

**UTILIZING DATA-DRIVEN AND KNOWLEDGE-BASED  
TECHNIQUES TO ENHANCE ARABIC SPEECH  
RECOGNITION**

BY

**DIA EDDIN MOHAMMAD ASAD ABUZEINA**

A Dissertation Presented to the  
DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the  
Requirements for the Degree of

**DOCTOR OF PHILOSOPHY**

In

**COMPUTER SCIENCE AND ENGINEERING**

**DECEMBER 2011**

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

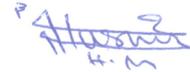
DEANSHIP OF GRADUATE STUDIES

This dissertation, written by DIA EDDIN M. ABUZEINA under the direction of his dissertation advisor and approved by his dissertation committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING.

Dissertation Committee



Thesis Advisor  
Prof. Moustafa Elsbafei



Co-advisor/Member  
Dr. Husni Al-Muhtaseb



Member  
Prof. Radwan Abdel-Aal



Member  
Dr. Mansour Alghamdi



Member  
Dr. Wasfi G. Al-Khatib

  
Department Chairman  
Dr. Umar Al-Turki

  
Dean of Graduate Studies  
Dr. Salam Adel Zummo

31/1/12  
Date



*Dedicated to My: Parents*

# ACKNOWLEDGEMENT

Acknowledgment is due to the King Fahd University of Petroleum & Minerals for supporting this research.

I am highly grateful to the overall support I received from my wife and my children throughout preparing this thesis. I also express my gratitude to my brothers and sisters for limitless help during these years.

I wish to express my deep appreciation to my major advisor Prof. Moustafa Elshafei. I also wish to thank the other members of my thesis/dissertation committee Dr. Husni Almuhtaseb, Prof. Radwan E. Abdel-Aal, Dr. Mansour Alghamdi, and Dr. Wasfi G. Al-Khatib for their excellent comments on my research that improved this dissertation a lot.

I also thank many of my colleagues in King Fahd University of Petroleum & Minerals for helping and supporting my work over the years. I also grateful thanks King Abdulaziz City for Science and Technology (KACST) for partially supporting this research work under NSTP project research grant # (08-INF100-4).

For everyone who had helped and supported me to reach this stage: I would say  
THANK YOU.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT .....</b>	<b>IV</b>
<b>TABLE OF CONTENTS .....</b>	<b>V</b>
<b>LIST OF TABLES .....</b>	<b>VIII</b>
<b>LIST OF FIGURES .....</b>	<b>IX</b>
<b>LIST OF APPENDICES .....</b>	<b>XI</b>
<b>PUBLICATIONS .....</b>	<b>XII</b>
<b>ABSTRACT .....</b>	<b>XV</b>
<b>خلاصة .....</b>	<b>XVII</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 INTRODUCTION .....	1
1.2 THESIS STATEMENT.....	3
1.3 MOTIVATION.....	4
1.4 OBJECTIVES.....	4
1.5 CONTRIBUTIONS.....	5
1.6 THESIS OUTLINE .....	6
<b>CHAPTER 2 PRELIMINARIES AND BACKGROUND.....</b>	<b>7</b>
2.1 THEORY AND BACKGROUND .....	7
2.2 SPEECH RECOGNITION ARCHITECTURE .....	10
2.2.1 <i>Front-End signal processing</i> .....	12
2.2.2 <i>Acoustic model</i> .....	14
2.2.3 <i>Decoding Using Viterbi algorithm</i> .....	16
2.2.4 <i>Training Using Baum-Welch algorithm</i> .....	18
2.2.5 <i>Language model</i> .....	22
2.2.6 <i>Pronunciation dictionary</i> .....	24
<b>CHAPTER 3 LITERATURE REVIEW .....</b>	<b>26</b>
3.1 OVERVIEW OF SPEECH RECOGNITION MODELING TECHNIQUES.....	26
3.2 LITERATURE OF ARABIC SPEECH RECOGNITION SYSTEMS .....	29
3.3 CHALLENGES OF ARABIC SPEECH RECOGNITION.....	35
<b>CHAPTER 4 THE BASELINE SYSTEM.....</b>	<b>36</b>
4.1 INTRODUCTION .....	36

4.2	ARABIC SPEECH CORPUSES .....	36
4.3	ARABIC PHONEME SET .....	38
4.4	ARABIC PRONUNCIATION DICTIONARY .....	38
4.5	ARABIC LANGUAGE MODEL .....	39
4.6	PERFORMANCE METRICS.....	40
4.6.1	<i>Word Error Rate (WER):</i> .....	40
4.6.2	<i>Out-Of-Vocabulary (OOV):</i> .....	41
4.6.3	<i>Perplexity (PP)</i> .....	41
4.7	SIGNIFICANCE MEASUREMENT .....	42
<b>CHAPTER 5 WITHIN-WORD PRONUNCIATION VARIATION MODELING .....</b>		<b>43</b>
5.1	INTRODUCTION .....	43
5.2	RELATED WORK.....	46
5.3	MOTIVATION.....	49
5.4	DYNAMIC PROGRAMMING.....	52
5.5	THE PROPOSED METHOD .....	53
5.6	TESTING AND EVALUATION .....	57
5.7	EXECUTION TIME .....	66
<b>CHAPTER 6 CROSS-WORD PRONUNCIATION VARIATION MODELING .....</b>		<b>67</b>
6.1	INTRODUCTION .....	67
6.2	EFFECTIVENESS OF COMPOUND-WORD ON PERFORMANCE .....	68
6.3	CROSS-WORD MODELING USING PHONOLOGICAL RULES .....	70
6.3.1	<i>Sources of cross-word problem</i> .....	72
6.3.2	<i>Arabic cross-word variations examples</i> .....	74
6.3.3	<i>Arabic Phonological Rules</i> .....	76
6.3.4	<i>Idgham</i> .....	77
6.3.5	<i>Iqlaab</i> .....	82
6.3.6	<i>Proposed method</i> .....	83
6.3.7	<i>Testing and evaluation</i> .....	86
6.3.8	<i>Execution time</i> .....	88
6.4	CROSS-WORD MODELING USING PART OF SPEECH TAGGING.....	96
6.4.1	<i>Proposed method</i> .....	97
6.4.2	<i>Testing and evaluation</i> .....	101
6.4.3	<i>Execution time</i> .....	104
6.5	CROSS-WORD MODELING USING SMALL WORDS MERGING .....	105
6.5.1	<i>Proposed method</i> .....	106
6.5.2	<i>Testing and evaluation</i> .....	108
6.5.3	<i>Execution time</i> .....	111
6.6	A COMPARISON BETWEEN CROSS-WORD MODELING APPROACHES .....	112
6.7	COMBINING OF WITHIN-WORD AND CROSS-WORD METHODS .....	113
<b>CHAPTER 7 RESCORING N-BEST HYPOTHESES .....</b>		<b>114</b>
7.1	INTRODUCTION .....	114

7.2	RELATED WORK.....	116
7.3	DATA-MINING APPROACH (WEKA TOOL).....	117
7.4	THE PROPOSED METHOD .....	118
	<b>CLOSING REMARKS .....</b>	<b>128</b>
	<b>APPENDICES .....</b>	<b>130</b>
	<b>NOMENCLATURE.....</b>	<b>140</b>
	<b>REFERENCES .....</b>	<b>142</b>
	<b>VITA .....</b>	<b>154</b>

# LIST OF TABLES

TABLE 3-1 ROUGH WORD ERROR RATES FOR A NUMBER OF ASRS (ENGLISH CORPUSES).....	29
TABLE 4-1 N-GRAMS IN THE BASELINE SYSTEM .....	40
TABLE 5-1 WITHIN-WORD PRONUNCIATION VARIATIONS EXAMPLES.....	50
TABLE 5-2 PRONUNCIATION VARIATION MODELING TECHNIQUES .....	51
TABLE 5-3 PROPOSED PRONUNCIATION VARIATION TECHNIQUE .....	52
TABLE 5-4 AN ALIGNMENT BETWEEN TWO SEQUENCES.....	53
TABLE 5-5 RECOGNITION OUTPUTS FOR DIFFERENT SPECIFICATIONS .....	58
TABLE 5-6 THE ACCURACY ACHIEVED USING WITHIN-WORD MODELING .....	59
TABLE 5-7 STATISTICAL INFORMATION ABOUT THE VARIANTS COLLECTED.....	60
TABLE 5-8 VARIANTS' FREQUENCIES.....	61
TABLE 5-9 PRONUNCIATION VARIATION MODELING WITHOUT LANGUAGE MODEL.....	62
TABLE 5-10 N-GRAMS IN THE BASELINE AND THE ENHANCED SYSTEMS .....	65
TABLE 5-11 RECOGNITION TIME OF THE BASELINE AND THE ENHANCED SYSTEMS.....	66
TABLE 6-1 IDGHAM CASES OF NUUN SAAKINA.....	79
TABLE 6-2 IDGHAM OF TWO CONSECUTIVE IDENTICAL LETTERS.....	80
TABLE 6-3 IDGHAM OF TWO CLOSE IN PRONUNCIATION LETTERS .....	81
TABLE 6-4 PERFORMANCE IMPROVEMENT USING PHONOLOGICAL RULES.....	87
TABLE 6-5 EXECUTION TIME COMPARISON OF THE ENHANCED AND THE BASELINE SYSTEMS.....	89
TABLE 6-6 A COMPARISON BETWEEN THE BASELINE AND THE ENHANCED SYSTEMS.....	91
TABLE 6-7 N-GRAMS OF BOTH SYSTEMS (BASELINE AND ENHANCED).....	93
TABLE 6-8 SAMPLES OF INDIRECT IMPROVEMENTS BY THE LANGUAGE MODEL.....	94
TABLE 6-9 THE ARABIC TAGS OF STANFORD TAGGER.....	98
TABLE 6-10 AN ARABIC SENTENCE AND ITS TAGS .....	99
TABLE 6-11 ACCURACY ACHIEVED .....	101
TABLE 6-12 PERPLEXITIES AND OOV IN DIFFERENT EXPERIMENTS MADE .....	102
TABLE 6-13 STATISTICAL INFORMATION FOR COMPOUND WORDS .....	103
TABLE 6-14 AN ERROR IN THE TAGGER .....	103
TABLE 6-15 EXECUTION TIME COMPARISON OF THE ENHANCED AND THE BASELINE SYSTEMS.....	105
TABLE 6-16 RESULTS FOR DIFFERENT SMALL WORD LENGTHS .....	109
TABLE 6-17 EXECUTION TIME COMPARISON OF THE ENHANCED AND THE BASELINE SYSTEMS.....	111
TABLE 6-18 A COMPARISON BETWEEN COMBINED PROPOSED TECHNIQUES .....	112
TABLE 6-19 A COMPARISON BETWEEN COMPOUND WORDS TECHNIQUES.....	113
TABLE 7-1 FIRST 5-BEST SYNTACTIC RULES OF 3000 EXTRACTED RULES .....	120
TABLE 7-2 TWO EXAMPLES OF TAGGED SENTENCES.....	126

