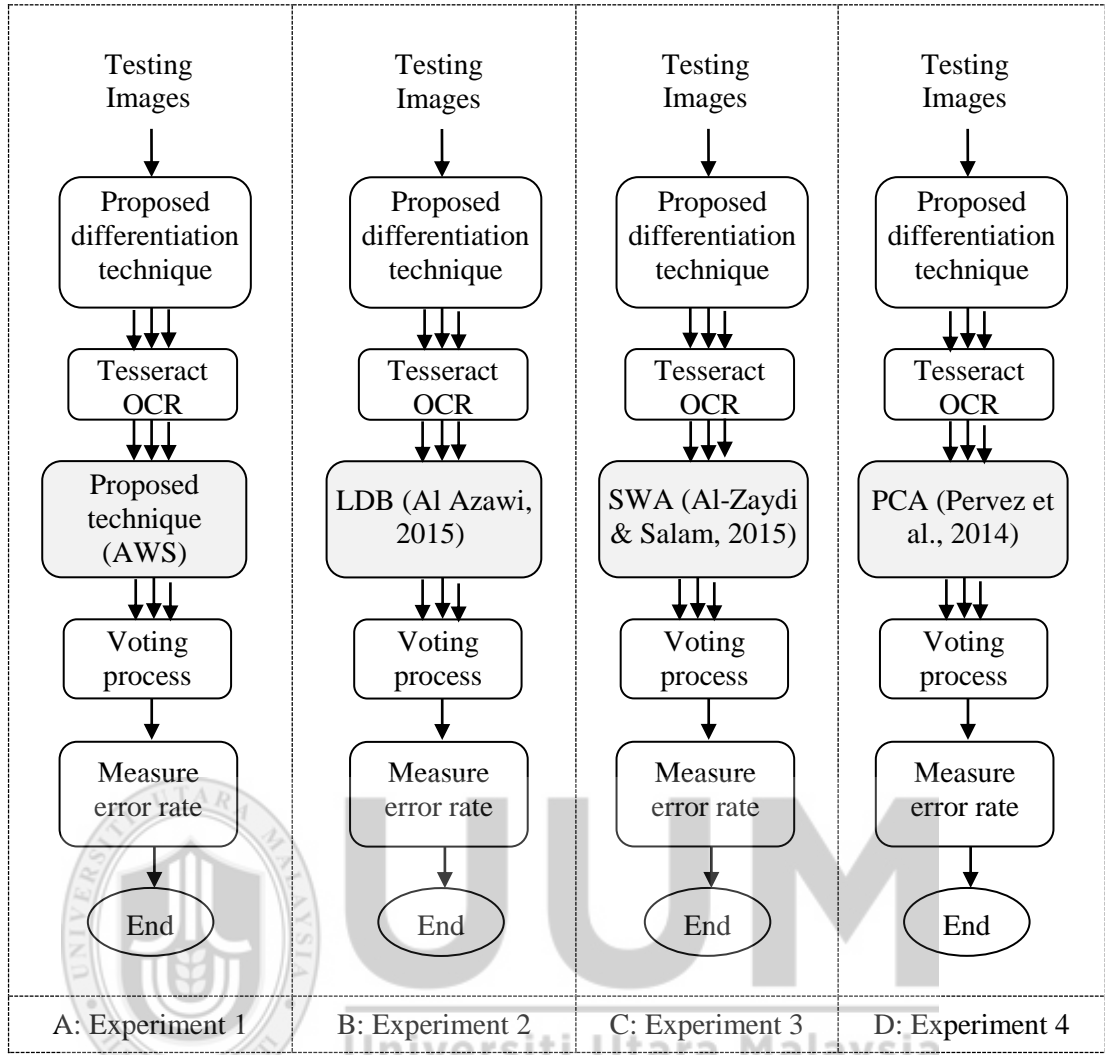


Figure 3.12. Experiments used to evaluate the proposed alignment technique

Figure 3.12 shows that the existing techniques using in the evaluation process are ProbCons alignment (PCA) used by Pervez et al. (2014), Smith–Waterman alignment (SWA) used by Al-Zaydi and Salam (2015), and Levenshtein distance with backtrack alignment (LDB) used by Al Azawi (2015). It also shows that all experiments used same testing dataset as described in Section 3.5.1.1. Furthermore, they also use same metrics in measuring OCR error rate. These metrics are word error rate, character error rate, and non-word error rate. The explanation on how can measure these metrics are described in Section 3.5.3. The voting process has been performed using the technique proposed by Al-Zaydi and Salam (2015).

### 3.5.2.3 Voting Technique Evaluation

This section explained the evaluation process of the proposed voting technique to reduce the error rate of the OCR. The evaluation process has been achieved by comparing the output of the OCR using the proposed voting technique with the outputs of OCR using three related existing techniques. Figure 3.13 shows the experiments used to evaluate the proposed voting technique.

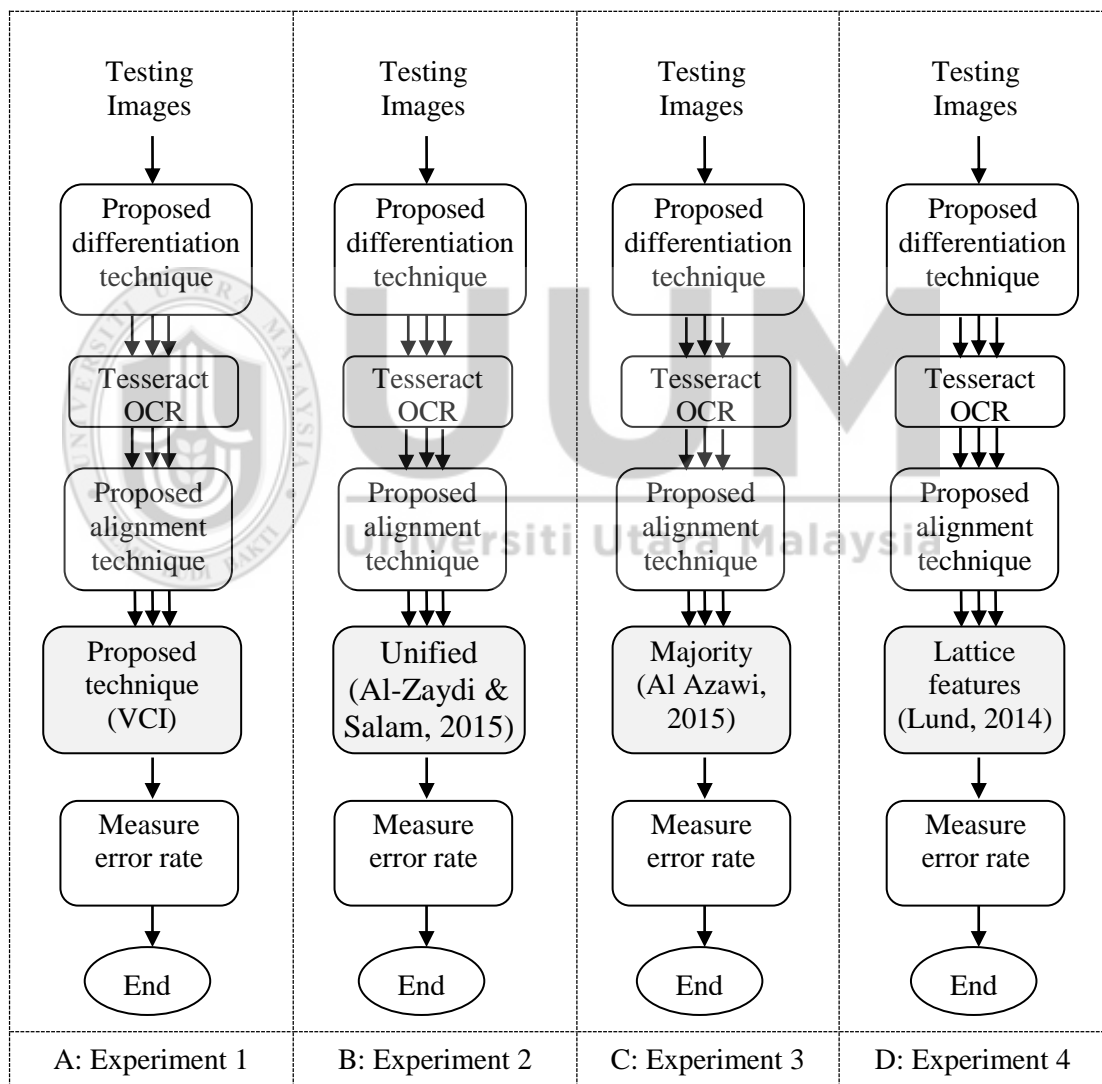


Figure 3.13. Experiments used to evaluate the proposed voting technique

Figure 3.13 shows that the existing techniques using in the evaluation process are the Majority technique (Al Azawi, 2015), Lattice features technique (Lund, 2014),



and a Unified technique (Al-Zaydi & Salam, 2015). It also shows that all experiments use same testing dataset as described in Section 3.5.1.1. Furthermore, they also use same metrics in measuring OCR error rate. These metrics are word error rate, character error rate, and non-word error rate. The explanation on how can measure these metrics are described in Section 3.5.3.

### 3.5.2.4 Hybrid Model Evaluation

This section explained the evaluation process of the proposed hybrid model to reduce the error rate of the OCR. The evaluation process has been achieved by comparing the output of the OCR using the proposed hybrid model with the outputs of OCR using three related existing hybrid works. Figure 3.14 shows the experiments used to evaluate the proposed hybrid model.

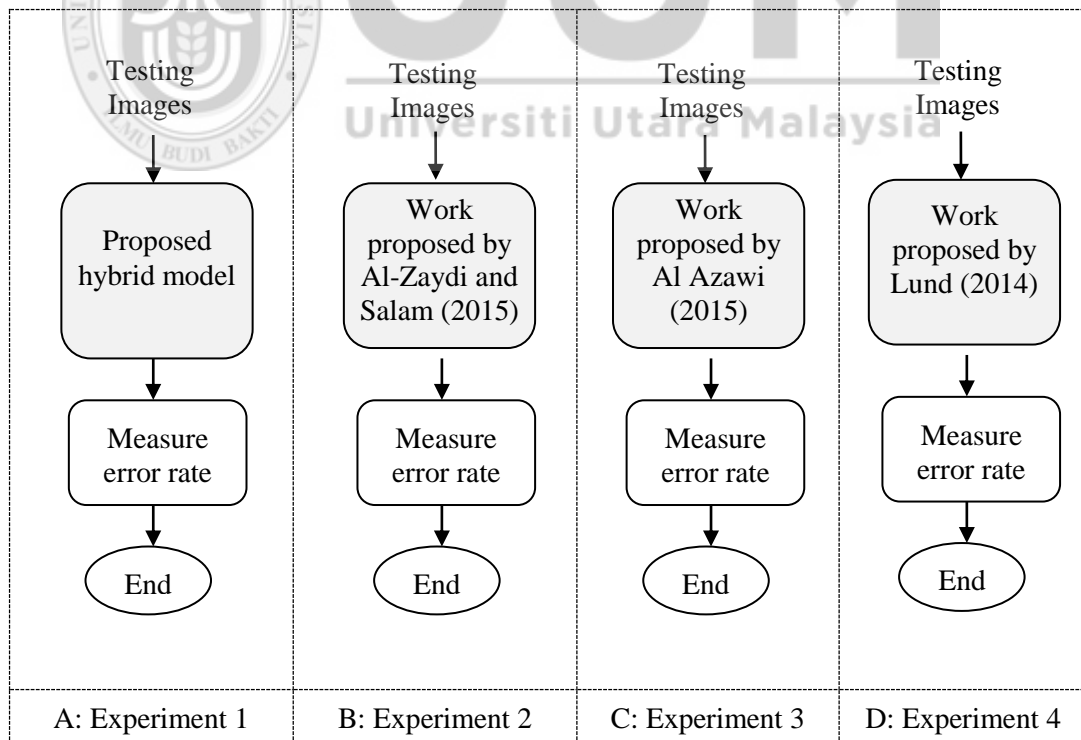


Figure 3.14. Experiments used to evaluate the proposed hybrid model

Figure 3.14 shows that the existing hybrid works using in the evaluation process are proposed by Al Azawi (2015), Lund (2014), and Al-Zaydi and Salam (2015). It also shows that all experiments use same testing dataset as described in Section 3.5.1.1. Furthermore, they also use same metrics in measuring OCR error rate. These metrics are word error rate, character error rate, and non-word error rate. The explanation on how can measure these metrics are described in the next section.

### 3.5.3 Measurements

In this study, three metrics were used in the evaluation process of the hybrid model and proposed techniques. These metrics are the word error rate (WER), character error rate (CER), and non-word error rate (NWER). WER is used to measure the rate of all wrong words in the OCR output text while NWER is only used to measure the rate of non-word errors in the OCR output text. CER is used to measure the rate of all wrong characters in the OCR output text. Equations 3.1, 3.2, and 3.3 shows how to compute the WER, CER, and NWER respectively (Al-Zaydi & Salam, 2015; Bassil & Alwani, 2012c; Dehkordi, 2014; El-Mahallawy, 2008; Kolak & Resnik, 2005; Raaid & Rafid, 2015).

$$WER = \frac{\text{No. of wrong words in output OCR text}}{\text{No. of all words in reference text}} * 100 \quad (3.1)$$

$$CER = \frac{\text{No. of wrong characters in output OCR text}}{\text{No. of all characters in reference text}} * 100 \quad (3.2)$$

$$NWER = \frac{\text{No. of non - words in output OCR text}}{\text{No. of all words in reference text}} * 100 \quad (3.3)$$

In addition to that, Equation 3.4 (Dehkordi, 2014; Lund, 2014) was used to measure the relative decrease in OCR error rate:

$$\text{Relative decrease} = \frac{\text{Error rate (A)} - \text{Error rate (B)}}{\text{Error rate (A)}} * 100 \quad (3.4)$$

Where the term “Error rate (A)” represents OCR error rate of the best existing technique while the term “Error rate (B)” represents OCR error rate of the proposed technique. Furthermore, the term “error rate” refers to the WER, CER, or NWER.

#### **3.5.4 Statistical Test**

In addition to three metrics used in the evaluation process, this research conducted a statistical test to show if the reduction in the terms of the OCR error rate is significant or not. A statistical test of the difference has been measured using analysis of variance (ANOVA). This test is a statistical method of comparing three samples or more in terms of their values (Howell, 2012). It shows the average of each group and the variance inside each group. It also shows whether this difference is statistically significant or due to chance and other circumstances. This research uses Microsoft Excel 2010 to analysis data, and to perform an ANOVA. Table 3.1 shows the major variables resulted from ANOVA with a description for each one.

Table 3.1

*Major variables resulted from ANOVA*

No.	Variables	Description
1	Average	Average of a group
2	Variance	Variance inside each group
3	Count	Number of items in a group
4	F	The F value that needs to be greater than (F critical in row 6) in order for the difference to be significant
5	P-value	P-value: the probability that the difference in mean for groups is real and not due to chance.
6	F-crit	The value that needs to exceed by (F value in row 4) in order for the difference to be significant at the 5% level

Table 3.1 shows that the most important variables are P-value in row 5, and “F-crit” in row 6. P-value means the probability that the difference in means for groups is real and not due to chance. “F-crit” in row 6 is a value that needs to exceed by “F” in row 4. P-value should be less than 0.05, and “F” value should be greater than F-crit in order for the difference in the means to be significant. Otherwise, it is not true, and it is not real (Howell, 2012).

### 3.6 Summary

This research has followed a methodology that has been used in developing the most successful OCR post-processing techniques, which is known as the experimental methodology. This kind of methodology can accept the feedback when some modifications are needed. It was suitable for this research because the proposed model adopted in this study can be changed according to the results of the testing.

This chapter began with a description of the research phases. After that, the design and development of the hybrid model were presented. Lastly, the details of the testing dataset, training dataset, experimental settings, and measurements were explained.



## **CHAPTER FOUR**

### **PROPOSED DIFFERENTIATION TECHNIQUE**

#### **4.0 Introduction**

As mentioned in Chapter 3, the proposed differentiation technique has been referred to as EDT by this research. The chapter begins by introducing the concept of the EDT. Furthermore, an example is provided to show how the EDT technique works. The chapter continues with a discussion on the algorithm to implement EDT. The comparison of the advantages and disadvantages between EDT and existing techniques are explained. The experimental results of the EDT were presented next, and the chapter ends with a summary.

#### **4.1 Differentiation Technique (EDT) Concept**

As mentioned before in Chapter 1, the differentiation process is used to generate MO of OCR for the same input image. The proposed differentiation technique of this study is based on new differentiation function to generate N-versions from the original image. The generated versions are similar, but not identical. Although several versions of the same original image can be generated by differentiation function, this study generates only three versions to reduce complexity. Figure 4.1 shows the differentiation function and its implementation for images 1, 2, and 3.

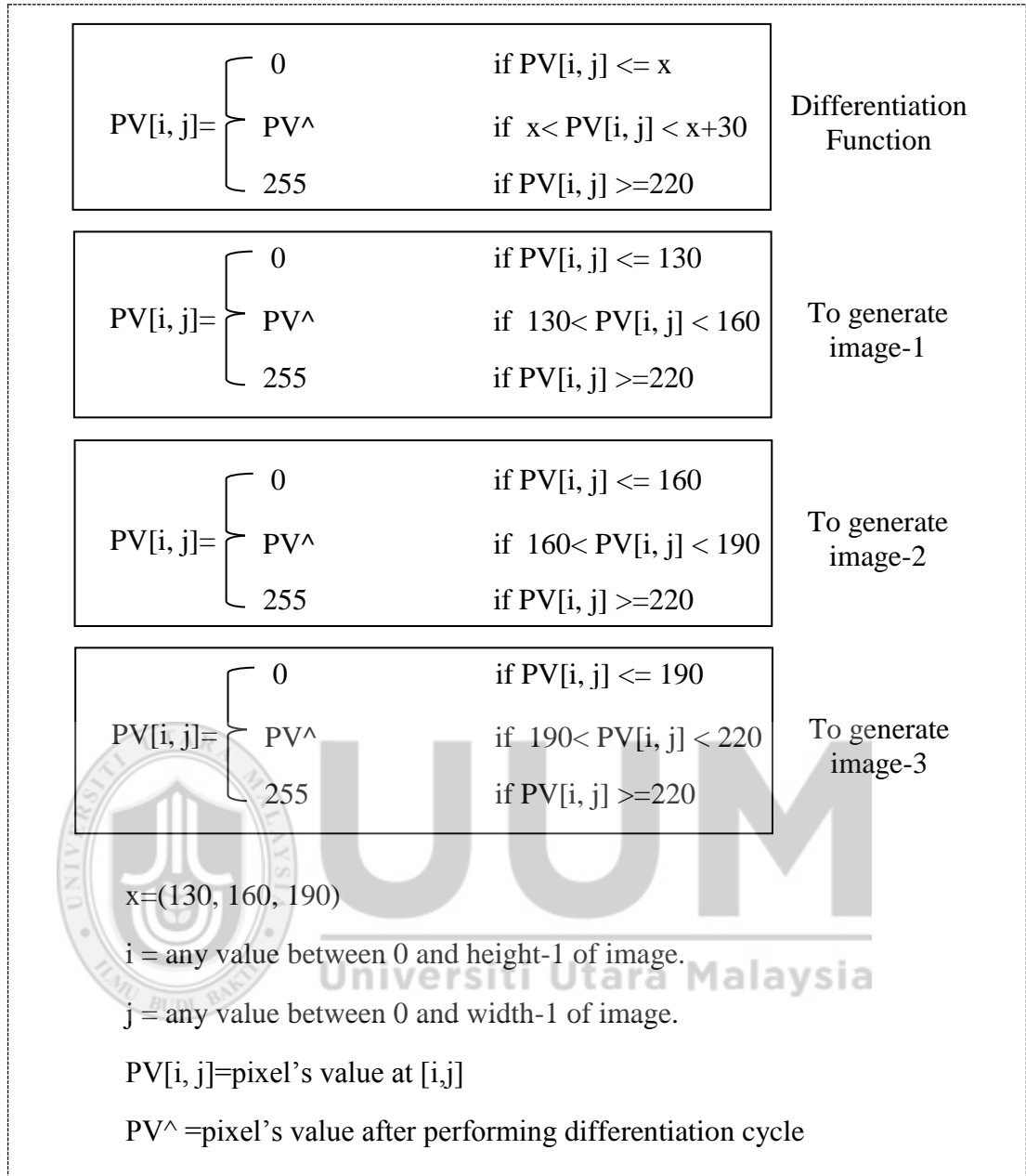


Figure 4.1. Differentiation function and its implementation for images 1, 2, and 3

Figure 4.1 shows that the differentiation function divided pixels' values of an image into three groups. The first group contains the pixels that have values close to the black, which range from (0 to  $x$ ). The second group contains the pixels that have values close to the white, which range from (220 to 255). The last group contains weak values that lie between the first and second group, which range between ( $x+1$  and  $x+29$ ).

The goal of differentiation function is to confirm the first group by making its members black and confirm the second group by making its members white while the last group will be subjected to the differentiation cycle. As mentioned in Section 3.3.1, the proposed technique focuses on gray pixels in the path of black pixels because white pixels do not represent any information about images of characters while black pixels give strong evidence that they may represent information about them. Therefore, the purpose of a differentiation cycle is to modify gray pixels values in the path of black pixels by performing several operations called a cycle on them. This will lead generating differences between resulting versions of the input image.

