AN AUTOMATIC DIACRITIZATION ALGORITHM FOR UNDIACRITIZED ARABIC TEXT

AYMAN AHMAD MOHAMMAD ZAYYAN

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

UNIVERSITI UTARA MALAYSIA

2017

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)

AYMAN AHMAD MOHAMMAD ZAYYAN

calon untuk Ijazah
(candidate for the degree of) MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

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

"AN AUTOMATIC DIACRITIZATION ALGORITHM FOR UNDIACRITIZED ARABIC TEXT"

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 : ***17 Mei, 2017.***
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: ***May 17, 2017.***

Pengerusi Viva: *(Chairman for VIVA)*	Assoc. Prof. Dr. Osman Ghazali	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	Assoc. Prof. Dr. Akram M Z M Khedher	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	Dr. Samry @ Mohd Shamrie Sainin	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	Dr. Husniza Husni	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	Dr. Shahrul Azmi Mohd Yusof	Tandatangan *(Signature)*

Tarikh:
(Date) May 17, 2017

Permission to Use

In presenting this thesis in fulfilment 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 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

Bahasa Arab Standard Moden (MSA) digunakan hari ini dalam kebanyakan media bertulis dan beberapa media pertuturan. Ia bagaimanapun, bukan dialek asal mana-mana negara. Kebanyakan teks ini telah ditulis dalam dialek Mesir, kerana ia dianggap dialek yang paling banyak digunakan dan difahami di seluruh Timur Tengah. Seperti Bahasa Semitik lain, dalam Bahasa Arab bertulis, vokal pendek tidak ditulis tetapi diwakili dengan tanda diakritik. Walau bagaimanapun, tanda ini tidak digunakan dalam kebanyakan teks bahasa Arab moden (buku, akhbar, dll.). Ketiadaan tanda diakritik mewujudkan kekaburan yang besar kerana perkataan yang tidak bertanda diakritik mungkin bersesuaian dengan lebih daripada satu bentuk *diacritization* yang betul (*vowelization*). Oleh itu, matlamat penyelidikan ini adalah untuk mengurangkan kekaburan ketiadaan tanda diakritik menggunakan algoritma hibrid dengan ketepatan yang lebih tinggi berbanding sistem terkini bagi MSA. Selain itu, kajian ini juga adalah untuk melaksanakan dan menilai ketepatan algoritma untuk teks Bahasa Arab dialek. Reka bentuk algoritma yang dicadangkan berdasarkan dua teknik utama seperti berikut: statistik n-gram bersama dengan anggaran kebarangkalian maksimum dan penganalisis morfologi. Menggabungkan perkataan, morfem, dan aras huruf serta sub-model mereka bersama-sama ke dalam satu platform untuk meningkatkan ketepatan *diacritization* automatik adalah cadangan penyelidikan ini. Selain itu, dengan menggunakan ciri *case ending diacritization*, iaitu mengabaikan tanda diakritik pada huruf terakhir perkataan, menunjukkan peningkatan signifikan terhadap penambahbaikan ke atas ralat. Sebab peningkatan yang luar biasa ini adalah bahawa Bahasa Arab melarang menambah tanda diakritik terhadap beberapa huruf. Algoritma yang dicadangkan menunjukkan prestasi yang baik sebanyak 97.9% apabila digunakan untuk korpora MSA (Tashkeela), 97.1% apabila diaplikasikan pada LDC's Arabic Treebank-Part 3 v1.0 dan 91.8% apabila digunakan bagi korpus dialektal Mesir (CallHome). Sumbangan utama penyelidikan ini ialah algoritma hibrid untuk *diacritization* automatik teks MSA yang tiada diakritik dan teks Bahasa Arab dialek. Algoritma yang dicadangkan digunakan dan dinilai pada dialek Bahasa harian Mesir, dialek yang paling luas difahami dan digunakan di seluruh dunia Arab yang dianggap sebagai kali pertama berdasarkan kajian literature.

Kata kunci: Diacritization automatik, tanda diakritik, penganalisis morfologi, Anggaran kebarangkalian maksimum, statistic n-gram.

Abstract

Modern Standard Arabic (MSA) is used today in most written and some spoken media. It is, however, not the native dialect of any country. Recently, the rate of the written dialectal Arabic text increased dramatically. Most of these texts have been written in the Egyptian dialectal, as it is considered the most widely used dialect and understandable throughout the Middle East. Like other Semitic languages, in written Arabic, short vowels are not written, but are represented by diacritic marks. Nonetheless, these marks are not used in most of the modern Arabic texts (for example books and newspapers). The absence of diacritic marks creates a huge ambiguity, as the un-diacritized word may correspond to more than one correct diacritization (vowelization) form. Hence, the aim of this research is to reduce the ambiguity of the absences of diacritic marks using hybrid algorithm with significantly higher accuracy than the state-of-the-art systems for MSA. Moreover, this research is to implement and evaluate the accuracy of the algorithm for dialectal Arabic text. The design of the proposed algorithm based on two main techniques as follows: statistical n-gram along with maximum likelihood estimation and morphological analyzer. Merging the word, morpheme, and letter levels with their sub-models together into one platform in order to improve the automatic diacritization accuracy is the proposition of this research. Moreover, by utilizing the feature of the case ending diacritization, which is ignoring the diacritic mark on the last letter of the word, shows a significant error improvement. The reason for this remarkable improvement is that the Arabic language prohibits adding diacritic marks over some letters. The hybrid algorithm demonstrated a good performance of 97.9% when applied to MSA corpora (Tashkeela), 97.1% when applied on LDC's Arabic Treebank-Part 3 v1.0 and 91.8% when applied to Egyptian dialectal corpus (CallHome). The main contribution of this research is the hybrid algorithm for automatic diacritization of undiacritized MSA text and dialectal Arabic text. The proposed algorithm applied and evaluated on Egyptian colloquial dialect, the most widely dialect understood and used throughout the Arab world, which is considered as first time based on the literature review.

Keywords: Automatic diacritization, Diacritic marks, morphological analyzer, maximum likelihood estimation, statistical n-gram.

Acknowledgement

All praise is due to Allah, who guided me to this.

I would like to express my sincere gratitude to my supervisor; Dr. Husniza binti Husni and Dr. Shahrul Azmi Mohd Yusof. I'm greatly indebted to their assistance, guidance and support.

I would like to thank the Arabic language expert Dr. Mohamed Elmahdy, German University in Cairo, for his generous help.

I am very grateful for my dear parents, wife, daughter and my friends whom I consider as my brothers. Thank you all for being always there when I needed you most. Thank you for believing in me and supporting me. I believe that without your support and your prayers, none of this work would be accomplished. Finally, I hope this thesis be a useful addition to the research activities of Arabic natural language processing.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1- **MSA:** Modern Standard Arabic.

2- **OOV:** Out of Vocabulary.

3- **WER1:** Word Error Rate, without considering the case ending.

4- **WER2:** Word Error Rate, with considering the case ending.

5- **DER1:** Diacritization Error Rate, without considering the case ending.

6- **DER2:** Diacritization Error Rate, with considering the case ending.

CHAPTER ONE
INTRODUCTION

1.1 Background

Arabic is the largest still living Semitic language in terms of number of speakers that exceeds 350 million [1]. Arabic is natively spoken by people in the Middle East as well as for religious texts by Muslims in many countries. Modern Standard Arabic (MSA) [2] is the form of Arabic closest to the classical Arabic used in the Qur'an and other ancient texts. MSA is used today in most written and some spoken media. It is, however, not the native dialect of any country. Recently the rate of the written dialectal Arabic text increased dramatically. It is being used as a daily life language communication and for expressing the ideas across the World Wide Web [3]. Most of these texts have been written in the Egyptian dialectal, as it is considered the most widely dialect used and understood throughout the Middle East [3]. Moreover, due to the limited availability of the dialectal data. Like other Semitic languages, in written Arabic, short vowels are not written, but are represented by diacritic marks. Nonetheless, these marks are not used in most of the modern Arabic texts (books, newspapers, etc).

The Arabic language is one of the languages where the intended pronunciation of a certain word cannot be fully determined by its standard orthographic representation. Therefore, a set of special diacritic marks is needed in order to indicate the intended correct pronunciation, see Table 1.1.

Table 1.1

Arabic language diacritic marks

Diacritic's type	Diacritic	Example of a letter
Short vowel	Fatha	بَ
	Kasra	بِ
	Damma	بُ
Doubled case ending (Tanween)	Tanween Fatha	بً
	Tanween Kasra	بٍ
	Tanween Damma	بٌ
Syllabification marks	Sukuun	بْ
	Shadda	بّ

The absence of diacritic marks creates a huge ambiguity, as the undiacritized word may correspond to more than one (correct) diacritization form. For example, the word كتب may be diacritized as كَتَبَ (wrote), كُتِبَ (was written), كُتُب (books), كَتَّب (made someone write), or كُتِّب (was forced to write), see Table 1.2.

Table 1.2

Illustrate the different meanings of diacritized Arabic word "كتب"

Diacritized Form	Transliteration	Meaning
كَتَبَ	Kataba	Wrote
كُتِبَ	Kutiba	Was written
كُتُب	Kutub	Books
كَتَّب	kattaba	Made someone write
كُتِّب	Kuttiba	Was forced to write

One of the major challenges for the Arabic language is its rich derivative and complex nature. It is completely difficult to build a complete vocabulary that covers all (or even most of) the Arabic general words. Arabic readers, however, are able to figure out the correct form of the word from the context. Other diacritic marks are also used in Arabic to show the absence of a vowel or the duplication of consonants. Nevertheless, the presence of diacritics is desired, or sometimes even crucial in most Natural Language Processing (NLP) tasks. These include text to speech engines, which cannot function correctly in the absence of diacritization. Data mining is another field where diacritization will help retrieves the exact words in the queries.

1.2 Problem Statement

Arabic text resources are mostly written without diacritic marks, because the manual addition of diacritic marks is tedious, expensive and impractical solution [4], as it requires a long time and a large number of Arabic language experts. These difficulties with the manual solution have created a need for an automated and accurate tool to help in restoring the diacritic marks [5]. Techniques for automatic diacritization have been in development since the late 1980s [6]. Most of the conducted research focused on MSA diacritization [7] [2] [1] [8] [5], though the dialectal Arabic is of the utmost importance as it's the everyday life communication language. There are significant differences between MSA and dialectal Arabic which prevent the researchers from investigating and evaluating their techniques on dialectal Arabic text. According to the literature review, due to the limited availability of dialectal Arabic text resources, most of the existing techniques have not been applied on dialectal Arabic.

The field has been continuously an active research field [9] with many rule-based techniques being implemented to tackle the problem, namely lexical analyzer [10], morphological analyzer [10], [7], syntax analyzer [7], and statistical-based techniques, namely Maximum Likelihood Estimation [9], [11], Hidden Markov Model (HMM) [9], [11], Statistical Machine Translation (SMT) [4], n-gram [9], [11], and Finite State Transducers (FST) [12]. The current automatic diacritization techniques still fall short of the desired outcome of a near perfect diacritic restoration, in particular, the rule-based techniques, namely lexical analyzer, morphological analyzer and syntax analyzer. The relatively low accuracy of the rule-based techniques can be attributed to the morphological complexity and the difficulty of keeping up a huge list of grammatical rules that maintain all the aspects of Arabic language including the difficulties in diacritizing the ending letter [13], [6]. Therefore, the main focus of this study is to propose, implement and evaluate a hybrid based algorithm that automatically retrieves the diacritic marks of the MSA with accuracy significantly higher than the current state-of-the-art systems. The algorithm will be based on a hybrid technique that combines the statistical n-gram alongside with the maximum likelihood estimate and the morphological analyzer.

Since Arabic is a morphologically very rich derivative and complex nature [7], vocabulary size can reach several billions of words. That is why a morphological analyzer is used to decompose Out-of-Vocabulary (OOV) words into morphemes.

Measures that are utilized in order to evaluate the performance of the algorithm will be the WER [1], [2], [3], [4], [5], [6] and the DER [1], [2], [3], [4], [5], [6]. Based on the previous research, WER is the percentage of the words that diacritized incorrectly (one

letter at least has an incorrect diacritic mark), while the DER is the percentage of the letters that diacritized incorrectly. WER cannot be utilized as the only measure for the diacritization accuracy, as it might provide inaccurate information about the system performance. For example, if there is a word diacritized incorrectly because of one diacritization mark, in this case, WER and DER will be equal to one, while if we have one word diacritized incorrectly because of four diacritization marks, WER will be equal to one and DER will be equal to four. Therefore, both measures WER and DER give more precise indication of the accuracy of the approach in use.

Due to the fact that the diacritic marks attached to the last letter of the words (case ending), do rarely affect the meaning of Arabic statement. Therefore, many authors considered this fact in their study in order to increase the diacritization accuracy.

1.3 Research Question

The main research question of this study is how to improve the automatic diacritization accuracy of the undiacritized MSA text, which will be positively reflected on the WER and DER. Moreover, to evaluate the diacritization accuracy, WER and DER of the proposed algorithm when applied to dialectal Arabic text, especially the Egyptian dialectal Arabic text, as it is being used as a daily life language communication, and for expressing the ideas across the World Wide Web. Therefore, the strengths and weakness of the current diacritization algorithms have to be addressed in order to answer the main research question of this research.

1.4 Research Objectives

The main objective of this study is to design and develop a hybrid based algorithm for automatically retrieving the diacritic marks for undiacritized MSA text and dialectal Arabic text, with accuracy significantly higher than the current state-of-the-art systems. The followings are the sub objectives:

a) To propose an improved hybrid algorithm that combines the rule-based approach, namely morphological analyzer along with a statistical-based approach, namely statistical n-gram and maximum likelihood estimate.

b) To implement the proposed hybrid algorithm on widely available MSA datasets for restoring the diacritic marks, and displaying the correct form of the word.

c) To evaluate the proposed hybrid algorithm using the diacritization accuracy, WER and DER and compare it with the current state-of-the-art algorithms.

1.5 Research Scope

The aim of this study is to propose and implement an automatic diacritization hybrid algorithm for MSA with significantly higher accuracy than the state-of-the-art systems. Moreover, to implement and evaluate the accuracy of the proposed algorithm for dialectal Arabic text, as the rate of the written dialectal Arabic text increased dramatically. It is being used as a daily life language communication and for expressing the ideas between Arab people across the World Wide Web [3]. Most of these texts have been written in the Egyptian dialectal, as it is considered the most widely dialect used and understood throughout the Middle East [3]. Therefore, the proposed algorithm will be implemented and evaluated on Egyptian dialectal.

1.6 Deliverables

A hybrid algorithm is proposed comprising four main modules:

The first one is a pre-processing script to confirm that the corpus is containing only the alphabetic letters, diacritic and punctuation marks. The second module builds a dictionary of n-grams models at the levels of word, morpheme and letter. The third module which is the main contribution and research objective of this study will be used for automatic diacritization of the undiacritized Arabic text, while the last module will be used for testing and evaluating the results on widely available data sets. These modules will be then utilized to achieve an improved algorithm for automatic diacritization with significantly higher accuracy than the state-of-the-art systems.

1.7 Significance of Research

The significance of this study is increasing the diacritization accuracy of the undiacritized Arabic text, which will significantly ease the understanding of non-native Arabic speakers for undiacritized Arabic text. Moreover, the proposed algorithm will be applied and evaluated on Egyptian dialect Arabic text, the most widely dialect understood and used throughout the Arab world, which is considered as first time based on the literature review.