# LIST OF FIGURES

FIGURE 2-1 SPHINX-ENGINE ARCHITECTURE .....	10
FIGURE 2-2 FEATURE VECTORS EXTRACTION.....	12
FIGURE 2-3 MFCCS OF A SPEECH FILE .....	14
FIGURE 2-4 A 3-STATE PHONE ACOUSTIC MODEL.....	15
FIGURE 2-5 THE VARIOUS TASKS INVOLVED IN BUILDING THE ACOUSTIC MODEL.....	19
FIGURE 2-6 STEPS FOR CREATING AND TESTING LANGUAGE MODEL.....	24
FIGURE 5-1 PRONUNCIATION VARIATION MODELING TECHNIQUES .....	45
FIGURE 5-2 THE BASELINE CORPUS TRANSCRIPTION AND ITS PHONEMES.....	54
FIGURE 5-3 BASELINE DICTIONARY AFTER TRANSFORMING THE PHONEMES.....	55
FIGURE 5-4 FINDING VARIANTS PROCESS.....	55
FIGURE 5-5 ACCURACY ACHIEVED USING PURE DATA-DRIVEN VARIANTS .....	59
FIGURE 5-6 A DELETION CASE PROBLEM FIXED IN THE ENHANCED SYSTEM.....	64
FIGURE 5-7 AN INSERTION CASE PROBLEM FIXED IN THE ENHANCED SYSTEM.....	64
FIGURE 5-8 INDIRECT ENHANCEMENT IN THE ENHANCED SYSTEM. ....	65
FIGURE 5-9 THE NEGATIVE EFFECT OF RECALCULATING N-GRAMS .....	66
FIGURE 6-1 PRONUNCIATION VARIATIONS AND MODELING TECHNIQUES .....	68
FIGURE 6-2 THE DIFFERENCE BETWEEN SMALL AND LONG WORDS DURING DECODING .....	69
FIGURE 6-3 CROSS-WORD ADAPTATION LEVELS .....	70
FIGURE 6-4 CROSS-WORD PROBLEM .....	71
FIGURE 6-5 CROSS-WORD VARIATIONS SOURCES.....	74
FIGURE 6-6 THE EFFECT OF IDGHAM IN ARABIC SPEECH .....	75
FIGURE 6-7 THE EFFECT OF IQLAAB IN ARABIC SPEECH.....	75
FIGURE 6-8 IDGHAM OF TWO CLOSE IN PRONUNCIATION CASE.....	76
FIGURE 6-9 GENERATING A COMPOUND WORD .....	77
FIGURE 6-10 IQLAAB EXAMPLES.....	82
FIGURE 6-11 A SAMPLE OF THE TRANSCRIPTION CORPUS USED .....	83
FIGURE 6-12 A SAMPLE OF THE ENHANCED CORPUS TRANSCRIPTION.....	84
FIGURE 6-13 A SAMPLE OF THE DICTIONARY ENTRIES .....	84
FIGURE 6-14 CROSS-WORD MODELING USING PHONOLOGICAL RULES .....	86
FIGURE 6-15 IDGHAM CASE: UNVOWELLED NUUN (NUUN SAAKINAH) FOLLOWED BY RAA' .....	89
FIGURE 6-16 IDGHAM CASE: UNVOWELLED NUUN (NUUN SAAKINAH) FOLLOWED BY MIIM.....	90
FIGURE 6-17 IQLAAB CASE: UNVOWELLED NUUN (NUUN SAAKINAH) FOLLOWED BY BAA'.....	90
FIGURE 6-18 A COMPARISON BETWEEN THE BASELINE AND THE ENHANCED SYSTEMS .....	91
FIGURE 6-19 A CONNECTION SPOT BETWEEN A NOUN AND AN ADJECTIVE .....	97
FIGURE 6-20 A NOUN-ADJECTIVE COMPOUND WORD GENERATION .....	100
FIGURE 6-21 CROSS-WORD MODELING ALGORITHM USING TAGS MERGING .....	100
FIGURE 6-22 AN EXAMPLE OF ENHANCEMENT IN THE ENHANCED SYSTEM.....	104
FIGURE 6-23 A SMALL-WORD PROBLEM EXPLANATION .....	105
FIGURE 6-24 THE CONCEPT OF MODELING SMALL-WORD .....	106
FIGURE 6-25 CROSS-MODEL PRONUNCIATION VARIATION ALGORITHM USING SMALL WORDS .....	108

FIGURE 6-26 A COMPARISON OF ACCURACY FOR DIFFERENT COMPOUND WORDS LENGTHS.....	109
FIGURE 6-27 COMPOUND WORDS USAGE .....	110
FIGURE 7-1 AN EXAMPLE OF 6-BEST HYPOTHESES OF A SENTENCE .....	115
FIGURE 7-2 ILLUSTRATION OF RESCORING N-BEST LIST.....	115
FIGURE 7-3 GENERATION OF RESCORED N-BEST LIST .....	118
FIGURE 7-4 A PERFECT ENHANCEMENT IN A TESTED FILE.....	123
FIGURE 7-5 A PERFECT ENHANCEMENT IN A TESTED FILE.....	123
FIGURE 7-6 A PARTIAL ENHANCEMENT IN A TESTED FILE.....	124
FIGURE 7-7 A PARTIAL ENHANCEMENT IN A TESTED FILE.....	124
FIGURE 7-8 A WRONG HYPOTHESIS SELECTION EXAMPLE .....	125
FIGURE 7-9 NOT-SELECTED CORRECT HYPOTHESIS EXAMPLE.....	125
FIGURE 7-10 10-BEST LIST OF A TESTED FILE.....	127

# LIST OF APPENDICES

<b>APPENDIX 1 : ARABIC TERMINOLOGIES .....</b>	<b>130</b>
<b>APPENDIX 2 : ARABIC–ROMAN LETTERS MAPPING TABLE .....</b>	<b>132</b>
<b>APPENDIX 3 : THE PHONEMES SET USED IN THE BASELINE SYSTEM (IPA) .....</b>	<b>133</b>
<b>APPENDIX 4: PHONEME-CHARACTER MAPPING .....</b>	<b>134</b>
<b>APPENDIX 5: RULES USAGE IN THE ENTIRE TRANSCRIPTION CORPUS .....</b>	<b>136</b>
<b>APPENDIX 6: STANFORD TAGGING SET .....</b>	<b>138</b>

# PUBLICATIONS

Following is the list of publications, at the time of submitting this dissertation, based on the work in this PhD dissertation.

## **Refereed Journal Papers:**

- AbuZeina D., Al-Khatib W., Elshafei M., Al-Muhtaseb H., "Cross-word Arabic pronunciation variation modeling for speech recognition" , International Journal of Speech Technology ,2011.
- AbuZeina D., Al-Khatib W., Elshafei M., "Modeling of Cross-word Pronunciation Variation for Arabic ASRs: A Knowledge-Based Approach", Journal of Communications and Computer Engineering, 2011.
- AbuZeina D., Al-Khatib W., Elshafei M., Al-Muhtaseb H., "Within-Word Pronunciation Variation Modeling for Arabic ASRs: A Direct Data-Driven approach", International Journal of Speech Technology, 2011.
- AbuZeina D., Al-Khatib W., Elshafei M., Al-Muhtaseb H.," Toward enhanced Arabic speech recognition using part of speech tagging ”, International Journal of Speech Technology ,2011.

## **Submitted:**

- AbuZeina D., Elshafei M., Al-Muhtaseb H., Al-Khatib W.,"Rescoring N-Best Hypotheses for Arabic Speech Recognition: A Syntax-Mining Approach", International Journal of Speech Technology, 2011.

### **International Conferences:**

- AbuZeina D., Elshafei M., "Cross-word Arabic pronunciation variation modeling for speech recognition", IEEE GCC Graduate Forum 2011 UAE, Dubai, 2011.
- AbuZeina D., Elshafei M., "Within-Word Pronunciation Variation Modeling for Arabic ASRs: A Direct Data-Driven approach", 3rd International Conference on Computer Engineering and Technology, Malaysia, 2011.
- AbuZeina D., Al-Khatib W., Elshafei M., "Modeling of Cross-word Pronunciation Variation for Arabic ASRs: A Knowledge-Based Approach", 7th International Computing Conference In Arabic, Riyadh, 2011.
- AbuZeina D., Elshafei M., Al-Khatib W., "Part of Speech Tagging Approach to Designing Compound Words for Arabic Continuous Speech Recognition Systems", The International Conference on Informatics Engineering & Information Science, Malaysia, 2011.
- AbuZeina D., Al-Khatib W., Elshafei M., "Small-Word Pronunciation Modeling for Arabic Speech Recognition: A Data-Driven Approach", Seventh Asian Information Retrieval Societies Conference, Dubai, 2011.

### **Books:**

- Cross-Word Modeling for Arabic Speech Recognition.  
(SpringerBriefs in Electrical and Computer Engineering / SpringerBriefs in Speech Technology) by Dia AbuZeina and Moustafa Elshafei.
- In progress ( A book chapter):

A chapter in a forthcoming Speech Recognition book, ISBN 979-953-307-790-0, to be publish by InTech, a global leader in Open Access publishing for the international science, technology and medical community.

# ABSTRACT

NAME: DIA EDDIN MOHAMMAD ASAD ABUZEINA  
TITLE OF STUDY: UTILIZING DATA-DRIVEN AND KNOWLEDGE-BASED TECHNIQUES TO ENHANCE ARABIC SPEECH RECOGNITION  
MAJOR FIELD: COMPUTER SCIENCE AND ENGINEERING  
DATE OF DEGREE: DECEMBER, 2011

Pronunciation variation is a well-known phenomenon which leads to performance reduction in speech recognition systems. This performance reduction factor occurs mainly in two forms: within-word pronunciation variation, and cross-word pronunciation variation. The within-word variation occurs inside the word, while the cross-word variation occurs when two successive words interact leading to a different pronunciation in one or two letters. Furthermore, the two words could merge together creating one continuous utterance with no clear boundary between them. In speech recognition, within-word and cross-word pronunciation variations alter the phonetic spelling of words beyond their listed forms in the pronunciation dictionary, leading to a number of out-of-vocabulary word forms, and consequently reducing the speech recognition performance. Pronunciation variation problems could also arise in the form of an incorrectly recognized word sequence with out-of-language syntax. In this thesis we propose knowledge-based and data-driven techniques to solve these three problems (i.e. within-word, cross-word, and out of correct order syntactical structures).

The proposed methods were investigated on a modern standard Arabic speech recognition system using Carnegie Mellon University Sphinx speech recognition engine. The first problem (within-word variations) was modeled using the data-driven approach which utilizes a dynamic programming method (sequence alignment for phonemes) to distill variants from the pronunciation corpus. The results showed that this technique achieved significant improvements of 1.82%.

The second problem (cross-word variations) was modeled using three different tracks: a knowledge-based approach (using Arabic phonological rules), a knowledge-based approach (using part of speech tagging), and a data-driven approach (by merging small words). The results showed that the three above mentioned tracks achieved significant improvements. The part of speech tagging approach achieved the highest improvement of 2.39%, followed by the phonological rules approach, achieving 2.30% and finally the merging small words approach achieving 2.16%, over the baseline system.