A variable named  $x$  is used as an initial step to achieve a differentiation function goal of dividing pixels' values of an image into three groups. The value of  $x$  is changed during the production of each image. It takes the values 130, 160, and 190 to produce image 1, 2, and 3 respectively. The values of  $x$  are selected based on two factors. The first factor is that they should lie between black's pixels and white's pixels. Therefore, the differentiation function selected values of  $x$  between 130 and 220. This is because pixels' values under than 130 are close to the black, and pixels' values greater than 220 are close to the white. The second factor is that this research conducted series of experiments as described in Section 3.3.1 and based on results; it found that the best values of  $x$  are 130, 160, and 190.

To produce any version of the original image, several operations are performed. Firstly, each pixel's value that is equal or smaller than  $x$  will change to zero, while each pixel's value that is equal or greater than 220 will change to 250. The reason for this is to confirm the stronger pixels. Secondly, any pixel's value that is greater than  $x$  and less than 220 will remain unchanged until performing differentiation cycle on



them. Differentiation cycle will start by identifying all pixels having values between  $(x+1)$  and  $(x+29)$ , located beside pixels having values equal to zero as shown in part A in Figure 4.2. These will become primary starting pixels for the process of differentiation.

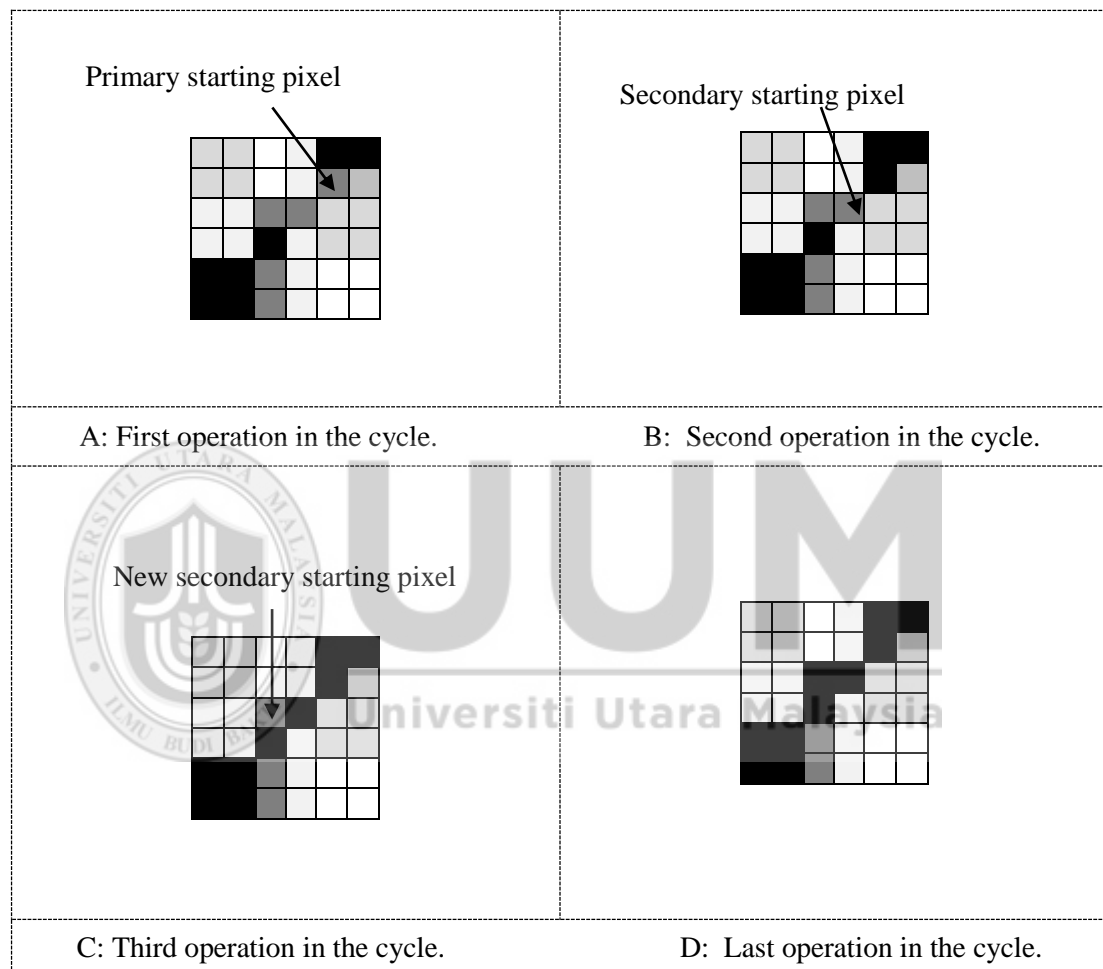


Figure 4.2. Simple example on differentiation cycle for one primary starting pixel

In the proposed differentiation cycle, each primary starting pixel has a cycle of operations: (1) value of starting pixel is changed to zero, and (2) all the neighboring pixels from all sides with non-zero values are arranged in ascending order, so that the pixel having the smallest value becomes a secondary starting pixel, on condition that its value lies between  $(x+1)$  and  $(x+29)$ . If these conditions are met, then all previous operations (1 and 2) are performed for the new starting pixel and so on, otherwise the

cycle is ended for current starting pixel, and another cycle for next primary starting pixel is initiated. At the end of a differentiation cycle of each primary starting pixel, the value of a single pixel or the values of multiple pixels are changed based on the proposed differentiation cycle.

There are three advantages resulting from the proposed differentiation technique. The first advantage is that it does not require scanning image multiple times that is considered a boring process as mentioned by Al-Zaydi and Salam (2015). The second advantage is that it does not require connecting different OCR systems that is considered a difficult and manual process as stated by Al-Zaydi and Salam (2015). The last advantage is that it is better than existing technique proposed by Lund (2014) that generates seven outputs by using seven threshold values. Figure 4.3 shows why the proposed differentiation technique is better than existing technique proposed by Lund (2014).

From Figure 4.3, it can be seen that resulting image produced by existing technique proposed by Lund (2014) is not similar to the original image while resulting image produced by proposed differentiation technique is similar. As mentioned previously, this is because existing technique makes all pixel values above threshold value white. This will lead to loss of some important features from characters' images (Al-Zaydi & Salam, 2015). The effect of losing some features from characters' images is that the number of wrong words in OCR outputs will be increased. In contrast, the proposed differentiation technique preserves features of characters' images. Furthermore, it restores them to the original shape.

For example, Figure 4.3 contains only one character “n”. Therefore, existing technique proposed by Lund (2014) will produce seven outputs for this character. However, most of them will be wrong as shown in Figure 4.3. Hence, selecting the best OCR output by the voting process will become difficult, and may produce errors. In contrast, the proposed differentiation technique produces three outputs, and most of them similar to the original. Hence, selecting the best OCR output by the voting process will become easier.

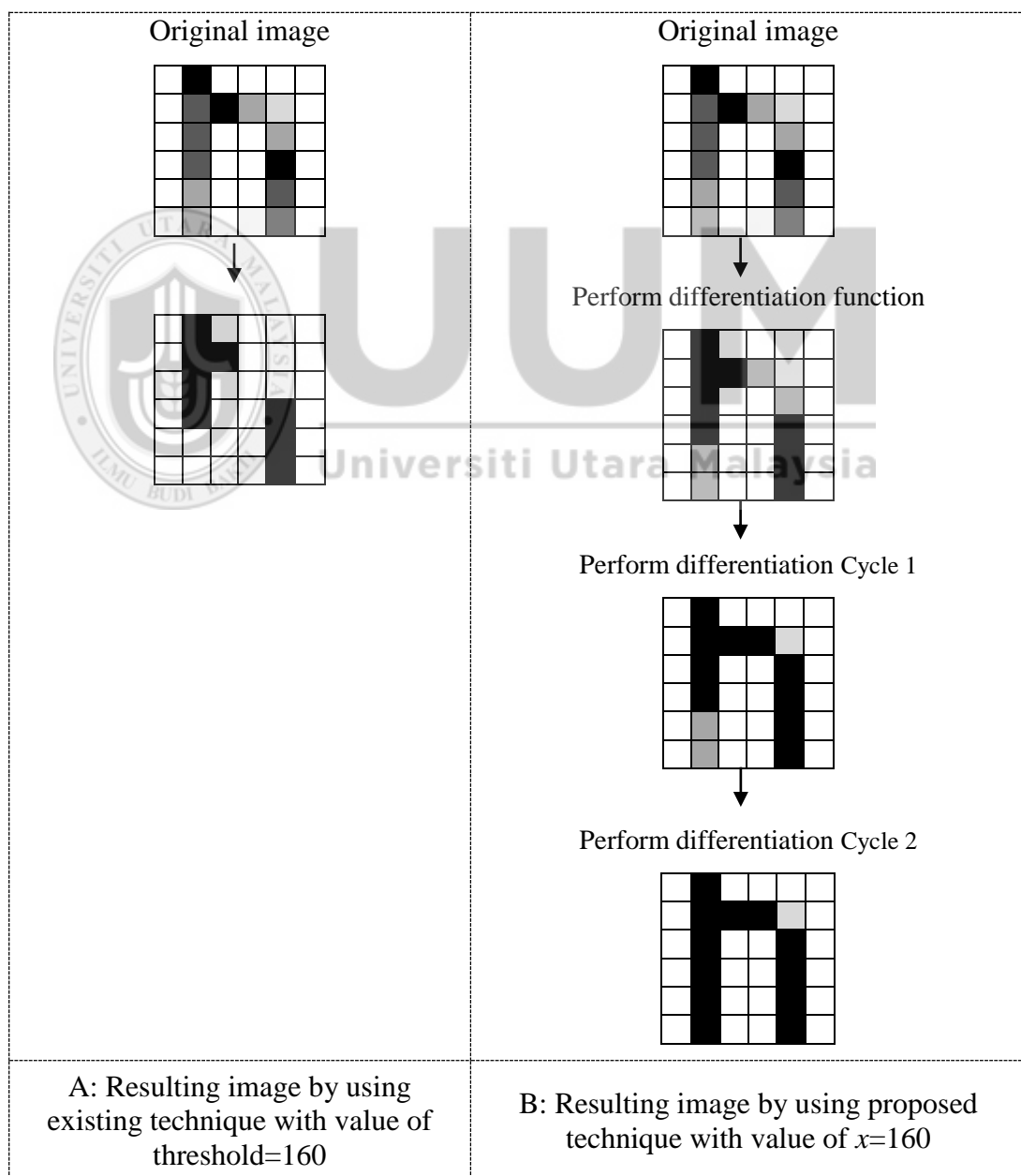


Figure 4.3. Simple example on proposed differentiation technique.

## 4.2 EDT Algorithm

Algorithm 4.1 represents the pseudo code for proposed differentiation technique.

<b>Algorithm 4.1: Proposed differentiation technique</b>	
<i>S1</i>	Let <i>z</i> is array of [130, 160, 190]
<i>S2</i>	Let <i>k</i> =0 // counter for elements in the array <i>z</i>
<i>S3</i>	Let <i>x</i> = <i>z</i> [ <i>k</i> ] // <i>x</i> is used to save the threshold value
	// Produce <i>N</i> -versions of Arabic dataset images
<i>S4</i>	Let <i>c</i> =0 // counter for images in the Arabic dataset images
<i>S5</i>	Select image[ <i>c</i> ] from the dataset images
<i>S6</i>	If each pixel value $\leq x$ , then this value will be changed to zero
<i>S7</i>	Let <i>PSP</i> is array of pixels having values between ( <i>x</i> +1) and ( <i>x</i> +29), located beside pixels having values equal to zero
<i>S8</i>	Let <i>y</i> = 0 // counter for elements in the array <i>PSP</i>
<i>S9</i>	Let Pixel( <i>i</i> , <i>j</i> ) refers to the position of <i>PSP</i> [ <i>y</i> ] in image[ <i>c</i> ]
<i>S10</i>	Pixel( <i>i</i> , <i>j</i> )=0
<i>S11</i>	Let <i>SSP</i> is array of neighboring pixels to the Pixel( <i>i</i> , <i>j</i> ) from all sides with non-zero values
<i>S12</i>	Let <i>h</i> = minimum value in array <i>SSP</i>
<i>S13</i>	If value of <i>h</i> between ( <i>x</i> +1) and ( <i>x</i> +29) then Let Pixel( <i>i</i> , <i>j</i> ) refers to the position of minimum value in array <i>SSP</i> in image[ <i>c</i> ] and go to <i>S10</i> else if <i>y</i> is the last element in <i>PSP</i> array then go to <i>S14</i> else <i>y</i> = <i>y</i> +1 and go to <i>S9</i>
<i>S14</i>	Save image[ <i>c</i> ] with different name // to maintain original dataset images
<i>S15</i>	if the image[ <i>c</i> ] is the last one in the dataset images then go to <i>S16</i> else <i>c</i> = <i>c</i> +1 and go to <i>S5</i>
<i>S16</i>	if <i>z</i> [ <i>k</i> ] is the last element in the array of <i>z</i> then go to <i>S17</i> else <i>k</i> = <i>k</i> +1 and go to <i>S3</i>
<i>S17</i>	end

From step S1 of Algorithm 4.1, it can be seen that the proposed differentiation technique used an array called “z”, which has three elements 130, 160, and 190. These elements were used to produce 3-versions of Arabic dataset images as explained in Section 4.1. Steps S2 to S4 define the variables used in this algorithm. In Step S5, each image from the testing dataset was selected in order to process by the proposed differentiation technique. In step S6, if the value of each pixel  $\leq x$ , then this value will be changed to zero.

After that, some gray pixels values will be changed to the black if they located in the path of black pixels (Steps S7 to S14). As mentioned previously, the proposed differentiation technique focuses on gray pixels in the path of black pixels because white pixels do not represent any information about images of characters while black pixels give strong evidence that they may represent information about them. This study finds the path of black pixels by focusing only on gray pixels that must be located beside black pixels while other gray pixels that do not satisfy this condition will be ignored. In Step S15, next image from Arabic dataset images would select to handle by this technique while in step S16 all previous steps are repeated for the next element in the array “z”.

After performing Algorithm 4.1, three similar but non-identical dataset images are produced. These three dataset images will be sent to the three versions of same OCR engine in order to turn them into three outputs text. The voting process will select the best between them to produce a single OCR output text.

### 4.3 Experimental results

This section describes the results of the experiment performed on proposed differentiation technique (EDT). In this section, the proposed differentiation technique EDT has been implemented. Furthermore, three related existing techniques have also been implemented to be used in the evaluation of EDT. They are the MOUMO, which is used by Al Azawi (2015), MOUMT, which is used by Lund (2014), and MOUMS, which is used by Al-Zaydi and Salam (2015). As mentioned previously, this study used three metrics in the evaluation process. They are word error rate, character error rate, and non-word error rate. The design of evaluation process and the details of testing dataset were explained in Chapter 3.

#### 4.4.1 Word Error Rate (WER)

Table 4.1 presents the experimental results of the EDT evaluation using the WER metric, while Figure 4.4 shows the clustered column graph for the WER values listed in this table. Note the gray column in Table 4.1 represents the results of the proposed differentiation technique (EDT).

Table 4.1

*Experimental results of the EDT evaluation using the WER metric*

	<b>MOUMS</b> (Al-Zaydi & Salam, 2015)	<b>MOUMT</b> (Lund, 2014)	<b>MOUMO</b> (Al Azawi, 2015)	<b>Proposed technique (EDT)</b>
Total words	39048	39048	39048	39048
Wrong words	20254	24315	21511	17938
WER	51.87%	62.27%	55.09%	45.94%

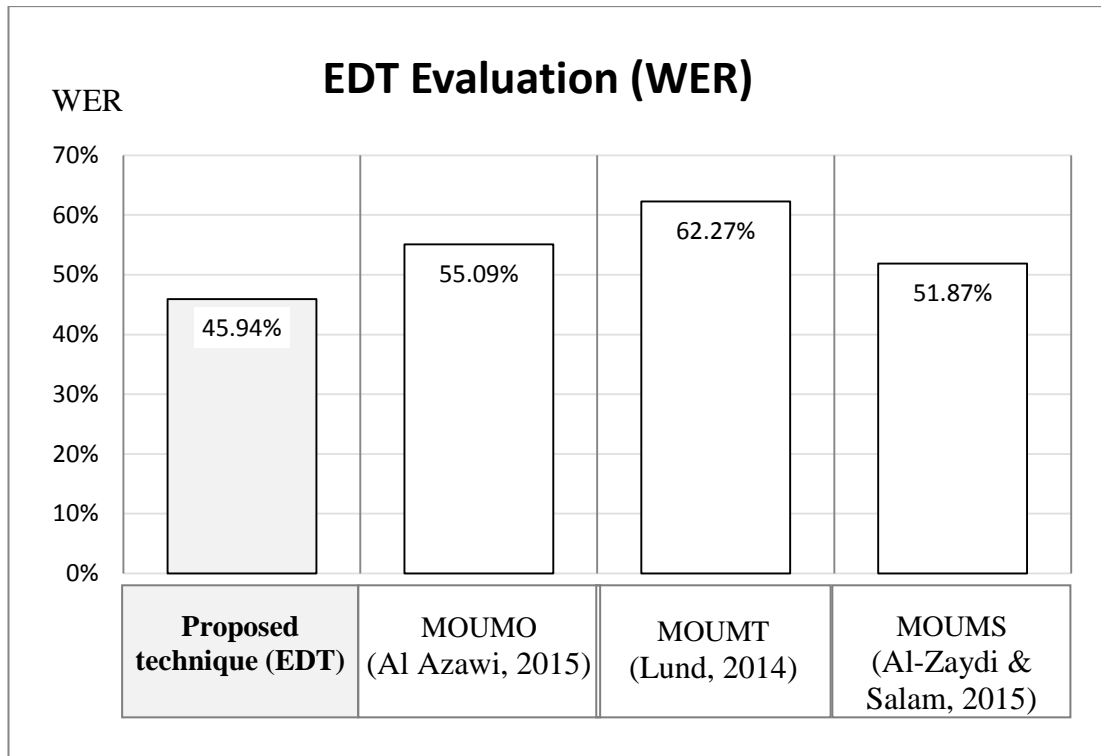


Figure 4.4. Clustered column graph for the WER values listed in Table 4.1.

From Table 4.1 and Figure 4.4, it can be clearly seen that the WER value for each differentiation technique is different from the others. Overall, they show that MOUMT technique had the highest percentage value of WER with the rate of 62.27%. This is followed by MOUMO 55.09%, and MOUMS 51.87%. Furthermore, it can be seen that WER of the proposed technique EDT had the lowest percentage value of OCR error rate than the others with the rate of 45.94%. This technique has an 18.09% relative decrease on the mean WER of the three existing differentiation techniques and 11.43% relative decrease on the best WER of them. The relative decrease in WER has been measured using Equation 3.4. This indicates that the proposed technique EDT had a reduction in the WER metric compared to the existing techniques. As mentioned previously, this research performed a statistical method using an ANOVA-test to show if the reduction in WER of the EDT is significant or not. Figure 4.5 shows ANOVA-test results using the WER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (EDT)	130	45.94	582.10
MOUMO (Al Azawi, 2015)	130	55.09	494.64
MOUMT (Lund, 2014)	130	62.27	432.38
MOUMS (Al-Zaydi & Salam, 2015)	130	51.87	477.76

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	18772.08	3	6257.36	12.60	0.00	2.62
Within Groups	256307.26	516	496.72			
Total	275079.34	519				

Figure 4.5. ANOVA-test results for the WER values

Figure 4.5 shows that the number of tests for each technique is 130 as shown in the term “Count” in column 2. The input for each test is a single image, and the output is OCR error rate. Figure 4.5 also shows that the average of OCR error rate for EDT is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 12.60 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the OCR error rate among four techniques is real and not due to chance. Therefore, EDT is better than other in terms of the WER.



#### 4.4.2 Character Error Rate (CER)

Table 4.2 displays the results of the four experiments in terms of the least CER value, while Figure 4.6 shows the clustered column graph for the CER values listed in this table.