1.8 Thesis Organization

Aiming to enhance the accuracy of automatic diacritization for undiacritized MSA text; a hybrid algorithm is proposed that combines the rule-based approach, namely morphological analyzer along with statistical-based approach, namely statistical n-gram and maximum likelihood estimate, trained at different lexical unit levels (words,

morphemes, and letters). Thus, this thesis presents five chapters, including this chapter, that explain in detail what has been done. Chapter 1 includes the necessary information for understanding the concepts that are used in the next chapters. Chapter 2 discussed the literature review with a description of the different aspects relating to the research area. Chapter 3 presents the methodology steps that were used in this study. Chapter 4 presents the proposed algorithm and the results. Finally, Chapter 5 includes the achieved objects, limitations and recommendation, contribution of this study and the future work.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

Due to the importance of automatic restoration for diacritic marks, many attempts have been tackled by research teams to approach the Arabic diacritization problem over the past two decades [1], [2], [3], [4], [5], [6], [7], [8]. These attempts are divided mainly into two categories: first category concerns with the systems developed by project researchers as part of their academic activities at academic research centers; the second category of these attempts concerns the commercial companies for realizing market applications. However, current automatic diacritization techniques still fall short of the desired outcome of near perfect diacritic restoration. The techniques used in automatic diacritization are divided mainly into three approaches: rule-based approach, statistical-based approach, and hybrid approach. This chapter will review the previous work carried out according to each approach and will identify the shortcomings and gaps in this research area.

2.2 Diacritization approaches

In this section, and referring to the previous works carried out, the state-of-the-art systems for automatic diacritization of Arabic texts will be discussed according to their approaches.

2.2.1 Rule-based approach

The rule-based systems for automatic diacritization depend on a core of solid linguistic knowledge, in order to provide a solution for a problem. These systems are solving the

diacritization problem intelligently and heuristically by exploiting the human knowledge. However, the high level of ambiguity and a large number of morphological and syntactic rules is the main drawback of this approach; hence, it's difficult to develop an automatic diacritization system based only on grammar rules.

One of the major challenges for the Arabic language is its rich derivative and complex nature. It is completely difficult to build a complete vocabulary that covers all the Arabic general words. Thus, many words could not be diacritized based on statistical n-gram and maximum likelihood estimate, and these words will be considered as OOV. Therefore, it was very important to overcome the OOV during the diacritization process. In this case, morphological analyzer could be used to handle the OOV, by factorize the OOV words into its possible morphological components (prefix, root and suffix), and then diacritize each segment separately using statistical n-gram and maximum likelihood estimate.

A tagging system was proposed which classifies the words into a non-vocalized Arabic text to their tags [10]. The system goes through three analysis levels. First one is a lexical analyzer, the second level is a morphological analyzer, and the last level is a syntax analyzer. They have tested the system performance using a data set with a total of 2355 non-vocalized words selected randomly from newspaper articles. The reported accuracy of the system was 94%. The author didn't clarify in his research the training corpus, also the testing corpus is relatively small in terms of size.

A rule-based diacritization system for written Arabic was presented by N. Habash [14]; this system based on a lexical resource, which combines a lexeme language and tagger

model. They used “ATB3-Train”, 288,000 corpus for training the system, and “ATB3-Devtest”, 52,000 words for testing purpose. The best result reported by their system was 14.9% as WER and 4.8% as DER. Authors also have considered the case ending and their system reported 5.5% as WER and 2.2% as DER.

Table 2.1 summarizes the diacritization accuracy, WER and DER for the above Rule-based approaches.

Table 2.1

The diacritization accuracy, WER and DER for the Rule-based approaches.

Author	Dataset	Accuracy	WER1	DER1	WER2	DER2
A. Al-Taani [10] - (2009)	Consists of 2355 non-vocalized Arabic words, selected randomly from newspaper articles.	94%	-	-	-	-
N. Habash [14] – (2007)	They used ATB3-Train with 288,000 for training purpose and ATB3-Devtest with 52,000 words for testing purpose.	-	14.9%	4.8%	5.5%	2.2%

2.2.2 Statistical approach

Probability prediction for a sequence of letters or sequence of words in this approach is based on certain statistics, such as letters or words frequency in the data resource. The main advantage of applying this approach is that there is no need for using the morphological or syntactic rules applied in rule-based approach. However, this approach requires a huge and fully diacritized Arabic corpus. This approach includes many sub-models, such as Hidden Markov Model (HMM), n-gram model, Statistical Machine Translation (SMT), and Finite State Transducers (FST).

Statistical n-gram and maximum likelihood estimate could be employed as a stand-alone approach to diacritize sentence, word and letter [6]. It’s one of the most commonly used approaches due to the difficulty in retrieving the missing diacritic marks of undiacritized Arabic text [6], statistical n-gram along with maximum likelihood estimate resolve the

ambiguity problem of Arabic language, that has been discussed in section 1.1 - The undiacritized word may correspond to more than one correct diacritization form. In this case, it would be easier to consider the right context or the left context of the selected word to be diacritized in order to get to correct diacritization form.

Therefore, the accuracy of the statistical n-gram algorithm determined based on the value of n, as the diacritization accuracy significantly increased with larger value of n.

A new statistical approach for Arabic diacritics restoration was presented [11], this system is based on two main models - the first one is a bi-gram-based model to handle vocalization, the second one is a 4-gram letter-based model to handle the OOV words. The diacritization probability for both models was calculated based on the following equation:

$$P(W_n|W_1^{n-1}) = P(W_n|W_{n-1})$$

The applied equation for n-gram Author used a corpus retrieved automatically from the URL http://www.al-islam.com/. This corpus is an Islamic religious corpus contains a number of vocalized subjects (Quran Commentaries, Hadith, etc.). Moreover, vocalized Holy Qur'an was also downloaded from the URL http://tanzil.net/ and merged with the corpus. Training to testing ratio was 90% to 10% respectively. The system reported WER varies from 11.53% to 16.87% based on the applied smoothing model, and DER varies from 4.30% to 8.10% based on the applied smoothing model. They have considered the case ending in their research and their system reported WER varies from 6.28% to 9.49% based on the applied smoothing model, and DER varies from 3.18% to 6.86% based on the applied smoothing model.

A statistical approach for automatic diacritization of MSA and Algiers dialectal texts was proposed [4]. This approach is based on statistical machine translation. Authors first

investigate this approach on MSA texts using several data sources and extrapolated the results on available dialectal texts. For MSA corpus, they used Tashkeela, a free corpus under GPL license. This corpus is a collection of classical Arabic books downloaded from an on-line library. It consists of more than 6 million words. They split data on training (80%), developing (10%) and testing sets (10%). For comparison purpose, they used LDC Arabic Treebank (Part3, V1.0). For dialect corpus, they created the Algiers dialect corpus by hand; initially, it did not contain diacritics, and proceed to vocalize it by hand. The vocalized corpus consists of 4,000 pairs of sentences, with 23,000 words. For MSA, WER reported by their system is 16.2% and 23.1% based on the corpus on use, while DER reported is 4.1% and 5.7% based on the corpus on use. For Algiers dialect corpus, WER reported by their system is 25.8%, DER reported by their system is 12.8%.

An algorithm was proposed in order to recover the diacritic marks using dynamic programming approach [9]. The possible word sequences with diacritics are assigned scores using statistical n-gram language modeling approach, different smoothing techniques used in this research such as Katz smoothing, Absolute Discounting and Kneser-Ney for Arabic diacritization restoration. For training and testing purpose, authors used Arabic vocalized text corpus Tashkeela. The corpus is free and collected from the internet using automatic web crawling method. It contains 54,402,229 words. The author divided the corpus into training and testing sets, the training set consists of 52,500,084 words, while the testing set consists of 1,902,145 words, which mean 96.5% of the corpus used for training purpose, and 3.5% used for testing purpose. The WER for this system varies from 8.9% to 9.5% based on the applied smoothing model. The WER in the case of considering the case ending varies from 3.4% to 3.7% based on the applied

smoothing model. The author in this research didn't mention the DER based on the applied system.

A new search algorithm was developed which supports higher order n-gram language models [15]. The search algorithm depends on dynamic lattices where the scores of different paths computed on the run time. For training and testing purpose, authors used Arabic vocalized text corpus Tashkeela. The corpus is free and collected from the internet using automatic web crawling method. It contains 6,149,726 words. The author divided the corpus into training and testing sets, the training set consists of 52,500,084 words, while the testing set consists of 1,902,145 words, which mean 96.5% of the corpus used for training purpose, and 3.5% used for testing purpose. The WER for this system varies for 8.9% to 9.2% based on the applied model, and the WER in the case of considering the case ending varies from 3.4% to 3.6% based on the applied model. Author in this research didn't mention the DER.

The empirical study for Arabic diacritization restoration, using different smoothing techniques commonly used in speech recognition and machine translation fields was proposed [16]. For training and testing purpose, authors used Arabic vocalized text corpus Tashkeela. The corpus is free and collected from the internet using automatic web crawling method. It contains 6,149,726 words. The author divided the corpus into training and testing sets, the training set consists of 52,500,084 words, while the testing set consists of 1,902,145 words, which mean 96.5% of the corpus used for training purpose, and 3.5% used for testing purpose. The WER for his system varies from 8.9% to 9.5% based on the applied smoothing model, the WER in the case of considering the case ending vary from 3.4% to 3.7% based on the applied smoothing model. The author in this research didn't mention the DER based on the applied system.

A baseline system which is small in terms of size, fast in terms of processing and independent from linguistic rules and other tools was proposed [17]. The system uses a statistical method that relies on quad-gram probabilities. For training purpose, authors have used KDATD corpus that developed by KACST to create the quad-gram list, the corpus contains 231 text files with 22 different subjects. Each file has an average of 1000 diacritized words. Authors tested their system using 15983 words from LDC corpus. Their system reported 46.83% as WER and 13.83% as character error rate, they have considered the case ending in their research and their system reported 26.03% as WER and 9.25% as character error rate. Authors in this research didn't mention the DER in both cases, with case ending and without case ending.

An innovative system for Arabic text diacritization was proposed [18], the system based on a statistical method that depends on a quad-gram probability and the applied technique in this system has mainly two steps. Step one is to create a very rich quad-grams list of Arabic words which is used frequently, step two is to utilize that list in discretizing almost any Arabic text. For training purpose, authors used a corpus developed by KACST in order to create the list of quad-gram, the corpus contains 231 files with 22 different subjects. Each file has 1000 diacritized words as an average. Authors tested their system using 5 different articles taken from KACST corpus and 10 articles from Alriyadh Newspaper. The error rate for the first set was 7.64% and for the second set was 8.87%, the average error rate for both sets was 8.52%. In this research the authors didn't clarify the meaning of the error rate, is it WER or DER, also the training to testing ratio wasn't mentioned. Moreover, authors didn't consider the case ending.

An HMM statistical approach for automatic generation of the diacritical marks of the Arabic text was proposed [19]. The used approach needs a fully diacritized large corpus of texts for retrieving the language n-gram for letters and words. Search algorithms are then utilized to retrieved the best diacritized word form of the given undiacritized word. Authors used the Holy Qur'an as Arabic text corpus that contains 78,679 words and 607,849 characters, for testing they used a set contains 995 words and 7657 characters, which mean 98.75% as training set and 1.25% as a testing set. Their system reported 4.1% as letter error rate. In this research the authors didn't mention the WER and DER, also they didn't consider the case ending in their research and the reflection on WER and DER.

A new statistical HMM approach was presented [20], authors used a corpus prepared by King Abdulaziz City of Science and Technology, it includes 100 articles different newspapers and magazines, covering a number of subjects. Their system operated at 0.5% when tested on the corpus, and 5.5% when tested on other corpora. In this research authors didn't mentioned the DER, and they didn't consider the case ending in their research. Moreover, the training to testing ratio was not clear.

A statistical approach that restores automatically the diacritics marks was presented [21]. It is based on the maximum entropy framework. Different sources of information were utilized. The model is based on learning the correlation between different types of output diacritics and information. The dataset used for training and testing purpose was LDC's Arabic Treebank, which includes complete vocalization with a total of 340,281 words. Authors split the corpus into training and testing data, the training contains 288,000 words, while the test data contains 52,000 words, which means 85% as training set and 15% as a test set. Their system reported 17.3% as WER and 5.1% as DER, also they

have considered the case ending and their system reported 7.2% as WER and 2.2% as DER.

A statistical and knowledge-based approach that implements a number of generative statistical models at the character and word levels, in order to recover the missing diacritics based on the context was proposed [22]. The approach was trained using Arabic Treebank catalogs released by the LDC. These corpora contain about 554,000 words, they used 541,00 words for training purpose, and 13,300 words for testing purpose, which means 97.5% for training and 2.5% for testing. Their system accuracy varies from 74.96% to 86.50% based on the applied model. In this research authors didn't mention the DER, also they didn't consider the case ending.

A statistical approach proposed for Arabic diacritization restoration was proposed [12], this approach based on finite-state transducers algorithm was proposed and integrated with a letter-based and word-based language models, along with the morphological model. The system was trained by 90% of LDC's Arabic Treebank. This corpus contains 501 news stories retrieved from Al-Hayat with a total of 144,199 words. The remaining 10% was used for testing purpose. The WER in that system varies from 23.61% to 30.39% based on the applied model, and the DER varies from 12.79% to 24.03% based on the applied model. Authors considered the case ending and it's reflection on WER and DER in their research. The WER after considering the case ending varies from 7.33% to 15.48% based on the applied model, and the DER varies from 6.35% to 17.33% based on the applied model.

An HMM was proposed [23] as a statistically based approach for vowel restoration in Semitic languages Arabic and Hebrew; Qur'an was used as Arabic text corpus and Bible as Hebrew text corpus. The proposed system was trained by 90% of Qur'an and Bible,

the remaining 10% was used for testing purpose. This system achieves an accuracy of 86% for Arabic texts and of 81% for Hebrew texts. The author didn't mention in his research the DER, he also didn't consider in his research the case ending and the reflection on WER and DER.

Table 2.2 summarizes the diacritization accuracy, WER and DER for the Statistical approaches.

Table 2.2

The diacritization accuracy, WER and DER for the Statistical approaches.