The third problem was modeled using a data mining algorithm to extract the best language syntax rules, that can be later used for rescoring the N-best hypotheses. A Stanford Arabic tagger was used for the tagging process. This method, nevertheless, did not lead to a significant improvement.

## خلاصة

### درجة الدكتوراة في الفلسفة

الاسم: ضياء الدين محمد أسعد أبوزينة  
عنوان الرسالة: تعزيز كفاءة أنظمة التعرف على الكلام العربي باستخدام معلومات اللغة وبيانات التدريب  
التخصص: هندسة وعلوم الحاسب  
تأريخ التخرج: ديسمبر 2011

تعتبر ظاهرة تغير نطق الكلمات واحدة من العوامل التي تؤدي إلى ضعف الأداء في أنظمة التعرف على الكلام العربي. وتظهر عوامل ضعف الأداء في شكلين أساسيين: الأول هو حدوث التغير في النطق داخل الكلمة نفسها، بينما يتجلى العامل الثاني في حدوث التغير بين كلمتين متجاورتين، بحيث تندمج هاتان الكلمتان مع بعضهما مما يؤدي إلى اختلاف النطق المفترض بسبب حدوث التداخل وفقدان الحد الفاصل بينهما. إن ظاهرة التغير في النطق سواء على مستوى الكلمة أو بين كلمتين متجاورتين تؤدي إلى ظهور كلمات جديدة غير مدرجة في القاموس الصوتي، وينتج عن ذلك زيادة في عدد الكلمات الخاطئة في النتائج والتي تؤدي أيضاً إلى إنتاج تراكيب لغوية خاطئة. نقترح في هذه الرسالة استخدام معلومات اللغة وبيانات التدريب من أجل نمذجة ظاهرة التغير في نطق الكلمات (على مستوى الكلمة، بين كلمتين، ومشكلة التراكيب اللغوية الخاطئة).

تم فحص الطرق المقترحة من خلال إستخدام نظام تعرف على الكلام تم بناؤه في جامعة الملك فهد للبتروك والمعادن بإستخدام وسائل التعرف على الكلام (سفنكس) المقدمة من جامعة "كارنيجي ميلون".

تم نمذجة ظاهرة التغير في النطق داخل الكلمة الواحدة وذلك بإستخدام طريقة البرمجة الديناميكية من اجل مطابقة سلاسل الفونيمات لإنتاج المتغيرات المقترحة من المدونة الصوتية. وقد أظهر استخدام هذه الطريقة تحسناً ملحوظاً في الأداء بنسبة 1.82 في المئة.

كما تمت نمذجة ظاهرة إندماج الكلمات باستخدام ثلاثة طرق منفصلة كالتالي: بإستخدام معلومات اللغة التي تحتوي على القواعد الفونولوجية، إستخدام أقسام الكلام لدمج الكلمات المتجاورة، و بيانات التدريب لدمج الكلمات الصغيرة. وقد أظهر إستخدام هذه الطريق تحسناً ملحوظاً في الأداء. إذ تحسن الاداء بأعلى نسبة بإستخدام طريقة أقسام الكلام حيث كانت النسبة 2.39 في المئة، تلتها طريقة القواعد الفونولوجية بنسبة 2.30 في المئة، وبعدها طريقة دمج الكلمات الصغيرة بنسبة 2.16 في المئة.

تم نمذجة ظاهرة الأخطاء التركيبية وذلك بإعادة تقييم الفرضيات الناتجة من نظام التعرف بحيث يتم إعتداد أفضل فرضية بعد عملية التقييم. تم إستخدام المدونة النصية (بعد توصيف كلماتها) من اجل التنقيب عن أكثر التراكيب شيوعاً في اللغة العربية وبالتالي ايجاد أفضل فرضية من حيث تطابقها مع تراكيب اللغة. ولم تظهر هذه الطريقة تحسناً في الأداء.

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

The fast pace of the advancement in information and communications technology is reshaping our society and vastly increasing our capabilities for faster learning, higher achievements, better and wider communication, in addition to more effective and productive human-computer interface.

One of the important frontiers of communication technology is the user-interface, namely how the man-machine interface can be designed in a more natural environment and immersive environment, which captures the essential attributes of a human-like exchange between human and machine. To address this important issue, researchers from various areas have been hard at work to equip machines with vital human-like capabilities, such as speech communication and vision. It is fair to say that despite many staggering technological successes achieved in these areas, the machine capabilities developed so far remain rather primitive compared to their human counterparts. This has propelled human-machine system designers to continue their relentless effort to achieve this far reaching goal.

One such general area where research is continuing persistently is the speech processing area. Speech is the natural form of communication between humans. Its

production is a highly nonlinear process that is strongly influenced by the high variability of factors such as, age, gender, rate of speech, different dialects and regional accents, emotional state, and more. Speech perception is a hard task in that, in addition to the above-cited production-related difficulties, it has to contend with other equally variable and adverse factors such as background noise, interference from other speakers, room acoustics, recording equipment, and channel characteristics in the case of telephone conversation. Automatic Speech Recognition (ASR) is a key technology for a variety of applications, such as automatic translation, hands-free operation and control (as in cars and airplanes), automatic query answering, telephone communication with information systems, automatic dictation (speech-to-text transcription), government information systems, etc. In fact, speech communication with computers and household appliances is envisioned to be the dominant human-machine interface in the near future. However, despite many impressive achievements in the area of speech recognition, reaching well-functioning human performance levels still remains a possibly unattainable goal.

During the last few decades, much research was carried out in the ASR area resulting in numerous practical and commercial successes with impressive high recognition performances, but only if the environment and the speaking manner are constrained such as with using isolated keywords.

No doubt, conversational or continuous speech recognition introduces many challenges to ASRs. One of these challenges is the pronunciation variation problem, which is known to reduce recognition accuracy. Pronunciation variation appears in the form of insertions, deletions, or substitutions of phoneme(s) relative to the canonical transcription of the words in the pronunciation dictionary. Within-word variations and

cross-word variations (words' junctures merging) are well known variation problems in continuous speech. Additionally, syntactically incorrect ASRs outputs are also another types of error sources in ASRs. Accordingly, handling these phenomena is a major requirement to have robust ASRs.

This thesis focuses on Arabic speech recognition, which has gained increasing importance in the last few years. Arabic is a Semitic language spoken by more than 330 million people as a native language [1]. In this thesis, we consider the modern standard Arabic (MSA) which is currently used in writing and in most formal speech. MSA is also the major medium of communication for public speaking and news broadcasting [2] and is considered to be the official language in most Arabic-speaking countries [3].

This thesis contains necessarily many examples in Arabic; Appendix 1 is provided for the Arabic terminologies used in this thesis.

## 1.2 Thesis Statement

In this thesis, the most noticeable Arabic ASRs performance reduction factors were investigated. These factors include within-word and cross-word pronunciation variations, which also lead to syntactically incorrect ASRs outputs. To enhance speech recognition accuracy, data-driven and knowledge-based techniques have been utilized to model the above mentioned problems at two ASRs components: the pronunciation dictionary and the language model. While modeling the within-word and cross-word variations shows a significant enhancement, our investigations show that knowledge-based technique to model syntactically incorrect ASRs outputs does not enhance the recognition Accuracy.

### 1.3 Motivation

Speech recognition is often used as the front-end for many natural language processing (NLP) applications. Some of these typical applications include voice dialing, call routing, data entry, dictation, control, commands, and computer-aided language learning. Intuitively, improving the speech recognition performance will improve the related NLP applications. Generally, this thesis explores new methods to improve the recognition performance of Arabic ASR systems.

### 1.4 Objectives

The main objective of this thesis is to enhance the accuracy of Arabic ASRs systems. The objectives are divided as follows.

First, the direct data-driven approach was investigated to model within-word pronunciation variations, in which the pronunciation variants were distilled from the training speech corpus. The proposed method consists of performing phoneme recognition, followed by a sequence alignment between the observation phonemes generated by the phoneme recognizer and the reference phonemes obtained from the pronunciation dictionary. A phoneme-to-grapheme conversion is then used to generate the transcription forms of the unique variants, which will be added to the pronunciation dictionary and the language model.

Second, the cross-word problem was investigated and modeled in three different ways: Arabic phonological rules, speech tags merging, and small words merging. The small words' merging is considered as a data-driven approach while the phonological rules and tags merging are considered as knowledge-based methods. Using these

methods, the cross-word problem is tackled by merging the consequent words, according to pre-specified rules, to be then added to the pronunciation dictionary and the language model.

Third, we present a syntax-mining approach to rescore N-best hypotheses for Arabic speech recognition systems. The method depends on a machine learning tool (weka-3-6-5) to extract the N-best syntactic rules from the baseline tagged transcription corpus. The extracted rules are then used to rescore N-best hypotheses to choose the best one.

Carnegie Mellon University (CMU) Sphinx speech recognition engine was used to investigate the above cited objectives. The Sphinx engine was applied on the baseline system, which contains a pronunciation dictionary of 14,234 words from a 5.4 hours corpus of Arabic broadcast news.

## 1.5 Contributions

The main contribution of this thesis is the enhancements achieved in the Arabic speech recognition over the baseline system. These enhancements are pursued by utilizing data-driven and knowledge-based techniques as a preprocessing and a post-processing stages. Our results show the following findings:

- For within-word variation: Data-driven approach which is based on extracting variants from pronunciation corpus, leads to a significant enhancement.
- For cross-word variation: Knowledge-based (phonological rules and part of speech tagging) approaches to combine consecutive words lead to significant enhancements.

- For cross-word variation: Data-Driven (compounding consecutive small-words) leads to a significant enhancement.
- For N-best hypotheses rescoring: Rescoring N-best hypotheses using data-mining syntactic structures does not lead to a performance enhancement (for Arabic).
- A set of tools has been developed specifically for Arabic language. these tools will be made available for the academic community.

## 1.6 Thesis outline

The rest of this thesis is organized as follows. Chapter 2 presents the preliminaries and the background of this research work. Chapter 3 presents the literature review and the Arabic speech recognition challenges. Then, in chapter 4, the baseline system is described. Chapter 5 discusses the within-word pronunciation variations phenomenon, the suggested solution, and the results. Chapter 6 presents the cross-word pronunciation variations, the modeling techniques, and the results. Chapter 7 discusses the N-best hypotheses and the rescoring procedure as well as our findings. Finally, the closing remark concludes the thesis with the recommended research directions in Arabic speech recognition research area.

# **CHAPTER 2**

## **PRELIMINARIES AND**

## **BACKGROUND**

### **2.1 Theory and background**

A speech recognizer is a program that converts speech into texts for many purposes; facilitating human computer interface is the major advantage. A wider reach of the information technology (IT) in the society can be achieved if users can verbally communicate with computer. In fact, being able to speak fluently with computer may eliminate handwriting problems and, therefore, increases the productivity of people. Nowadays, big companies utilize this technology to automate their processes. With huge number of customers, companies tend to offer their services more smoothly as a user can verbally inquire, order, and pay. In addition to the commercial applications, speech recognition is also employed in eLearning, training, and education of students with learning disabilities. Khasawneh et al. in [4] listed some speech recognition applications, which include banking by telephone, automatic teller machines, compact size computers, browsing computer networks and databases by voice, and operating machinery from a distance in dangerous working sites. However, there are drawbacks. Speech recognition systems require high computational machines with large memory. Additionally, a high

rate of misrecognitions and errors is still a major problem in speech recognition systems, which hinders its widespread adaptation in the IT applications.