Table 4.2

*Experimental results of the EDT evaluation using the CER metric*

	<b>MOUMS</b> (Al-Zaydi & Salam, 2015)	<b>MOUMT</b> (Lund, 2014)	<b>MOUMO</b> (Al Azawi, 2015)	<b>Proposed technique (EDT)</b>
Total characters	231896	231896	231896	231896
Wrong characters	64206	84370	67499	53039
CER	27.69%	36.38%	29.11%	22.87%

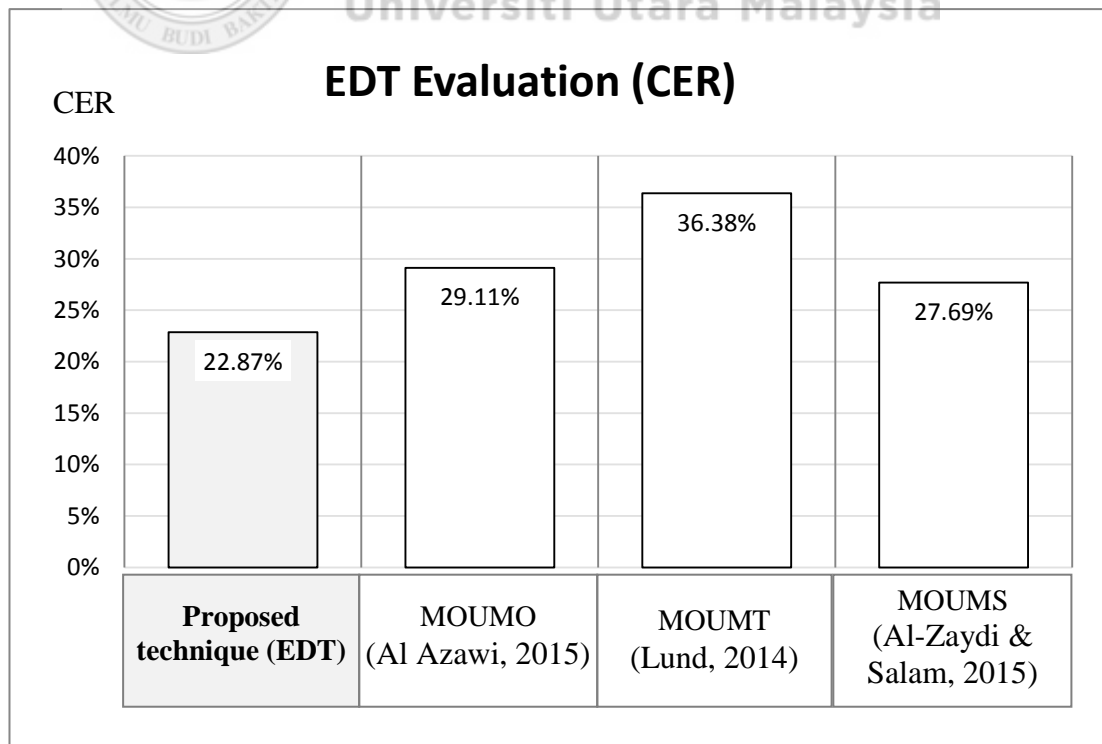


Figure 4.6. Clustered column graph for the CER values listed in Table 4.2.

As it can be seen in Table 4.2 and Figure 4.6, the differentiation techniques of MOUMS and MOUMO show slightly a difference in the values of CER with rates of 27.69% and 29.11% respectively. Furthermore, the CER values for both MOUMS and MOUMO are less than CER value for MOUMT technique, which has a rate of 36.38%. The proposed differentiation technique EDT outperformed the existing techniques in terms of CER with the rate of 22.87%. The proposed technique has a 25.32% relative decrease on the mean CER of the three existing differentiation techniques and 17.39% relative decrease on the best CER of them. The previous values of CER show that the proposed technique EDT had the highest percentage decrease in the number of wrong characters compared to the MOUMS, MOUMT and MOUMO techniques.

This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the CER of the EDT is significant or not. Figure 4.7 shows ANOVA-test results using the CER values. From this figure, it can be clearly seen that the average of CER for EDT is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 15.19 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the CER among four techniques is real and not due to chance. Therefore, EDT is better than other in terms of the CER.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (EDT)	130	22.87	240.52
MOUMO (Al Azawi, 2015)	130	29.11	224.69
MOUMT (Lund, 2014)	130	36.38	205.00
MOUMS (Al-Zaydi & Salam, 2015)	130	27.69	432.16

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	12562.74	3	4187.58	15.19	0.00	2.62
Within Groups	142205.09	516	275.59			
Total	154767.84	519				

Figure 4.7. ANOVA-test results for the CER values

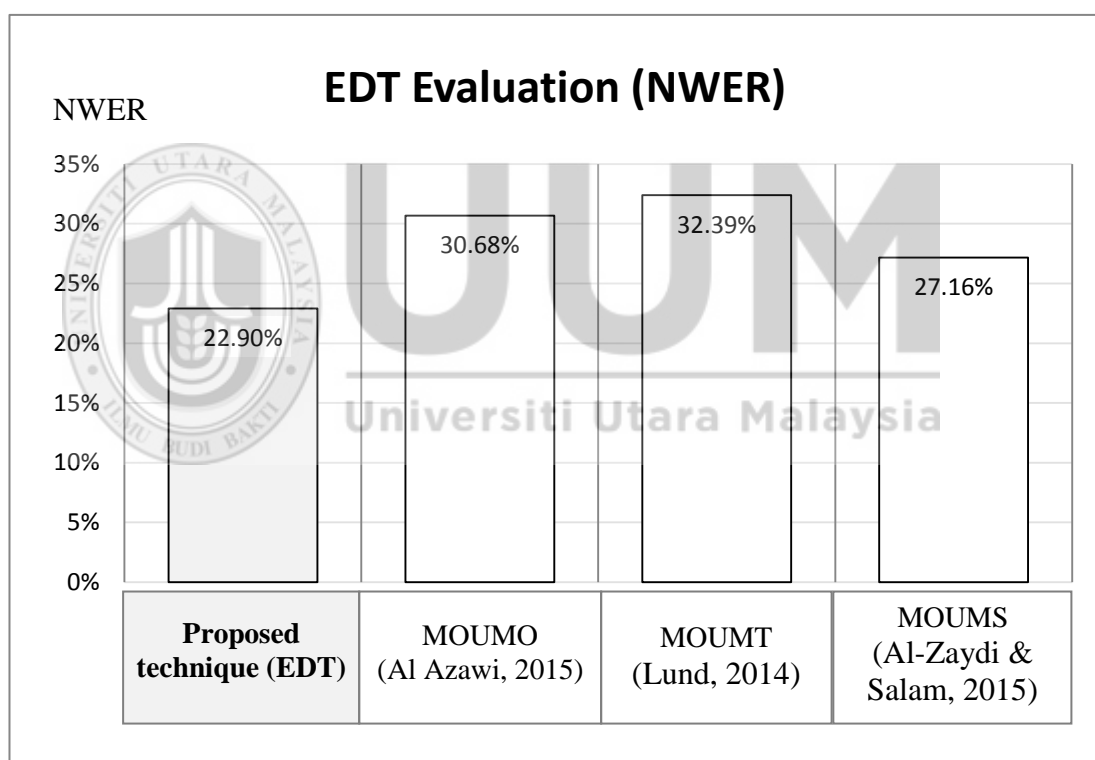
#### 4.4.3 Non-Word Error Rate (NWER)

Table 4.3 presents the experimental results of the EDT evaluation using the NWER metric, while Figure 4.8 shows the clustered column graph for the NWER values listed in this table.

Table 4.3

*Experimental results of the EDT evaluation using the NWER metric*

	<b>MOUMS</b> (Al-Zaydi & Salam, 2015)	<b>MOUMT</b> (Lund, 2014)	<b>MOUMO</b> (Al Azawi, 2015)	<b>Proposed technique (EDT)</b>
Total words	39048	39048	39048	39048
Non-word errors	10606	12646	11980	8943
NWER	27.16%	32.39%	30.68%	22.90%



*Figure 4.8. Clustered column graph for the NWER values listed in Table 4.3.*

Table 4.3 and Figure 4.8 show that the worse performance was produced by MOUMT technique with the rate of 32.39%. The techniques MOUMS and MOUMO show performances better than MOUMO technique, with rates of 27.16% and 30.68% respectively. The proposed differentiation technique EDT achieved the best

performance in terms of NWER value with a rate of 22.90%. The proposed differentiation technique EDT has a 23.44% relative decrease on the mean NWER of the three existing differentiation techniques and 15.68% relative decrease on the best NWER of them. This indicates that the proposed technique EDT is the best compared to the three existing differentiation techniques in terms of NWER metric. This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the NWER of the EDT is significant or not. Figure 4.9 shows ANOVA-test results using the NWER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (EDT)	130	22.90	182.62
MOUMO (Al Azawi, 2015)	130	30.68	264.80
MOUMT (Lund, 2014)	130	32.39	197.90
MOUMS (Al-Zaydi & Salam, 2015)	130	27.16	137.99

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	7412.57	3	2470.86	12.62	0.00	2.62
Within Groups	101046.60	516	195.83			
Total	108459.17	519				

Figure 4.9. ANOVA-test results for the NWER values

From Figure 4.9, it can be clearly seen that the average of NWER for EDT is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented

in column 6, which is less than 0.05, and the value of “F” is 12.62 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the NWER among four techniques is real and not due to chance. Therefore, EDT is better than other in terms of the NWER.

#### **4.4.4 Results Discussion**

The experimental results of this study show that the error rate of OCR systems is still high for the Arabic language. High OCR error rate values of this study are similar to the high error rate values for the Arabic language mentioned in previous related studies (Al-Masoudi & Al-Obeidi, 2015; Al-Zaydi & Salam, 2015). The experimental results also show that the MOUMS technique is better from others existing differentiation techniques. These results are similar to the results obtained by Al-Zaydi and Salam (2015).

The worst performances in terms of WER, CER, and NWER values were achieved by MOUMT technique. As mentioned previously, this is because it generates seven outputs using seven threshold values, and it makes all pixel values above threshold value white. This leads to the loss of some important features from characters' images. The effect of losing some features from characters' images is that the number of wrong words in OCR outputs will be increased (Al-Zaydi & Salam, 2015; Lund, 2014). In contrast, the proposed differentiation technique EDT maintains the features of characters' images as explained in Section 4.1. Furthermore, it achieved the lowest values of WER, CER, and NWER compared to the others existing techniques. This is due to the effective design of proposed technique. This design has several advantages as explained in Section 4.3.

#### **4.4 Summary**

The goal of this chapter is to improve OCR accuracy for the Arabic language by proposing a solution to the limitations of the existing differentiation techniques. Therefore, in this chapter, the design details regarding the proposed differentiation technique are presented and discussed. The details explained the concept of EDT, its flowchart, and its contributions. In addition to that, this study conducted four experiments to evaluate this technique using three metrics. Furthermore, this research conducted a statistical test to show if the reduction in the OCR error rate is significant or not. The statistical test of this research has been measured using ANOVA-test. The results of the evaluation process are presented in detail. The experimental results are very encouraging. The proposed technique EDT outperforms other existing related techniques in terms of WER, CER, and NWER. Therefore, the practical results of this chapter indicate that the objective one of this study is achieved. Lastly, the experiments also show that the error rate is high for Arabic text. This presents a fact that it is difficult for OCR accuracy to be 100% for the Arabic language.

## **CHAPTER FIVE**

### **PROPOSED ALIGNMENT TECHNIQUE**

#### **5.0 Introduction**

As mentioned in Chapter 3, the proposed technique has been referred to as ASW by this research. The chapter begins by introducing the concept of the ASW. Furthermore, an example is provided to show how the ASW technique works. The chapter continues with a discussion on the algorithm to implement ASW. The comparison of the advantages and disadvantages of ASW and existing techniques are explained. The experimental results of the ASW were presented next, and the chapter ends with a summary.

#### **5.1 Alignment Technique (AWS) Concept**

As mentioned in Chapter 1, alignment process usually contains errors after processing by existing alignment techniques. Furthermore, most existing alignment techniques, such as the works proposed by Lund et al. (2011), Volk et al. (2011), Lund et al. (2013b), Pervez et al. (2014), Lund (2014), and Al-Zaydi and Salam (2015), require executing a character alignment algorithm between each pair of OCR outputs. However, Lund (2014) and Lopresti and Zhou (1997) mentioned that increase number of executing the character alignment algorithm will increase the probability of errors in the alignment process. Therefore, a novel alignment technique has been designed by this research to make the alignment process is exact and to prevent executing any character alignment algorithm. The first step in designing a solution for alignment problem is to know why it occurred. Alignment problem occurs because existing alignment techniques deal with the output texts after



OCR engines have produced them (Lund & Ringger, 2009). This means the existing techniques do not address the origin of a problem but try to address its effect on OCR output texts. To handle an alignment problem, there is a need to process an image before passing it to the OCR engines, and this has been done in the proposed technique.

As mentioned in Chapter 1, the origin of the alignment problem is that characters in each OCR engine may be deleted, or inserted. The deleted and inserted characters make the number of characters in each OCR output to be unequal to the others. This will lead to changes in locations of words in each OCR-output text. Furthermore, the process of finding locations of words will become difficult due to the misrecognition of characters in each OCR-output text. Figure 5.1 shows a simple example of the loss of words' locations in MO for an image containing only the sentence “*Arabic language is complex*”.

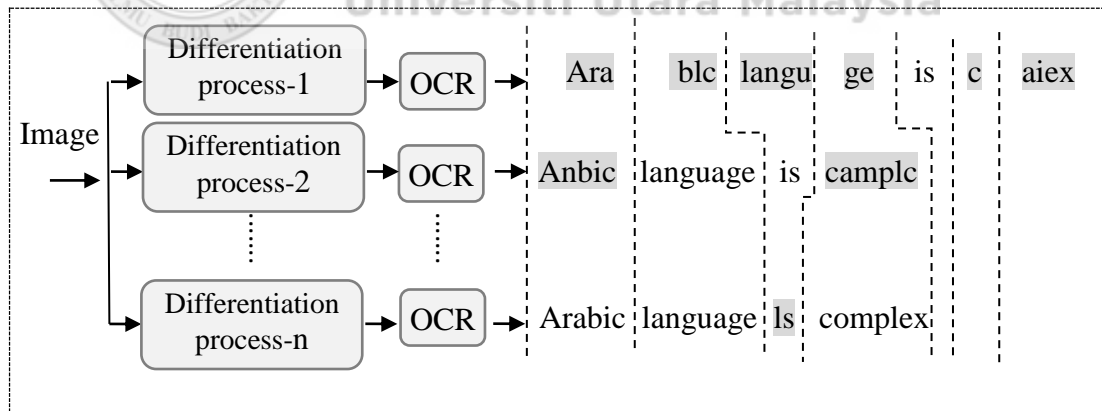


Figure 5.1. Loss of words' locations in MO of OCR

Figure 5.1 shows that the locations of words in MO of OCR are lost for input image containing only four words. This problem becomes difficult for input image containing large numbers of words. Therefore, the idea of solving the alignment problem in the proposed technique is based on pre-saving the locations of words

before sending any image to OCR engines. This means sending words' images after extracting them from the original image to the OCR instead of sending the complete original image. Figure 5.2 and Figure 5.3 show the difference in work between the proposed and existing techniques regarding an alignment problem.

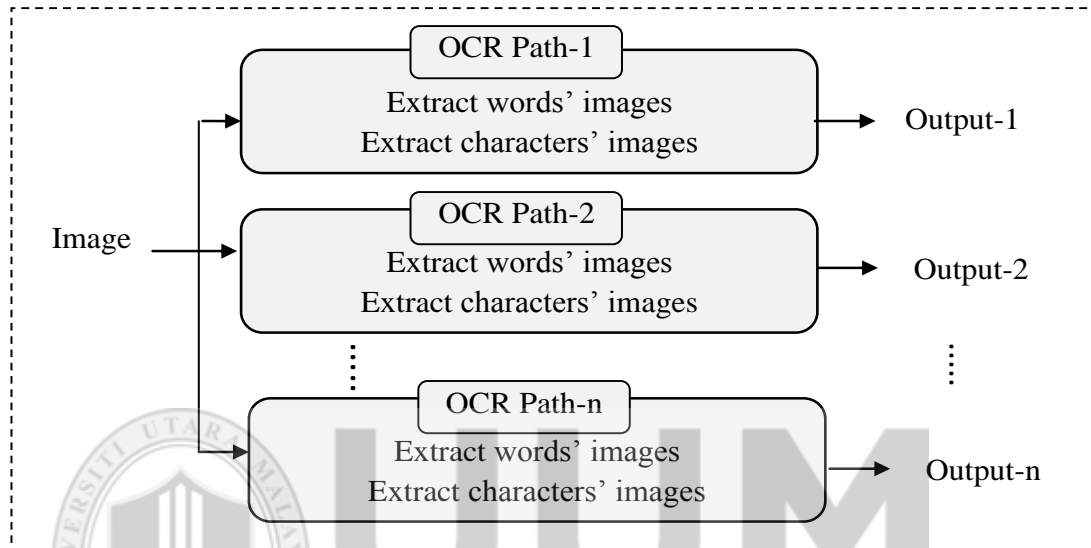


Figure 5.2. Extraction of words' images in the existing techniques

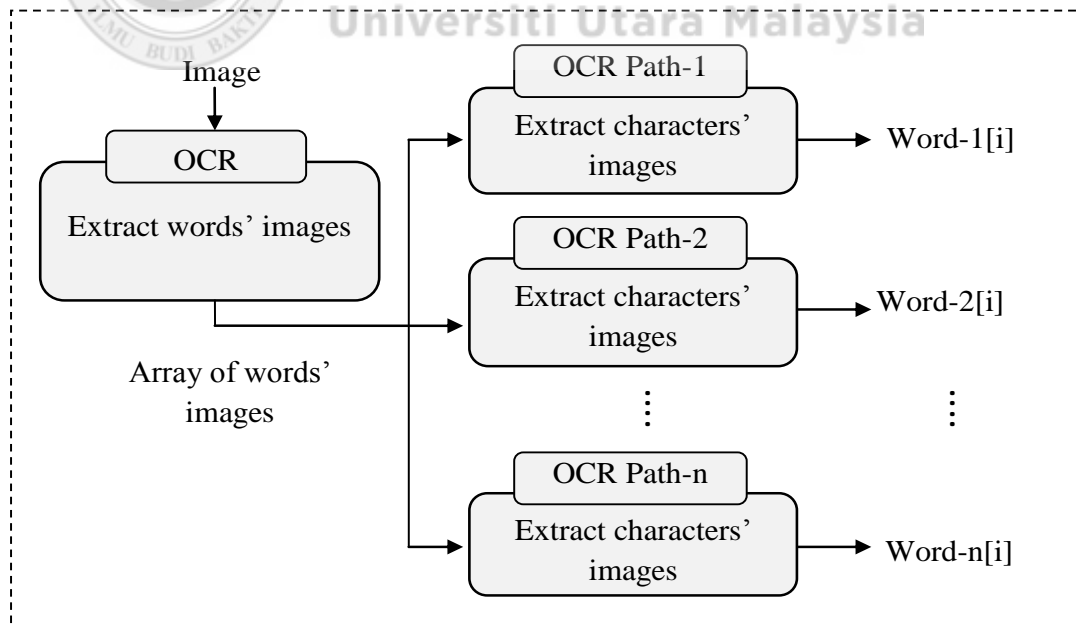
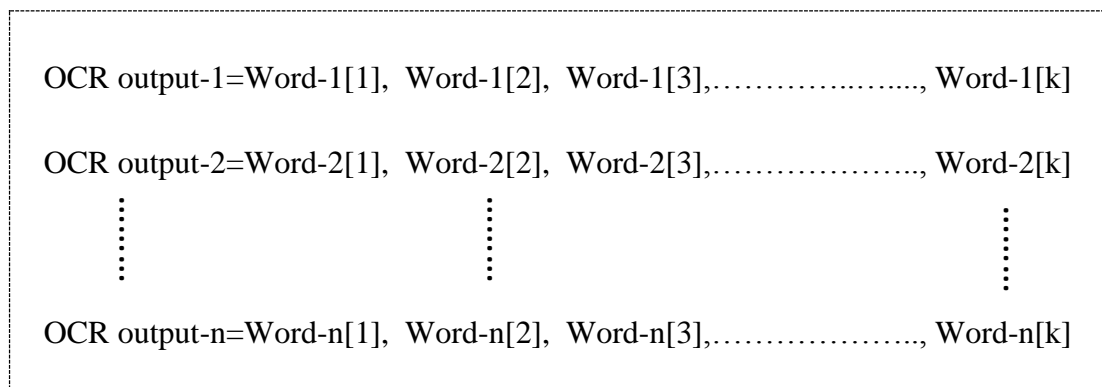


Figure 5.3. Extraction of words' images in the proposed technique

Figure 5.2 shows that the extraction process of words' images is repeated multiple times in existing techniques (Lund, 2014; Lund & Ringger, 2009; Volk et al., 2011), while it is performed once in the proposed technique as shown in Figure 5.3. Furthermore, the proposed technique does not be affected by the number of OCR outputs, and it will also not be affected by the number of characters in each OCR output. This is because words' locations are saved before sending any image to OCR engines. Therefore, deleting, misrecognizing, and inserting characters will not change the locations of words in each OCR output. In addition to that, there is no need to run any character alignment algorithm to align resulting texts of MO because they are already aligned.

There are two advantages that can be derived from the proposed technique. The first advantage is that the solution to the alignment problem because OCR outputs will be represented as a sequence of multiple candidates for the same single word as shown below:



Where the character “n” represents the number of MO used, and the character “k” represents the number of words in each OCR output. For example, by assuming Word [2] is divided into three parts in first OCR output, and it is divided into two

parts in second OCR output, while it is not divided in third OCR output, then the proposed technique will deal with Word [2] as single word in three OCR outputs, while existing techniques will deal with Word [2] as three words in first OCR output, two words in second OCR output, and single word in third OCR output. This means there is no need to implement any character alignment algorithm between resulting texts in MO of OCR in the proposed technique.

The second advantage is that the process of generating differences between MO of OCR will be performed on words' images instead of the whole original image. For example, if an image contains only one word, then the operations of making differences will be performed on this word's image after extracting it instead of doing differentiation's operations on the whole original image. It is to be noted that words of the input image are easy to be identified because of the presence of spaces between them. The spaces between the letters are very small, and sometimes attached to each other, especially when the scanning resolution of the image is low (Ma & Agam, 2013).

## 5.2 AWS Algorithm

Algorithm 5.1 below represents the pseudo code for proposed alignment technique (AWS).