Author	Dataset	Accuracy	WER1	DER1	WER2	DER2
M. Ameur [11] – (2015)	Retrieved automatically from http://www.al-islam.com/	-	11.53% to 16.87%	4.30% to 8.10%	6.28% to 9.49%	3.18% to 6.86%
S. Harrat [4] – (2013)	**MSA corpus:** Tashkela, free corpus under GPL license. **Dialect corpus:** Created the Algiers dialectal corpus by hand.	-	**MSA:** 16.2% and 23.1% **Dialects:** 25.8%	**MSA:** 4.1% and 5.7% **Dialects:** 12.8%	-	-
Y. Hifny [9] – (2013)	Tashkela With 54,402,229 words. Training set 52,500,084 words. Testing set consists of 1,902,145 words.	-	8.9% to 9.5%	-	3.4% to 3.7%	-
Y. Hifny [15] – (2012)	Tashkela With 6,149,726 words. Training set 52,500,084 words. Testing set consists of 1,902,145 words.	-	8.9% to 9.2%	-	3.4% to 3.6%	-
Y. Hifny [16] – (2012)	Tashkeela With 6,149,726 words. Training set 52,500,084 words. Testing set consists of 1,902,145 words.	-	8.9% to 9.5%	-	3.4% To 3.7%	-
M. Alghamdi [17] – (2010)	Developed by KACST, contains 231 text files, around 1000 diacritized words per file. Testing using 15983 words from LDC corpus.	-	46.83%	-	26.03%	-
M. Alghamdi [18] – (2007)	Developed by KACST, contains 231 text files, around 1000 diacritized words per file. Testing using 5 articles taken from KACST corpus and 10 articles from Alriyadh Newspaper.	-	7.64% to 8.87%	-	-	-
M. Elshafei [19] – (2006)	Holy Qur'an, for training 78,679 words 995 words for testing.	-	-	-	-	-
M. Elshafei [20] – (2006)	Developed by king Abdulaziz City of Science and Tech., consists of 100 articles collected from magazines and newspapers covering various subjects. Testing was manually diacritized by Arabic language specialist	-	0.5% to 5.5%	-	-	-
I. Zitouni [21] – (2006)	Trained and evaluated on the LDC's Arabic Treebank, with total of 340,281 words. Training contains 288,000 words, testing contains 52,000 words.	-	17.3%	5.1%	7.2%	2.2%
S. Ananthakrishnan [22] – (2005)	Arabic Treebank with totaling about 554,000 words. Training 541,000 words and testing 13,300 words.	74.96% to 86.50%	-	-	-	-
R. Nelken	Trained by 90% of LDC's Arabic	-	23.61%	12.79%	7.33% to	6.35%

[12] – (2005)	Treebank of diacritized news stories (Part 2).The remaining 10% used for testing purpose.		to 30.39%	to 24.03%	15.48%	to 17.33%
Y. Gal [23] – (2002)	Holy Qur'an. Training 90%, the remaining 10% was used for testing.	86%	-	-	-	-

2.2.3 Hybrid approach

In this study, hybrid algorithm will combine the statistical n-gram along with maximum likelihood estimate and the morphological analyzer in order to retrieve the missing diacritic marks of undiacritized Arabic text. In this case, the diacritization process will be based on three levels, first level is word level, and the diacritization based on statistical n-gram along with maximum likelihood estimate. In case of OOV, the algorithm will switch to the second level, morphological analyzer, and factorize the OOV words into its possible morphological components, prefix, root and suffix, and then and then diacritize each segment separately using statistical n-gram and maximum likelihood estimate. In case of morphological analyzer OOV, the algorithm will switch to the third level, letter level, and will split each segment from morphological analyzer in to set of letters, and then and then diacritize each letter separately using statistical n-gram and maximum likelihood estimate.

A Hybrid approach that uses the strengths of rule-based approach and statistical approach was presented [1], from the important work in this field, a solution developed and tackled the Arabic diacritization under a deep learning framework that includes the Confused Sub-set Resolution (CSR) method to improve the classification accuracy, in addition to an Arabic Part-of-Speech (PoS) tagging framework using deep neural nets. Authors used TRN_DB_I and TRN_DB_II for training purpose, with 750,000- word dataset and 2,500,000- word dataset respectively, collected from different sources and diacritized manually by expert linguists, for testing purpose they used TST_DB with

11,000- word test set. Their system reported syntactical accuracy varies from 88.2% to 88.4% based on the dataset on use, and 97% as morphological accuracy.

An approach based on a sequence transcription was developed for the automated diacritization of Arabic text [2]. A recurrent neural network is trained to recover the diacritics marks of undiacritized Arabic text. Authors used a deep bidirectional long short-term memory network that builds high-level linguistic abstractions of text and exploits long- range context in both input directions. Authors used data from the books of Islamic religious heritage, along with Holy Qur'an. These 11 books are written with full diacritization marks. 88% used for training purpose and the remaining 12% for testing purpose. The WER in their system varies from 5.82% to 15.29% based on the data in use, the DER varies from 2.09% to 4.71% based on the data in use. They considered the case ending and the WER in varies from 3.54% to 10.23% based on the data in use, the DER varies from 1.28% to 3.07% based on the data in use.

A hybrid diacritization system utilized data-driven and rule-based techniques was developed [5]. This system was based on morphological analysis, POS tagging, automatic correction and out of vocabulary diacritization components. Authors used LDC's Arabic Treebank #LDC2004T11 for training and testing purpose, the training set 288K words and a test set 52 K words, which means 85% to 15% training to testing ratio respectively. The best reported WER was 11.4% and DER 3.6%. By considering the case ending, the best reported WER was 4.4% and DER 1.6%.

The issue of retrieving the missing diacritic marks to undiacritized MSA Arabic text was addressed [24], using a hybrid approach that relies on lexicon retrieval, bigram, and SVM-statistical prioritized techniques. The diacritization system trained and evaluated

on the LDC's Arabic Treebank Part 2 v2.0, where this corpus includes 501 stories collected from the Ummah Arabic News Text, with a total number of 144,199 words. Training to testing ratio was 92.5% to 7.5% respectively. The proposed system reported 17.31% as WER and 4.41% as DER. By considering the case ending, the system reported 12.16% as WER and 3.78% as DER.

A hybrid approach for automatically diacritize MSA Arabic text was presented [25]. Presented approach combines the rule-based technique and data-driven technique in order to recover the missing diacritic marks in MSA text. For training and testing purpose, the author used ATB corpus that contains around 350K works. The author didn't mention the training to the testing ratio in his research. The proposed system reported 11.4% as WER and 3.6% as DER. By considering the case ending, the system reported 4.4% as WER and 1.6% as DER.

A large-scale dual-mode stochastic hybrid system was presented [26], the proposed system is based on two main steps. The first one was simple maximum-likelihood-unigram probability estimation; each undiacritized word in the test set was replaced by the corresponding diacritized one that occurs most frequently in the training set. In the case of OOV, system set to switch to the second step, which split each Arabic word into all its possible morphological constituents, then applied the same technique simple maximum-likelihood-unigram probability estimation, hence the most likely diacritization. Authors used for training purpose TRN_DB_I with 750,000- word dataset, collected from different sources and manually annotated by expert linguists with every word PoS and Morphological quadruples, and TRN_DB_II with 2,500,000- word dataset. For testing purpose, they have used TST_DB with 11,000- word test set. Their system reported a WER vary from 3.1% till 18% based on the used model.

The diacritization problem was treated as an SMT problem and sequence labeling problem [27]. The proposed translation system uses a pure SMT with several models. The translation model is built for a phrase-based system, where phrases were diacritized with a word level model. For training and testing purpose, the author used two data sources, the diacritized LDC's Arabic Treebank as well as data provided by AppTek. The training to testing ratio was not defined in this research. The best WER reported for this system was 21.9%, the DER 4.7%. By considering the case ending, the best WER reported 8.3% as WER and 1.9% as DER.

A new hybrid based algorithm presented for automatically diacritize MSA Arabic text [28]. The presented system is based on two layers in which, the first layer tries to decide the most likely diacritic marks by selecting the sequence of full-form Arabic word diacritization with the highest probability via A* lattice and m-gram probability estimation. If the case of OOV from the first layer, the second layer is resorted to factorizes each selected word into its possible morphological structure (prefix, root and suffix), then uses m-gram probability estimation and A* lattice for selecting the most likely diacritization marks. For training purpose, the author used TRN_DB_I and TRN_DB_II, with a total number of words ≈ 750K and ≈ 2500K respectively. For testing purpose, the author used TST_DB with a total number of words ≈ 11K. The best reported WER by the proposed algorithm was 2.1%, and DER wasn't mentioned by the author. Moreover, the author didn't consider the case ending in this study and the reflection on WER and DER.

A hybrid methodology for language modeling was proposed [29]. The system factored language modeling (FLM) and morphological decomposition were exploited to work with the complex morphology of Arabic language. Authors evaluate the results of the

GALE 2007 development and evaluation sets dev07 2.5h and eval07 4h. WER reported by their system varies from 13.9% to 16.5% based on the applied model and the corpus in use. They didn't consider the case ending in this research and the reflection on WER and DER. A two-layer statistical system is proposed to diacritize automatically Arabic text [30]. The first layer was based on simple maximum-likelihood n-gram probability estimation and long A* lattice search. When full-form words happen to be out-of-vocabulary, system set to switch to the second layer which was split each Arabic word into its prefix, root, pattern and suffix, then uses A* lattice search and n-gram probability estimation to select among the diacritize forms of the selected word. For training and testing purpose, authors used LDC's Arabic Treebank with 340,281 words; they split the data into two sets, a training set and testing set. The training set contains 288,000 words, whereas the test data contains 52,000 words, which means 85% as training set and 15% as a test set. WER reported by their system was 12.5%, while the DER varies was 3.8%, based on the applied model. They have considered the case ending in their research, WER reported after considering the case ending was 3.1%, while DER was 1.2%, based on the applied model. A new Hybrid diacritization module was proposed [31], using a new combination of techniques, tagger and a lexeme language model. Author trained the proposed approach using Arabic Treebank catalogs ("ATB3-Train"), released by the LDC, it contains about 288,000 words. For testing purpose, the author used ("ATB3-Devtest"), released by the LDC, it contains about 52,000 words. The system reported WER 14.9% and DER 4.8%, and by considering the case ending in his research, the system reported WER 5.5% and DER 2.2%.

Arabic automatic diacritization approach that integrates syntactic analysis with morphological tagging through improving the prediction of case and state features was

proposed [7]. The system increases the accuracy of word diacritization by 2.5% absolute on all words, and 5.2% absolute on nominals over a state-of-the-art baseline. Authors didn't consider the case ending in their study. A new hybrid approach for automatic vowelization of Arabic texts was proposed [13], the proposed approach depends on two phases, the first one is morphological analysis, which provides all possible vowelization for each word of the text taken out of context. The second one is statistical analysis; it consists of a statistical treatment based on the hidden Markov model and the Viterbi algorithm, and allows obtaining the most likely vowelization of words in the sentence. The training carried out with 90% of a corpora consisting of 2,463,351 vowelized words, divided between NEMLAR corpus (460,000 words), (Tashkeela) corpus (780,000 words) and RDI corpus (1,223,351 words). The remaining 10% used testing phase. The WER for this system was 21.11%, the DER 7.37%. By considering the case ending, the system reported 9.93% as WER and 3.75% as DER. The Arabic diacritization under a deep learning framework was presented [7], it includes the Confused Sub-set Resolution (CSR) method to improve the classification accuracy, in addition to an Arabic Part-of-Speech (PoS) tagging framework using deep neural nets. Authors used TRN_DB_I and TRN_DB_II for training purpose, with 750,000- word dataset and 2,500,000- word dataset respectively, collected from many sources and annotated manually by expert linguists, for the testing purpose they used TST_DB with 11,000- word test set. Their system reported syntactical accuracy varies from 88.2% to 88.4% based on the dataset on use, and 97% as morphological accuracy.

Table 2.3 summarizes the diacritization accuracy, WER and DER for the Hybrid approaches.

Table 2.3

The diacritization accuracy, WER and DER for the Hybrid approaches.

Author	Dataset	Accuracy	WER1	DER1	WER2	DER2
M. Rashwan [1] - (2015)	TRN_DB_I and TRN_DB_II for training purpose, with 750,000 and 2,500,000 word. For testing purpose TST_DB with 11,000 word.	88.2% to 88.4%	-	-	-	-
G. Abandah [2] - (2015)	Data drawn from ten books of the Tashkeela collection of Islamic religious heritage books. 88% for training and the remaining 12% for testing.	-	5.82% to 15.29%	2.09% To 4.71%	3.54% to 10.23%	1.28% to 3.07%
A. Said [5] - (2013)	LDC's Arabic Treebank #LDC2004T11. Training set 288K words and testing 52K words.	-	11.4%	3.6%	4.4%	1.6%
M. Rashwan [26] - (2011)	TRN_DB_I and TRN_DB_II for training with 2,500,000 with 750,000 words. Testing TST_DB with 11,000 words.	-	3.1% To 18%	-	-	-
A. El-Desoky [29] - (2010)	GALE 2007 development and evaluation sets dev07 2.5h and eval07 4h.	-	13.9% to 16.5%	-	-	-
M. RASHWAN [30] - (2009)	LDC's Arabic Treebank - Part 3 v1.0. With total of 340,281 words. Training contains 288,000 words testing contains 52,000 words.	-	12.5%	3.8%	3.1%	1.2%
A. Shahrour [7] - (2015)	Penn Arabic Treebank (PATB, parts 1, 2 and 3). Divide Dev into two parts with equal number of sentences: DevTrain (30K words) for training and DevTest (33K words) for development testing. The Test set has 63K words.	-	11%	4%	-	-
M. Rashwan [8] - (2014)	TRN_DB_I and TRN_DB_II with 750,000 and 2,500,000 words. Testing TST_DB with 11,000 words.	88.2% to 88.4%	-	-	-	-
Habash [31] - (2007)	They used ATB3-Train with 288,000 for training purpose and ATB3-Devtest with 52,000 words for testing purpose.	-	14.9%	4.8%	5.5%	2.2%
Shaalan [24] - (2009)	LDC's Arabic Treebank Part 2 v2.0, this corpus includes 501 stories collected from the Ummah Arabic News Text, with total number of 144,199 words. Training to testing ratio was 92.5% to 7.5% respectively.	-	17.31%	4.41%	12.16%	3.78%
Rashwan [28] - (2009)	TRN_DB_I and TRN_DB_II, with total number of words ≈ 750K and ≈ 2500K respectively. For testing purpose, author used TST_DB with total number of words ≈ 11K.	-	2.1%	-	-	-
Said [25] - (2013)	ATB corpus that contains around 350K works.	-	11.4%	3.6%	4.4%	1.6%
Bebah [13] - (2014)	The training carried out with 90% of a corpora consisting of 2,463,351 vowelized words, divided between NEMLAR corpus (460,000 words), (Tashkeela) corpus (780,000 words) and RDI corpus (1,223,351 words). The remaining 10% used testing phase.	-	21.11%	7.37%	9.93%	3.75
Schlippe [27] - (2008)	For training and testing purpose, author used two data sources, the diacritized LDC's Arabic Treebank as well as data provided by AppTek.	-	21.9%	4.7%	8.3%	1.9%

A number of papers utilized high training and testing ratios which negatively affect the certainty of the results as in [11] and [24], while others didn't mention the training and testing ratios as in [18], [20], [25] and [27].

Although some approaches which didn't consider the case ending concept yielded good diacritization accuracy, WER and DER, their results would have improved if they employed this concept in combination with their approaches, as in [10], [4], [18], [20], [26], [7] and [28].

2.3 Research Gap

Having reviewed a broad range of relevant literature, a conclusion can be drawn that the vast majority of the reviewed papers investigated their proposed approach on MSA. A single paper has investigated its proposed approach on dialectal Arabic [4], as the main challenge in dialectal text is the limited availability of dialectal corpora. Therefore, research gap has been identified in investigating the accuracy of the existing diacritization approaches when implemented to Dialectal Arabic.

Referring to Tables 2.1, 2.2 and 2.3, we can conclude that hybrid approach yield higher accuracy than Statistical approach and Rule-based approach. Thus, in this study, the main focus is to propose and implement a hybrid based approach that combines rule-based approach with statistical approach adapting the morphological analyzer along with maximum likelihood estimate and statistical n-gram for automatically retrieving the diacritic marks with accuracy higher than the state-of-the-art systems.