Benzeghiba et al. in [5] presented a comprehensive study on pronunciation variations as major sources of errors in automatic speech recognition. They demonstrated some of the speech variability sources: foreign and regional accents, speaker physiology, speaking style and spontaneous speech, rate of speech, children speech, emotional state, and more.

A typical large vocabulary speech recognizer would first convert speech waveform into a sequence of feature vectors to be used to identify the phones (the acoustic speech unit). The recognized phones are used to specify the words and then the sequence of words.

Rabiner and Juang [6] demonstrated that the statistical approach has dominated ASR research over the last few decades. The statistical approach is itself dominated by the powerful statistical technique called Hidden Markov Model (HMM). Based on the initiating research work of Baker [8], the HMM-based ASR technique has led to numerous successful applications requiring large vocabulary speaker-independent continuous speech recognition as mentioned by Jelinek in [7], Morgan and Bourlard in [9], and Young in [10].

The HMM-based technique essentially consists of recognizing speech by estimating the likelihood of each phone at contiguous, small frames of the speech signal ([6], [11]). Words in the target vocabulary are modeled into a sequence of phonemes and then a search procedure is used to find, among the words in the vocabulary list, the phoneme sequence that best matches the sequence of phones of the spoken word. Each

phoneme is modeled as a sequence of HMM states. In standard HMM-based systems, the likelihoods (also known as the emission probabilities) of a certain frame observation being produced by a state are estimated using traditional Gaussian mixture models (GMMs). The use of HMM with Gaussian mixtures has several notable advantages such as a rich mathematical framework, efficient learning and decoding algorithms, and an easy integration of multiple knowledge sources.

Two notable successes in the academic community in developing high performance large vocabulary, speaker-independent, continuous speech recognition systems are the HMM tools, known as the Hidden Markov Model Toolkit (HTK), developed at Cambridge University ([12], [13]), and the Sphinx system developed at CMU ([14], [15]). HTK is a general purpose toolkit for building HMMs and has been used in many applications. On the contrary, CMU Sphinx system was built specifically for speech recognition applications. In this thesis, we used Sphinx-based ASR system for testing and evaluation.

The Sphinx Group at CMU has been supported for many years by funding from the Defense Advanced Research Projects Agency (DARPA) and industries to assess and develop speech recognition techniques. In 2000, the Sphinx group released Sphinx-II, a real-time, large vocabulary, speaker-independent speech recognition system as free software. The source code is freely available for educational institutions. The extensive source code resources represent an excellent research infrastructure and a powerful test bed for researchers to pursue further state-of-the-art research in the area of speech recognition techniques. CMU Sphinx toolkit has a number of packages for different tasks

and applications, Open Source Toolkit for Speech Recognition [16]. Some tools are as follows:

- PocketSphinx—recognizer library written in C
- Sphinxbase—support library required by PocketSphinx
- Sphinx 3—adjustable, modifiable recognizer written in C
- Sphinx 4—adjustable, modifiable recognizer written in Java
- CMUclmtk—language model tools
- SphinxTrain—acoustic model training tools

## 2.2 Speech recognition architecture

Modern large vocabulary, speaker-independent, continuous speech recognition systems have three knowledge sources: acoustic model, language model (LM), and pronunciation dictionary (also called lexicon). A lexicon provides pronunciation information for each word in the vocabulary in phonemic units, which are modeled in detail by the acoustic models. The language model provides the priori probabilities of word sequences. Figure 2-1 shows Sphinx-engine architecture.

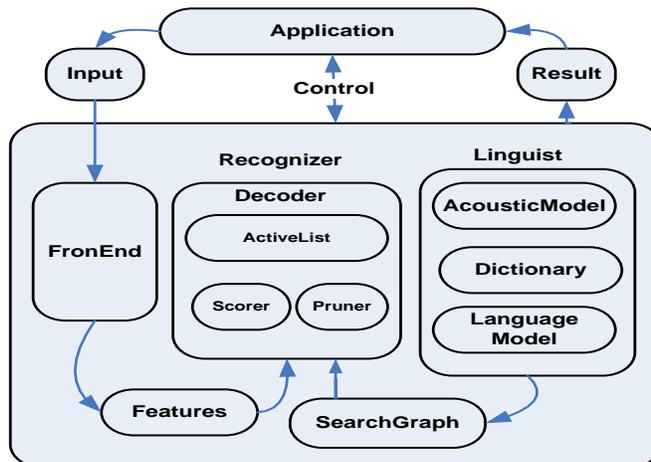


Figure 2-1 Sphinx-engine architecture

Figure 2-1 illustrates the sub-systems available in Sphinx tools and the relationships between them. The following is a brief description of the main sub-functions of Sphinx engine:

**The Front-End:** The purpose of this sub-system is to extract speech features, and it plays a crucial role for better recognition performance. Speech features includes Linear Predictive Cepstral Coefficients (LPCC), Mel-Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Predictive (PLP) coefficients. The Sphinx engine used in this work relies on the (MFCCs).

**The Linguist:** This part contains the modifications required for a particular language. It contains three parts: acoustic model, language model, and pronunciation dictionary. Acoustic model contains the HMMs used in recognition process. The language model contains language's words and its combinations, each combination has two words or above. A pronunciation dictionary contains the words of the language. The dictionary represents each word in terms of phonemes.

**The Decoder (Recognizer):** With help from the linguistic part, the decoder is the module where the recognition process takes place. The decoder uses the speech features presented by the Front-End to search for the most probable words and, then, sentences that correspond to the observation speech features. Hence fore, the recognition process starts by finding the likelihood of a given sequence of speech features based on the phonemes HMMs.

The speech recognition problem is to transcribe the most likely spoken words given the acoustic observations. If  $O = o_1, o_2, \dots, o_n$  is the acoustic observation, and  $W = w_1, w_2, \dots, w_n$  is a word sequence, then:

$$\hat{W} = \underset{\text{for all words}}{\operatorname{arg\,max}} P(W)P(O/W)$$

Where  $\hat{W}$  is the most probable word sequence of the spoken words, which is also called maximum posteriori probability.  $P(W)$  is the prior probability computed in the language model, and  $P(O/W)$  is the probability of observation likelihood computed using acoustic model. The following subsections contain more details of a typical speech recognition system.

### 2.2.1 Front-End signal processing

The features extraction stage aims to produce the spectral properties (features vectors) of the speech signal. These properties consist of a set (39 coefficients) of MFCCs. The speech signal is divided into overlapping short segments that will be represented using MFCCs, the widely used feature vectors for speech signals. Figure 2-2 shows the steps to extract the MFCCs of a speech signal [17].

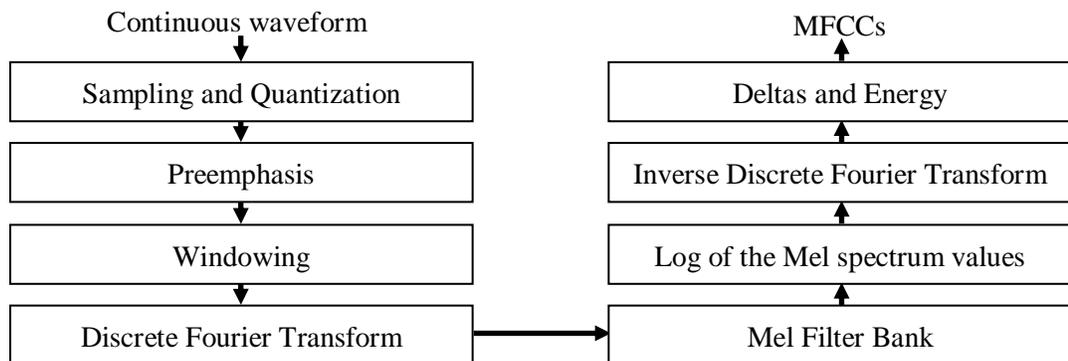


Figure 2-2 Feature vectors extraction

*Sampling and Quantization:* sampling and quantization are the two steps for analog-to-digital conversion. The sampling rate is the number of samples taken per second, while quantization is the process of representing real-valued numbers as integers.

*Preemphasis:* this stage is to boost the high frequency part that was suppressed during the sound production mechanism, so making the information more available to the acoustic model.

*Windowing:* a stationary portion of speech is extracted using a window which can be characterized by width (20~30ms), offset or optional overlap (around 10ms), frame size (around 320 sample points), and frame rate (around 100 frames per second).

*Discrete Fourier Transform:* the goal of this step is to obtain the magnitude frequency response of each frame. Therefore, the output is a complex number representing the magnitude and phase of the frequency component in the original signal.

*Mel Filter Bank:* A set of triangular filter banks is used to approximate the frequency resolution of the human ear. The Mel frequency scale is linear up to 1000 Hz and logarithmic thereafter. For 16 KHz sampling rate, Sphinx uses a set of 40 Mel filters [18].

*Log of the Mel spectrum values:* The range of the values generated by the Mel filter bank is reduced by replacing each value by its natural logarithm. This is done to make the statistical distribution of the spectrum approximately Gaussian [18].

*Inverse Discrete Fourier Transform:* This transform is used to compress the spectral information into a set of low order coefficients. This representation is called the Mel-cepstrum [18].

*Deltas and Energy*: the previous step provides the 12 cepstral coefficient for each frame. This step is to add the 13<sup>th</sup> feature: the energy from the frame. It is useful to identify phone identity.

Figure 2-3 shows the feature vector of a speech file after completing the feature extraction process. Each column represents the 13 features of a 25.6 milliseconds frame.

	1	2	3	4	5
1	-1.3030	-1.1439	-1.2332	-1.1225	-1.3957
2	-1.2602	-1.2450	-1.2588	-1.1915	-1.2575
3	-0.0487	-0.2290	-0.0345	-0.0404	-0.0538
4	-0.1191	-0.2934	-0.1264	-0.1076	-0.1466
5	0.0237	-0.1424	0.0330	0.1451	-0.0068
6	-0.0395	-0.0243	0.0356	0.0135	-0.0487
7	-0.0970	-0.0554	0.0969	0.0282	0.0642
8	0.0092	-0.0489	-0.0104	0.0079	0.1567
9	0.1431	0.0946	0.1227	0.0366	-0.0841
10	0.0034	0.0942	0.1810	0.1100	-0.0273
11	0.0925	-0.0315	0.0419	0.0439	0.0590
12	0.0653	0.0674	-0.0610	0.0697	-0.0102
13	0.0459	-0.0267	-0.0058	0.0878	0.0270
14					

Figure 2-3 MFCCs of a speech file

## 2.2.2 Acoustic model

Acoustic model is a statistical representation of the phone. Precise acoustic model is a key factor to improve recognition accuracy as it characterizes the HMM of each phone. Sphinx uses 39 English phonemes [19]. The acoustic model uses a 3- to 5-state Markov chain to represent the speech phone [14]. Figure 2-4 shows a representation of a 3-state phone's acoustic model. In Figure 2-4, S1 is the representation of phone at the beginning, while S2 and S3 is a representation of the phone at the middle and the end states, respectively. S1, S2, and S3 are mixture Gaussian densities that describe the behavior of the feature vectors of the phone.