<b>Algorithm 5.1: Proposed alignment technique</b>	
<i>S1</i>	Read input image
<i>S2</i>	Extract words' images from input image and save them in an array <i>z</i>
<i>S3</i>	Let <i>i</i> =0 // counter for elements in the array <i>z</i>
<i>S4</i>	Select <i>z</i> [ <i>i</i> ] // <i>z</i> [ <i>i</i> ] is word image in the array <i>z</i>
<i>S5</i>	Send <i>z</i> [ <i>i</i> ] to the differentiation process to produce N-versions of <i>z</i> [ <i>i</i> ]

<i>S6</i>	Send N-versions of $z[i]$ to the OCR engine to turn them into N-outputs of words
<i>S7</i>	Save N-outputs of words in an N-array of words
<i>S8</i>	if $z[i]$ is the last word image in the array of $z$ then go to <i>S9</i> else $i=i+1$ and go to <i>S4</i>
<i>S9</i>	Apply voting process to select the best words among N-array of words
<i>S10</i>	End

The important step in the proposed alignment technique is to extract words' images from the input image and store them in an array as shown in S2. After that, each word's image in an array will send sequentially to the differentiation process (S5). As was mentioned previously, the differentiation process is used to generate the differences between the OCR multiple outputs. The results of the differentiation process are several similar words' images but not identical. In step S6, each word's image will pass to a single OCR engine to convert it to a word while the next step is to combine the sequence of words resulted from each OCR engine in a single array (S7). In step S9, the resulting arrays of all OCR outputs are sent to the voting process to select the best OCR output.

From Algorithm 5.1, it can be seen that the goal of the alignment process, which is preparing OCR outputs to the voting process by aligning each word in the OCR output with corresponding in other OCR outputs has been achieved. This is because words' locations are saved before sending any image to OCR engines (S2). Therefore, deleting, misrecognizing, and inserting of characters will not change the locations of words in each OCR output. Furthermore, resulting texts of OCR multiple outputs are already aligned according to the words of the input image (S7). In

addition to that, the proposed technique does not need to run any character alignment algorithm to align resulting characters of OCR multiple outputs because they are already aligned.

### 5.3 AWS Contributions

The contributions of the proposed alignment technique can be more clarified if there is a comparison between the proposed alignment technique and other existing techniques. Therefore, this research presents this comparison as shown in Table 5.1.

Table 5.1

*Comparison between AWS technique and other existing techniques*

Existing alignment techniques	Proposed technique
Alignment process is approximate (Al-Zaydi & Salam, 2015; Lund, 2014; Lund et al., 2013b; Lund et al., 2011; Pervez et al., 2014; Volk et al., 2011).	Alignment process is exact
It requires executing a character alignment algorithm between each pair of OCR outputs (Al-Zaydi & Salam, 2015; Lund, 2014; Lund et al., 2013b; Lund et al., 2011; Pervez et al., 2014; Volk et al., 2011). Problems resulted from character alignment algorithm are described in Section 2.3.1.2.	It does not require executing any character alignment algorithm.
Less accuracy because alignment process usually contains errors.	Better accuracy

## 5.4 Experimental results

This section describes the results of the experiment performed on proposed alignment technique (AWS). In this section, the proposed alignment technique AWS has been implemented. Furthermore, three related existing techniques have also been implemented to be used in the evaluation of AWS. They are the ProbCons alignment (PCA), which is used by Pervez et al. (2014), Smith–Waterman alignment (SWA), which is used by Al-Zaydi and Salam (2015), and Levenshtein distance with backtrack (LDB), which is used by Al Azawi (2015). As mentioned previously, this research used three metrics in the evaluation process. They are word error rate (WER), character error rate (CER), and non-word error rate (NWER). The design of evaluation process and the details of testing dataset were explained in Chapter 3.

### 5.4.1 Word Error Rate (WER)

Table 5.2 presents the experimental results of the AWS evaluation using the WER metric, while Figure 5.4 shows the clustered column graph for the WER values listed in this table.

Table 5.2

*Experimental results of the AWS evaluation using the WER metric*

	<b>PCA</b> (Pervez et al., 2014)	<b>SWA</b> (Al-Zaydi & Salam, 2015)	<b>LDB</b> (Al Azawi, 2015)	<b>Proposed alignment (AWS)</b>
Total words	39048	39048	39048	39048
Wrong words	18445	17938	17254	15185
WER	47.24%	45.94%	44.19%	38.89%

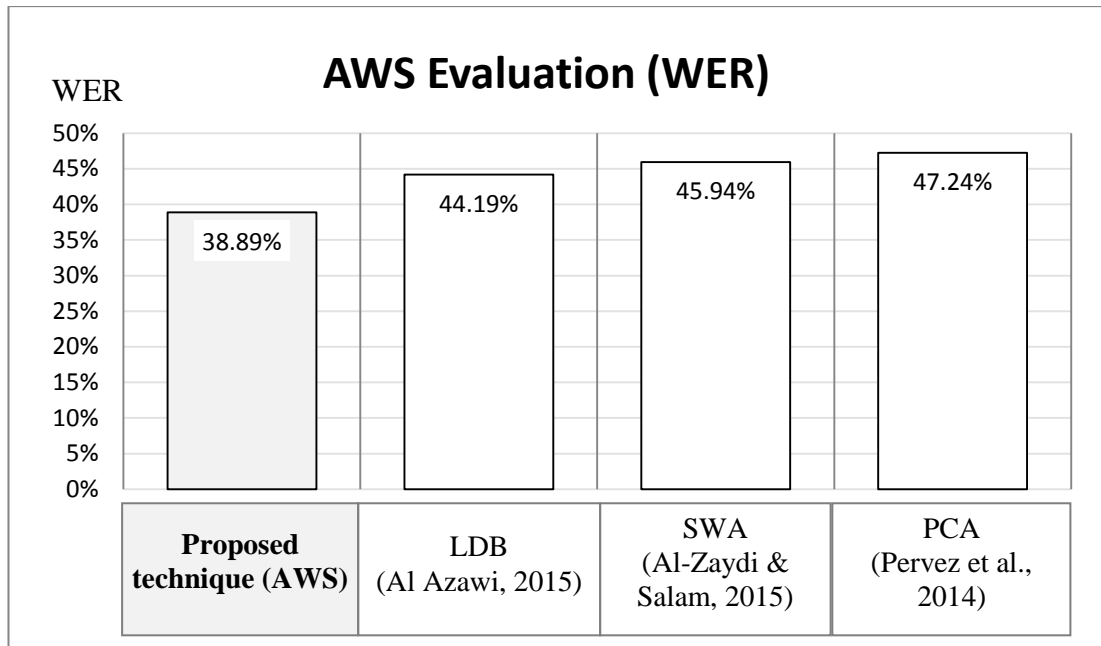


Figure 5.4. Clustered column graph for the WER values listed in Table 5.2.

From Table 5.2 and Figure 5.4, it can be clearly seen that the WER value for each alignment technique is different from the others. Overall, they show that PCA technique had the highest percentage value of WER with the rate of 47.24%. This is followed by SWA 45.94%, and LDB 44.19%. Furthermore, it can be seen that WER of the proposed technique AWS had the lowest percentage value of OCR error rate than the others with the rate of 38.89%. This technique has a 15% relative decrease on the mean WER of the three existing alignment techniques and 11.99% relative decrease on the best WER of them. This indicates that the proposed technique AWS had a reduction in the WER metric compared to the existing techniques.

As mentioned previously, this research performed a statistical method using an ANOVA-test to show if the reduction in WER of the AWS is significant or not.

Figure 5.5 shows ANOVA-test results using the WER values.



Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (AWS)	130	38.89	515.82
LDB (Al Azawi, 2015)	130	44.19	512.96
SWA (Al-Zaydi & Salam, 2015)	130	45.94	582.10
PCA (Pervez et al., 2014)	130	47.24	870.02

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	6911.94	3	2303.98	3.71	0.01	2.62
Within Groups	320035.94	516	620.22			
Total	326947.87	519				

Figure 5.5. ANOVA-test results for the WER values

Figure 5.5 shows that the number of tests for each technique is 130 as shown in the term “Count” in column 2. The input for each test is a single image, and the output is OCR error rate. Figure 5.5 also shows that the average of OCR error rate for AWS is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.01 as presented in column 6, which is less than 0.05, and the value of “F” is 3.71 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the OCR error rate among four techniques is real and not due to chance. Therefore, AWS is better than other in terms of the WER.

### 5.4.2 Character Error Rate (CER)

Table 5.3 displays the results of the four experiments in terms of the least CER value, while Figure 5.6 shows the clustered column graph for the CER values listed in this table.

Table 5.3

*Experimental results of the AWS evaluation using the CER metric*

	<b>PCA</b> (Pervez et al., 2014)	<b>SWA</b> (Al-Zaydi & Salam, 2015)	<b>LDB</b> (Al Azawi, 2015)	<b>Proposed alignment (AWS)</b>
Total characters	231896	231896	231896	231896
Wrong characters	57258	53039	51850	33415
CER	24.69%	22.87%	22.36%	14.41%

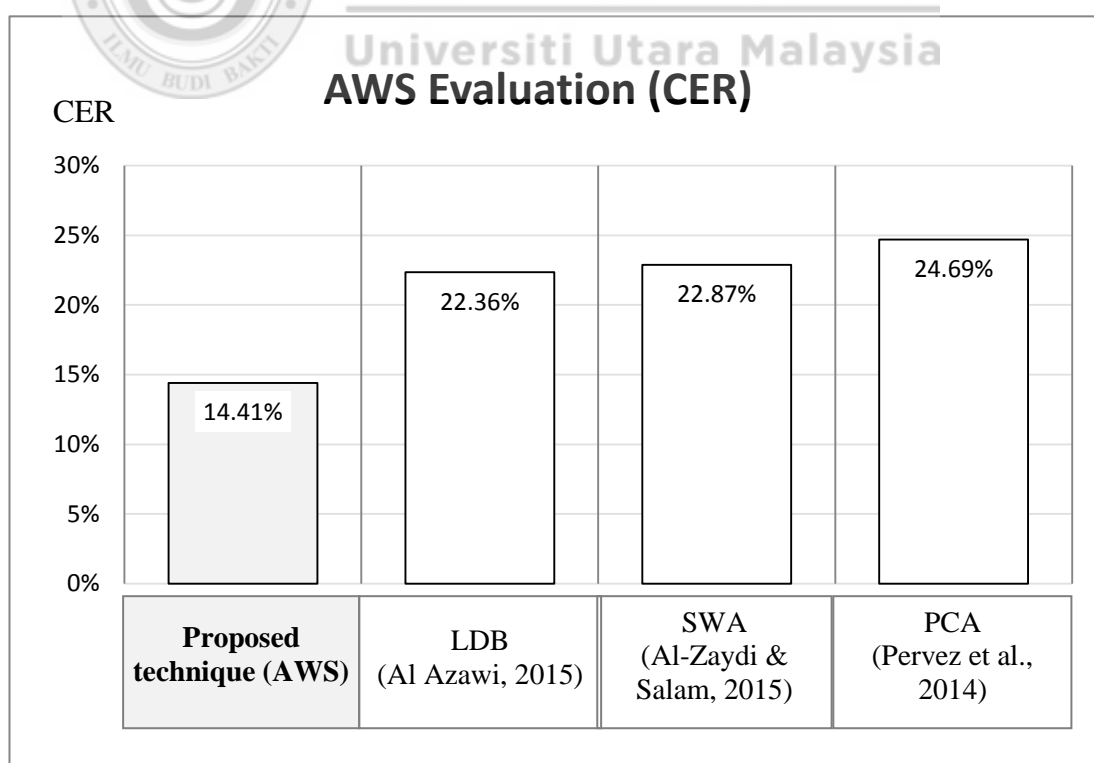


Figure 5.6. Clustered column graph for the CER values listed in Table 5.3.

As it can be seen in Table 5.3 and Figure 5.6, the alignment techniques of SWA and LDB show almost the same values of CER with rates of 22.87% and 22.36% respectively. Furthermore, the CER values for both SWA and LDB are less than CER value for PCA technique, which has a rate of 24.69%. The proposed alignment technique AWS outperformed the existing techniques in terms of CER with a rate of 14.41%. The proposed technique has a 38.07% relative decrease on the mean CER of the three existing alignment techniques and 35.55% relative decrease on the best CER of them. The previous values of CER show that the proposed technique AWS had the highest reduction percentage in the number of wrong characters compared to the PCA, LDB, and SWA techniques.

This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the CER of the AWS is significant or not. Figure 5.7 shows ANOVA-test results using the CER values. From Figure 5.7, it can be clearly seen that the average of CER for AWS is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 10.61 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the CER among four techniques is real and not due to chance. Therefore, AWS is better than other in terms of the CER.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (AWS)	130	14.41	171.66
LDB (Al Azawi, 2015)	130	22.36	239.14
SWA (Al-Zaydi & Salam, 2015)	130	22.87	240.52
PCA (Pervez et al., 2014)	130	24.69	349.33

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	7962.82	3	2654.27	10.61	0.00	2.62
Within Groups	129082.24	516	250.16			
Total	137045.06	519				

Figure 5.7. ANOVA-test results for the CER values

### 5.4.3 Non-Word Error Rate (NWER)

Table 5.4 presents the experimental results of the AWS evaluation using the NWER metric, while Figure 5.8 shows the clustered column graph for the NWER values listed in this table.

Table 5.4

*Experimental results of the AWS evaluation using the NWER metric*

	<b>PCA</b> (Pervez et al., 2014)	<b>SWA</b> (Al-Zaydi & Salam, 2015)	<b>LDB</b> (Al Azawi, 2015)	<b>Proposed alignment (AWS)</b>
Total words	39048	39048	39048	39048
Non-word errors	9414	8943	8917	6594
NWER	24.11%	22.90%	22.84%	16.89%

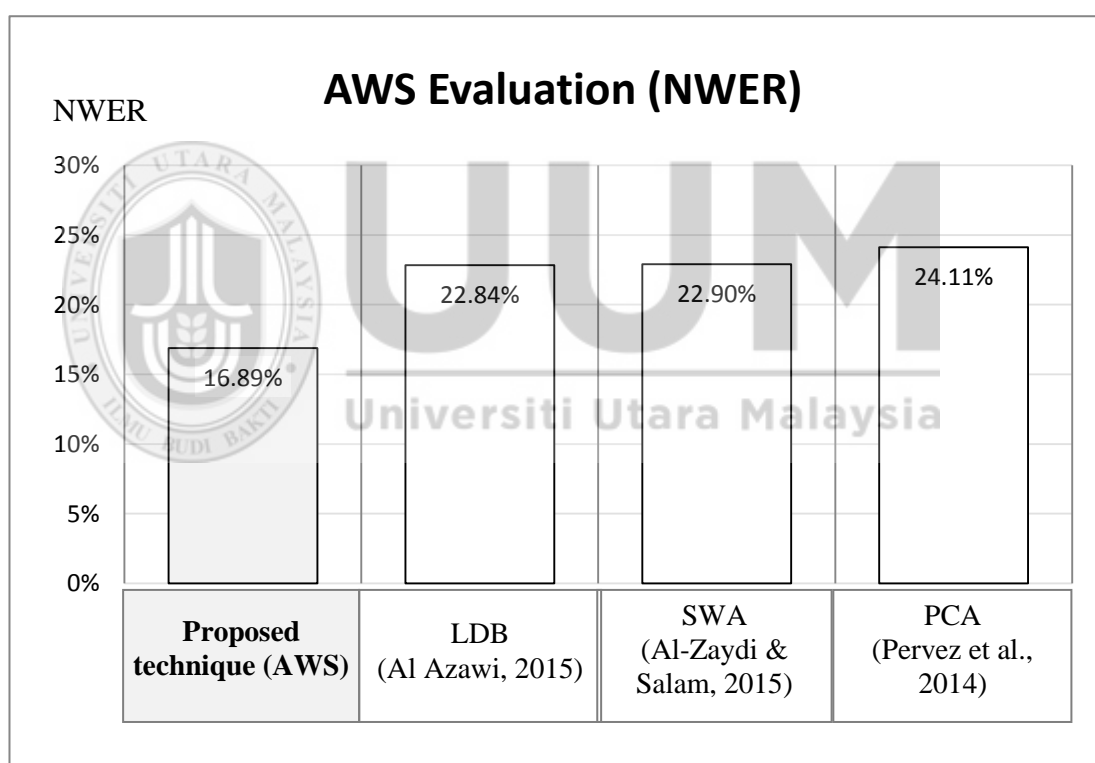


Figure 5.8. Clustered column graph for the NWER values listed in Table 5.4.

Table 5.4 and Figure 5.8 show that the worse performance was produced by PCA technique with the rate of 24.11%. The techniques SWA and LDB show similar performance to each other, which is better than PCA technique, with rates of 22.90% and 22.84% respectively. The proposed alignment technique AWS achieved the best

performance in terms of NWER value with a rate of 16.89%. The proposed alignment technique AWS has a 27.42% relative decrease on the mean NWER of the three existing alignment techniques and 26.05% relative decrease on the best NWER of them. This indicates that the proposed technique AWS is the best compared to the three existing alignment techniques in terms of NWER metric.

This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the NWER of the AWS is significant or not. Figure 5.9 shows ANOVA-test results using the NWER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (AWS)	130	16.89	207.48
LDB (Al Azawi, 2015)	130	22.84	176.28
SWA (Al-Zaydi & Salam, 2015)	130	22.90	182.62
PCA (Pervez et al., 2014)	130	24.11	245.04

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	5423.78	3	1807.93	8.91	0.00	2.62
Within Groups	104674.60	516	202.86			
Total	110098.38	519				

Figure 5.9. ANOVA-test results for the NWER values

From Figure 5.9, it can be clearly seen that the average of NWER for AWS is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as

presented in column 6, which is less than 0.05, and the value of “F” is 8.91 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the NWER among four techniques is real and not due to chance. Therefore, AWS is better than other in terms of the NWER.

#### **5.4.4 Results Discussion**

The experimental results of this study show that the worse performances in terms of WER, CER, and NWER values were achieved by PCA technique. Both SWA and LDB techniques are better than PCA technique in aligning multiple texts due to the difference in their work. SWA and LDB are based on dynamic programming approach in their work, while PCA is based on hidden Markov model (Pervez et al., 2014). Dynamic programming approach is better than hidden Markov model in the alignment of multiple texts. However, this approach is not suitable for aligning more than four outputs due to the huge main memory used and long processing time required, while hidden Markov model can deal with more than four outputs with fewer requirements (Just, 2001; Lopresti & Zhou, 1997; Pervez et al., 2014).

The proposed alignment technique AWS achieved the lowest values of WER, CER, and NWER compared to the others existing techniques. This is due to the effective design of proposed alignment technique. As mentioned in previous sections, this design does not be affected by any number of OCR outputs used. Furthermore, it also does not be affected by any number of characters in each OCR output. This is because existing alignment techniques deal with the output texts after OCR engines have produced them (Lund, 2014). This means the existing techniques do not address the origin of a problem but try to address its effect on OCR output texts. In contrast,

the proposed technique is better than other existing techniques because it handles the origin of the alignment problem as described in Section 5.1.

## **5.5 Summary**

The goal of this chapter is to improve OCR accuracy for the Arabic language by proposing a solution to the alignment problem. Therefore, in this chapter, the design details regarding the proposed alignment technique are presented and discussed. The details explained the concept of AWS, flowchart, and its contributions. In addition to that, this study conducted four experiments to evaluate this technique using three metrics. Furthermore, this research conducted a statistical test to show if the reduction in the OCR error rate is significant or not. The statistical test of this research has been measured using ANOVA-test. The results of the evaluation process are presented in detail. The experimental results are very encouraging. The proposed technique AWS outperforms other existing related techniques in terms of WER, CER, and NWER. Therefore, the practical results of this chapter indicate that the objective two of this study is achieved.



## CHAPTER SIX

### PROPOSED VOTING TECHNIQUE

#### 6.0 Introduction

As mentioned in Chapter 3, the proposed technique has been referred to as VCI by this research. The chapter begins by introducing the concept of the VCI with some examples. Furthermore, an algorithm to implement VCI is provided. The chapter continues with a comparison of the advantages and disadvantages of VCI and existing techniques are explained. The experimental results of the VCI were presented next, and the chapter ends with a summary.

#### 6.1 Voting Technique (VCI) Concept

The first step in designing the improved technique is to analysis the limitations of existing techniques. As mentioned before in Chapter 2, most existing voting techniques are based on two techniques: Majority alone or Majority & Lexicon. Both techniques do not perform well in some cases as explained in Section 2.3.1.3. This is because they do not give any attention to the context of a sentence around an incorrect word. The design of the enhanced voting technique is based on context information of a sentence around the incorrect word. Therefore, this research used N-gram language model to capture context information of sentences through training to benefit from it in the voting process.

Table 6.1 shows simple examples that explain why context information of a sentence around the incorrect word is important in the voting process. Note the example in Table 6.1 assumed that input image contains only one sentence, which is “*Swim \_\_\_\_\_ for long hair*”, and there are three OCR outputs to complete it.

Table 6.1

*Voting process example*

Original word	OCR outputs			Majority	Majority & Lexicon	Proposed technique (VCI)
	OCR-1	OCR-2	OCR-3			
cap	cap	cop	cop	cop	cop	cap
cap	cap	cop	cep	?	?	cap
cap	cap	cep	cep	cep	cap	cap
cap	cep	cep	cep	cep	?	cap
cap	cap	cap	cap	cap	cap	cap

Table 6.1 shows that in the first row, the existing Majority technique selects the word “cop” rather than “cap” because first-word “cop” exists in two from three OCR outputs. Of course, this is wrong because “cap” is suitable to complete the sentence, while “cop” is not. On the other hand, existing Majority & Lexicon technique selects also “cop” even if it is unsuitable for the sentence. The reason is that both “cap”, and “cop” are found in the lexicon and “cop” is more frequent than “cap”. In contrast, the proposed voting technique selects the word “cap” because the proposed technique gives attention to the context of a sentence around the incorrect word by using an N-gram language model.