2.4 Summary

Arabic is a highly complex language, even for the Arabic native speakers. It is a very rich language in terms of morphology and syntax. In this chapter, we discussed many of the automatic diacritization techniques for undiacritized Arabic text. Techniques used in automatic diacritization can be divided mainly into three approaches, rule-based approach, statistical approach, and hybrid approach. It was noticed that hybrid approach yield higher accuracy than other approaches. Current automatic diacritization techniques still fall short of the desired outcome of near perfect diacritic restoration, especially the rule-based techniques, such as a lexical analyzer, morphological analyzer, and syntax analyzer. This is due to of the morphological complexity and the difficulty of keeping up a huge list of grammatical rules that maintain all the aspects of Arabic language and affect the diacritization especially on ending letters. Thus, the field is still an active research one and needs much work in both finding new approaches, and enhancing the old ones.

In this study, based on the literature review, a hybrid based algorithm for automatically retrieving the diacritic marks will be proposed; this algorithm will be based on a hybrid approach that combines the morphological analyzer along with maximum likelihood estimate and statistical n-gram to achieve the main research objective of this study, which is an improved approach for automatic diacritization.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

In this study, a hybrid algorithm which combines the statistical n-gram along with maximum likelihood estimate and morphological analyzer is adapted for solving the diacritization problem by predicting the diacritized version of undiacritized Arabic text.

3.2 Research Phases

This study consisted of four main phases as follows, a theoretical study of the current diacritization algorithm, design phase of the proposed algorithm, development phase of the proposed algorithm, and evaluation of the proposed algorithm. Figure 3.1 explain the four main phases of this study

.

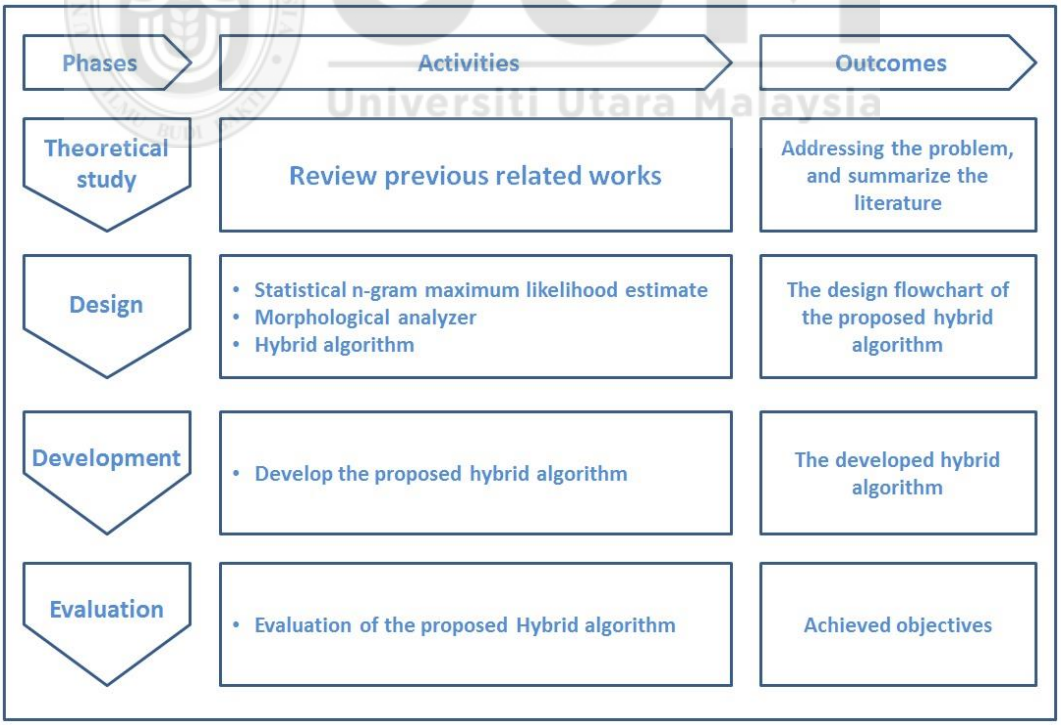

Figure 3.1 The four main phases of this study

3.2.1 Theoretical Study

The first step in this study is theoretical study. In this step, two main points were considered in analyzing the research problem. The first point was studying the state-of-the-art algorithms developed in order to restore the missing diacritic marks. The second point was studying the characteristics of the Arabic language. This information collected from the literature review conducted on different types of publications, such as journals, technical reports, conference proceedings and books, with focus on recent publications, in order to identify the limitation of each algorithm, research gap in this field, problem statement, research questions, research objectives and research scope.

3.2.2 Design Phase

In this phase, the design of the proposed algorithm was constructed. This design based on two main techniques as follows: statistical n-gram along with maximum likelihood estimate, and morphological analyzer. The following sub-section presents the design steps of the proposed algorithm. Merging the word, morpheme, and letter levels and their sub-models together into one platform in order to improve the automatic diacritization accuracy is the proposition of this research. Moreover, by utilizing the feature of the case-ending diacritization, which is, ignoring the diacritic mark on the last letter of the word, shows a significant error improvement. The reason for this remarkable improvement is that the Arabic language prohibits adding diacritic marks over some letters. For example, a fatha on "و" is phonetically prohibited. Also, it favors some diacritics over other letters like a tanween on "ا". Figure 3.2 illustrates the proposed algorithm for this study.

3.2.1.1 Word-level

In this type of model, four different models are used to re-introduce the missing diacritization marks for a certain word. They are as follows:

(i) Four-gram Model

In this model, the maximum-likelihood is applied; each undiacritized word in the test-set is replaced by the corresponding diacritized one that occurs most frequently in the training set, given the word history and the words next to the one to be diacritized based on the selected sub-model. This model has also been split into two sub-models:

1) Four-gram - right context

First, in this sub-model, the number of times each diacritized word occurs in the training set is counted, given the history-right content, of each word. Then, for each undiacritized word that appears on the test set, the search is done through all of the words with the same undiacritized structure and given the same word history. The diacritized word with the highest occurrence in the table is chosen.

More formally, in this case, the diacritizer chooses $word_i^d$ as the diacritized form of the input word represented by $word_i^u$ considering the previous history of that word represented by $word_{i-1}^u$, $word_{i-2}^u$ and $word_{i-3}^u$, as per Equation 3.1

$$word_i^d = \max \mathrm{p}\left(word_i^d \middle| word_i^u, word_{i-1}^u, word_{i-2}^u, word_{i-3}^u\right) \quad \text{Eq. 3.1}$$

Where $word_i^d$ represents the selected diacritized form of the i^{th} undiacritized word represented by $word_i^u$, given the previous history of the word represented by $word_{i-1}^u$, $word_{i-2}^u$ and $word_{i-3}^u$. In case the word was not found to have a Four-gram - right context, the system defers to the next sub-model, Four-gram - left context.

2) Four-gram - left context

As a further measure to improve the word level accuracy and diacritization level accuracy, and similar to the previous sub model (Four-gram - right context) this sub-model was adapted to consider the words next to the given one to be diacritized - left context.

In this case, the diacritizer chooses $word_i^d$ as the diacritized form of the input word $word_i^u$ and the words next to the given one to be diacritized $word_{i+1}^u$, $word_{i+2}^u$ and $word_{i+3}^u$ as per Equation 3.2

$$word_i^d = \max \mathrm{p}\left(word_i^d \middle| word_{i+3}^u, word_{i+2}^u, word_{i+1}^u, word_i^u\right) \quad \text{Eq. 3.2}$$

Where $word_i^d$ represents the selected diacritized form of the i^{th} undiacritized word represented by $word_i^u$, given the words next to $word_i^u$ represented by $word_{i+1}^u$, $word_{i+2}^u$ and $word_{i+3}^u$. In case the word is not found in any of the Four-gram models, the system defers to Tri-gram models.

(ii) Tri-gram Model

Similar to the Four-gram model, and for the improvement of the diacritization accuracy, context continues to be used for diacritizing a certain word, but less than previously. This model has also been split into two sub-models, as follows:

1) Tri-gram - right context

In this model, a word history less than the history utilized for the Four-gram - right context sub-model is considered. More formally, in this case, the diacritizer chooses $word_i^d$ as the diacritized form of the input word represented by $word_i^u$ considering the previous history of that word represented by $word_{i-1}^u$ and $word_{i-2}^u$, as per Equation 3.3

$$word_i^d = \max p\left(word_i^d \middle| word_i^u, word_{i-1}^u, word_{i-2}^u\right) \quad \text{Eq. 3.3}$$

Where $word_i^d$ represents the selected diacritized form of the i^{th} undiacritized word represented by $word_i^u$, given the previous history of the word represented by $word_{i-1}^u$ and $word_{i-2}^u$. In case the word is not found to have a Tri-gram - right context, the system defers to the next sub-model, Tri-gram - left context.

2) Tri-gram - left context

As a further measure to improve word level accuracy and diacritization level accuracy, and similar to the previous sub model Tri-gram - right context, this sub model was adapted to consider the words next to the given one to be diacritized - left context.

In this case, the diacritizer chooses $word_i^d$ as the diacritized form of the input word represented by $word_i^u$ and the words next to the given one to be diacritized represented by $word_{i+1}^u$ and $word_{i+2}^u$, as per Equation 3.4

$$word_i^d = \max p\left(word_i^d \middle| word_{i+2}^u, word_{i+1}^u, word_i^u\right) \quad \text{Eq. 3.4}$$

Where $word_i^d$ represents the selected diacritized form of the i^{th} undiacritized word represented by $word_i^u$, given the words next to $word_i^u$ represented by $word_{i+1}^u$ and $word_{i+2}^u$. In case of the word was not found in any of the Tri-gram models, the system defers to Bigram models.

(iii) Bigram Model

In this model, context continues to be used for diacritizing a certain word but less than previously. This model has also been split into two sub-models, as follows

1) Bigram - right context

In this case, the diacritizer chooses $word_i^d$ as the diacritized form of the input word represented by $word_i^u$ and the previous history of that word represented by $word_{i-1}^u$, as per Equation 3.5

$$word_i^d = \max p(word_i^d | word_i^u, word_{i-1}^u) \quad \text{Eq. 3.5}$$

Where $word_i^d$ represent the selected diacritized form of the i^{th} undiacritized word represented by $word_i^u$, given the previous history of the word represented by $word_{i-1}^u$.

In case the word was not found to have a Bigram right-context, the system defers to the next sub-model, Bigram - left context.

2) Bigram - left context

In this case, the diacritizer chooses $word_i^d$ as the diacritized form of the input word represented by $word_i^u$ and the word next to the given one to be diacritized represented by $word_{i+1}^u$, as per Equation 3.6

$$word_i^d = \max p(word_i^d | word_{i+1}^u, word_i^u) \quad \text{Eq. 3.6}$$

Where $word_i^d$ represents the selected diacritized form of the i^{th} undiacritized word represented by $word_i^u$, given the word next to $word_i^u$ represented by $word_{i+1}^u$.

In the case of the word was not found in any of the Bigram models, the system defers to the Baseline model - Unigram.

(iv) Unigram - Baseline Model

In this model, each undiacritized word in the test set is replaced by the corresponding diacritized one that occurs most frequently in the training set, based on Equation 3.7 [36]

$$word_i^d = \max p(word_i^d | word_i^u) \quad \text{Eq. 3.7}$$

Where $word_i^d$ represents the selected diacritized form of the word, and the i^{th} word in the undiacritized Arabic text is represented by $word_i^u$.

Figure 3.2 illustrates the Automatic diacritization based on the word-level.

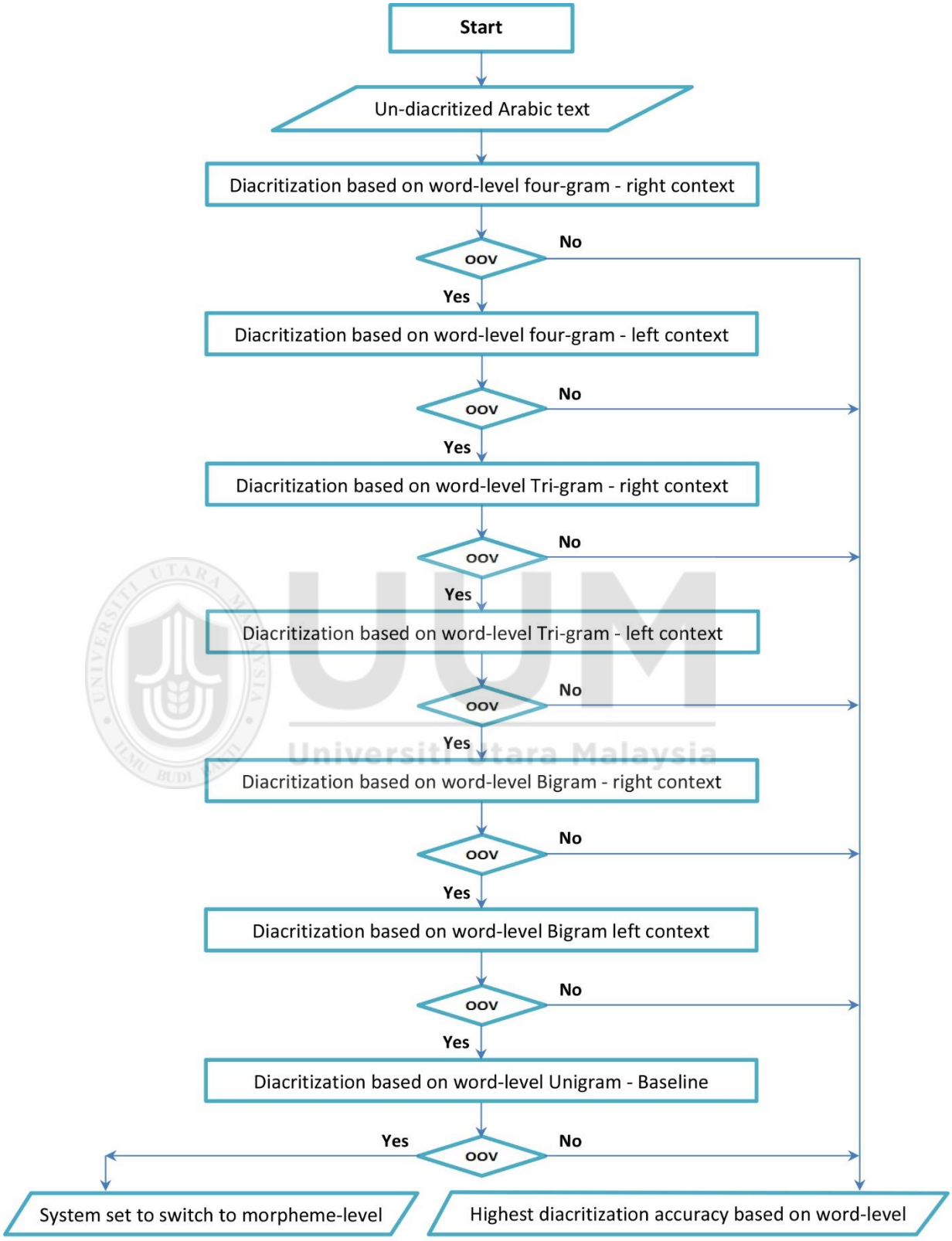

Figure 3.2 Illustrate the Automatic diacritization based on the word-level.

Due to the relatively small size of our training corpus, and the highly inflected nature of the Arabic language [7], about 6% of the undiacritized words in the test set do not occur in the training set. Concerning the OOV that have no statistics associated with them, some of the previous research works were simply copying the undiacritized form of the word into the diacritized version. In this study, in the case of OOV words, algorithm set to switches to the next diacritization level, morphemes-level.