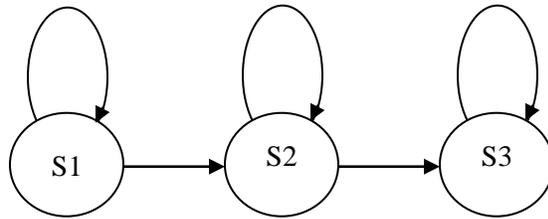


Figure 2-4 A 3-state phone acoustic model

An HMM,  $\lambda$ , is described by the following set of parameters [11]:

- The number of states  $N$ .
- The state transition probabilities,  $A$ ,  $a_{ij} = P(s_{t+1} = j | s_t = i)$ , where  $s_t$  is the state at time  $t$ .
- The observation symbol probability,  $B$ ,  $b_j(x_t) = P(x_t | s_t = j)$ , where  $x_t$  is the observation at time  $t$ .
- The initial state probabilities,  $\Pi$ .  $\pi_i = P(s_1 = i)$

In continuous speech, each phoneme is influenced in different degrees by its neighboring phonemes. Therefore, for better acoustic modeling, Sphinx uses triphones. Triphones are context dependent models of phonemes; each triphone represents a phoneme surrounded by specific left and right phonemes [20]. For example the phoneme /B/ when /EY/ appears on its left and /L/ appears on its right is the triphone /B(EY, L)/.

Sphinx uses two different techniques for parametrizing the probability distributions of the state emission probabilities: continuous HMM (CHMM), and semi-continuous HMM (SCHMM) ([21], [22], [23]). The semi-continuous technique requires substantially smaller number of parameters and is faster in decoding, but is only good for

limited vocabulary. The continuous HMM, however, uses more parameters, slower in decoding, but proves to be successful for large vocabulary applications.

In CHMM, for example, the Gaussian mixture density is used. The probability of generating the observation  $\mathbf{x}_t$  given the transition state  $j$ ,  $P(x_t | j)$  becomes

$$b_j(x_t) = p(x_t | q_t = j) = \sum_{k=1}^M w_{j,k} N_{j,k}(x_t) \quad (1)$$

Where  $N_{j,k}$  is the  $k$ -th Gaussian distribution,  $w_{j,k}$  are the mixture weights, and  $\sum_k w_{j,k} = 1$ . CHMM is the most popular method today for large vocabulary speech recognition systems. However, its main drawback is the extremely large number of parameters needed to describe the Gaussian distributions.

Reducing the number of parameters to describe all the acoustic models of all triphones can be achieved by using the concept of shared distributions [20]. In this technique, all the states of all triphones of a given phoneme share a common pool of probability distributions. These shared distributions are called Senones.

### 2.2.3 Decoding Using Viterbi algorithm

Given the acoustic model, the purpose of the decoding phase is to find the HMMs sequence that is more likely to have the observation sequence. The Baum-Welch (any path) and Viterbi (best path) are two approaches used to find the best-state sequence. The HMMs scoring the maximum are considered as the most probable sequence of the observation speech. Therefore, a basic step in recognition is to calculate the probability of observing a sequence of speech features  $X = \{x_1, x_2, \dots, x_T\}$ , given a phoneme HMMs,  $\lambda$ ,  $P(X | \lambda)$ . We need then to enumerate every possible state sequence of length  $T$ .

Consider the sequence  $S = [s_1, s_2, \dots, s_T]$ , the probability of observing such sequence of feature vectors given the model is obtained by summing up all possible state sequences of length T.

$$P(X | \lambda) = \sum_{all S} P(X | S, \lambda) P(S | \lambda)$$

$$P(X | \lambda) = \sum_{all S} \pi_{s_1} b_{s_1}(x_1) \prod_{t=2}^T a_{s_{t-1}, s_t} b_{s_t}(x_t) \quad (2)$$

Equation (2) can be efficiently calculated using an iterative procedure called Forward-Backward procedure. For isolated word recognition or recognition of limited number of sentences, Forward-Backward procedure can be performed by selecting the model of the sentence which gives the highest probability of observations. In large vocabulary system, where there could be large possibilities of phoneme sequences, a recognition procedure is needed for matching the observed sound wave with the nearest sequence of phones.

Viterbi algorithm is used to find the highest scoring state sequence,  $q = s_1, s_2, \dots, s_T$  for a given observation sequence  $X = x_1, x_2, \dots, x_t, \dots, x_T$  i.e. find  $S_{best} = \arg \max_S P(S | X)$  which is equal to:

$$\arg \{ \max_S \prod_{i=1, \dots, K} P(x_i | s_i, s_{i-1}) p(s_i | s_{i-1}) \} \quad (3)$$

Let us define  $\phi(t, i)$  to be the probability of the most likely partial state sequence or path until time t, and ending at the  $i^{\text{th}}$  state, the algorithm proceeds in the following steps ([6], [11], [23]):

**Step 1:** Initialization  $\phi(1, j) = a_{1,j} b_j(x_1)$  (4)

**Step 2:** Induction

$$\phi(t, j) = \max_i \{ \phi(t-1, i) a_{i,j} \} b_j(x_t) \quad (5)$$

$i = 1, 2, \dots, N; \text{ and } t = 2, 3, \dots, T$

$$U(t, i) = \arg \max_j \{ \phi(t-1, j) a_{i,j} \} b_j(x_t), \quad (6)$$

$j = 1, 2, \dots, N; \text{ and } t = 2, 3, \dots, M$

**Step 3:** Best Path: The maximum likelihood of the best path is then given by:

$$P(X|Model) = \varphi(N, T) = \{ \max_j \{ \phi(N, j) \} \quad j = 1, 2, \dots, n_v(M) \}$$

$$U(M, i_{best}) = \arg \max_j \{ \phi(M, j) \} \quad j = 1, 2, \dots, n_v(M) \} \quad (7)$$

**Step 4:** Backtracking

$$i_M = i_{best}$$

$$i_{t-1} = U(t, i_t); \text{ for } t = M, M-1, \dots, 2 \quad (8)$$

$$S = s_{i_1} s_{i_2} \dots s_{i_M}$$

#### 2.2.4 Training Using Baum-Welch algorithm

Training speech recognition system consists of building two models, the language model and the acoustic model. In natural language speech recognition system, the language model is statistically based model using unigram, bigrams, and trigrams of the language for the text to be recognized. On the other hand, the acoustic model builds the HMMs for all the triphones and the probability distribution of the observations for each state in each HMM.

Sphinx training tools have a set of executable and Perl scripts that cooperate to create acoustic models for Sphinx speech applications. The models can be built and configured directly using the provided scripts, or by manually running the executable.

The training process for the acoustic model consists of three phases, as shown in Figure 2-5, each phase consists of three stages (model definition, model initialization, and model training) and makes use of the output of its previous phase. The following phases are:

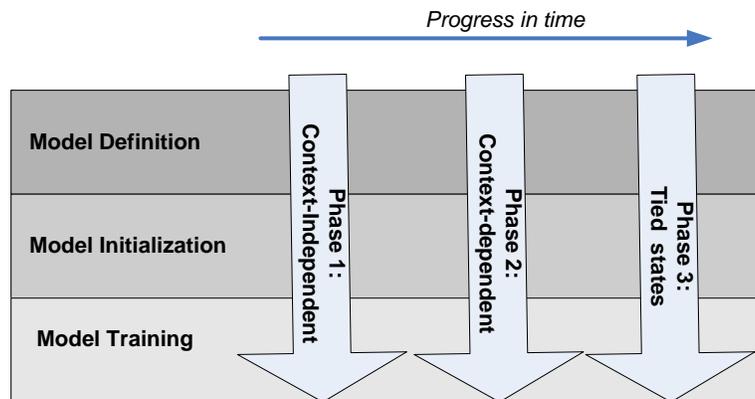


Figure 2-5 The various tasks involved in building the acoustic model

**Context-independent phase (CI):** The context-independent phase creates a single HMM for each phoneme in the phoneme list. The number of states in an HMM model can be specified by the developer; in the model definition stage, a serial number is assigned for each state in the whole acoustic model. Additionally, the main topology for the HMMs is created. The topology of an HMM specifies the possible state transitions in the acoustic model, the default is to allow each state to loop back and move to the next state; however, it is possible to allow states to skip to the second next state directly. In the model initialization, some model parameters are initialized to some calculated values. The model training stage consists of number of executions of the Baum-Welch algorithm (5 to 8 times) followed by a normalization process.

**Untied context-dependent phase (CD):** In this phase, triphones are added to the HMM set. In the model definition stage, all the triphones appearing in the training set will be created, and then the triphones below a certain frequency are excluded. Specifying a reasonable threshold for frequency is important for the performance of the model.

After defining the needed triphones, states are given serial numbers as well (continuing the same count). The initialization stage copies the parameters from the CI phase. Similar to the previous phase, the model training stage consists of number of executions of the Baum-Welch algorithm followed by a normalization process.

**Tied context-dependent phase:** This phase aims to improve the performance of the model generated by the previous phase by tying some states of the HMMs. These tied states are called Senones. The process of creating these Senones involves building some decision trees based on some "linguistic questions" provided by the developer. For instance, these questions could be about the classification of phonemes according to some acoustic property. If the user did not supply these questions, SphinxTrain could guess these questions by analyzing the voice transcriptions provided in the training data. In this research work, we used the Sphinx 3 default setting. After the new model is defined, the training procedure continues with the initializing and training stages. The training stage for this phase may include modeling with a mixture of normal distributions. This may require more iterations of Baum-Welch algorithm.

Determination of the parameters of the acoustic model is referred to as training the acoustic model. Estimation of the parameters of the acoustic models is performed using Baum-Welch Re-Estimation, which tries to maximize the probability of the

observation sequence given the model. The algorithm proceeds iteratively, starting from an initial model  $\lambda$ . The steps in this algorithm may be summarized as follows

**Step 1:** Calculate the forward and backward probabilities for all states  $j$  and times  $t$ .

**Step 2:** Update the parameters of the new model as follows:

$$\bar{\pi}_j = \text{expected frequency of the state } j \text{ at time } t=1 \quad (9)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transition from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \quad (10)$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observation symbol } x_k}{\text{expected number of times in state } j} \quad (11)$$

If for each state the output distribution is a single component Gaussian, the parameters of the distribution can be found by:

$$\bar{\mu}_j = \frac{\sum_{t=1}^T L_j(t)x_t}{\sum_{t=1}^T L_j(t)}; \quad \text{The mean value of the observation vectors emitted at state } j.$$

$$\bar{\Sigma}_j = \frac{\sum_{t=1}^T L_j(t)(x_t - \bar{\mu}_j)(x_t - \bar{\mu}_j)'}{\sum_{t=1}^T L_j(t)}; \quad \text{The covariance matrix of the observation}$$

vectors emitted at state  $j$

Where  $L_j(t)$  is probability of being in state  $j$  at the time  $t$ , given the observation sequence and the model.

**Step 3:** If the value of  $P(X | \lambda)$  for this iteration is not higher than the value at the previous iteration then stop, otherwise repeat the above steps using the new re-estimated parameter values.