In the second row of Table 6.1, the existing Majority technique cannot decide which word is suitable for the sentence because there is no majority among three OCR outputs. On the other hand, the existing Majority & Lexicon technique cannot also decide which word is suitable for the sentence because of both “cap”, and “cop” are found in the lexicon and there is no majority among them. In third row of Table 6.1,

the existing Majority technique selects the wrong word “cep” rather than “cap” because it exists in two from three OCR outputs while the existing Majority & Lexicon technique can select the correct word “cap” because it is only one from three OCR outputs that exist in lexicon. In fourth row, the existing Majority technique selects also the wrong word “cep” because it exists in all OCR outputs while existing Majority & Lexicon technique cannot decide which word is suitable for the sentence because after generating candidates list for the wrong word “cep” there is no majority between candidates list (Al Azawi, 2015; Batawi & Abulnaja, 2012; Lund, 2014). From Table 6.1, it can clearly be seen that proposed voting technique can handle more cases of OCR outputs text.

## 6.2 VCI Algorithm

Algorithm 6.1 below represents the pseudo code for proposed voting technique (VCI).

<b>Algorithm 6.1: Proposed voting technique</b>	
<i>S1</i>	Let $z1[0..k]$ is array of words for first OCR output
<i>S2</i>	Let $z2[0..k]$ is array of words for second OCR output
<i>S3</i>	Let $z3[0..k]$ is array of words for third OCR output
<i>S4</i>	Let $i=0$ // counter for words in the arrays $z1$ , $z2$ , and $z3$
<i>S5</i>	Select $z1[i]$ , $z2[i]$ , and $z3[i]$
<i>S6</i>	Select unique words from ( $z1[i]$ , $z2[i]$ , and $z3[i]$ )
<i>S7</i>	If only one word from unique words belongs to a unigram then: send this word to the output text and go to <i>S10</i>
<i>S8</i>	If more than one word from unique words belongs to the unigram then: begin select the word from unique words based on N-gram language model, go to <i>S10</i> end if

<i>S9</i>	If no word from unique words belongs to the unigram then: begin generate candidates list to the ( $z1[i]$ , $z2[i]$ , and $z3[i]$ ) based on N-gram language model, select the candidate word based on Levenshtein algorithm end if
<i>S10</i>	if $i=k$ then go to <i>S11</i> else $i=i+1$ and go to <i>S5</i>
<i>S11</i>	End

Algorithm 6.1 shows that the proposed voting technique receives several arrays of words. Each array represents the words of a single OCR output. After that, each word from each array will pass to the other operations in voting technique in sequence (*S5*), and unique words are selected from all OCR outputs (*S6*). Next, each word will be checked if it belongs to the unigram (*S7*). This checking will cause three cases. The first case will only occur when a single correct word belonging to the unigram resulted from OCR engines (*S7*) while the second case will occur when more than one correct word belonging to the unigram (*S8*). The last case occurred when no correct word belonging to the unigram (*S9*). Note existing technique uses a lexicon in checking step. As mentioned previously, unigram is better than the lexicon because unigram contains all words of the lexicon and it contains also additional words representing most frequent words of specific language such as names, new words, etc (Bassil & Alwani, 2012c).

In case one (*S7*), if there is just one word belongs to the unigram, then it is marked as correct, and it is sent to the output text. Otherwise, in case two (*S8*), if more than one word belonging to the unigram, then the best word will be selected based on the probability of an N-gram language model, which is used context information of a

sentence in the voting process. In existing technique, the best word will be selected based on shared characters among OCR outputs without giving any attention to the context information. In case three (S9), the candidate list is generated using a language model. If the language model fails, then the Levenshtein algorithm will be used to generate candidates list. In existing technique, the candidate list is generated using Levenshtein algorithm. As mentioned in chapter 2, N-gram language model produces candidate list based on context information while Levenshtein algorithm does not. According to the Naseem (2004), context-based correction is better than isolated word correction. After generating candidates list, the last step is to choose any word from suggestions' list having the least edit distance to the incorrect word using Levenshtein distance. The resulting words will be used to build a final OCR output text.

### **6.3 VCI Contributions**

The contributions of the proposed voting technique can be more clarified if there is a comparison between the proposed voting technique and other existing techniques. Therefore, this study presents this comparison as shown in Table 6.2 (Al Azawi, 2015; Batawi & Abulnaja, 2012; Lund, 2014).

Table 6.2

*Comparison between VCI technique and other existing techniques*

<b>Cases of OCR outputs</b>	<b>Majority</b>	<b>Majority &amp; Lexicon</b>	<b>Proposed technique</b>
All outputs (words) are wrong	Fail (majority between wrong words) (Batawi & Abulnaja, 2012)	Fail if there are two or more candidate words have the same edit distance to the incorrect word (Lund, 2014)	Can handle this case
One output (word) is correct and other outputs are wrong.	Fail (single correct word does not have the majority) (Al Azawi, 2015)	Can handle this case	Can handle this case
More than one output (word) is correct and other outputs are wrong.	Fail if a majority of a wrong word is greater than majority of correct word (Lopresti & Zhou, 1997)	Fail if the majority of each correct word is equal (Al-Zaydi & Salam, 2015)	Can handle this case
Accuracy	Low accuracy because it cannot handle most OCR outputs cases	Medium accuracy because it can handle some OCR outputs cases	Better accuracy because it can handle most OCR outputs cases

## 6.4 Experimental results

This section describes the results of the experiment performed on the proposed voting technique (VCI). In this section, the proposed voting technique VCI has been implemented. Furthermore, three related existing techniques have also been implemented to be used in the evaluation of VCI. They are the Majority (Al Azawi, 2015), Lattice features (Lund, 2014), and Unified technique (Al-Zaydi & Salam, 2015). As mentioned previously, this study used three metrics in the evaluation process. They are word error rate (WER), character error rate (CER), and non-word error rate (NWER). The design of evaluation process and the details of testing dataset were explained in Chapter 3.

### 6.4.1 Word Error Rate (WER)

Table 6.3 presents the experimental results of the VCI evaluation using the WER metric, while Figure 6.1 shows the clustered column graph for the WER values listed in this table. Note the gray column in Table 6.6 represents the results of the proposed voting technique (VCI).

Table 6.3

*Experimental results of the VCI evaluation using the WER metric*

	<b>Lattice features (LF)</b> (Lund, 2014)	<b>Majority (MT)</b> (Al Azawi, 2015)	<b>Unified (UT)</b> (Al-Zaydi & Salam, 2015)	<b>Proposed technique (VCI)</b>
Total words	39048	39048	39048	39048
Wrong words	14946	16334	15460	12785
WER	38.28%	41.83%	39.59%	32.74%

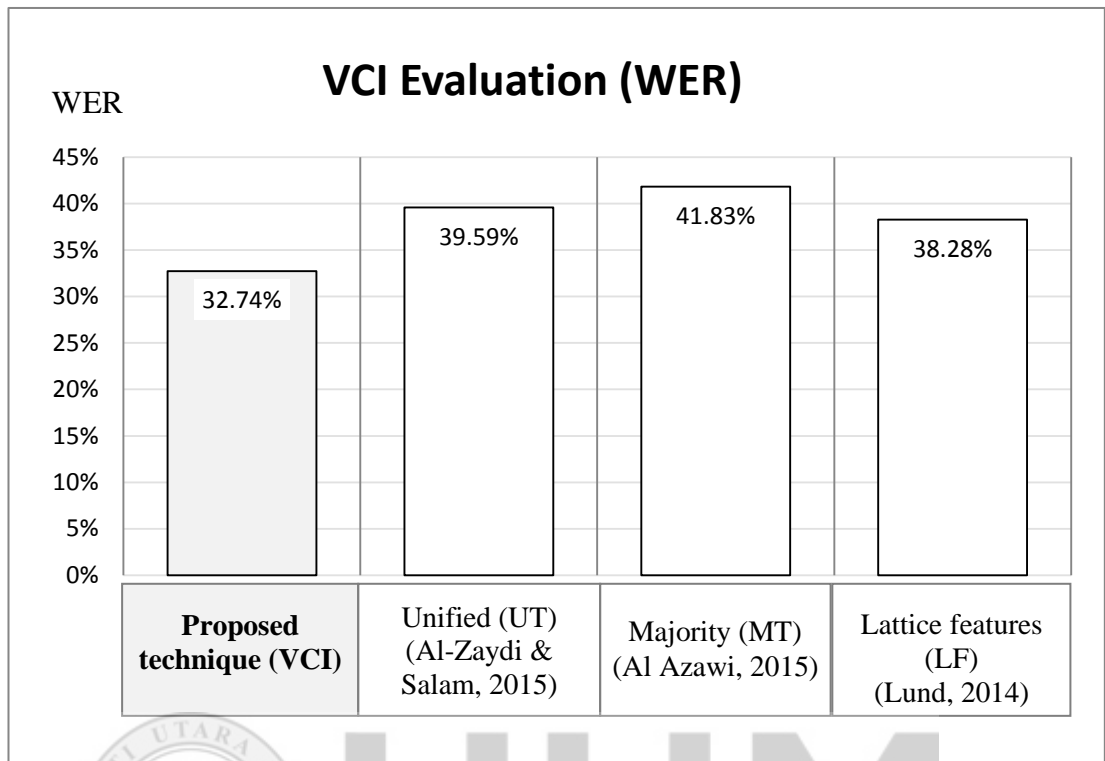


Figure 6.1. Clustered column graph for the WER values listed in Table 6.3.

From Table 6.3 and Figure 6.1, it can be clearly seen that the WER value for each voting technique is different from the others. Overall, they show that the MT technique had the highest percentage value of WER with the rate of 41.83%. This is followed by UT 39.59%, and LF 38.28%. Furthermore, it can be seen that WER of the proposed technique VCI had the lowest percentage value of OCR error rate than the others with the rate of 32.74%. This technique has a 17.83% relative decrease on the mean WER of the three existing voting techniques and 14.46% relative decrease on the best WER of them. This indicates that the proposed technique VCI had a significant reduction in the WER metric compared to the existing techniques. As mentioned previously, this research performed a statistical method using an ANOVA-test to show if the reduction in WER of the VCI is significant or not. Figure 6.2 shows ANOVA-test results using the WER values.



Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (VCI)	130	32.74	465.50
Unified (Al-Zaydi & Salam, 2015)	130	39.59	512.27
Majority (Al Azawi, 2015)	130	41.83	520.43
Lattice features (Lund, 2014)	130	38.28	531.11

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	6460.75	3	2153.58	4.24	0.01	2.62
Within Groups	261780.55	516	507.33			
Total	268241.29	519				

Figure 6.2. ANOVA-test results for the WER values

Figure 6.2 shows that the number of tests for each technique is 130 as shown in the term “Count” in column 2. The input for each test is a single image, and the output is OCR error rate. Figure 6.2 also shows that the average of OCR error rate for VCI is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.01 as presented in column 6, which is less than 0.05, and the value of “F” is 4.24 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the OCR error rate among four techniques is real and not due to chance. Therefore, VCI is better than other techniques in terms of the WER.

### 6.4.2 Character Error Rate (CER)

Table 6.4 displays the results of the four experiments in terms of the least CER value, while Figure 6.3 shows the clustered column graph for the CER values listed in this table.

Table 6.4

*Experimental results of the VCI evaluation using the CER metric*

	<b>Lattice features (LF)</b> (Lund, 2014)	<b>Majority (MT)</b> (Al Azawi, 2015)	<b>Unified (UT)</b> (Al-Zaydi & Salam, 2015)	<b>Proposed technique (VCI)</b>
Total characters	231896	231896	231896	231896
Wrong characters	35475	38015	33956	29916
CER	15.30%	16.39%	14.64%	12.90%

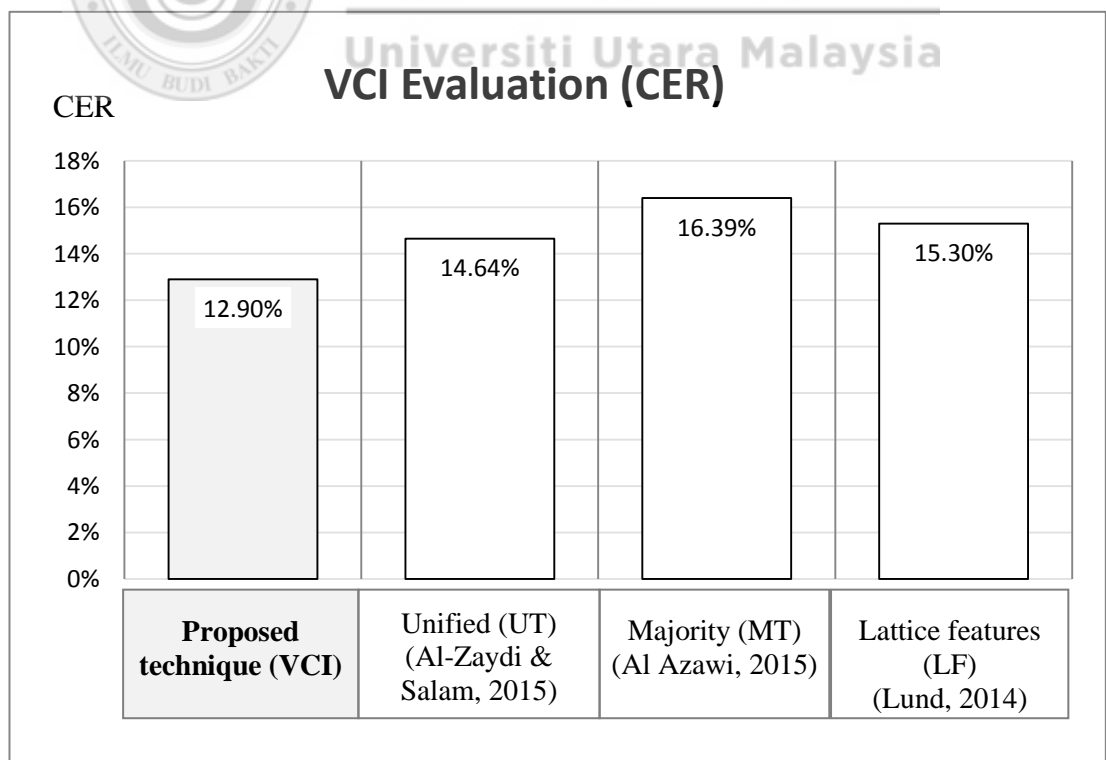


Figure 6.3. Clustered column graph for the CER values listed in Table 6.4.

As it can be seen in Table 6.4 and Figure 6.3, the voting techniques of UT, LF and MT show a slightly difference in the values of the CER with rates of 14.64%, 15.30%, and 16.39% respectively. The proposed voting technique VCI outperformed the existing techniques in terms of CER with the rate of 12.90%. The proposed technique has a 16.29% relative decrease on the mean CER of the three existing voting techniques and 11.90% relative decrease on the best CER of them. The previous values of CER show that the proposed technique VCI had the highest reduction percentage in the number of wrong characters compared to the UT, LF, and MT. This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the CER of the VCI is significant or not. Figure 6.4 shows ANOVA-test results using the CER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (VCI)	130	12.90	164.06
Unified (Al-Zaydi & Salam, 2015)	130	14.64	170.01
Majority (Al Azawi, 2015)	130	16.39	158.18
Lattice features (Lund, 2014)	130	15.30	166.20

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	925.18	3	308.39	3.91	0.01	2.62
Within Groups	84940.04	516	164.61			
Total	85865.22	519				

Figure 6.4. ANOVA-test results for the CER values

From Figure 6.4, it can be clearly seen that the average of CER for VCI is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.01 as presented in column 6, which is less than 0.05, and the value of “F” is 3.91 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the CER among four techniques is real and not due to chance. Therefore, VCI is better than other in terms of the CER.

### 6.4.3 Non-Word Error Rate (NWER)

Table 6.5 presents the experimental results of the VCI evaluation using the NWER metric, while Figure 6.5 shows the clustered column graph for the NWER values listed in this table.

Table 6.5

*Experimental results of the VCI evaluation using the NWER metric*

	<b>Lattice features (LF)</b> (Lund, 2014)	<b>Majority (MT)</b> (Al Azawi, 2015)	<b>Unified (UT)</b> (Al-Zaydi & Salam, 2015)	<b>Proposed technique (VCI)</b>
Total words	39048	39048	39048	39048
Non-word errors	8586	10533	6680	6218
NWER	21.99%	26.97%	17.11%	15.92%

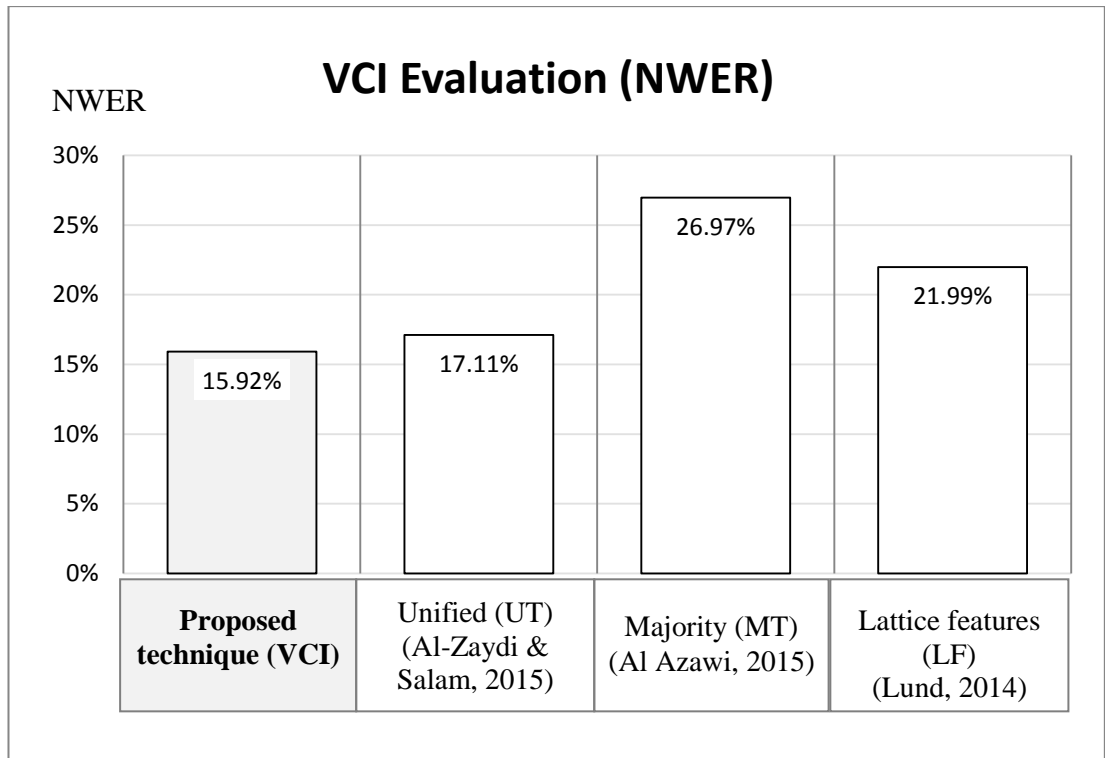


Figure 6.5. Clustered column graph for the NWER values listed in Table 6.5.

Table 6.5 and Figure 6.5 show that the worse performance was produced by MT with the rate of 26.97%. The techniques of LF and UT show performances better than MT, with rates of 21.99% and 17.11% respectively. The proposed voting technique VCI achieved the best performance in terms of NWER value with a rate of 15.92%. The proposed voting technique VCI has a 25.15% relative decrease on the mean NWER of the three existing voting techniques and 6.92% relative decrease on the best NWER of them. This indicates that the proposed technique VCI is the best compared to the three existing voting techniques in terms of NWER metric.

This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the NWER of the VCI is significant or not. Figure 6.6 shows ANOVA-test results using the NWER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed technique (VCI)	130	15.92	168.47
Unified (Al-Zaydi & Salam, 2015)	130	17.11	200.58
Majority (Al Azawi, 2015)	130	26.97	209.77
Lattice features (Lund, 2014)	130	21.99	188.01

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	9359.47	3	3119.82	16.27	0.00	2.62
Within Groups	98919.53	516	191.70			
Total	108279.00	519				

Figure 6.6. ANOVA-test results for the NWER values

From Figure 6.6, it can be clearly seen that the average of NWER for VCI is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 16.27 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the NWER among four techniques is real and not due to chance. Therefore, VCI is better than other in terms of the NWER

#### 6.4.4 Results Discussion

The experimental results of this chapter show that the worse performances in terms of WER, CER, and NWER values were achieved by MT technique. This is because this technique fails in three cases in the voting process (Al Azawi, 2015; Lund,

2014). The first is when all outputs (words) are wrong. Therefore, the majority will be between wrong words. The second is when one output (word) is correct and other outputs are wrong. Therefore, the single correct word does not have the majority. The last is when more than one output (word) is correct and other outputs are wrong. In this case, it will fail if the majority of wrong words is greater than of correct words.