3.2.1.2 Morphemes-level

The morphological structure of any valid Arabic word consists of zero (or more) prefixes, word root and zero (or more) suffixes. Using the Buckwalter Transliteration, and as per the following two different groups of affixes, the morphological structure of the selected word can be achieved within two main steps.

Basic affixes

Prefixes {Al, b, f, k, l, ll, w}

Suffixes {h, hA, hm, hmA, hn, k, km, kmA, kn, nA}

Compound affixes

Prefixes {Al, b, bAl, f, fAl, fb, fbAl, fk, fl, fll, k, kAl, lll, w, wAl, wb, wbAl, wk, wkAl, wl, wll}

Suffixes {h, hA, hm, hmA, hn, k, km, kmA, kn, nA}

The first step is the stripping phase of the prefix. In this phase, the prefix with the largest matching number of letters is selected.

The second step is the stripping phase of the suffix. In this phase, the suffix with the largest matching number of letters is selected.

The remaining letters, after stripping the prefix and suffix, are selected as the word root.

For example, based on the above steps the morphological structure of the valid Arabic word "وكالسماوات" will be "وكالـ" as prefix, "وات" as suffix and "سما" as word root.

Based on that, in this type (morphemes level) and similar to what was done in the first type (word level) the diacritizer then considers each segment that has been separated in the morphological structure as a word, as follows

<s> وكال سما وات </s>

Hence, the four different models adapted to the word level are used to re-introduce the missing diacritization marks for each segment in the morphological structure, as follows

(i) Four-gram Model

The approach employed in this model is similar to the approaches employed in the previous model, Four-gram word level. More contexts were used for diacritizing a certain segment. This model has been split also to two sub-models as follows

1) Four-gram - right context

In this model history-right context - of the given segment was mainly considered. In this case, the diacritizer chooses segment_i^d as the diacritized form of the input segment represented by $segment_i^u$ considering the previous history of that segment represented by $segment_{i-1}^u$, $segment_{i-2}^u$ and $segment_{i-3}^u$, as per Equation 3.8

$$\text{segment}_i^d = \max \text{p}\left(segment_i^d \middle| segment_i^u, segment_{i-1}^u, segment_{i-2}^u, segment_{i-3}^u\right) \quad \text{Eq. 3.8}$$

In case the segment was not found to have a Four-gram - right context, the system defers to the next sub-model, Four-gram - left context.

2) Four-gram - left context

In this sub model, segments next to the given one to be diacritized - left context – is considered. In this case, the diacritizer chooses $segment_i^d$ as the diacritized form of the input segment represented by $segment_i^u$ and the segments next to the given one to be diacritized represented by $segment_{i+1}^u$, $segment_{i+2}^u$ and $segment_{i+3}^u$ as per Equation 3.9

$$segment_i^d = \max \mathrm{p}\left(segment_i^d \middle| segment_{i+3}^u, segment_{i+2}^u, segment_{i+1}^u, segment_i^u\right) \quad \text{Eq. 3.9}$$

In case the segment was not found to have a Four-gram - left context, the system defers to the next sub-model, Trigram - right context.

(ii) Tri-gram Model

Similar to what has been done in the Four-gram model; context continues to be used for diacritizing a certain segment but less than before. This model has also been split into two sub-models, as follows

1) Tri-gram - right context

In this model, the history-right context, of the given segment is considered. In this case, the diacritizer chooses $segment_{\mathrm{i}}^{\mathrm{d}}$ as the diacritized form of the input segment represented by $segment_i^u$, considering the previous history of that segment represented by $segment_{i-1}^u$ and $segment_{i-2}^u$, as per Equation 3.10

$$segment_i^d = \max \mathrm{p}\left(segment_i^d \middle| segment_i^u, segment_{i-1}^u, segment_{i-2}^u\right) \quad \text{Eq. 3.10}$$

In case the segment was not found to have a Tri-gram - right context, the system defers to the next sub-model, Tri-gram - left context.

2) Tri-gram - left context

In this case, the diacritizer considers the segments next to the one to be diacritized. The diacritizer chooses $segment_i^{\mathrm{d}}$ as the diacritized form of the input segment represented by $segment_i^u$ and the segments next to the given one to be diacritized represented by $segment_{i+1}^u$ and $segment_{i+2}^u$, as per Equation 3.11

$$segment_i^d = \max \mathrm{p}\left(segment_i^d \middle| segment_{i+2}^u, segment_{i+1}^u, segment_i^u\right) \quad \text{Eq. 3.11}$$

In case the segment was not found to have a Tri-gram - left context, the system defers to the next sub-model, Bigram - right context.

(iii) Bigram Model

Bigram model is adapted to keep using the context for diacritizing a certain segment. This model is split into two sub-models, as follows

1) Bigram - right context

In this case the diacritizer will choose the corresponding diacritized segment that occurs most frequently in the training set, given the previous history of that segment, as per Equation 3.12

$$Segment_i^d = \max \mathrm{p}\left(segment_i^d \middle| segment_i^u, segment_{i-1}^u\right) \quad \text{Eq. 3.12}$$

In case the segment was not found to have a Bigram - right context, the system defers to the next sub-model, Bigram - left context.

2) Bigram - left context

The diacritizer chooses the corresponding diacritized segment that occurs most frequently in the training set, given the segment next to the one to be diacritized, as per Equation 3.13

$$Segment_i^d = \max \mathrm{p}\left(segment_i^d \middle| segment_{i+1}^u, segment_i^u\right) \quad \text{Eq. 3.13}$$

In case the segment is not found to have a Bigram - left context, the system defers to the next model, Baseline - Unigram.

(iv) Unigram - Baseline Model

In this approach, the same simple maximum-likelihood-unigram-baseline is be applied; each undiacritized segment in the test set is replaced by the corresponding diacritized one that occurs most frequently in the training set, as per Equation 3.14 [36]

$$segment_i^d = \max \mathrm{p}\left(segment_i^d \middle| segment_i^u\right) \quad \text{Eq. 3.14}$$

In case the segment is not found to have a Unigram, the system defers to the next level, Letter level.

Figure 3.3 illustrates the Automatic diacritization based on the morpheme-level.

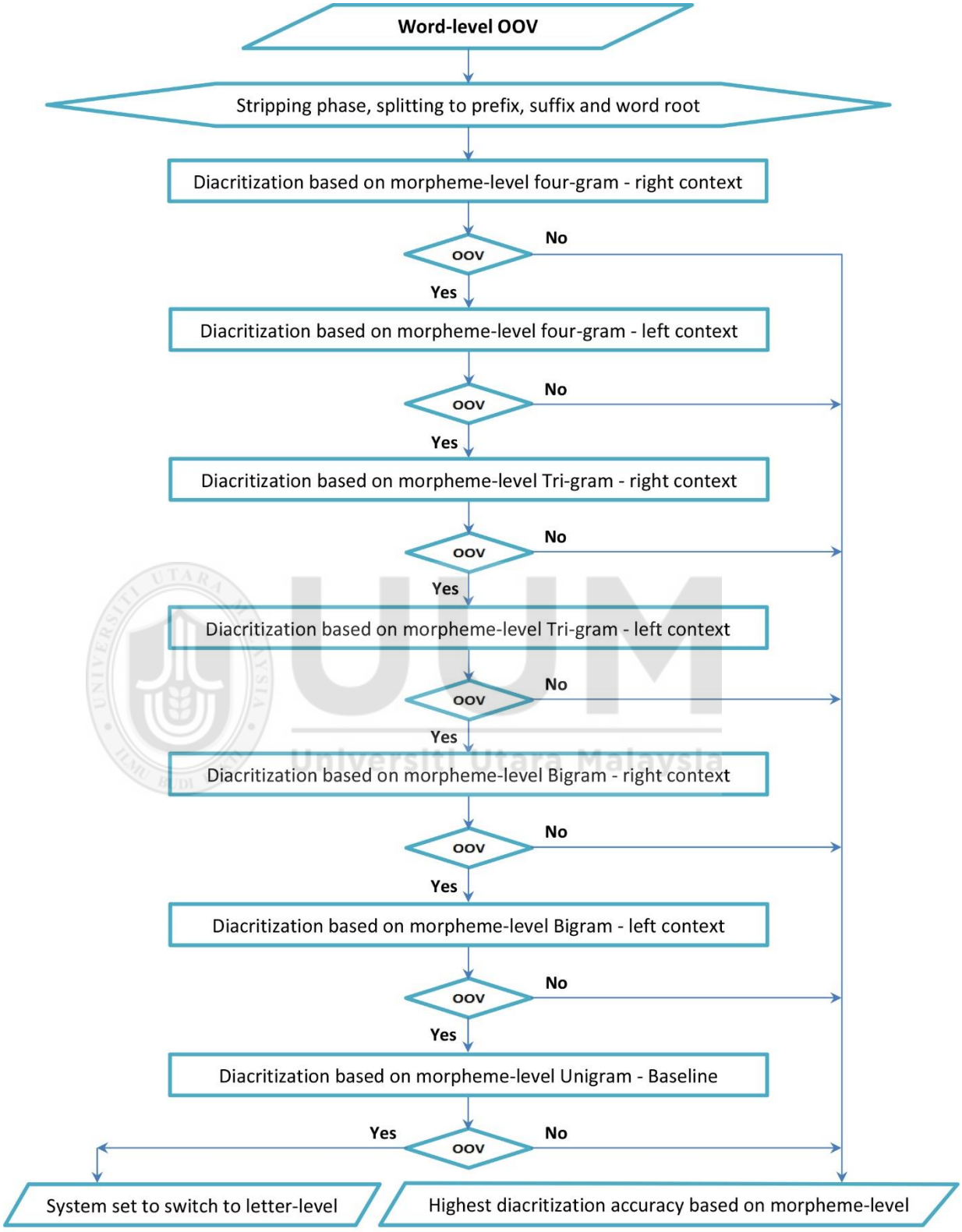

Figure 3.3 Illustrate the Automatic diacritization based on the morpheme-level.

3.2.1.3 Letter-level

In this diacritization level, the diacritizer will split each word or segment into a set of letters. Based on that, in this type (letter level) and similar to what was done in the previous types (word level and morphemes level) the diacritizer considers each letter in the word as a separated word, as follows

<s> و ك ا ل س م ا و ا ت </s>

Hence, the four different models and sub-models adapted in the previous types (word level and morphemes level) is used to re-introduce the missing diacritization marks for each letter in the word, as follows

(i) Four-gram Model

The approach employed in this model is similar to the approach employed at the word level and morphemes level. More contexts are used for diacritizing a certain letter. This model has also been split into two sub-models as follows

1) Four-gram - right context

In this model history-right context, of the given letter is mainly considered. In this case, the diacritizer chooses $letter_i^d$ as the diacritized form of the input letter represented by $letter_i^u$ considering the previous history of that letter represented by $letter_{i-1}^u$, $letter_{i-2}^u$ and $letter_{i-3}^u$, as per Equation 3.15

$$letter_i^{\boldsymbol{d}} = \max \mathrm{p}\left(letter_i^d \middle| letter_i^u, letter_{i-1}^u, letter_{i-2}^u, letter_{i-3}^u\right) \quad \text{Eq. 3.15}$$

In case the letter was not found to have a Four-gram - right context, the system defers to the next sub-model, Four-gram - left context.

2) Four-gram - left context

In this sub model, letters next to the given one to be diacritized - left context – is considered. In this case, the diacritizer chooses $letter_i^d$ as the diacritized form of the input letter represented by $letter_i^u$ and the letters next to the given one to be diacritized represented by $letter_{i+1}^u$, $letter_{i+2}^u$ and $letter_{i+3}^u$ as per Equation 3.16

$$letter_i^d = \max p\left(letter_i^d \middle| letter_{i+3}^u, letter_{i+2}^u, letter_{i+1}^u, letter_i^u\right) \quad \text{Eq. 3.16}$$

In case the letter was not found to have a Four-gram - left context, the system defers to the next sub-model, Tri-gram - right context.

(ii) Tri-gram Model

Similar to what has been done in the Four-gram model, context continues to be used for diacritizing a certain letter, but with less emphasis than before. This model has also been split into two sub-models, as follows

1) Tri-gram - right context

In this case, the diacritizer chooses $letter_i^d$ as the diacritized form of the input letter represented by $letter_i^u$, considering the previous history of that letter represented by $letter_{i-1}^u$ and $letter_{i-2}^u$, as per Equation 3.17

$$letter_i^d = \max p\left(letter_i^d \middle| letter_i^u, letter_{i-1}^u, letter_{i-2}^u\right) \quad \text{Eq. 3.17}$$

In case the letter was not found to have a Tri-gram - right context, the system then defers to the next sub-model, Tri-gram - left context.

2) Tri-gram - left context

In this case, the diacritizer chooses $letter_{\mathrm{i}}^{\mathrm{d}}$ as the diacritized form of the input letter represented by $letter_i^u$ and the letters next to the given one to be diacritized represented by $letter_{i+1}^u$ and $letter_{i+2}^u$, as per Equation 3.18

$$letter_i^d = \max \mathrm{p}\left(letter_i^d \middle| letter_{i+2}^u, letter_{i+1}^u, letter_i^u\right) \quad \text{Eq. 3.18}$$

In case the letter is not found to have a Trigram - left context, the system then defers to the next sub-model, Bigram - right context.

(iii) Bigram Model

Bigram model is adapted to keep using the context for diacritizing a certain letter. This model is split into two sub-models, as follows

1) Bigram - right context

In this case, the diacritizer chooses the corresponding diacritized letter that occurs most frequently in the training set, given the previous history of that letter, as per Equation 3.19

$$letter_i^d = \max \mathrm{p}\left(letter_i^d \middle| letter_i^u, letter_{i-1}^u\right) \quad \text{Eq. 3.19}$$

In case the letter was not found to have a Bigram - right context, the system then defers to the next sub-model, Bigram - left context.

2) Bigram - left context

Similar to the previous sub model Bigram - right context, this sub-model has been adapted and considers the letters next to the given one to be diacritized - left context. In this case, the diacritizer chooses the corresponding diacritized letter that occurs most

frequently in the training set, given the letter next to the one to be diacritized, as per Equation 3.20

$$letter_i^d = \max \mathrm{p}\left(letter_i^d \middle| letter_{i+1}^u, letter_i^u\right) \quad \text{Eq. 3.20}$$

In case the letter is not found to have a Bigram - right context, the system then defers to the next model, Unigram - Baseline model.

(iv) Unigram - Baseline Model

In this approach, the same simple maximum-likelihood-unigram-baseline is applied; each undiacritized letter in the test set is replaced by the corresponding diacritized one that occurs most frequently in the training set, as per Equation 3.21 [36]

$$letter_i^d = \max \mathrm{p}\left(letter_i^d \middle| letter_i^u\right) \quad \text{Eq. 3.21}$$

Figure 3.4 illustrates the Automatic diacritization based on the letter-level.