### 2.2.5 Language model

Speech recognition systems treat the recognition process as one of maximum a-posteriori estimation, where the most likely sequence of words is estimated, given the sequence of feature vectors for the speech signal. Mathematically, this can be represented as [91]:

Word1 Word2 Word3 ... =

$$\operatorname{argmax}_{Wd1 \ Wd2 \ \dots} \{P(\text{feature vectors} | Wd1 \ Wd2 \ \dots) P(Wd1 \ Wd2 \ \dots)\} \quad (12)$$

Where Word1.Word2... is the recognized sequence of words and Wd1.Wd2... is any sequence of words. The argument on the right hand side of Equation (12) has two components: the probability of the feature vectors, given a sequence of words  $P(\text{feature vectors} | Wd1 \ Wd2 \ \dots)$ , and the probability of the sequence of words itself,  $P(Wd1 \ Wd2 \ \dots)$ . The first component is provided by the acoustic model. The second component, also called the language component, is provided by a language model. The most commonly used language models are N-gram language models. These models assume that the probability of any word in a sequence of words depends only on the previous N words in the sequence. Thus, a bigram language model would compute  $P(Wd1 \ Wd2 \ \dots)$  as:

$$P(Wd1 \ Wd2 \ Wd3 \ Wd4 \ \dots) = P(Wd1)P(Wd2 | Wd1)P(Wd3 | Wd2)P(Wd4 | Wd3) \dots \quad (13)$$

Similarly, a trigram model would compute it as

$$P(Wd1.Wd2.Wd3...)=P(Wd1)P(Wd2|Wd1)P(Wd3|Wd2,Wd1)P(Wd4|Wd3,Wd2).. \quad (14)$$

The N-gram language model is trained by counting N-gram occurrences in a large transcription corpus to be then smoothed and normalized. In general, an N-gram language model is constructed by calculating the following probability for all combinations that exist in the transcription corpus:

$$P(w_1^n) = \prod_{k=1}^n p(w_k | w_1^{k-1})$$

Where n is limited to include the words' history as bigram (two consequent words), trigram (three consequent words), 4-gram (four consequent words), etc. for example, by assigning n=2, the bigram is calculated for the words sequence as follows:

$$P(w_1 w_2) = p(w_2 | w_1) p(w_1)$$

The CMU statistical language tool is described in [24]. The CMU statistical language tool kit is used to generate our Arabic statistical language model. The steps for creation and testing the language model [38], shown in Figure 2-6, are as follows:

- Compute the word unigram counts.
- Convert the word unigram counts into a vocabulary list.
- Generate bigram and trigram tables based on this vocabulary.

The tool generates the language model in two formats; a binary format to be used by the Sphinx decoder, and a portable text file in the standard ARPA format.

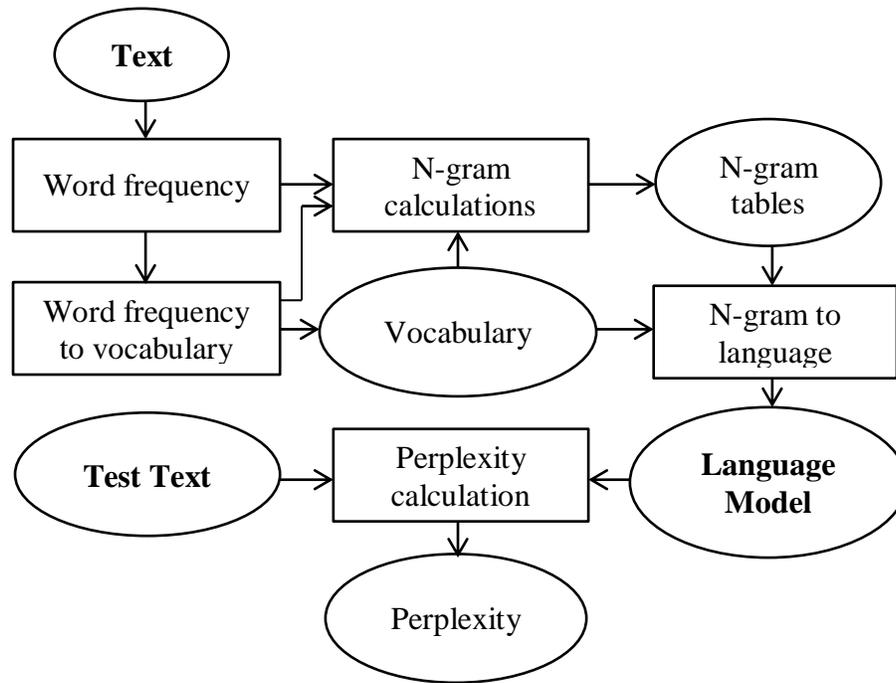


Figure 2-6 Steps for creating and testing language model

The language modeling tool comes with a tool for evaluation the language model. The evaluation measures the perplexity as indication of the goodness of the language model. For more information of the perplexity, please refer section 4.6.3 in chapter 4.

### 2.2.6 Pronunciation dictionary

Both training and recognition stages require a pronunciation dictionary which is a mapping table that maps words into sequences of phonemes. A pronunciation dictionary is basically designed to be used with a particular set of words. It provides the pronunciation of the vocabulary for the transcription corpus using the defined phoneme set. Like acoustic model and language model, the performances of the speech recognition systems depend critically on the dictionary and its phoneme set. In decoding stage, the

dictionary serves as intermediary between the acoustic model and the language model. There are two types of dictionary, closed vocabulary and open vocabulary. In closed vocabulary, all corpus transcription words are listed in the dictionary. In contrast, it is possible to have non-corpus transcription words in the open vocabulary dictionary. Typically, Phoneme set, that is used to represent dictionary words, is manually designed by language experts. However, when human expertise is not available, the phoneme set is possible to be selected using data-driven approach as demonstrated by [24]. In addition to providing the words phonemic transcriptions of the target vocabulary, the dictionary is the place where alternative pronunciation variants are added.

# CHAPTER 3

## LITERATURE REVIEW

### 3.1 Overview of speech recognition modeling techniques

The statistical approach using HMM has been the dominant technique for speech recognition systems for the last two decades. HMM-based speech recognition systems started around 1975 when James Baker applied statistical method to speech recognition ([8],[94]). Rabiner and Juang in [17] outlined the major components of a HMM-based modern speech recognition and spoken language understanding systems. Bilmes in [25] presented a list of possible HMM properties. From speech recognition point of view, Bilmes found that HMMs are extremely powerful, given enough hidden states and sufficiently rich observation distributions. Baker in [26] presented a report to survey historically significant events in speech recognition and understanding which have enabled this technology to become progressively more capable and cost effective in a growing number of everyday applications. Deng and Huang in [27] demonstrated a number of fundamental and practical limitations in speech recognition technology, which hinder ubiquitous adoption of this widely used technology. Gales and Young in [28] demonstrated that almost all present day large vocabulary continuous speech recognition (LVCSR) systems are based on HMMs. They described the various refinements which are needed to achieve state-of-the-art performance. Ye-Yi et al. in [29] categorized

spoken dialog technology into form filling, call routing, and voice search, and reviewed the voice search technology. Sainath et al. in [30] explored applying a complete LVCSR HMM-based system to a small vocabulary corpus. By taking advantage of speaker adaptation and discriminative training techniques commonly used in LVCSR systems, they achieved an error rate of 20%, the best results reported on the TIMIT corpus to date. TIMIT is a speech corpus worked on by many sites, including Texas Instruments and Massachusetts Institute of Technology (MIT). Recent results have shown that HMMs are remarkably good even for difficult conversational speech-to-text ,the latest Switchboard word error rates are at around 13% [25].

Zweig and Nguyen in [31] proposed a segmental conditional random fields (CRF) approach to large vocabulary continuous speech recognition systems. They achieved improvement of 2% compared to the HMM-based baseline. Luo in [32] proposed an improved speech recognition algorithm based on a hybrid support vector machine (SVM) and HMM architecture. The experimental results showed that the recognition rate had increased greatly. To overcome the flaws of the HMM paradigm, Xi et al. [33] designed a hybrid HMM/artificial neural networks (ANN) model where the nonparametric probabilistic model (a BP neural network) was used to substitute the Gauss blender to calculate the observed probability that was necessary for computing the states of the HMM. Sloin and Burshtein [35] presented a discriminative training algorithm that used SVMs, to improve the classification of discrete and continuous output probability HMMs. The presented algorithm used a set of maximum-likelihood (ML)-trained HMMs as a baseline system, and an SVM training scheme to rescore the results of the baseline HMMs. Xian in [36] presented the use of a hybrid HMM and ANNs for ASR. The

proposed hybrid system for ASR was to take advantage from the properties of both HMM and ANN, improving flexibility and recognition performance. Schwenk in [41] described the use of a neural network language model for large vocabulary continuous speech recognition. The underlying idea of his approach was to alleviate the data sparseness problem by performing the language model probability estimation in a continuous space. Yuecheng et al. in [42] suggested using a gating network to modulate the effects of the context to improve the performance of a neural network language model. It was found that it was a very effective way.

Beutler in [40] demonstrated a method to bridge the gap between statistical language models and elaborate linguistic grammars. He introduced precise linguistic knowledge into a medium vocabulary continuous speech recognizer. His results showed a statistically significant improvement of recognition accuracy on a medium vocabulary continuous speech recognition dictation task.

Xiao and Qin in [34] demonstrated that feature coefficients based on MFCC were not fully reflecting speech information as a result of speech signal movement and overlap of frames, especially noisy effect. They presented a new method for noise robust speech recognition based on a hybrid model of HMM and Wavelet Neural Network (WNN). Their experimental results showed a better noise robustness model. Middag et al. in [37] presented a novel methodology that utilized phonological features to assess the pathological state of the speaker using ASR. Table 3-1 shows the word error rate (WER) from state-of-the-art systems on different English pronunciation corpuses [94].

Pronunciation Corpus	Vocabulary	WER %
TI Digits	11 (zero-nine, oh)	0.5
<i>Wall Street Journal</i> read speech	5,000	3
<i>Wall Street Journal</i> read speech	20,000	3
Broadcast News	64000+	10
Conversational Telephone Speech (CST)	64000+	20

Table 3-1 Rough word error rates for a number of ASRs (English corpuses)

### 3.2 Literature of Arabic speech recognition Systems

This section presents a literature survey of Arabic speech recognition systems. Development of an Arabic speech recognition is a multidiscipline effort, which requires integration of Arabic phonetics ([43],[44],[45]), Arabic speech processing techniques ([46],[47],[45]), and natural language processing [48]. A number of researchers have recently addressed development of Arabic speech recognition systems.

Al-Otaibi in [49] provided a single-speaker speech dataset for MSA. He also proposed a technique for labeling Arabic speech. He reported a recognition rate for speaker dependent ASR of 93.78% using his technique. The ASR was built using the HTK. Hyassat and Abu Zitar in [50] described an Arabic speech recognition system based on Sphinx 4. They also proposed an automatic toolkit for building pronunciation dictionaries for the Holy Qur'an and standard Arabic language. Three corpuses were developed in Hyassat and Abu Zitar [50] work, namely, the Holy Qura'an corpus of about 18.5 hours, the command and control corpus of about 1.5 hours, and the Arabic digits corpus of less than 1 hour of speech.