The techniques of LT and UT are better than MT because they can handle more cases in the voting process (Al-Zaydi & Salam, 2015; Al Azawi, 2015; Lund, 2014). For example, when one output (word) is correct and other outputs are wrong. Therefore, the single correct word will be selected by these techniques as the best correction. However, they also fail in some cases: such as if there are two or more correct words as candidates to the incorrect word and their majorities are equal. Lastly, the proposed technique VCI achieved the lowest values of WER, CER, and NWER compared to the others existing techniques because it gives high attention to the context of a sentence around the incorrect word. This leads to handle more cases in the voting process than what others existing techniques can do as explained in Section 6.1.

## **6.5 Summary**

The goal of this chapter is to improve OCR accuracy for the Arabic language by proposing a solution to the limitations of the voting process. Therefore, in this chapter, the design details regarding the proposed voting technique are presented and discussed. The details explained the concept of VCI, its flowchart, and its contributions. In addition to that, this study conducted four experiments to evaluate this technique using three metrics. Furthermore, this research conducted a statistical

test to show if the reduction in the OCR error rate is significant or not. The statistical test of this research has been measured using ANOVA-test. The results of the evaluation process are presented in detail. The experimental results are very encouraging. The proposed technique VCI outperforms other existing related techniques in terms of WER, CER, and NWER. Therefore, the practical results of this chapter indicate that the objective three of this study is achieved.





## **CHAPTER SEVEN**

### **PROPOSED HYBRID MODEL**

#### **7.0 Introduction**

In this chapter, the design detail and experimental results of the proposed hybrid model are presented. As mentioned in Chapter 3, the proposed model has been referred to as HMNL by this research. The chapter begins by introducing the design of the HMNL. Furthermore, a diagram of HMNL is provided to show how the HMNL works. The chapter continues with a discussion on the interaction between components and proposed techniques in the hybrid model. The solutions for Arabic challenges that are related to the OCR post-processing are explained. The experimental results of the HMNL were presented next, and the chapter ends with a summary.

#### **7.1 Interaction in the Hybrid Model (HMNL)**

The diagram of the hybrid model is shown in Figure 7.1. The hybrid model consists of three stages. The first stage is to describe the major steps used in building the hybrid model as mentioned in Chapter 3. The second stage is to explain the proposed techniques that are included in this model as mentioned in Chapters 4, 5, and 6. The last stage is to discuss the interaction between the components and proposed techniques in the hybrid model.

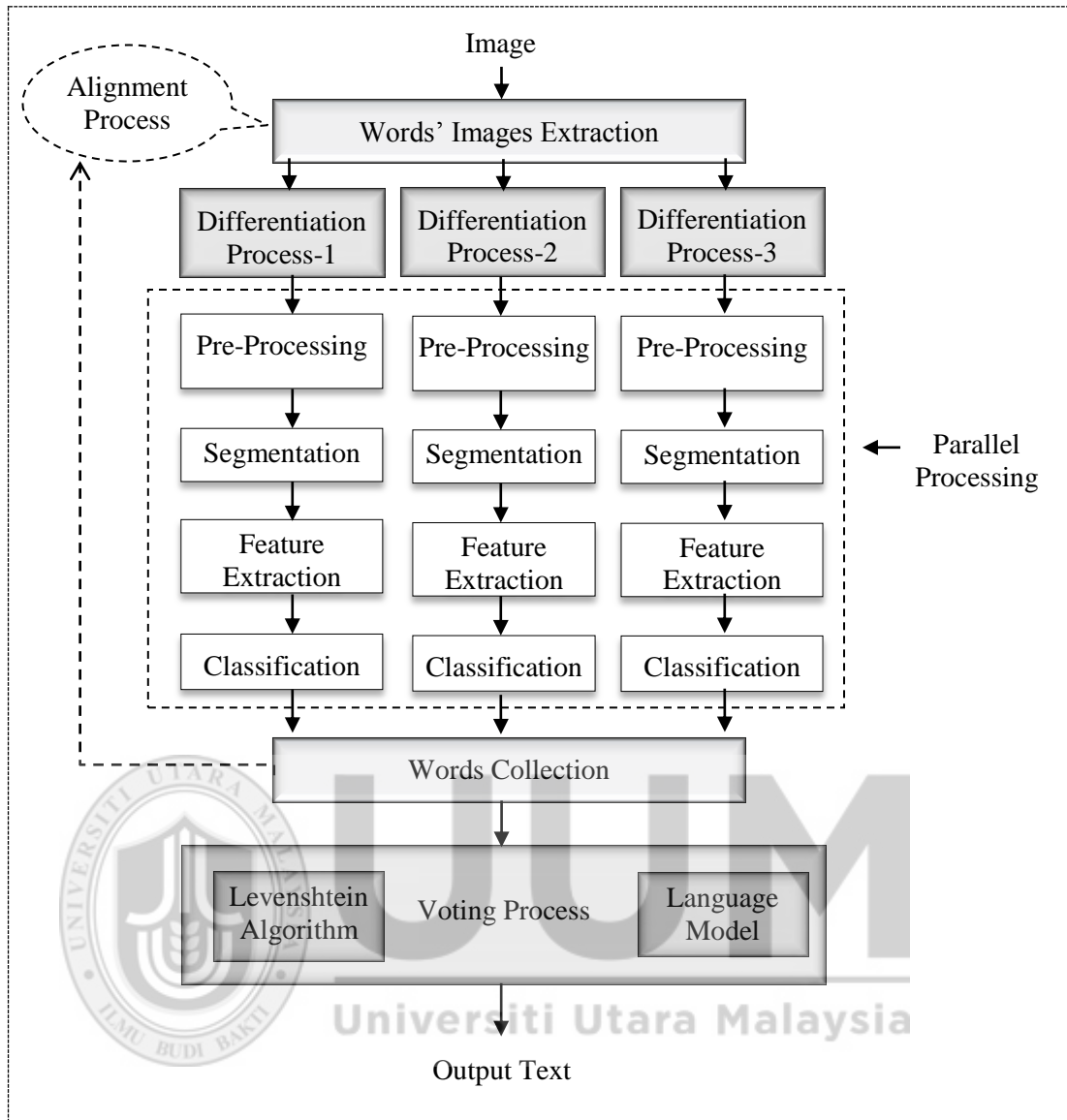


Figure 7.1. The Proposed hybrid model

In “Words’ Images Extraction”, the input image is searched thoroughly to locate words and denote the words as blocks. Thus, each block will contain a word image. At the end, words of an input image are extracted and stored in an array of words’ images. The reason for extracting only words’ images is that, if the input image is passed to the multiple OCR engines directly, then alignment problem of output texts will occur. The proposed alignment techniques remove this type of alignment as described previously in Chapter 4. The output of “Words’ Images Extraction” is an array of words’ images.

Next, each word's image is passed in sequence to the differentiation process. Here, in the proposed differentiation process, some pixels' values of word image will be changed. This means differentiation process will produce three words' images as described previously in Chapter 4. Note proposed differentiation technique could generate several words' images for the same word. However, this research uses only three OCR outputs to reduce complexity. After that, each word's image is passed through one OCR engine to turn into a word, so that the results are three words.

In "Words Collection", each sequence of words resulting from each OCR engine is combined in a single array so that three arrays will be obtained. Previous steps are performed in a multi-threads manner, (in parallel) in order to reduce processing time (Akhter & Roberts, 2006). Furthermore, OCR engines in the hybrid model are not different, but they are multiple copies of the same OCR engine. Lastly, the voting process of the hybrid model receives three arrays of words. It uses an N-gram language model and Levenshtein distance to select the best words between multiple outputs of OCR.

In voting process, if the differences between multiple outputs of OCR are not enough to choose the best word among them, then the proposed voting technique of this study will depend on an N-gram language model to find the best suggestion. Finally, if N-gram language model failed to choose the best among the candidates' list, then the proposed voting technique will depend on Levenshtein distance to find the best solution. The detail of the proposed voting technique performed in the voting process is described in details in Chapter 6.

## **7.2 Arabic Challenges**

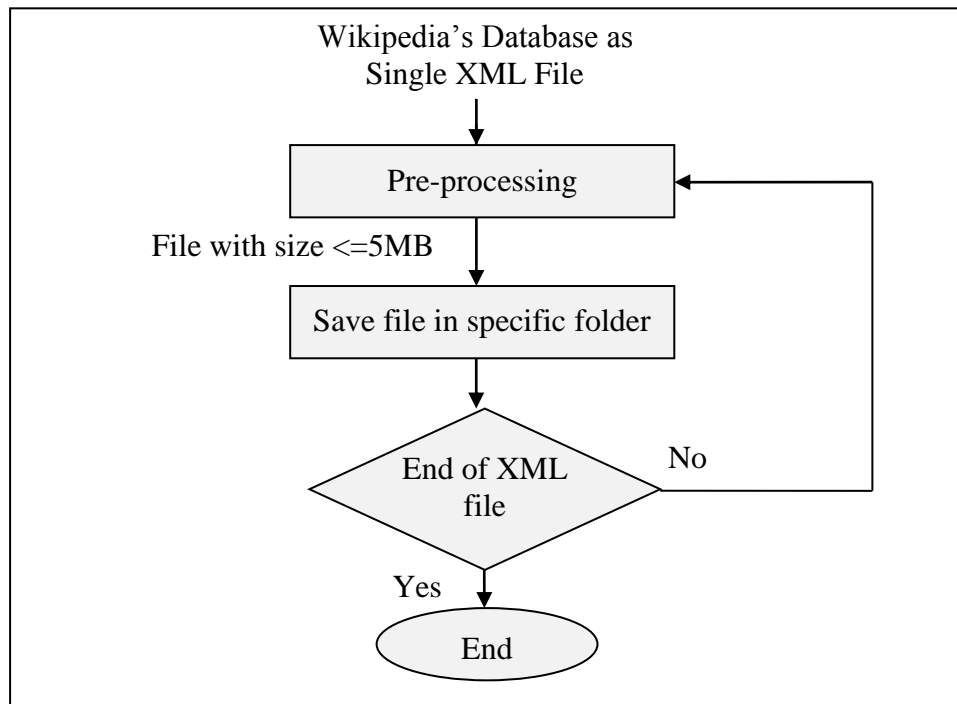
This section proposes the solutions for two Arabic challenges: lack of available Arabic web corpus and diacritics in the Arabic words. These challenges are related to the OCR post-processing stage as described in Chapter 2. Other Arabic challenges and characteristics mentioned in Chapter 2 are related to other OCR stages, which are out of the scope of this research. As mentioned in Chapter 2, lack of available Arabic web corpus is related to the N-gram language model structure, while diacritics are related to all OCR post-processing techniques. Therefore, this section contains two subsections. The first is related to the N-gram language model and the second is related to the diacritics.

### **7.2.1 N-gram Language Model Challenges**

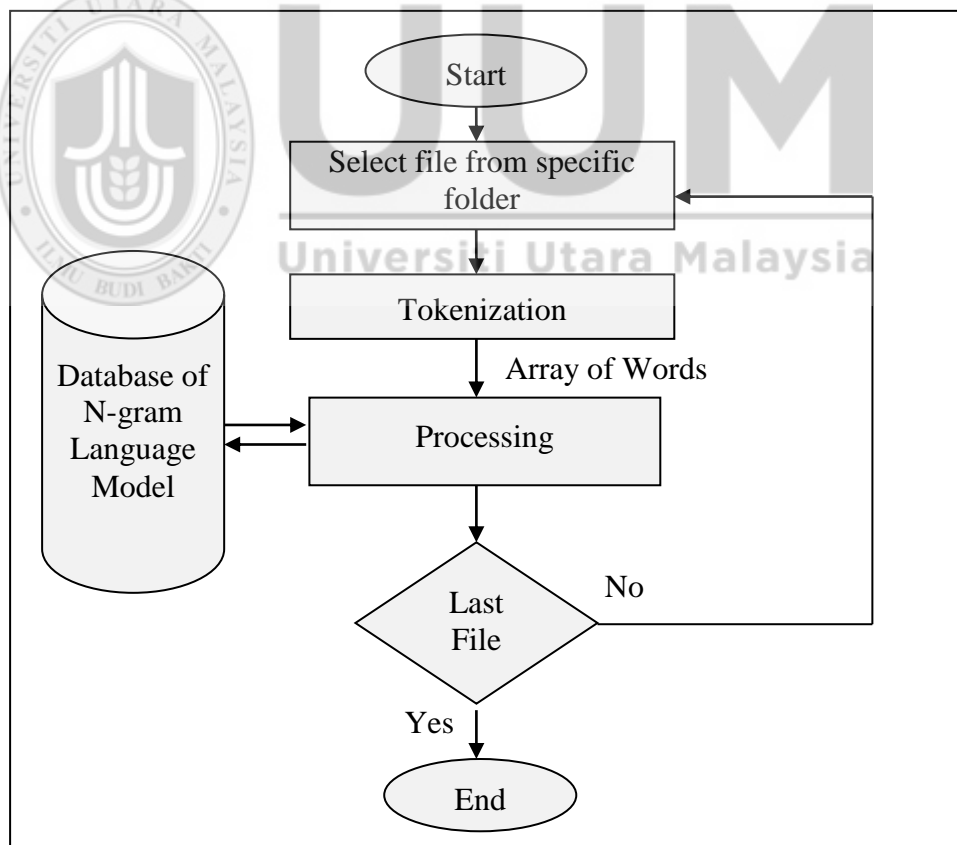
It is difficult to design a large Arabic N-gram language model for three reasons mentioned in Chapter 2. Firstly, the Arabic language has a lack of available web corpus. Secondly, the similarity between the words of Arabic is high. High similarity of Arabic valid words requires a greater amount of corpus to create an accurate language model (Shaalán et al., 2012; Zribi & Ahmed, 2003). Lastly, large N-gram gives better accuracy than short N-gram. However, seeing of short N-gram in the corpus is easier than seeing large N-gram in the same corpus. Based on previous reasons, this study does not design large N-gram. However, it designs and uses unigram, bigram, and trigram models together. The data source for building these models is Wikipedia database. This database is chosen for several reasons mentioned in Chapter 3.

As previously mentioned, English language and many other languages that use Latin characters, do not suffer from the major problem when extracting text from Wikipedia's database because there are many programs that can be used to extract text directly from a Wikipedia dump file. However, these programs are designed to eliminate any non-Latin letter (Vrandečić, Sorg, & Studer, 2011), and therefore, cannot be used to extract Arabic texts. The use of an indirect way in the extraction of Arabic texts by downloading pages and articles from Wikipedia's site and then extracting text from them takes a long time (Alkhalifa & Rodríguez, 2009). For previous reasons, an extraction method is developed that can extract only Arabic text from a Wikipedia's database. Figure 7.2 shows the proposed method.

In this method, the reading of Wikipedia XML file has been performed in parts because it is very large. In the preprocessing stage, addresses of images, navigation, and layout are ignored and only text from Wikipedia XML file is extracted. Then each text with size less or equal to 5MB is stored as a separate plain text file. The text of Wikipedia XML file is split into small pieces for two reasons. Firstly, a large file is not recommended to put in the main memory at once because this may cause hanging or stopping a computer.



A: Split Wikipedia xml database into small files.



B: Fill database of N-gram language model.

Figure 7.2. Extract Arabic text from Wikipedia database

The second reason is that, there are thousands of operations that are required to take place on each file and these processes are time-consuming. Therefore, when a file is a large and the process has an error, the work will be repeated from scratch, and this takes additional time. However, in the case of a small file, the program will have to re-process only the specific file and thus, takes a shorter time. At the end, all files are placed in a specific folder by the program automatically.

In the tokenization stage, the program extracts only text "T" from an input file and then split the text "T" into an array of words "W" using a space as a divider. Next, an array of words "W" is passed to the processing stage. In the processing stage, each unigram, bigram, or trigram has been stored in the database of a language model. On the other hand, there is a variable named "*frequency*" associated with each unigram, bigram, and trigram in the database. These variables save frequencies of them, where each occurrence of them in the text leads to increase the variable "*frequency*" by one. Frequencies of unigram, bigram, and trigram are stored within the database because they are used to calculate the probability for candidates list when it is needed.

Other operations in processing stage are replacing all the dates, times, numbers by special tokens as shown in Table 7.1. The words that contain numbers or special characters are ignored. If the middle word of tri-gram is ignored, then the whole trigram is ignored, and the same operation is applied for unigram and bigram. Figure 7.2 also shows that all unigrams, bigrams, and trigrams with their frequencies are stored in an N-gram language model database.

Table 7.1

*Special tokens in the classification stage*

<b>Tokens</b>	<b>Examples</b>	<b>Replaced By</b>
Date	11/12/1982	</date/>
Time	11:12:02	</time/>
Number	1212	</number/>

On the other hand, the goal of this study is not to find the optimized database structure of an N-gram language model. Therefore, this study does not design a complex structure for a database, but it designs a simple structure that can satisfy the objectives of this research. The Arabic documents of Wikipedia contain huge numbers of words. Therefore, it is not efficient to load them all in the main memories of computers. For this reason, the database structure of an N-gram language model is organized as shown in Figure 7.3.

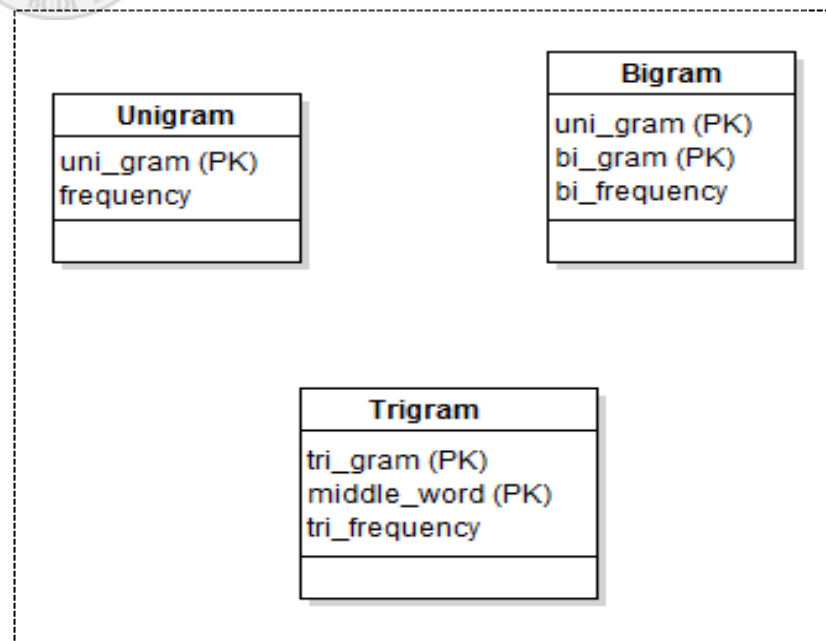
*Figure 7.3. Database structure of N-gram language model*



Figure 7.3 shows three tables in the database of an N-gram language model. The first table named “Unigram” contains all unigrams with their frequencies. The second table named “Bigram” containing three columns. Each cell in the first column contains the first word from bigram, while each cell in the second column contains the second word from bigram. The last column in table “Bigram” contains a frequency of each bigram.

The third table named “Trigram” also contains three columns. Each cell in the first column contains the first and the third word from any trigram, while each cell in the second column contains the second word from trigram. The last column in table “Trigram” contains a frequency of each trigram. As a simple example of how to store data in the database of an N-gram language model, the sentences “*Student plays football*” and “*Student plays tennis*” are stored in the database as shown in Table 7.2, Table 7.3, and Table 7.4.

Table 7.2

*Example of how to store sentences in Unigram table*

Uni_gram	Frequency
Student	2
plays	2
football	1
tennis	1

Table 7.3

*Example of how to store sentences in Bigram table*

Uni_gram	Bi_gram	Bi_frequency
Student	plays	2
plays	football	1
plays	tennis	1

Table 7.4

*Example of how to store sentences in Trigram table*

Tri_gram	Middle_word	Tri_frequency
Student football	plays	1
Student tennis	plays	1

This structure of database does not store any bigram or any trigram in a single column. This will avoid using the condition named “like %word%” in the SQL statement named “select-from-where”. The condition “like” takes a long time in retrieving any information from a database, while equal condition takes less time in execution. The type and size of columns of tables in the N-gram language model database are shown in Table 7.5.

Table 7.5

*Type and size of columns of tables in N-gram language model database*

Table Name	Column Name	Type and Size
Unigram	uni_gram	nvarchar(20)
	frequency	integer
Bigram	uni_gram	nvarchar(20)
	bi_gram	nvarchar(20)
	bi_frequency	integer

Trigram	tri_gram	nvarchar(41)
	middle_word	nvarchar(20)
	tri_frequency	integer

Finally, a comparison between the proposed corpus of this study and two related existing Arabic corpora is shown in Table 7.6. They are the NEMLAR Arabic written corpus (El-Mahallawy, 2008; Yaseen et al., 2006) and the KACST Arabic corpus (Al-Thubaity, 2015). Table 7.6 shows that the proposed corpus outperformed the existing corpora in terms of text size with a value of 2.2G. Furthermore, it also outperformed them in terms of the domain with a value of 25. The term “*domain*” refers to the number of text categories that use in producing the corpus, such as news, sports, sciences, etc.