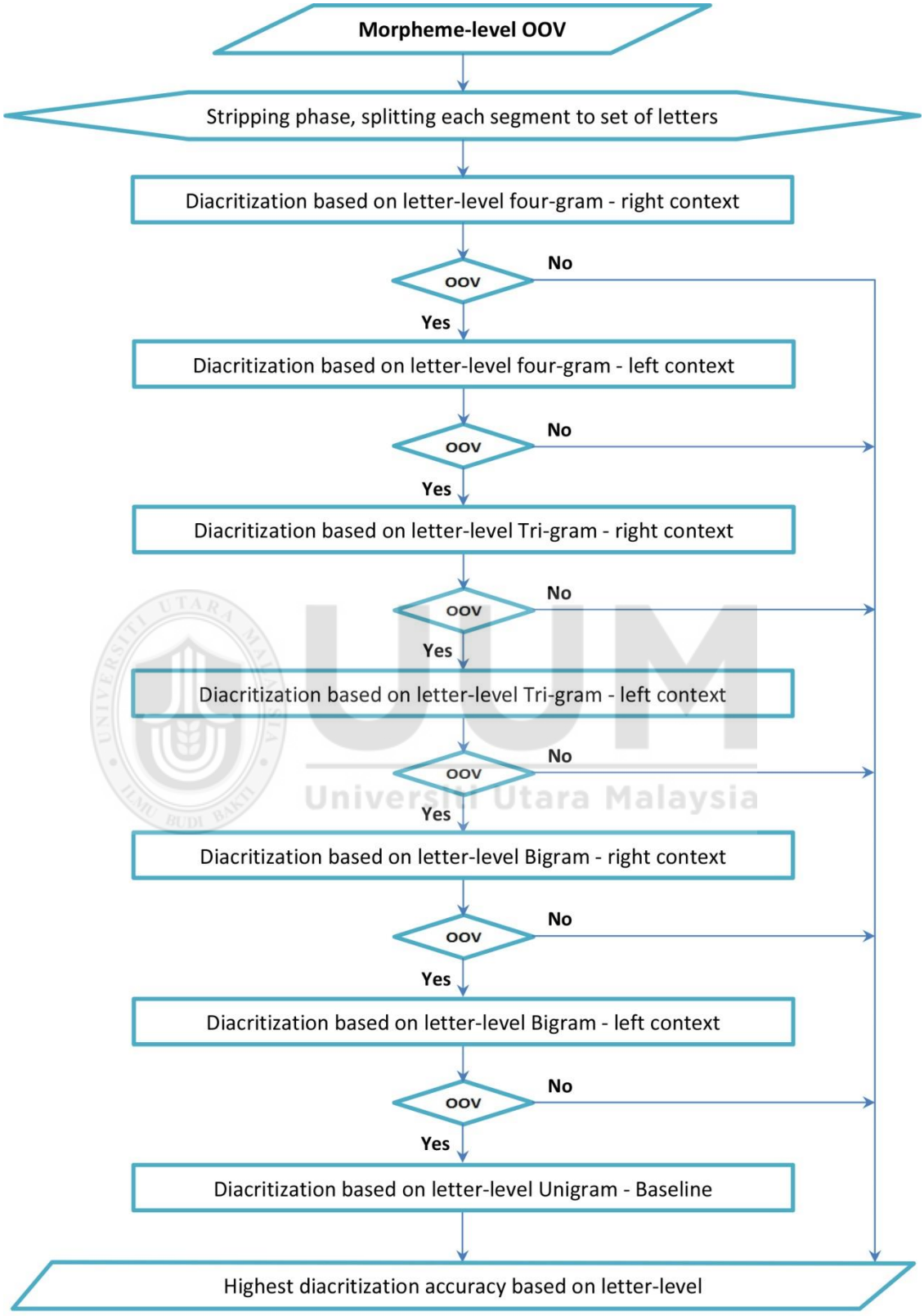

Figure 3.4 Illustrate the Automatic diacritization based on the letter-level.

Figure 3.5 illustrates the proposed Automatic diacritization algorithm based on based the word level, morpheme level and letter level.

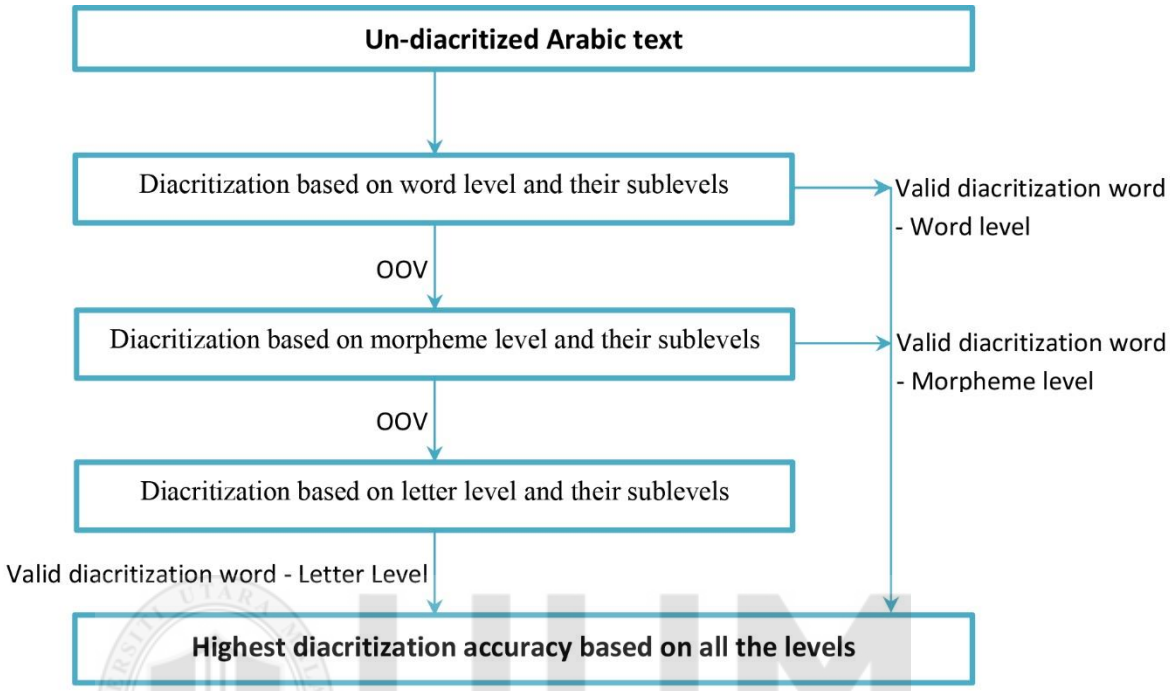

Figure 3.5 Proposed algorithm for this study.

In this algorithm, the diacritization of the given word will be based on three levels respectively, word level and their sublevels; in this level, statistical n-gram along with maximum likely hood estimate techniques will be applied. In order to overcome the OOV resulted from the word level, algorithm will switch to morpheme level and their sublevels. In this level, morphological analyzer technique will be applied first in order to strip the selected word into prefix, root and suffix, and then each segment will be diacritized based on statistical n-gram along with maximum likely hood estimate techniques. In order to overcome the OOV resulted from the morpheme level, algorithm will switch to letter level and their sublevels; in this level, each word will be split in to

set of letters, and each letter will be diacritized based on statistical n-gram along with maximum likely hood estimate techniques.

Pseudo code for automatic diacritization for Arabic words.

```
if (diacritization(word)==true) //based on word level and all sub-levels
       {print diacritization(word);
       //Based on statistical n-gram along with maximum likely hood estimate
       exit;}

strip(word);
//Stripping the word to Prefix, Root and Suffix using morphological analyzer.

if (diacritization(prefix)==true)
       print diacritization(prefix);
       //Based on statistical n-gram along with maximum likely hood estimate
else
       {split(prefix);
       for(int i=0;i<length(prefix);i++)
              diacritization(letter);}
       //Based on statistical n-gram along with maximum likely hood estimate

if (diacritization(root)==true)
       print diacritization(root);
       //Based on statistical n-gram along with maximum likely hood estimate
else
       {split(root);
       for(int i=0;i<length(root);i++)
              diacritization(letter);}
       //Based on statistical n-gram along with maximum likely hood estimate

if (diacritization(suffix)==true)
       print diacritization(suffix);
       //Based on statistical n-gram along with maximum likely hood estimate
else
       {split(suffix);
       for(int i=0;i<length(suffix);i++)
              diacritization(letter);}
       //Based on statistical n-gram along with maximum likely hood estimate
```

3.2.3 Development Phase Hybrid Algorithm

In this phase, a software prototype has been developed in order to measure the diacritization accuracy, WER and DER for the hybrid algorithm. The prototype was developed using the following:

- Python 3.4.1 programming language that was used to develop the prototype.
- Free source code editor notepad++ that was used to build the database.

3.2.4 Evaluation

Figure 3.6 shows the whole evaluation process that has been conducted to evaluate statistical n-gram technique along with maximum likelihood estimate technique, and morphological analyzer technique. The evaluation process has been performed individually for each technique, and then evaluating the hybrid algorithm.

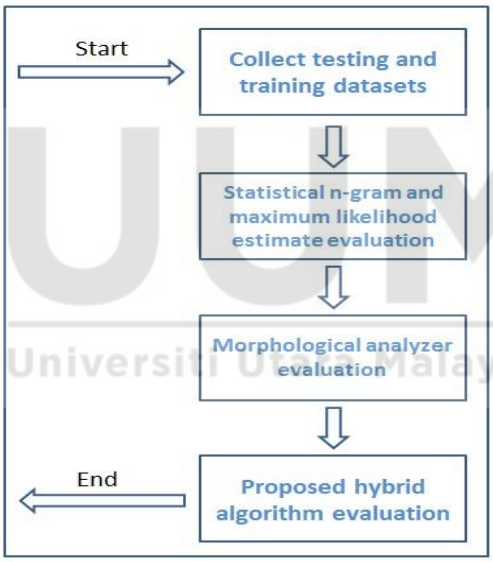

Figure 3.6 The whole evaluation process.

3.2.4.1 Data Collection

Based on the literature review, the best WER and DER were 3.54% and 1.28% achieved by G. Abandah [2], and 3.1% and 1.2% achieved by M. Rashwan [30]. In order to unify the compression criteria, the same corpora which were utilized by the researchers were utilized in this study. G. Abandah [2] used Tashkeela corpus [32], while M. Rashwan [30] used LDC's Arabic Treebank-Part 3 v1.0 [33].

Tashkeela is a classical Arabic text vocalized corpus, collected from Islamic religious books using an automatic web crawling methods. This corpus contains over 73 million words fully diacritized. LDC's Arabic Treebank-Part 3 v1.0 is an Arabic text vocalized corpus, consists of 600 documents (≈340K words) from AnNahar newspaper.

For dialectal Arabic, CallHome dialectal Arabic corpus of telephone speech [34] has been used, it is an Arabic corpus of telephone speech collected and transcribed by the Linguistic Data Consortium primarily in support of the project on Large Vocabulary Conversational Speech Recognition (LVCSR), sponsored by the U.S. Department of Defense. This release of the CallHome Arabic corpus consists of 120 unscripted telephone conversations between native speakers of Egyptian Colloquial Arabic (ECA), the spoken variety of Arabic found in Egypt. The dialect of ECA that this corpus represents is Cairene Arabic. The transcripts cover a contiguous 5 or 10- minute segment taken from a recorded conversation lasting up to 30 minutes.

3.2.4.2 Experimental Design

In the experimental design, the main goal is to evaluate the proposed algorithm in comparison with other algorithms. This goal is related to the research objectives of this study.

(v) Statistical n-gram and maximum likelihood estimate evaluation

Evaluation of statistical n-gram along maximum likelihood estimate techniques have been conducted, the goals of the enhancement technique has been achieved by comparing the output of the enhancement technique with the best output achieved based

on the literature review. Moreover, in order to unify the compression criteria, the experiments used the same training and testing dataset explained in section 3.2.4.a.

(i) Morphological Analyzer evaluation

Morphological analyzer technique cannot be evaluated individually, as it could not be employed as a stand-alone technique to diacritize sentence, word and letter. Morphological analyzer usually used to overcome the OOV during the diacritization process. In this case, morphological analyzer factorize the OOV words into its possible morphological components (prefix, root and suffix), and then diacritize each segment separately using statistical n-gram and maximum likelihood estimate.

(ii) Hybrid Algorithm

Evaluation of algorithm has been conducted, the goals of the proposed algorithm has been achieved by comparing the output of the algorithm with the best output achieved based on the literature review. Moreover, in order to unify the compression criteria, the experiments used the same training and testing dataset explained in section 3.2.4.a.

3.2.4.3 Measurement

Measures that are utilized in order to evaluate the performance of the systems will be the WER [1], [2], [3], [4], [5], [6] and the DER [1], [2], [3], [4], [5], [6]. Based on the previous works, WER is the percentage of the words that are diacritized incorrectly (one letter at least has an incorrect diacritic mark), while the DER is the percentage of the letters that are diacritized incorrectly. WER cannot be utilized as the only measure for diacritization accuracy, as it might provide inaccurate information about the system performance. For example, if there is a word diacritized incorrectly because of one

diacritization mark, in this case, WER and DER will be equal to one, while if we have one word diacritized incorrectly because of four diacritization marks, WER will be equal to one and DER will be equal to four.

Therefore, both measures WER and DER give more precise indication of the accuracy of the approach in use.

In this study, the following performance measures are utilized

i. **WER1 -** The percentage of the words that are diacritized incorrectly (considering the diacritic mark of the last letter).
ii. **WER2 -** The percentage of the words that are diacritized incorrectly (ignoring the diacritic mark of the last letter).
iii. **DER1** - The percentage of the letters diacritized incorrectly (considering the diacritic mark of the last letter).
iv. **DER2** - The percentage of the letters diacritized incorrectly (ignoring the diacritic mark of the last letter).

3.2.4.4 Statistical Test

In addition to the three main performance measures, this research conducted a statistical test to show if the diacritization accuracy significantly changed or not. Due to the impracticality of presenting the results for 50 dataset, the prediction of diacritization accuracy, WER and DER will be analyzed statistically. The statistical mean (μ), standard deviation (σ), and maximum and minimum values of diacritization accuracy, WER and DER were calculated to evaluate and summarize the effect of dataset change on diacritization accuracy, WER and DER, using the following equation.

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} Xi \qquad \text{Eq. 3.22}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Xi - \mu)^2} \qquad \text{Eq. 3.23}$$

Where ***X*** is the accuracy rate, WER, and DER. ***N*** is the number of samples.

According to the Empirical rule for normal data [35], approximately 99.7% of data lies within $\mu \pm 3\,\sigma$.

3.3 Summary

In this chapter, the methodology that has been followed in this study has been proposed. This methodology has been used in developing the most successful diacritization techniques. The proposed algorithm tested on different datasets, and compared with other well-known diacritization systems. The initial results of the proposed algorithm are very promising.

CHAPTER FOUR
EXPERIMENTAL RESULTS

At the beginning of this chapter, the proposed algorithm was presented, and then the specification of the corpora used for our experiments were discussed, the results after applying the proposed algorithm on MSA corpora were displayed. Moreover, a comparison with recent related works was performed. Finally, the results of applying the proposed algorithm on Dialectal Arabic corpus were displayed.

4.1 Training and Testing Datasets (Corpora)

Based on the literature review, the highest WER and DER were 3.54% and 1.28% achieved by G. Abandah [2], and 3.1% and 1.2% achieved by M. Rashwan [30]. In order to unify the compression criteria, the same corpora which were utilized by the researchers were utilized in this study. G. Abandah [2] used 88% of Tashkeela corpus [32] as training dataset, while the rest used as testing dataset. M. Rashwan [30] used 85% of LDC's Arabic Treebank-Part 3 v1.0 [33] as training dataset, while the rest used as testing dataset.

Tashkeela is a classical Arabic text vocalized corpus, collected from Islamic religious books using an automatic web crawling methods. This corpus contains over 73 million words fully diacritized. LDC's Arabic Treebank-Part 3 v1.0 is an Arabic text vocalized corpus, consists of 600 documents (≈340K words) from AnNahar newspaper.