A workshop was held in 2002 at John Hopkins University where Kirchhofl et al. in [51] proposed to use Romanization method for transcription of Egyptian dialectic of telephone conversations. Soltau et al. in [52] reported advancements in the IBM system for Arabic speech recognition as part of the continuous effort for the Global autonomous language exploitation (GALE) project. The system consisted of multiple stages that incorporate both diacritized and non-diacritized Arabic speech model. The system also incorporated a training corpus of 1,800 hours of unsupervised Arabic speech. Azmi et al. in [53] investigated using Arabic syllables for speaker-independent speech recognition system for Arabic spoken digits. The database used for both training and testing consisted of 44 Egyptian speakers. In a clean environment, experiments showed that the recognition rate obtained using syllables outperformed the rate obtained using monophones, triphones, and words by 2.68%, 1.19%, and 1.79%, respectively. Also in noisy telephone channel, syllables outperformed the rate obtained using monophones, triphones, and words by 2.09%, 1.5%, and 0.9%, respectively. Abdou et al. in [54] described a speech-enabled computer-aided pronunciation learning system. The system was developed for teaching Arabic pronunciations to non-native speakers. The system uses a speech recognizer to detect errors in user recitation. A phoneme duration classification algorithm was implemented to detect recitation errors related to phoneme durations. Performance evaluation using a dataset that includes 6.6% wrong speech segments showed that the system correctly identified the error in 62.4% of pronunciation errors, reported “Repeat Request” for 22.4% of the errors, and made false acceptance of 14.9% of total errors. Khasawneh et al. in [4] compared the polynomial classifier that was applied to isolated-word speaker-independent Arabic speech and dynamic time warping

(DTW) recognizer. They concluded that the polynomial classifier produced better recognition performance and much faster testing response than the DTW recognizer. Choi et al. in [55] presented recent improvements to their English/Iraqi Arabic speech-to-speech translation system. The presented system-wide improvements included user interface , dialog manager, ASR, and machine translation components. Rambow et al. in [56] addressed the problem of parsing transcribed spoken Arabic. They examined three different approaches: sentence transduction, treebank transduction, and grammar transduction. Overall, grammar transduction outperformed the other two approaches. Parsing can be used to check the speech recognizer N-best hypothesis to rescore them according to most syntactically accurate one. Nofal et al. in [57] demonstrated a design and implementation of stochastic-based new acoustic models suitable for use with a command and control system speech recognition system for the Arabic language. Park et al. in [58] explored the training and adaptation of multilayer perceptron (MLP) features in Arabic ASRs. Three schemes had been investigated. First, the use of MLP features to incorporate short-vowel information into the graphemic system. Second, a rapid training approach for use with the perceptual linear predictive (PLP) + MLP system was described. Finally, the use of linear input networks (LIN) adaptation as an alternative to the usual HMM-based linear adaptation was demonstrated. Shoaib et al. in [59] presented an approach to develop a robust Arabic speech recognition system based on a hybrid set of speech features. This hybrid set consists of intensity contours and formant frequencies. Imai et al. in [60] presented a new method for automatic generation of speaker-dependent phonological rules in order to decrease recognition errors caused by pronunciation variability dependent on speakers. Choueiter et al. in [61] concentrated their efforts on

MSA, where they built morpheme-based LMs and studied their effect on the OOV rate as well as the word error rate (WER). Bourouba et al. in [62] presented a new HMM/support vectors machine (SVM) (k-nearest neighbor) for recognition of isolated spoken words. Sagheer et al. in [63] presented a visual speech features representation system. They used it to comprise a complete lip-reading system. Taha et al. in [64] demonstrated an agent-based design for Arabic speech recognition. They defined the Arabic speech recognition as a multi-agent system where each agent had a specific goal and deals with that goal only. Elmisery et al. in [65] implemented a pattern matching algorithm based on HMM using field programmable gate array (FPGA). The proposed approach was used for isolated Arabic word recognition and achieved accuracy comparable with the powerful classical recognition system. Mokhtar and El-Abddin in [66] represented the techniques and algorithms used to model the acoustic-phonetic structure of Arabic speech recognition using HMMs. Gales et al. in [67] described the development of a phonetic system for Arabic speech recognition. A number of issues involved with building these systems had been discussed, such as the pronunciation variation problem. Bahi and Sellami in [68] presented experiments performed to recognize isolated Arabic words. Their recognition system was based on a combination of the vector quantization technique at the acoustic level and markovian modeling.

A number of researchers investigated the use of neural networks for Arabic phonemes and digits recognition ([69], [70], [59]). For example, El-Ramly et al. in [69] studied recognition of Arabic phonemes using an Artificial Neural Network. Alimi and Ben Jemaa in [71] proposed the use of a fuzzy neural network for recognition of isolated words. Bahi and Sellami in [70] investigated a hybrid of neural networks and HMMs for

NN/HMM for speech recognition. Alotaibi in [72] reported achieving high- performance Arabic digits recognition using recurrent networks. Essa et al. in [73] proposed different combined classifier architectures based on Neural Networks by varying the initial weights, architecture, type, and training data to recognize Arabic isolated words. Emami and Mangu in [74] studied the use of neural network language models (NNLMs) for Arabic broadcast news and broadcast conversations speech recognition.

Alghamdi et al. in [75] developed an Arabic broadcast news transcription system. They used a corpus of 7.0 h for training and 0.5 h for testing. The WER they obtained ranged from 14.9 to 25.1% for different types and sizes of test data. Satori et al. in [79] used Sphinx tools for Arabic speech recognition. They demonstrated the use of the tools for recognition of isolated Arabic digits. The data were recorded from six speakers. They achieved a digits recognition accuracy of 86.66%. Lamel et al. in [3] described the incremental improvements to a system for the automatic transcription of broadcast data in Arabic, highlighting techniques developed to deal with specificities (no diacritics, dialectal variants, and lexical variety) of the Arabic language. Afify et al. in [80] compared grapheme-based recognition system with explicitly modeling short vowels. They found that a short vowel modeling improves recognition performance. Billa et al. in [81] described the development of audio indexing system for broadcast news in Arabic. Key issues addressed in Billa's [81] work revolve around the three major components of the audio indexing system: automatic speech recognition, speaker identification, and named entity identification.

Messaoudi et al. in [82] demonstrated that by building a very large vocalized vocabulary and by using a language model including a vocalized component, the WER

could be significantly reduced. Elmahdy et al. in [83] used acoustic models trained with large MSA news broadcast speech corpus to work as multilingual or multi-accent models to decode colloquial Arabic. Vergyri et al. in [84] showed that the use of morphology-based language models at different stages in a large vocabulary continuous speech recognition (LVCSR) system for Arabic leads to WER reductions. To deal with the huge lexical variety, Xiang et al. in [85] concentrated on the transcription of Arabic broadcast news by utilizing morphological decomposition in both acoustic and language modeling in their system. Selouani and Alotaibi in [86] presented genetic algorithms to adapt HMMs for non-native speech in a large vocabulary speech recognition system of MSA. Saon et al. in [87] described the Arabic broadcast transcription system fielded by IBM in the GALE project. Key advances included improved discriminative training, the use of subspace Gaussian mixture models (SGMM), neural network acoustic features, variable frame rate decoding, training data partitioning experiments, unpruned n-gram language models, and neural network based language modeling (NNLMs). These advances were instrumental in achieving a WER of 8.9% on the evaluation test set. Kuo et al. in [88] studied various syntactic and morphological context features incorporated in an NNLM for Arabic speech recognition. Abushariah et al. in [90] reported the design, implementation, and evaluation of a research work for developing a high performance natural speaker-independent Arabic continuous speech recognition system. Muhammad et al. in [92] evaluated conventional ASR system for six different types of voice disorder patients speaking Arabic digits. MFCC and Gaussian mixture models (GMM)/HMM were used as features and classifier, respectively. Recognition result was analyzed for recognition for types of diseases.

### 3.3 Challenges of Arabic Speech Recognition

Arabic speech recognition faces many challenges. For example, Arabic has short vowels which are usually ignored in text. Therefore, more confusion will be added to the ASR decoder. Additionally, Arabic has many dialects where words are pronounced differently. Elmahdy et al. in [83] summarized the main problems in Arabic speech recognition which include Arabic phonetics, diacritization problem, grapheme-to-phoneme, and morphological complexity. Diacritization is represented by different possible diacritizations of a particular word. As modern Arabic is usually written in non-diacritized scripts, lots of ambiguities for pronunciations and meanings are introduced. Elmahdy et al. in [83] also showed that grapheme-to-phoneme relation is only true for diacritized Arabic script. Arabic morphological complexity is demonstrated by the large number of affixes (prefixes, infixes, and suffixes) that can be added to the three consonant radicals to form patterns. Farghaly and Shaalan in [1] provided a comprehensive study of Arabic language challenges and solutions. Lamel et al. in [3] presented a number of challenges for Arabic speech recognition such as no diacritics, dialectal variants, and very large lexical variety. Alotaibi et al. 2008 in [89] introduced foreign-accented Arabic speech as a challenging task in speech recognition. A number of Arabic speech challenges were presented in a workshop held in John Hopkins University [51]. Billa et al. 2002 in [81] discussed a number of research issues for Arabic speech recognition, e.g., absence of short vowels in written text and the presence of compound words that are formed by the concatenation of certain conjunctions, prepositions, articles, and pronouns, as prefixes and suffixes to the word stem.

# CHAPTER 4

## THE BASELINE SYSTEM

### 4.1 Introduction

This chapter presents the main components of the baseline system that was used to test the proposed method. A number of Arabic speech recognition components were described. These components include the Arabic speech corpus, Arabic phoneme set, Arabic language model, and Arabic pronunciation dictionary. The chapter also provides the details of how to build each one of these Arabic ASR components. The performance metrics (WER, Perplexity, and OOV) also provided in this chapter.

### 4.2 Arabic speech corpora

This research work utilized the large vocabulary, speaker independent, natural Arabic continuous speech recognition system developed at King Fahd University of Petroleum and Minerals (KFUPM), Alghamdi et al. in [75]. This system is based on CMU Sphinx 3 ASR system. The baseline system used 3-emitting states HMM for triphone-based acoustic models. The state probability distribution uses a continuous density of 8 Gaussian mixture distributions. The baseline system was trained using audio files recorded from several TV news channels at a sampling rate of 16 k samples per seconds. Two speech corpora were used in this work: the first speech corpus contains of

249 business/economics and sports stories (144 by male speakers, 105 by female speakers), summing up to 5.4 hours of speech. The 5.4 hours (1.1 hours used for testing) were split into 4572 files with an average file length of 4.5 seconds. The length of wave files ranges from 0.8 seconds to 15.6 seconds. An additional 0.1 second silence period is added to the beginning and end of each file. The 4572 wav files were completely transcribed with fully diacritized text. Although care was taken to exclude recordings with background music or excessive noise, some of the files still contain background noise such as low level or fainting music, environmental noise such as that of a reporter in an open area, e.g., a stadium or a stock market, and low level overlapping foreign speech, occurring when a reporter is translating foreign statements. The transcription is meant to reflect the way the speaker has uttered the words, even if they were grammatically wrong. It is a common practice in MSA and most Arabic dialects to drop the vowels at the end of words; this situation is represented in the transcription by either using a silence mark (Sukun or unvowelled) or dropping the vowel, which is considered equivalent to the silence mark. The transcription file contains 39,217 words. The vocabulary list contains 14,234 words. The baseline (first speech corpus) WER is 12.21%. using sphinx 3.