Table 7.6  
*Comparison between three Arabic corpora*

Arabic Corpus Name	Size	Unique words	Domain	Update	Availability
NEMLAR Arabic written corpus (El-Mahallawy, 2008)	500K	Not mentioned	13	Static	Not free to download
KACST Arabic corpus (Al-Thubaity, 2015)	731M	Not mentioned	11	Not mentioned	Free to download
Proposed Corpus	2.2G	1,260,617	25	Each 15 days	Free to download

### 7.2.2 Diacritics

As mentioned in Chapter 2, existing OCR post-processing techniques suffer from diacritics when used for Arabic language (Muaz, 2011). Diacritics are one of the Arabic characteristics, which are located above or below the letters. Most researchers ignore the diacritics through OCR post-processing stage (Al-Masoudi & Al-Obeidi, 2015; Al-Zaydi & Salam, 2015; Bassil & Alwani, 2012c; Lund, 2014) and this will reduce accuracy if they are existing in the output text. Other researchers try to restore them. However, most techniques cannot reach 100% accuracy (Hadj Ameer, Moulahoum, & Guessoum, 2015; Shahrour, Khalifa, & Habash, 2015). Therefore, the voting process, Levenshtein distance, and candidates' list generation will be affected if there is an error in diacritization restoration.

The solution of this study to this problem is based on using the filtering technique. This technique will filter Arabic words before saving them in the database of a language model. Furthermore, they will also be filtered before processing them by Levenshtein distance, candidates' list generation, and voting process. Filtering process means removing diacritics from any word that has them. There are two reasons to do this filtering. The first reason is that if the diacritics are written correctly or not written, the word still remains the same (Hadj Ameer et al., 2015). However, if they are written wrongly in a word, then this word is considered wrong.

The second reason is that Arabic texts are generally written without diacritics. This is the case for newspapers, books, etc (Hadj Ameer et al., 2015). Therefore, it is better to remove them instead of restoring them because the process of removing them cannot produce errors, while the process of restoring them can produce errors (Hadj Ameer et al., 2015; Shahrour et al., 2015). In this study, the filtering of the diacritics

is only performed on Levenshtein distance, candidates' list generation, and voting process, and if the user requires diacritics, then it can perform one of the diacritization restoration techniques on final OCR output using one of the practical applications to achieve the user's desire.

### **7.3 Experimental results**

This section describes the results of the experiment performed on the proposed hybrid model (HMNL). In this section, the proposed HMNL model has been implemented. Furthermore, three related existing OCR post-processing works have also been implemented to be used in the evaluation of HMNL. These works are proposed by Al Azawi (2015), Lund (2014), and Al-Zaydi and Salam (2015). As mentioned previously, this study used three metrics in the evaluation process. They are word error rate (WER), character error rate (CER), and non-word error rate (NWER). The design of evaluation process and the details of testing dataset were explained in Chapter 3.

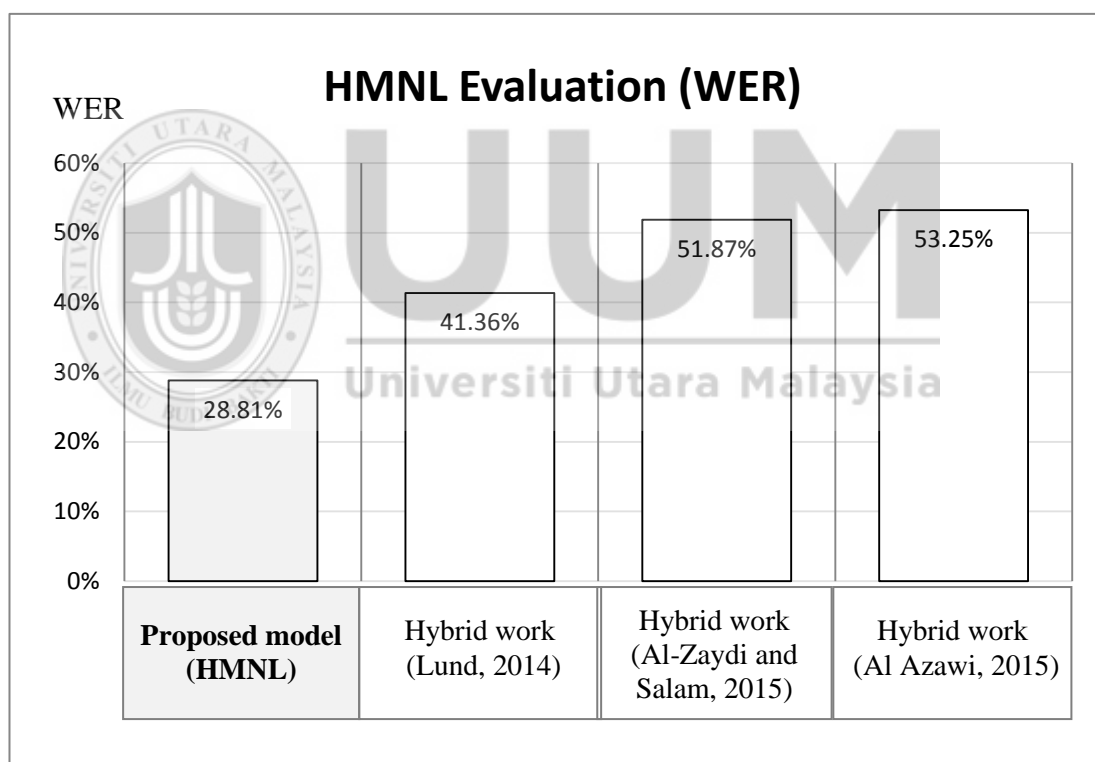
#### **7.3.1 Word Error Rate (WER)**

Table 7.7 presents the experimental results of the HMNL evaluation using the WER metric, while Figure 7.4 shows the clustered column graph for the WER values listed in this table.

Table 7.7

*Experimental results of the HMNL evaluation using the WER metric*

	Hybrid work (Al Azawi, 2015)	Hybrid work (Al-Zaydi and Salam, 2015)	Hybrid work (Lund, 2014)	Proposed Model (HMNL)
Total words	39048	39048	39048	39048
Wrong words	14946	16334	15460	12785
WER	53.25%	51.87%	41.36%	28.81%



*Figure 7.4. Clustered column graph for the WER values listed in Table 7.7.*

From Table 7.7 and Figure 7.4, it can be clearly seen that the WER values are different from one to another. Overall, they show that the work proposed by Al Azawi (2015) had the highest percentage value of WER with the rate of 53.25%. This is followed by Al-Zaydi and Salam (2015) 51.87%, and Lund (2014) 41.36%.

Furthermore, it can be seen that WER of the proposed hybrid model HMNL had the lowest percentage value of OCR error rate than the others with the rate of 28.81%. The proposed hybrid model HMNL has a 40.24% relative decrease on the mean WER of the three existing works and 30.35% relative decrease on the best WER of them. This indicates that the proposed hybrid model HMNL had a significant reduction in the WER metric compared to the existing works. As mentioned previously, this research performed a statistical method using an ANOVA-test to show if the reduction in WER of the HMNL is significant or not. Figure 7.5 shows ANOVA-test results using the WER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed model (HMNL)	130	28.81	490.72
(Lund, 2014)	130	41.36	598.93
Al-Zaydi and Salam, 2015)	130	51.87	477.76
(Al Azawi, 2015)	130	53.25	421.67

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	51245.85	3	17081.95	34.35	0.00	2.62
Within Groups	256590.11	516	497.27			
Total	307835.96	519				

Figure 7.5. ANOVA-test results for the WER values

Figure 7.5 shows that the number of tests for each hybrid work is 130 as shown in the term “Count” in column 2. The input for each test is a single image, and the output is OCR error rate. Figure 7.5 also shows that the average of OCR error rate for HMNL is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 34.35 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the OCR error rate among four works is real and not due to chance. Therefore, HMNL is better than other in terms of the WER.

### 7.3.2 Character Error Rate (CER)

Table 7.8 displays the results of the four experiments in terms of the least CER value, while Figure 7.6 shows the clustered column graph for the CER values listed in this table.

Table 7.8

*Experimental results of the HMNL evaluation using the CER metric*

	Hybrid work (Al Azawi, 2015)	Hybrid work (Al-Zaydi and Salam, 2015)	Hybrid work (Lund, 2014)	Proposed Model (HMNL)
Total characters	231896	231896	231896	231896
Wrong characters	69070	64206	52179	24828
CER	29.78%	27.69%	22.50%	10.71%



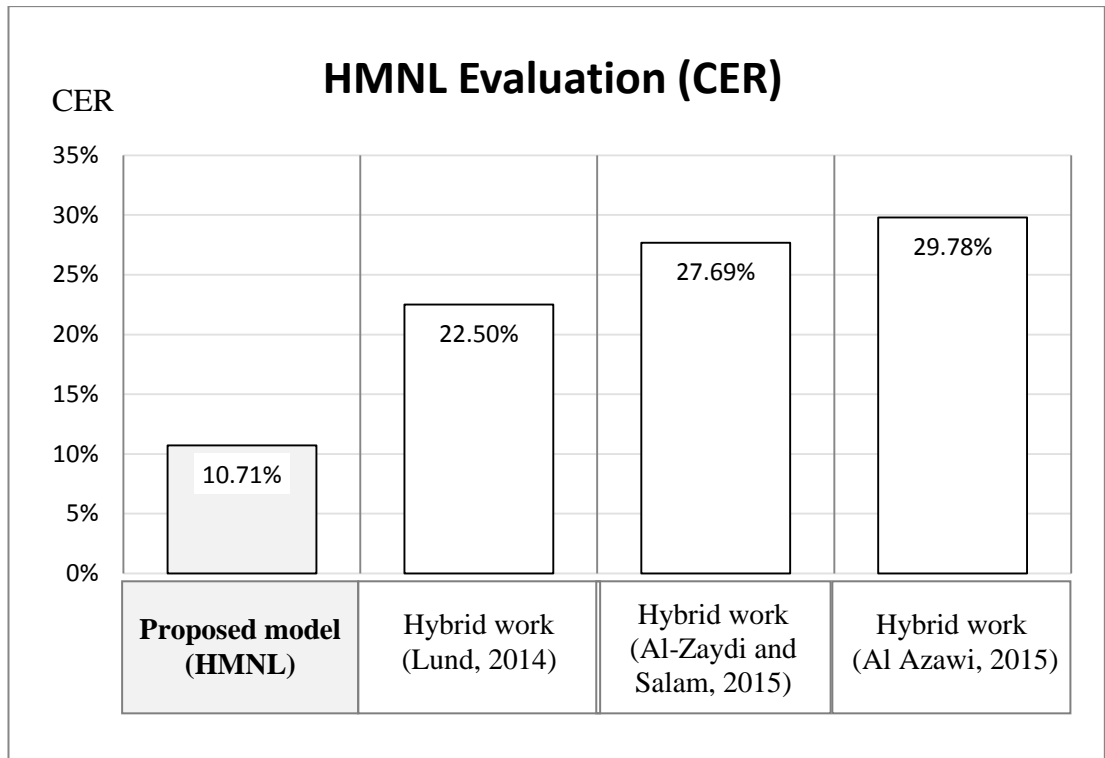


Figure 7.6. Clustered column graph for the CER values listed in Table 7.8.

As it can be seen in Table 7.8 and Figure 7.6, the existing works proposed by Al Azawi (2015) and Al-Zaydi and Salam (2015) show slightly a difference in values of CER with rates of 29.78% and 27.69% respectively. Furthermore, the CER values for them are higher than CER value for the work proposed by Lund (2014), which has a rate of 22.50%. The proposed hybrid model HMNL outperformed the existing works in terms of CER with the rate of 10.71%. The proposed hybrid model HMNL has a 59.27% relative decrease on the mean CER of the three existing works and 52.42% relative decrease on the best CER of them. The previous values of CER show that the proposed hybrid model HMNL had the highest percentage decrease in the number of wrong characters compared to the existing works.

This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the CER of the HMNL is significant or not. Figure 7.7 shows ANOVA-test results using the CER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed model (HMNL)	130	10.71	159.91
(Lund, 2014)	130	22.50	395.19
Al-Zaydi and Salam, 2015)	130	27.69	432.16
(Al Azawi, 2015)	130	29.78	224.02

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	30272.07	3	10090.69	33.32	0.00	2.62
Within Groups	156254.33	516	302.82			
Total	186526.40	519				

Figure 7.7. ANOVA-test results for the CER values

From Figure 7.7, it can be clearly seen that the average of CER for HMNL is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 33.32 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the CER among four hybrid works is real and not due to chance. Therefore, HMNL is better than other in terms of the CER.

### 7.3.3 Non-Word Error Rate (NWER)

Table 7.9 presents the experimental results of the HMNL evaluation using the NWER metric, while Figure 7.8 shows the clustered column graph for the NWER values listed in this table.

Table 7.9

*Experimental results of the HMNL evaluation using the NWER metric*

	Hybrid work (Al Azawi, 2015)	Hybrid work (Al-Zaydi and Salam, 2015)	Hybrid work (Lund, 2014)	Proposed Model (HMNL)
Total words	39048	39048	39048	39048
Non-word errors	13356	10606	10579	5516
NWER	34.20%	27.16%	27.09%	14.13%

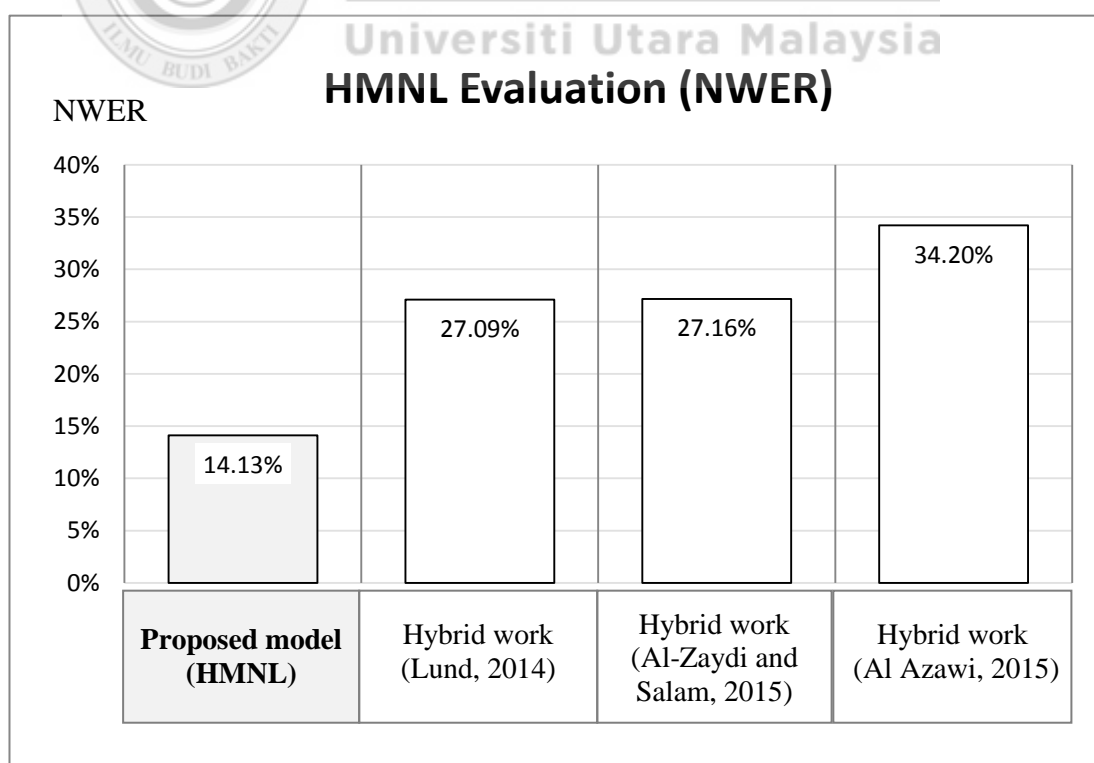


Figure 7.8. Clustered column graph for the NWER values listed in Table 7.9.

Table 7.9 and Figure 7.8 show that the worse performance was produced by the work proposed by Al Azawi (2015) with the rate of 34.20%. The works proposed by Lund (2014) and Al-Zaydi and Salam (2015) show performances better than first work, with rates of 27.09% and 27.16% respectively. The proposed hybrid model HMNL achieved the best performance in terms of NWER value with a rate of 14.13%. The proposed hybrid model HMNL has a 51.52% relative decrease on the mean NWER of the three existing works and 47.86% relative decrease on the best NWER of them. This indicates that the proposed hybrid model HMNL is the best compared to the three existing works in terms of NWER metric. This research has also been performed a statistical method using an ANOVA-test to show if the reduction in the NWER of the HMNL is significant or not. Figure 7.9 shows ANOVA-test results using the NWER values.

Anova: Single Factor

SUMMARY

Groups	Count	Average	Variance
Proposed model (HMNL)	130	14.13	171.80
(Lund, 2014)	130	27.09	326.81
Al-Zaydi and Salam, 2015)	130	27.16	137.99
(Al Azawi, 2015)	130	34.20	241.45

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	27807.58	3	9269.19	42.23	0.00	2.62
Within Groups	113268.27	516	219.51			
Total	141075.85	519				

Figure 7.9. ANOVA-test results for the NWER values

From Figure 7.9, it can be clearly seen that the average of NWER for HMNL is less than others as is clear in column 3. Furthermore, it shows that the P-value is 0.0 as presented in column 6, which is less than 0.05, and the value of “F” is 42.23 as presented in column 5, which is larger than the value of “F-crit” (2.62) in column 7. This indicates that the difference in the means of the NWER among four works is real and not due to chance. Therefore, HMNL is better than other in terms of the NWER.

#### **7.3.4 Results Discussion**

The experimental results of this chapter show that the worse performances in terms of WER, CER, and NWER values were achieved by the work proposed by Al Azawi (2015). This is because this work is based on the Majority technique in the voting process. As mentioned in Chapter 6, the Majority technique fails in some cases in the voting process (Al Azawi, 2015; Lund, 2014). The experimental results also show that the works proposed by Lund (2014) and Al-Zaydi and Salam (2015) are better than previous work because they depend on both lexicon and majority in the voting process.

The proposed hybrid model HMNL achieved the lowest values of WER, CER, and NWER compared to the others existing works because it gives high attention to the context of a sentence around the incorrect word. This leads to handle more cases in the voting process than what others existing works can do. Furthermore, the hybrid model combined the MO of the OCR, N-gram language model, and the Levenshtein algorithm to benefit from their strengths. In addition to that, each technique used in a hybrid model is improved by proposing solutions for its limitations and to make it suit Arabic challenges as explained in Chapters 4, 5, and 6.

## 7.4 Summary

This chapter is a complement to the Chapters: 4, 5, and 6 to achieve the main goal, which is to improve OCR accuracy for the Arabic language. Therefore, in this chapter, the design details regarding the proposed hybrid model are presented and discussed. Furthermore, this chapter presents solutions for Arabic challenges that are related to the OCR post-processing. In addition to that, this study conducted four experiments to evaluate this model using three metrics. Furthermore, this research conducted a statistical test to show if the reduction in the OCR error rate is significant or not. The statistical test of this research has been measured using ANOVA-test. The results of the evaluation process are presented in detail. They confirm that hybridizing OCR post-processing techniques are very useful and efficient. The proposed hybrid model outperforms other existing OCR post-processing works in terms of WER, CER, and NWER. Therefore, the practical results of this chapter indicate that the objective four of this study is achieved.

## **CHAPTER EIGHT**

### **CONCLUSION**

#### **8.0 Introduction**

This chapter included the conclusion and a brief recommendation of this study. It reviews the overall progress of the study and giving a full view based on objectives of the research. In addition to that, this chapter also contains the limitations and the directions of the future work.

#### **8.1 Achievement**

The main purpose of this study is to design and develop a hybrid model for OCR post-processing techniques to improve characters recognition for the Arabic language. The first step in designing this model is to study the strengths and weaknesses of existing techniques used in solving a research problem (refer to Section 2.3). This is followed by identifying the best techniques among them based on their strengths and weaknesses (refer to Section 2.4). After that, the hybrid model is designed to combine the selected techniques to benefits from their strength and to overcome their limitations (refer to Section 2.3.4 and Chapter 7).

On the other hand, some techniques used in the hybrid model are improved by proposing solutions for their limitations, and to make them suit Arabic challenges (refer to Section 3.3 and Chapters 4, 5, 6, and 7). Experiments were conducted to illustrate how the proposed hybrid model can be employed to yield a promising result. The evaluation process includes comparing the proposed hybrid model with three existing hybrid works using three metrics (refer to Section 3.5.3).

The first objective of this study has been achieved by designing and evaluating the enhanced differentiation technique. The algorithm and experimental results of this technique are shown in Section 4.2 and Section 4.3 respectively. The second objective has been achieved by designing and evaluating the proposed alignment technique. The algorithm and experimental results of this technique are shown in Section 5.2 and Section 5.3 respectively. The designing and evaluating of the enhanced voting technique is the third objective of this study. The algorithm and experimental results of this technique are shown in Section 6.3 and Section 6.2 respectively. The last objective has been achieved with the designing and evaluating of the hybrid model of OCR post-processing techniques (refer to Chapters 7). The proposed hybrid model has significantly reduced the error rate compared to the existing works in this area (refer to the experimental results of Chapter 7).

## 8.2 Research Contributions

This section summarizes the major contributions of this research. The contributions are:

- i. Differentiation technique:** It has three advantages as explained in the following sentences. The first is that it does not require connecting multiple OCR software, such as the works of Volk et al. (2011), Lund et al. (2013b), Lund (2014), and Al Azawi (2015). Their works are considered as a difficult process (Al-Zaydi & Salam, 2015). In contrast, the enhanced differentiation technique requires only connecting three copies of the same OCR engine. The second is that it does not require combining four different classifiers such as the work of Kittler et al. (1998), that reduces the performance of the best classifier (Lund, 2014). The last



is that the enhanced differentiation technique produces better OCR accuracy compared to the existing techniques.