For dialectal Arabic, CallHome dialectal Arabic corpus of telephone speech [34] has been used, it is an Arabic corpus of telephone speech collected and transcribed by the Linguistic Data Consortium primarily in support of the project on Large Vocabulary

Conversational Speech Recognition (LVCSR), sponsored by the U.S. Department of Defense. This release of the CallHome Arabic corpus consists of 120 unscripted telephone conversations between native speakers of Egyptian Colloquial Arabic (ECA), the spoken variety of Arabic found in Egypt. The dialect of ECA that this corpus represents is Cairene Arabic. The transcripts cover a contiguous 5 or 10- minute segment taken from a recorded conversation lasting up to 30 minutes.

4.2 Results for MSA

The proposed algorithm in chapter three yielded the results shown in Table 4.1 when applied to Tashkeela and LDC's Arabic Treebank. Having applied the word level Four-gram model and it's sub-models yielded the accuracy of 91.1%, WER of 8.9% and DER of 5.7% when applied to Tashkeela, and the accuracy of 89.9%, WER of 10.1% and DER of 6.9% when applied to LDC's Arabic Treebank. Having applied the Morpheme-level Four-gram model and it's sub-models on the Word-level OOV, the proposed algorithm yielded the accuracy of 93.7%, WER of 6.3% and DER of 3.9% when applied to Tashkeela, and the accuracy of 92.1%, WER of 7.9% and DER of 4.7% when applied to LDC's Arabic Treebank. Having applied the Letter-level Four-gram model and it's sub-models on the Morpheme-level OOV, the proposed algorithm yielded the accuracy of 95.9%, WER of 4.1% and DER of 3.1% when applied to Tashkeela, and the accuracy of 94.8%, WER of 5.2% and DER of 3.6% when applied to LDC's Arabic Treebank. However, by considering the case ending, the proposed algorithm yielded the accuracy of 97.9%, WER of 2.1% and DER of 1.1% when applied to Tashkeela, and the accuracy of 97.1%, WER of 2.9% and DER of 1.2% when applied to LDC's Arabic Treebank.

Table 4.1

Results of applying the proposed algorithm on MSA corpora.

	Tashkeela			LDC's Arabic Treebank		
	Accuracy	**WER**	**DER**	**Accuracy**	**WER**	**DER**
Word-level n-gram	91.1%	8.9%	5.7%	89.9%	10.1%	6.9%
Morpheme-level n-gram	93.7%	6.3%	3.9%	92.1%	7.9%	4.7%
Letter-level n-gram	95.9%	4.1%	3.1%	94.8%	5.2%	3.6%
By considering the case ending	97.9%	2.1%	1.1%	97.1%	2.9%	1.2%

4.3 Comparison with Other Methods

Based on Table 4.2, the proposed algorithm demonstrated a good performance when applied to Tashkeela and LDC's Arabic Treebank-Part 3 v1.0, as the comparison results with G. Abandah [2] and M. Rashwan [30] were in favor of the proposed algorithm.

Table 4.2

Comparisons between the proposed algorithm and other algorithms

	Tashkeela			LDC's Arabic Treebank		
	Accuracy	**WER**	**DER**	**Accuracy**	**WER**	**DER**
G. Abandah [2]	96.46%	3.54%	1.28%	-	-	-
M. Rashwan [30]	-	-	-	96.9%	3.1%	1.2%
Proposed algorithm	**97.9%**	**2.1%**	**1.1%**	**97.1%**	**2.9%**	**1.2%**

4.4 Results for dialectal Arabic

The proposed algorithm in chapter three yielded the results shown in Table 4.3 when applied to CallHome dialectal Arabic corpus of telephone speech [34]. Having applied the word level Four-gram model and its sub-models yielded the accuracy of 82.5%, WER of 17.5% and DER of 13.2%. Having applied the Morpheme-level Four-gram model and it's sub-models on the Word-level OOV, the proposed algorithm yielded the accuracy of 83.9%, WER of 16.1% and DER of 11.8%. Having applied the Letter-level

Four-gram model and it's sub-models on the Morpheme-level OOV, the proposed algorithm yielded the accuracy of 86.1%, WER of 13.9% and DER of 8.9%. However, by considering the case ending, the proposed algorithm yielded the accuracy of 88.8%, WER of 11.2% and DER of 6.1%.

Table 4.3

Results of applying the proposed algorithm on CallHome dialectal Arabic corpus

	CallHome dialectal Arabic corpus of telephone speech		
	Accuracy	**WER**	**DER**
Word-level n-gram	82.5%	17.5%	13.2%
Morpheme-level n-gram	83.9%	16.1%	11.8%
Letter-level n-gram	86.1%	13.9%	8.9%
By considering the case ending	88.8%	11.2%	6.1%

To enhance the results reported after applying the proposed algorithm on CallHome dialectal Arabic corpus of telephone speech, several configurations were also investigated. The best configuration was the diacritization through dialectal Arabic corpus word-level, MSA corpus word-level, dialectal Arabic corpus morpheme-level, MSA corpus morpheme-level, then dialectal Arabic corpus letter-level, with consideration of sub-models for each one. The best reported results were a WER of 9.7% and DER of 4.9%. When case ending is ignored, the system resulted in a WER and DER of 8.2% and 3.7% respectively. Table 4.4 present the best reported accuracy, WER and DER for Dialectal Arabic.

Table 4.4

Best reported accuracy, WER and DER for CallHome Dialectal Arabic.

	CallHome dialectal corpus AND Tashkeela corpus		
	Accuracy	**WER**	**DER**
Proposed algorithm without considering the case ending	90.3%	9.7%	4.9%
Proposed algorithm with considering the case ending	91.8%	8.2%	3.7%

The reported results for CallHome dialectal Arabic corpus are far away from the same algorithm applied to MSA corpus because of the limited amount of dialectal Arabic data used as training set.

4.5 Statistical Test

In order to statistically present the diacritization accuracy, WER and DER, testing corpus was split into 25 datasets. Due to the impracticality of presenting the results for 25 datasets, the diacritization accuracy, WER and DER will be analyzed statistically. This statistical test has been conducted on the MSA corpus Tashkeela, with 88% as training dataset, and 12% as testing dataset. Table 4.5 represents part of the diacritization accuracy, WER and DER after splitting the testing dataset.

Table 4.5

Diacritization accuracy, WER and DER for 10 datasets of Tashkeela corpus.

	Dataset1	**Dataset2**	**Dataset3**	**Dataset4**	**Dataset5**	**Dataset6**	**Dataset7**	**Dataset8**	**Dataset9**	**Dataset10**
Accuracy %	97.1	96.1	96.8	97.4	97.8	98.3	97.5	96.5	97.4	96.9
WER %	2.9	3.9	3.2	2.6	2.2	1.7	2.5	3.5	2.6	3.1
DER %	1.4	1.9	1.5	1.2	0.9	0.7	1.2	2	1.4	1.5

The mean, standard deviation, maximum and minimum values of the diacritization accuracy are 97.5%, 0.83%, 98.9% and 95.9%, respectively. Thus, the diacritization accuracy will fall in the range of $97.5\% \pm 2.49\%$. The mean, standard deviation, maximum and minimum values of the WER are 2.5%, 0.83%, 4.1% and 1.1%, respectively. Thus, the WER will fall in the range of $2.5\% \pm 2.49\%$. The mean, standard deviation, maximum and minimum values of the DER are 1.3%, 0.41%, 2.3% and 0.8%, respectively. Thus, the DER will fall in the range of $1.3\% \pm 1.23\%$.

Table 4.6 present the proposed algorithm evaluation in terms of diacritization accuracy, WER and DER, in comparison with the best reported result based on Tashkeela corpus.

Table 4.6

Evaluation of the proposed algorithm in comparison with G. Abandah [2]

Tashkeela corpus [32]	**88% used as training dataset, 12% as testing dataset**		
	Proposed Algorithm Statistical test	**Proposed Algorithm Actual test**	**G. Abandah** [2]
Total words	8,760,000	8,760,000	8,760,000
Wrong words	218,998	183,961	310,104
Diacritization accuracy	97.5%	97.9%	96.46%
WER	2.5%	2.1%	3.54%
DER	1.3%	1.1%	1.28%

Figure 4.1 present column graphs for diacritization accuracy, WER and DER in comparison with the best reported result based on Tashkeela corpus, G. Abandah [2].

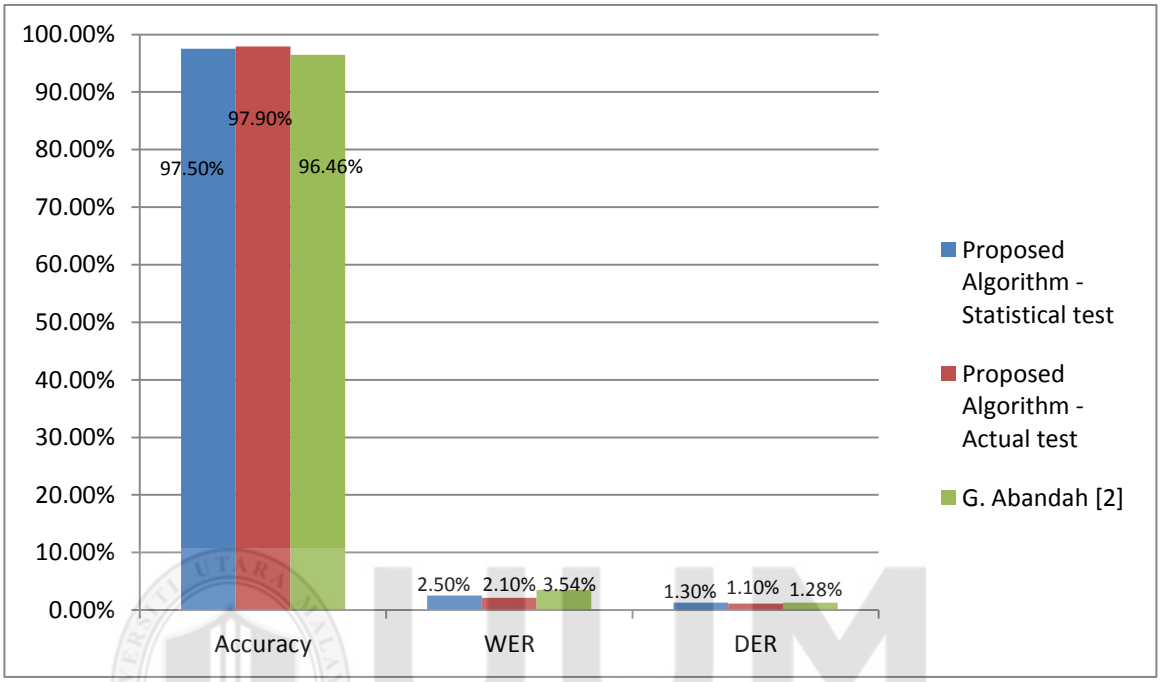

Figure 4.1. Graph for accuracy, WER and DER in comparison with Abandah [2]

The second statistical test has been conducted on the LDC's Arabic Treebank-Part 3 v1.0 [33] corpus, with 85% as training dataset, and 15% as testing dataset. Table 4.7 represents part of the diacritization accuracy, WER and DER after splitting the testing dataset.

Table 4.7

Diacritization accuracy, WER and DER for 10 datasets of LDC Arabic tree bank.

	Dataset1	**Dataset2**	**Dataset3**	**Dataset4**	**Dataset5**	**Dataset6**	**Dataset7**	**Dataset8**	**Dataset9**	**Dataset10**
Accuracy %	97.1	97.3	96.4	96.9	96.5	96.8	97.1	96.3	97.2	97.5
WER %	2.9	2.7	3.6	3.1	3.5	3.2	2.9	3.7	2.8	2.5
DER %	1.2	1.1	1.6	1.5	1.4	1.4	1.2	1.9	1.2	0.9

The mean, standard deviation, maximum and minimum values of the diacritization accuracy are 96.9%, 0.55%, 97.9% and 95.4%, respectively. Thus, the diacritization accuracy will fall in the range of $96.8\% \pm 1.65\%$. The mean, standard deviation, maximum and minimum values of the WER are 3.2%, 0.55%, 4.6% and 2.3%, respectively. Thus, the WER will fall in the range of $3.1\% \pm 1.65\%$. The mean, standard deviation, maximum and minimum values of the DER are 1.4%, 0.36%, 2.3% and 0.8%, respectively. Thus, the DER will fall in the range of $1.3\% \pm 1.08\%$.

Table 4.8 present the proposed algorithm evaluation in terms of diacritization accuracy, WER and DER, in comparison with the best reported result based on LDC's Arabic Treebank-Part 3 v1.0 [33] corpus.

Table 4.8

Evaluation of the proposed algorithm in comparison with M. Rashwan [30]

LDC's Arabic Treebank-Part 3 v1.0 [33] **corpus**	**85% used as training dataset, 15% as testing dataset**		
	Proposed Algorithm Statistical test	**Proposed Algorithm Actual test**	**M. Rashwan** [30]
Total words	51,000	51,000	51,000
Wrong words	1632	1479	1581
Diacritization accuracy	96.8%	97.1%	96.9%
WER	3.2%	2.9%	3.1%
DER	1.4%	1.2%	1.2%

Figure 4.2 present column graphs for diacritization accuracy, WER and DER in comparison with the best reported result based on LDC's Arabic Treebank-Part 3 v1.0, M. Rashwan [30].

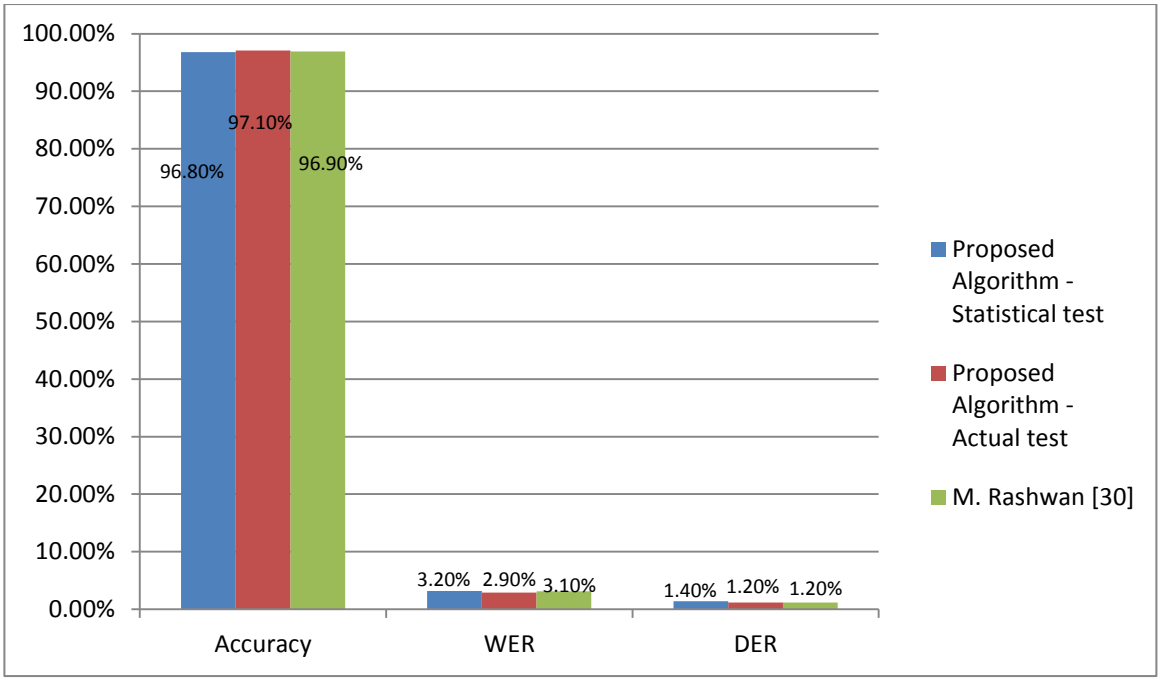

Figure 4.2 Graph for accuracy, WER and DER in comparison with Rashwan [30]

The third statistical test has been conducted on the dialectal Arabic corpus CallHome, with 85% as training dataset, and 15% as testing dataset. Table 4.9 represents part of the diacritization accuracy, WER and DER after splitting the testing dataset.