**ii. Alignment technique:** It has three advantages as explained in the following sentences. The first is that the alignment process is exact by the proposed technique while it is approximate by the existing techniques. The second is that the proposed alignment technique produces better OCR accuracy compared to the existing techniques. The last is that the proposed alignment technique does not require executing any character alignment algorithm that requires high computer resources as explained in Section 1.1.

**iii. Voting technique:** It has three advantages as explained in the following sentences. The first is that the enhanced voting technique produces better OCR accuracy compared to the existing techniques. The second is that the voting technique can correct real word errors, while existing techniques cannot. This is because they do not give any attention to the context of a sentence around the incorrect word. The last is that the proposed voting technique can handle most cases of resulting words of OCR outputs while existing techniques fail in some of these cases as explained in Chapter 6.

**iv. Hybrid model of the OCR post-processing techniques:** The proposed model has been successfully crafted to include various components that collaborate with each other to achieve the goal of reducing OCR error rate.

### 8.3 Research Limitations

Limitations are the conditions and shortcomings that cannot be controlled by the researcher, and this research has no exceptions.

- i. For Arabic OCR evaluation purpose, there is no large standard testing dataset that is available to download (Ahmad et al., 2016; Batawi & Abulnaja, 2012). Therefore, this research followed the same scenario used by (Al-Masoudi and Al-Obeidi (2015); Al-Zaydi and Salam (2015); Batawi and Abulnaja (2012); El-Mahallawy (2008)) to create the testing dataset.
- ii. The codes of some existing OCR techniques are available to download from formal websites, such as Microsoft, Google, and universities websites, while the codes of others do not. Hence, the codes of some existing OCR techniques that are not available to download have been implemented using the descriptions on them in literature.

### 8.4 Future Work

Although the experimental results are favorable there are some directions on which further research should focus. One of the directions of development, the hybrid model still generates errors for the Arabic language even if images are noise-free and have high scanning resolutions. Therefore, it is possible to improve the error rate through further research and development. This can be done by improving other stages of OCR for the Arabic language, such as pre-processing, segmentation, feature extraction, and classification.

The second direction is to test this model on low-resolution images. These images can be extracted from a sequence of low-quality video (Ma & Agam, 2012, 2013) or

when dealing with available images having a low resolution like what is available in thousands of documents' images on the Internet. The results of testing low-resolution images by proposed model can decide if it needs to be modified or not.

The third direction is to test the proposed hybrid model on noisy images. Different types of noises can result from old papers, low-quality printing. Therefore, differentiation process should take into account designing different techniques programmatically for handling various types of noises. This will lead modifying the hybrid model to improve the accuracy of these types of images.

Finally, this model cannot handle some characteristics of the Arabic language. This is because these characteristics are not related to the OCR post-processing stage, but it related to other stages. Therefore, it can modify techniques of pre-processing, segmentation, future extraction, and classification to suit the Arabic language effectively. For example, in pre-processing stage, it should modify its techniques so that they can distinguish between dots and diacritics of Arabic from noises of images. Another example, as was discussed previously, in segmentation stage, Arabic has cursive and overlapping between neighbor characters. Therefore, the techniques of segmentation stage should be improved to address these problems.

## **8.5 Summary**

As a conclusion for this research, this chapter has discussed and concluded the research summary, the contributions and the limitations of the research. At the end of the discussion, the recommendations for further research were outlined. Overall, this research provided a hybrid model to improve the accuracy of optical character

recognition for the Arabic language. Furthermore, it offers three techniques for improving components of this model, and to make it suits Arabic challenges.



## REFERENCES

- AbdelRaouf, A., Higgins, C. A., Pridmore, T., & Khalil, M. (2010). Building a multi-modal Arabic corpus (MMAC). *International Journal on Document Analysis and Recognition (IJDAR)*, 13(4), 285-302.
- Abdulkader, A. E., & Casey, M. R. (2015). Efficient identification and correction of optical character recognition errors through learning in a multi-engine environment: Google Patents.
- Abulnaja, O. A., & Batawi, Y. A. (2012). Improving Arabic Optical Character Recognition Accuracy Using N-Version Programming Technique. *Canadian Journal on Image Processing and Computer Vision*, 3(2), 44-46.
- Ahmad, I., Mahmoud, S. A., & Fink, G. A. (2016). Open-vocabulary recognition of machine-printed Arabic text using hidden Markov models. *Pattern recognition*, 51, 97-111.
- Akhter, S., & Roberts, J. (2006). *Multi-core programming* (Vol. 33): Intel press Hillsboro.
- Akila, G., El-Menisy, M., Khaled, O., Sharaf, N., Tarhony, N., & Abdennadher, S. (2015). Kalema: Digitizing Arabic Content for Accessibility Purposes Using Crowdsourcing. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing* (Vol. 9042, pp. 655-662): Springer International Publishing.
- Al-Badr, B., & Mahmoud, S. A. (1995). Survey and bibliography of Arabic optical text recognition. *Signal processing*, 41(1), 49-77.
- Al-Masoudi, A. F. R., & Al-Obeidi, H. S. R. (2015). Smoothing Techniques Evaluation of N-gram Language Model for Arabic OCR Post-processing. *Journal of Theoretical and Applied Information Technology*, 82(3), 432-439.
- AL-Shatnawi, A. M., AL-Salaimeh, S., AL-Zawaideh, F. H., & Omar, K. (2011). Offline arabic text recognition—an overview. *World of Computer Science and Information Technology Journal (WCSIT)*, 1(5), 184-192.
- Al-Thubaity, A. O. (2015). A 700M+ Arabic corpus: KACST Arabic corpus design and construction. *Language Resources and Evaluation*, 49(3), 721-751. doi: 10.1007/s10579-014-9284-1
- Al-Zaydi, Z. Q., & Salam, H. (2015). Multiple Outputs Techniques Evaluation for Arabic Character Recognition. *International Journal of Computer Techniques (IJCT)*, 2(5), 1-7.
- Al Azawi, M. (2015). *Statistical Language Modeling for Historical Documents using Weighted Finite-State Transducers and Long Short-Term Memory*. (PhD dissertation), Technical University of Kaiserslautern, Kaiserslautern, Germany.

- Al Azawi, M., & Breuel, T. M. (2014). *Context-dependent confusions rules for building error model using weighted finite state transducers for OCR post-processing*. Paper presented at the Proceeding of the 11th IAPR International Workshop on Document Analysis Systems (DAS) Loire Valley, France.
- Alex, B., Grover, C., Klein, E., & Tobin, R. (2012). *Digitised Historical Text: Does it have to be mediOCRe?* Paper presented at the Proceeding of the 11th Conference on Natural Language Processing (KONVENS), Vienna, Austria.
- Aljarrah, I., Al-Khaleel, O., Mhaidat, K., Alrefai, M. a., Alzu'bi, A., & Rabab'ah, M. (2012). Automated System for Arabic Optical Character Recognition with Lookup Dictionary. *Journal of Emerging Technologies in Web Intelligence*, 4(4), 362-370.
- Alkhalifa, M., & Rodríguez, H. (2009). *Automatically extending NE coverage of Arabic WordNet using Wikipedia*. Paper presented at the Proceeding of the 3rd International Conference on Arabic Language Processing (CITALA2009), Rabat, Morocco.
- Alobaedy, M. M. T. (2015). *Hybrid Ant Colony System Algorithm For Static And Dynamic Job Scheduling In Grid Computing*. (PhD thesis), Universiti Utara Malaysia, Kedah, Malaysia.
- Andoni, A., & Krauthgamer, R. (2012). The smoothed complexity of edit distance. *ACM Transactions on Algorithms (TALG)*, 8(4), 44.
- Attia, M., Rashwan, M., & Khallaaf, G. (2002). *On stochastic models, statistical disambiguation, and applications on Arabic NLP problems*. Paper presented at the Proceedings of the 3rd Conference on Software Language Engineering (CLE'2002), Cairo, Egypt.
- Attia, M., Toral, A., Tounsi, L., Monachini, M., & van Genabith, J. (2010). *An automatically built Named Entity lexicon for Arabic*. Paper presented at the Proceedings of the 7th conference on International Language Resources and Evaluation (LREC 2010) Valletta, Malta.
- Attia, M. E. (2000). *A large-scale computational processor of the Arabic morphology*. (Master thesis), Cairo University, Cairo, Egypt.
- Badawi, E.-S. M. (1996). *Understanding Arabic: essays in contemporary Arabic linguistics in honor of El-Said Badawi*: American Univ in Cairo Press.
- Bard, G. V. (2007). *Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric*. Paper presented at the Proceedings of the fifth Australasian symposium on ACSW frontiers, Darlinghurst, Australia.
- Barnes, D. N. (2011). *The Text Contains its Own Lexicon: Extracting a Spelling Reference in the Presence of OCR Errors*. (Master dissertation), The Open University, Milton Keynes, United Kingdom.

- Bassil, Y., & Alwani, M. (2012a). Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information. *Computer and Information Science*, 5(3), 37-48.
- Bassil, Y., & Alwani, M. (2012b). Ocr context-sensitive error correction based on google web 1t 5-gram data set. *arXiv preprint arXiv:1204.0188*.
- Bassil, Y., & Alwani, M. (2012c). Ocr post-processing error correction algorithm using google online spelling suggestion. *Journal of Emerging Trends in Computing and Information Sciences*, 3(1), 90-99.
- Batawi, Y., & Abulnaja, O. (2012). Accuracy Evaluation of Arabic Optical Character Recognition Voting Technique: Experimental Study. *IJECS: International Journal of Electrical & Computer Sciences*, 12(1), 29-33.
- Boyell, R. L., & Ruston, H. (1963). *Hybrid techniques for real-time radar simulation*. Paper presented at the Proceedings of the November 12-14, 1963, fall joint computer conference (AFIPS '71), Las Vegas, USA.
- Cai, X. (2013). *Approximate Sequence Alignment*. (Master thesis), Louisiana State University, Louisiana, USA.
- Daðason, J. F. (2012). *Post-Correction of Icelandic OCR Text*. (Master thesis), University of Iceland, Reykjavik, Iceland.
- Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3), 171-176.
- Dehkordi, Y. H. (2014). *Incorporating User Reviews as Implicit Feedback for Improving Recommender Systems*. (Master thesis), University of Victoria, Victoria, Canada.
- Do, C. B., Mahabhashyam, M. S., Brudno, M., & Batzoglou, S. (2005). ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2), 330-340.
- El-Mahallawy, M. S. M. (2008). *A large scale HMM-based omni front-written OCR system for cursive scripts*. (PhD thesis ), Cairo University, Cairo, Egypt.
- Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4), 14.
- Golding, A. R., & Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine learning*, 34(1), 107-130.
- Goswami, R., & Sharma, O. (2013). A Review on Character Recognition Techniques. *International Journal of Computer Applications*, 83(7), 19-23.

- Govindan, V., & Shivaprasad, A. (1990). Character recognition—a review. *Pattern recognition*, 23(7), 671-683.
- Habash, N., & Roth, R. M. (2011). *Using deep morphology to improve automatic error detection in Arabic handwriting recognition*. Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Stroudsburg, USA.
- Habeeb, I. Q., Yusof, S. A., & Ahmad, F. B. (2014). Two Bigrams Based Language Model for Auto Correction of Arabic OCR Errors. *International Journal of Digital Content Technology and its Applications*, 8(1), 72 - 80.
- Hadj Ameur, M. S., Moulahoum, Y., & Guessoum, A. (2015). Restoration of Arabic Diacritics Using a Multilevel Statistical Model. In A. Amine, L. Bellatreche, Z. Elberrichi, J. E. Neuhold & R. Wrembel (Eds.), *Computer Science and Its Applications* (pp. 181-192). Saida, Algeria: Springer International Publishing.
- Herceg, P., Huyck, B., Johnson, C., Van Guilder, L., & Kundu, A. (2005). *Optimizing OCR accuracy for bi-tonal, noisy scans of degraded Arabic documents*. Paper presented at the Proceedings of the International Society for Optical Engineering (SPIE) on Visual Information Processing, Florida, USA.
- Howell, D. C. (2012). *Statistical methods for psychology* (8th ed.): Cengage Learning.
- Islam, A., & Inkpen, D. (2009). *Real-word spelling correction using Google Web IT n-gram with backoff*. Paper presented at the IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE2009), Dalian, China.
- Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2nd ed.): Pearson Education India.
- Jurafsky, D., Martin, J. H., Kehler, A., Vander Linden, K., & Ward, N. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (Vol. 2): MIT Press.
- Just, W. (2001). Computational complexity of multiple sequence alignment with SP-score. *Journal of computational biology*, 8(6), 615-623.
- Kai, N. (2010). *Unsupervised Post-Correction of OCR Errors*. (Diploma thesis), Leibniz University, Hannover, Germany.
- Kanoun, S., Alimi, A. M., & Lecourtier, Y. (2011). Natural language morphology integration in off-line Arabic optical text recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(2), 579-590.



- Kenter, T., Erjavec, T., & Fišer, D. (2012). *Lexicon construction and corpus annotation of historical language with the CoBaLT editor*. Paper presented at the Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2012), Avignon, France.
- Khorsheed, M. S. (2002). Off-line Arabic character recognition—a review. *Pattern analysis & applications*, 5(1), 31-45.
- Kittler, J., Hatef, M., Duin, R. P., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3), 226-239.
- Knopp, J. (2010). *Classification of named entities in a large multilingual resource using the Wikipedia category system*. (Master thesis), University of Heidelberg, Heidelberg, Baden-Württemberg, Germany.
- Kolak, O., & Resnik, P. (2005). *OCR post-processing for low density languages*. Paper presented at the Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, Canada.
- Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4), 377-439.
- Lee, Y.-S., & Chen, H.-H. (1996). Analysis of error count distributions for improving the post-processing performance of OCR. *Communication of Chinese and Oriental Languages Information Processing Society*, 6(2), 81-86.
- Lopresti, D., & Zhou, J. (1997). Using consensus sequence voting to correct OCR errors. *Computer Vision and Image Understanding*, 67(1), 39-47.
- Lund, W. B. (2014). *Ensemble Methods for Historical Machine-Printed Document Recognition*. (PhD dissertation), Brigham Young University, Utah, USA.
- Lund, W. B., Kennard, D. J., & Ringger, E. K. (2013a). *Combining multiple thresholding binarization values to improve OCR output*. Paper presented at the Proceedings of the International Society for Optical Engineering (SPIE) on Document Recognition and Retrieval XX, San Francisco, California.
- Lund, W. B., Kennard, D. J., & Ringger, E. K. (2013b). *Why multiple document image binarizations improve OCR*. Paper presented at the Proceedings of the Workshop on Historical Document Imaging and Processing (HIP 2013), Washington, USA.
- Lund, W. B., & Ringger, E. K. (2009). *Improving optical character recognition through efficient multiple system alignment*. Paper presented at the Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries, Austin, USA.
- Lund, W. B., & Ringger, E. K. (2011, 18-21 Sept. 2011). *Error Correction with In-Domain Training Across Multiple OCR System Outputs*. Paper presented at the

Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011), Beijing, China.

Lund, W. B., Ringger, E. K., & Walker, D. D. (2014). *How well does multiple OCR error correction generalize?* Paper presented at the Proceedings of Document Recognition and Retrieval XXI (DRR 2014), San Francisco, USA.

Lund, W. B., Walker, D. D., & Ringger, E. K. (2011). *Progressive alignment and discriminative error correction for multiple OCR engines.* Paper presented at the Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011), Beijing, China.

Ma, D., & Agam, G. (2012). *Lecture video segmentation and indexing.* Paper presented at the Proceedings of the International Society for Optical Engineering (SPIE) on Document Recognition and Retrieval XIX, California, USA.

Ma, D., & Agam, G. (2013). *A super resolution framework for low resolution document image OCR.* Paper presented at the Proceedings of the International Society for Optical Engineering (SPIE) on Document Recognition and Retrieval XX, California, USA.

Magdy, W., & Darwish, K. (2008). Effect of OCR error correction on Arabic retrieval. *Information Retrieval*, 11(5), 405-425.

Mai, B. Q. Q., Huynh, T. H., & Doan, A. D. (2014). *A study about the reconstruction of remote, low resolution mobile captured text images for OCR.* Paper presented at the Proceeding of the International Conference on Advanced Technologies for Communications (ATC 2014), Saigon, Vietnam.

Muaz, A. (2011). *Urdu Optical Character Recognition System* (Master thesis), National University of Computer & Emerging Sciences, Islamabad, Pakistan.

Naseem, T. (2004). *A Hybrid Approach for Urdu Spell Checking.* (Master thesis), National University of Computer & Emerging Sciences, Islamabad, Pakistan.

Naseem, T., & Hussain, S. (2007). A novel approach for ranking spelling error corrections for Urdu. *Language Resources and Evaluation*, 41(2), 117-128. doi: 10.1007/s10579-007-9028-6

Navarro, G. (2001). A Guided tour to approximate string matching. *ACM Computing Surveys (CSUR)*, 33(1), 31-88.

Notredame, C. (2002). Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1), 131-144.

Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: A case study. *International Journal of Computer Applications*, 55(10), 50-56.

- Pervez, M. T., Babar, M. E., Nadeem, A., Aslam, M., Awan, A. R., Aslam, N., . . . Waheed, U. (2014). Evaluating the Accuracy and Efficiency of Multiple Sequence Alignment Methods. *Evolutionary bioinformatics online*, 10, 205-217.
- Pratt, W. K. (1991). *Digital image processing*: John Wiley & Sons, Inc.
- Raaid, A. F., & Rafid, H. S. (2015). Performance Evaluation of Smoothing Techniques for Arabic Character Recognition. *International Journal of Research in Information Technology (IJRIT)*, 3(11), 22-28.
- Ramanan, M., Ramanan, A., & Charles, E. (2014). *A performance comparison and post-processing error correction technique to OCRs for printed Tamil texts*. Paper presented at the Proceeding of the 9th International Conference on Industrial and Information Systems (ICIIS) Gwalior, India.
- Rardin, R. L., & Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7(3), 261-304.
- Saber, S., Ahmed, A., Elsis, A., & Hadhoud, M. (2016). Performance Evaluation of Arabic Optical Character Recognition Engines for Noisy Inputs. In T. Gaber, A. E. Hassanien, N. El-Bendary & N. Dey (Eds.), *The 1st International Conference on Advanced Intelligent System and Informatics (AISII2015), November 28-30, 2015, Beni Suef, Egypt* (Vol. 407, pp. 449-459): Springer International Publishing.
- Sattar, S. A. (2009). *A Technique for the Design and Implementation of an OCR for Printed Nastaliue Text*. (PhD thesis), NED University of Engineering & Technology, Karachi, Pakistan.
- Shaalán, K., Samih, Y., Attia, M., Pecina, P., & van Genabith, J. (2012). Arabic Word Generation and Modelling for Spell Checking. *Language Resources and Evaluation (LREC)*, 719-725.
- Shafii, M. (2014). *Optical Character Recognition of Printed Persian/Arabic Documents*. (Doctoral dissertation ), University of Windsor, Ontario, Canada.
- Shahrour, A., Khalifa, S., & Habash, N. (2015). *Improving Arabic Diacritization through Syntactic Analysis*. Paper presented at the Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal.
- Shannon, C., & Weaver, W. (2002). *A Mathematical Theory of Communication*: University of Illinois Press.
- Silfverberg, M., & Rueter, J. (2015). *Can Morphological Analyzers Improve the Quality of Optical Character Recognition?* Paper presented at the Proceeding of 1st International Workshop in Computational Linguistics for Uralic Languages (IWCLUL 2015), Tromsø, Norway.

- Singh, A., Bacchuwar, K., & Bhasin, A. (2012). A Survey of OCR Applications. *International Journal of Machine Learning and Computing (IJMLC)*, 2, 314-318.
- Springmann, U., Najock, D., Morgenroth, H., Schmid, H., Gotscharek, A., & Fink, F. (2014). *OCR of historical printings of Latin texts: problems, prospects, progress*. Paper presented at the Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, Madrid, Spain.
- Strohmaier, C., Ringlstetter, C., Schulz, K. U., & Mihov, S. (2003). *Lexical postcorrection of OCR-results: The web as a dynamic secondary dictionary*. Paper presented at the Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR), Edinburgh, UK.
- Taghva, K., & Stofsky, E. (2001). OCRSpell: an interactive spelling correction system for OCR errors in text. *International Journal on Document Analysis and Recognition*, 3(3), 125-137.
- Volk, M., Furrer, L., & Sennrich, R. (2011). Strategies for reducing and correcting OCR errors *Language Technology for Cultural Heritage* (pp. 3-22): Springer press.
- Vrandečić, D., Sorg, P., & Studer, R. (2011). *Language resources extracted from Wikipedia*. Paper presented at the Proceeding of the sixth international conference on Knowledge capture (K-CAP '2011), Banff, AB, Canada.
- Vu Hoang, C. D., & Aw, A. T. (2012). *An unsupervised and data-driven approach for spell checking in Vietnamese OCR-scanned texts*. Paper presented at the Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, Avignon, France.
- Watson, J. C. (2007). *The phonology and morphology of Arabic*: Oxford university press.
- Yaseen, M., Attia, M., Maegaard, B., Choukri, K., Paulsson, N., Haamid, S., . . . Rashwan, M. (2006). *Building annotated written and spoken Arabic LR's in NEMLAR project*. Paper presented at the Proceeding of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy.
- Zribi, C. B. O., & Ahmed, M. B. (2003). *Efficient automatic correction of misspelled Arabic words based on contextual information*. Paper presented at the Proceeding of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2003), Oxford, UK.