Table 4.9

Diacritization accuracy, WER and DER for 10 dataset group of CallHome corpus.

	Dataset1	**Dataset2**	**Dataset3**	**Dataset4**	**Dataset5**	**Dataset6**	**Dataset7**	**Dataset8**	**Dataset9**	**Dataset10**
Accuracy %	91.3	92.1	92.3	91.6	91.4	91.9	89.9	92	92.8	90.6
WER %	8.7	7.9	7.7	8.4	8.6	8.1	10.1	8	7.2	9.4
DER %	3.9	3.2	2.9	3.4	3.9	4.1	4.9	3.9	3.2	4.8

The mean, standard deviation, maximum and minimum values of the diacritization accuracy are 91.2%, 0.91%, 92.8% and 89.3%, respectively. Thus, the diacritization accuracy will fall in the range of $91.2\% \pm 2.73\%$. The mean, standard deviation, maximum and minimum values of the WER are 8.8%, 0.91%, 10.7% and 7.2%,

respectively. Thus, the WER will fall in the range of 8.8% $\pm$ 2.73%. The mean, standard deviation, maximum and minimum values of the DER are 4%, 0.63%, 5.1% and 2.9%, respectively. Thus, the DER will fall in the range of 4% $\pm$ 1.89%.

4.6 Summary

A hybrid based algorithm that combines the rule-based approach, namely morphological analyzer along with statistical approach, namely statistical n-gram and maximum likelihood estimate has been proposed. The algorithm has been tested on two different MSA corpora, namely Tashkeela and LDC's Arabic Treebank-Part 3 v1.0. The proposed algorithm demonstrated a good performance when applied to Tashkeela and LDC's Arabic Treebank-Part 3 v1.0, as the comparison results with best reported WER and DER as per the literature review were in favor of the proposed algorithm. Table 4.2 illustrates the comparison between the proposed algorithm and the best reported results based on the literature review.

The algorithm has been tested on CallHome dialectal Arabic corpus of telephone speech and proposed a good performance. Several configurations were also investigated in order to enhance the results yielded from applying the proposed algorithm on CallHome dialectal Arabic corpus. The best configuration was the diacritization through dialectal Arabic corpus word-level, MSA corpus word-level, dialectal Arabic corpus morpheme-level, MSA corpus morpheme-level, then dialectal Arabic corpus letter-level, with consideration of sub-models for each one. The best reported results were 90.3% of diacritization accuracy, WER of 9.7% and DER of 4.9%. When case ending is ignored,

the system resulted in 91.8% of diacritization accuracy, WER and DER of 8.2% and 3.7% respectively.

CHAPTER FIVE
CONCLUSION AND FUTURE WORK

At the beginning of this chapter, the objectives of this research that have been achieved were outlined, and then the limitations and recommendations were discussed. Finally, the contribution of this research and the future work are discussed in this chapter.

5.1 Achieved Objectives

The main findings of this research are as follows:

i) An improved hybrid based algorithm for automatic diacritization of undiacritized Arabic text. This objective has been achieved by designing and evaluating an enhanced algorithm with accuracy higher than the state-of-the-art systems.

ii) The proposed hybrid based algorithm was implemented on widely available MSA dataset, for restoring the diacritic marks and displaying the correct form of the word. The results of applying the proposed algorithm on MSA datasets are shown in Table 4.1.

iii) The reported results of diacritization accuracy, WER and DER on MSA dataset was higher that the state-of-the-art algorithms. The comparison between the proposed algorithm and the state-of-the-art systems are shown in section 4.3 and section 4.4 respectively.

5.2 Limitations and Recommendations

The reported results of the proposed algorithm are limited to the size of the training dataset and the Arabic varieties in use. It has got obvious, after extensive research and

experimentation, that in order to increase diacritization accuracy, the training dataset should be increased. Unfortunately, the development of large manually diacritized gold standard datasets is very costly. Thus, we were only limited to the existing datasets provided by ELRA, LDC, and Tashkeela. Moreover, the proposed approach results in higher accuracy of testing data of the same varieties as the training set. In other words, if training is only done using Modern Standard Arabic (MSA), the system will perform better on MSA testing data than on Egyptian Arabic data. We were only able to cover the varieties of MSA and Egyptian Arabic as we couldn't find diacritized data sets for the other varieties or dialects. That is why the algorithm is expected to result in lower accuracy on the other Arabic varieties like Moroccan, Levantine, Iraqi, Jordanian, and Gulf Area. As a general recommendation, larger data sets that cover as many Arabic dialects as possible are required in order to boost the performance of the proposed algorithm.

5.3 Contribution of this Research

The main contribution of this research is the hybrid algorithm for automatic diacritization of undiacritized MSA text and dialectal Arabic text. The proposed algorithm combines the rule-based approach, namely morphological analyzer along with statistical-based approach, namely statistical n-gram and maximum likelihood estimate. The proposed algorithm reported results higher than the state of the arts when applied on MSA as well as dialectal Arabic text.

Arabic is a highly complex language, even for Arabic native speakers. The absence of diacritic marks creates a huge ambiguity, especially for non-native Arabic speakers, as

the undiacritized word may correspond to more than one correct diacritization form. This proposed algorithm increase the automatic diacritization accuracy for undiacritized Arabic text, which will significantly ease the understanding of non-native Arabic speakers for undiacritized Arabic text. Moreover, the proposed algorithm applied and evaluated on Egyptian colloquial dialect, the most widely dialect understood and used throughout the Arab world, which is considered as first time based on the literature review.

5.4 Future Work

For future work, we are plaining to increase the accuracy of automatic diacritization for MSA and dialectal Arabic text, decrease the WER and DER as well. Moreover, to expand our training dataset to cover other dialectal Arabic forms, such as Moroccan Colloquial Dialect, Levantine Colloquial Dialect, Iraqi Colloquial Dialect, Jordanian Colloquial Dialect, and Gulf Area Colloquial Dialect.

REFERENCES

[1] M. Rashwan, A. Al Sallab, H. Raafat and A. Rafea, "Deep Learning Framework with Confused Sub-Set Resolution Architecture for Automatic Arabic Diacritization," *IEEE/ACM Transactions On Audio, Speech, And Language Processing,* vol. 23, no. 3, pp. 505-516, 2015.

[2] G. Abandah, A. Graves and B. Al-Shag, "Automatic diacritization of Arabic text using recurrent neural networks," *International Journal on Document Analysis and Recognition (IJDAR),* vol. 18, no. 2, pp. 183-197, 2015.

[3] H. Abo Bakr, K. Shaalan and I. Ziedan, "A hybrid approach for converting written Egyptian colloquial dialect into diacritized Arabic," in *The 6th international conference on informatics and systems, infos2008*, Cairo, Egypt, 2008.

[4] S. Harrat , M. Abbas , K. Meftouh and K. Smaïli, "Diacritics Restoration for Arabic Dialects," in *14th Annual Conference of the International Speech Communication Association* , Lyon, France, 2013.

[5] A. Said, M. El-Sharqwi, A. Chalabi and E. Kamal, "A Hybrid Approach for Arabic Diacritization," in *18th International Conference on Applications of Natural Language to Information Systems*, Salford, UK, 2013.

[6] A. Azmi and R. Almajed, "A survey of automatic Arabic diacritization techniques," *Natural Language Engineering,* vol. 21, no. 3, pp. 477-495, 2013.

[7] A. Shahrour, S. Khalifa and N. Habash, "Improving Arabic Diacritization through Syntactic Analysis," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[8] M. Rashwan, A. Al Sallab, H. Raafat and A. Rafea, "Automatic Arabic diacritics restoration based on deep nets," in *Empirical Methods in Natural Language Processing*, Doha - Qatar, 2014.

[9] Y. Hifny, "Restoration of Arabic Diacritics using Dynamic Programming," in *8th International Conference on Computer Engineering & Systems (ICCES)*, Cairo, Egypt, 2013.

[10] A. Al-Taani and S. Abu Al-Rub, "A Rule-Based Approach for Tagging Non-Vocalized Arabic Words," *The International Arab Journal of Information Technology,* vol. 6, no. 3, pp. 320-328, 2009.

[11] M. Ameur, Y. Moulahoum and A. Guessoum, "Restoration of Arabic Diacritics Using a Multilevel Statistical Model," *Springer International Publishing,* vol. 456, pp. 181-192, 2015.

[12] R. Nelken and S. Shieber, "Arabic diacritization using weighted finite-state transducers," in *In Proceedings of the 2005 ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan, 2005.

[13] M. Bebah, C. Amine, M. Azzeddine and L. Abdelhak, "Hybrid Approaches For Automatic Vowelization Of Arabic Texts," *International Journal on Natural Language Computing,* vol. 3, no. 4, pp. 53-71, 2014.

[14] N. Habash and O. Rambow, "Arabic Diacritization through Full Morphological Tagging," in *Proceedings of The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, NY, 2007.

[15] Y. Hifny, "Higher Order n-gram Language Models for Arabic Diacritics Restoration," in *The Twelfth Conference on Language Engineering*, Cairo, Egypt, 2012.

[16] Y. Hifny, "Smoothing Techniques for Arabic Diacritics Restoration," in *The Twelfth Conference on Language Engineering*, Cairo, Egypt, 2012.

[17] M. Alghamdi, Z. Muzaffar and H. Alhakami, "Automatic Restoration Of Arabic Diacritics: A Simple, Purely Statistical Approach," *The Arabian Journal for Science and Engineering,* vol. 35, no. 2C, pp. 125-135, 2010.

[18] M. Alghamdi and Z. Muzafar, "KACST Arabic Diacritizer," in *the First International Symposium on Computers and Arabic Language*, Riyadh, Saudi Arabia, 2007.

[19] M. Elshafei, H. Al-Muhtaseb and M. Alghamdi, "Statistical Methods for Automatic diacritization of Arabic text," in *Saudi 18th National Computer Conference*, Riyadh, Saudi Arabia, 2006.

[20] M. Elshafei, H. Al-Muhtaseb and M. Alghamdi, "Machine Generation Of Arabic Diacritical Marks," in *The 2006 International Conference on Machine Learning; Models, Technologies & Applications*, Las Vegas, Nevada, USA, 2006.

[21] I. Zitouni, J. Sorensen and R. Sarikaya, "Maximum Entropy Based Restoration of Arabic Diacritics," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, Sydney, Australia, 2006.

[22] S. Ananthakrishnan, S. Bangalore and S. Narayanan, "Automatic Diacritization of Arabic Transcripts for Automatic Speech Recognition," in *In Proceedings of the International*

Conference on Natural Language Processing (ICON-05), Kanpur, India, 2005.

[23] Y. Gal, "An hmm approach to vowel restoration in arabic and hebrew," in *Workshop on Computational Approaches to Semitic Languages*, Philadelphia, USA, 2002.

[24] K. Shaalan, H. M Abo Bakr and I. Ziedan, "A Hybrid Approach for Building Arabic Diacritizer," in *The Proceedings of the 12th European Chapter of the Association for Computational Linguistics (EACL 2009) Workshop on Computational Approaches to Semitic Languages*, Athens, Greece, 2009.

[25] A. Said, M. El-Sharqwi, A. Chalabi and E. Kamal, "A Hybrid Approach for Arabic Diacritization," in *18th International Conference on Applications of Natural Language to Information Systems*, Salford, UK, 2013.

[26] M. Rashwan, M. Al-Badrashiny, M. Attia, S. Abdou and A. Rafea, "A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features," *IEEE Transactions On Audio, Speech, And Language Processing,* vol. 19, no. 1, pp. 166-175, 2011.

[27] T. Schlippe, T. Nguyen and S. Vogel, "Diacritization as a Machine Translation Problem and as a Sequence Labeling Problem," in *The Eighth Conference of the Association for Machine Translation in the Americas - AMTA 2008*, Hawaii, 2008.

[28] M. Rashwan, M. Elbadrashiny, M. Attia and S. Mahdy Abdou, "A hybrid system for automatic arabic diacritization," in *The 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2009.

[29] A. El-Desoky, R. Schluter and H. Ney, "A Hybrid Morphologically Decomposed Factored Language Models for Arabic LVCSR," in *The 2010 Annual Conference of the North American Chapter of the ACL*, Los Angeles, California, 2010.

[30] M. Rashwan, M. Al Badrashiny, M. Attia, S. Abdou and A. Rafea, "Stochastic Arabic hybrid diacritizer," in *Natural Language Processing and Knowledge Engineering, 2009*, Dalian, China, 2009.

[31] N. Habash and O. Rambow, "Arabic Diacritization through Full Morphological Tagging," in *The Annual Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, Rochester, New York, 2007.

[32] T. Zerrouki, "Arabic corpora resources," Tashkeela collection from the Arabic Al-Shamela library, 19 July 2011. [Online]. Available: http://aracorpus.e3rab.com. [Accessed 27 November 2014].

[33] "Linguistic Data Consortium," LDC, [Online]. Available: https://www.ldc.upenn.edu. [Accessed 17 March 2016].

[34] "Linguistic Data Consortium," LDC, [Online]. Available: https://www.ldc.upenn.edu/. [Accessed 7 May 2016].

[35] S. Ross, Introductory Statistics, 3rd edition, Academic Press, 2005.

[36] D. Jurafsky and J. H. Martin, Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, New Jersey: Pearson Education, 2009.

[37] "Catalogue of Language Resources," European Land Registry Association (ELRA), [Online]. Available: http://catalog.elra.info/index.php?language=en. [Accessed 17 12 2015].

[38] A. Chennoufi, A. Mazroui and A. Lakhouaja, "HYBRID APPROACHES FOR AUTOMATIC," *International Journal on Natural Language Computing,* vol. 3, no. 4, pp. 53-71, 2014.

[39] E. Kamal, A. Said, M. El-Sharqwi and A. Chalabi, "A Hybrid Approach for Arabic Diacritization," in *18th International Conference on Applications of Natural Language to Information Systems*, Salford, UK, 2013.

[40] M. A. Rashwan, M. Elbadrashiny and S. Mahdy Abdou, "A Hybrid System for Automatic Arabic Diacritization," in *The 2nd International Conference on Arabic Language Resources and Tools.*, Cairo, Egypt., 2009.

[41] K. Shaalan, H. M Abo Bakr and I. Ziedan, "A Hybrid Approach for Building Arabic Diacritizer," in *The Proceedings of the 12th European Chapter of the Association for Computational Linguistics*, Athens, Greece, 2009.

[42] M. Bebah, C. Amine, M. Azzeddine and L. Abdelhak, "Hybrid Approaches For Automatic Vowelization Of Arabic Texts," *International Journal on Natural Language Computing (IJNLC),* vol. 3, no. 4, pp. 53-71, 2014.