The second speech corpus summing up to 7.57 hours (0.57 hours used for testing). The recorded speech was divided into 6146 audio files. The total words in the corpus are 52,714 words, while the vocabulary is 17,236 words. other specifications are same as the first speech corpus. The Baseline (second corpus) system WER is reported at 16.04% using PocketSphinx.

### 4.3 Arabic phoneme set

Before proceeding in discussing the Arabic phoneme set, it would be appropriate for the reader if we start first by providing a Romanization [2] of the Arabic letters and diacritical marks as shown in Appendix 2. The short vowels Fatha, Damma, and Kasra are represented using a, u, and i, respectively.

A phoneme is the basic unit of speech that is used in ASR systems. Appendix 3 shows the listing of the Arabic phoneme set (40 phonemes) used in the training, and the corresponding phoneme symbols. This phoneme set is chosen based on the previous experience with Arabic text-to-Speech systems ([43], [76], [46]), and the corresponding phoneme set which was successfully used in the CMU English pronunciation dictionary [77]. Although the Arabic phoneme set was found to be good enough, we believe that this set is far from being optimal, and further work is needed to derive an optimized phoneme set for Arabic.

### 4.4 Arabic pronunciation dictionary

Pronunciation dictionaries are essential components of ASRs. They contain the phonetic transcriptions of all the vocabulary in the target domain of the conversation. A phonetic transcription is a sequence of phonemes that describes how the corresponding word should be pronounced. Ali et al. in [78] developed a software tool to generate pronunciation dictionaries for Arabic texts using Arabic pronunciation rules. We utilized this tool to generate the enhanced dictionary (i.e. after modeling cross-word problem). This tool takes care of some of within-word variation such as: The context in which the words are uttered, for example, Hamzat Al-Wasl ( ء ) at the beginning of the word and the

Ta'al marbouta ( ة ) at the end of the word, and words and letters that have multiple pronunciations due to dialect issues. They also defined a set of rules based on regular expressions to define the phonemic definition of words. The tool scans the word letter by letter, and if the conditions of a rule for a specific letter are satisfied, then the replacement for that letter is added to a tree structure that represents all the possible pronunciations for that words.

The baseline dictionary contains 14234 words (without variants) and 23840 words (with within-word variants). A sample from the developed pronunciation dictionary is listed below. This example shows the within-word variants of (أَدْنَبْرَة < > 'dinbara ), in the baseline dictionary:

أَدْنَبْرَة E AE D IH M B R AA H (default)

أَدْنَبْرَة(2) E AE D IH M B R AA T

أَدْنَبْرَة(3) E AE D IH N B R AA H

أَدْنَبْرَة(4) E AE D IH N B R AA T

#### 4.5 Arabic language model

The CMU language toolkit (Open Source Toolkit for Speech Recognition 2011,[16]) was used to build a statistical language model from the transcription of the full diacritized transcription of 5.4 hours of the audio. Table 4-1 shows the total count of 1-grams, 2-grams, and 3-grams of the Arabic baseline language model with examples. for more information of language models, please refer to section 2.2.5.

Table 4-1 N-grams in the baseline system

n-grams Type	n-grams count	Examples
1-grams	14234	أَضَحَوْ < 'DHaw أَضْعَافِ < 'D'aafi أَضْحَت < 'DHat
2-grams	32813	المَجْلِسِ الإِتِّحَادِيِّ < almajlis al'tHaadyi المَجْلِسِ العَالَمِيِّ < almajlis al'aalamyi المَجْلِسِ تَعَامُلَاتِهَا < almajlis t'amulatiha
3-grams	37771	المَعْنِيَةِ وَالتَّكْيِيدِ عَلَى < alma'niya walta'kiid 'ala المَعْنِيَةِ خَمْسَةَ مِليَارَات < alma'niya khmsh mlyarat المَعْنِيَةِ فِي المَطَار < alma'niya fy almatar

## 4.6 Performance Metrics

Three performance metrics were used to measure the performance enhancement: the word error rate (WER), out of vocabulary (OOV), and perplexity (PP).

### 4.6.1 Word Error Rate (WER):

WER is a common metric to measure performance of ASRs. WER is computed using the following formula:

$$WER = \frac{S + D + I}{N}$$

Where:

- S is the number of substitutions words errors,

- D is the number of the deletions words errors,
- I is the number of the insertions words errors,
- N is the number of words in the testing set.

The word accuracy can also be measured using WER as the following formula:

$$\text{Word Accuracy} = 1 - \text{WER}$$

#### 4.6.2 Out-Of-Vocabulary (OOV):

OOV is a metric to measure the performance of ASRs. OOV is known as a source of recognition errors, which in turn could lead to additional errors in the words that follow [93]. Hence fore, increasing OOVs plays a significant role in increasing WER and deteriorating performance. In this research work, the baseline system is based on a closed vocabulary. The closed vocabulary assumes that all words of the testing set are already included in the dictionary. Jurafsky and Martin in [94] explore the differences between open and closed vocabulary. In our method, we calculate OOV as the percentage of recognized words that are not belonging to the testing set, but to the training set. The following formula is used to find OOV:

$$OOV_{\text{Baseline system}} = \frac{|\text{Non-Testing Set Words}|}{|\text{Testing Set Words}|} * 100$$

#### 4.6.3 Perplexity (PP)

The perplexity of the language model is defined in terms of the inverse of the average log likelihood per word [95]. It is an indication of the average number of words that can follow a given word, a measure of the predictive power of the language model, [96]. Measuring the perplexity is the common way to evaluate N-gram language model. It

is a way to measure the quality of a model independent of any ASR system. The measurement is performed on the testing set. The lower perplexity system is considered better than one of higher perplexity. The perplexity formula is:

$$PP(W) = N \sqrt{\frac{1}{P(w_1, w_2, \dots, w_N)}}$$

Where PP is the perplexity, P is the probability of the word set to be tested  $W=w_1, w_2, \dots, w_N$ , and N is the total number of words in the testing set.

#### 4.7 Significance measurement

The performance detection method proposed by Plötz in [97] was used to investigate the achieved recognition results. A 95% is used as a level of confidence. The WER of the baseline system (12.21 %) and the total number of words in the testing set (9288 words ) are used to find the confidence interval  $[\epsilon_l, \epsilon_h]$ . The boundaries of the confidence interval are found to be  $[12.21 - 0.68, 12.21 + 0.68] \rightarrow [11.53, 12.89]$ . If the changed classification error rate is outside this interval, this change can be interpreted as statistically significant. Otherwise, they were most likely caused by chance.

# CHAPTER 5

## WITHIN-WORD PRONUNCIATION VARIATION MODELING

### 5.1 Introduction

The main goal of automatic speech recognition systems (ASRs) is to enable people to communicate more naturally and effectively. However, this ultimate dream faces many obstacles such as variability in speaking styles and pronunciation variations, as explored in Chapter 2. Accordingly, handling these obstacles is a major requirement to enhance ASR performance.

In speech recognition, pronunciation variation causes recognition errors in the form of insertions, deletions, or substitutions of phoneme(s) relative to the phonemic transcription in the pronunciation dictionary. Pronunciation variations which reduce recognition performance, as indicated by McAllester et al. in [98], occur in continuous speech in two types: cross-word variation and within-word variation. Within-word variations cause alternative pronunciation(s) within words. In contrast, a cross-word variation occurs in continuous speech in which a sequence of words forms a compound word that should be treated as one entity. Hofmann et al. in [99] demonstrated that

conversational speech poses high challenge to nowadays' ASR as people tend to combine or even miss words out.

The pronunciation variations are often modeled using two approaches: knowledge based and data driven. The knowledge-based approach depends on linguistic criteria that have been developed over decades. These criteria are presented as phonetic rules that can be used to find the possible pronunciation alternative(s) for word utterances. On the contrary, data-driven methods depend solely on the training pronunciation corpus to find the pronunciation variants (direct data-driven) or transformation rules (indirect data-driven). That is, the direct data-driven approach distils variants, while the indirect data-driven approach distils rules that are used to find variants. As pros and cons of both approaches, the knowledge-based approach is not exhaustive; not all of the variations that occur in continuous speech can be described, whereas obtaining reliable information using the data-driven approach is extremely difficult [100]. However, Amdal and Fosler-Lussier in [101] mentioned that there is a growing interest in data-driven methods over knowledge-based methods due to the lack of domains' expertise. Wester and Fosler-Lussier in [102] compared between knowledge- based and data-driven approaches. The comparison showed that the latter leads to more significant improvement than knowledge-based methods which lead to a small improvement in recognition accuracy. Figure 5-1 illustrates the two types of pronunciation variations and the modeling techniques. In Figure 5-1, the bold text (i.e., modeling within-word pronunciation variation using data-driven) shows the goal of this chapter.

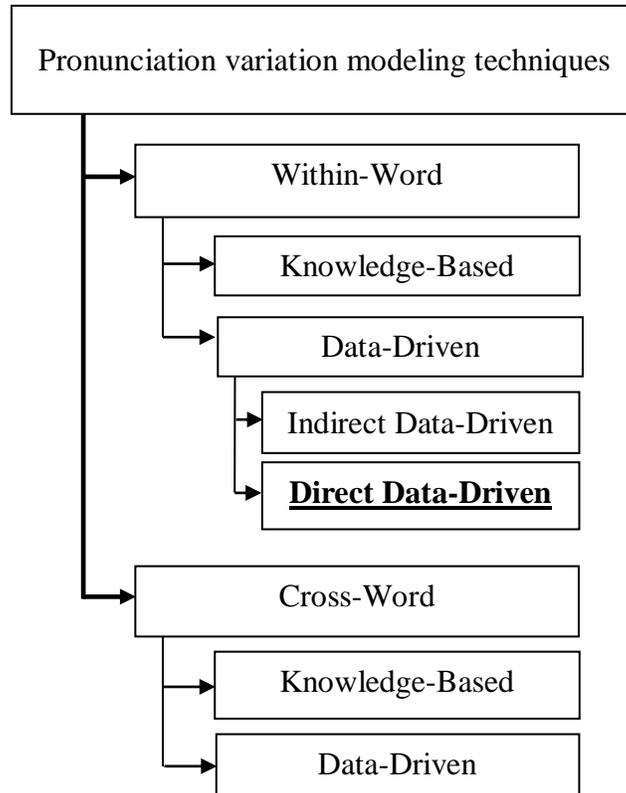


Figure 5-1 Pronunciation Variation Modeling Techniques

This chapter presents a direct data-driven approach to model within-word pronunciation variations, in which the pronunciation variants are distilled from the training speech corpus. The proposed method consists of performing phoneme recognition, followed by a sequence alignment between the observation phonemes generated by the phoneme recognizer and the reference phonemes obtained from the pronunciation dictionary. The unique collected variants are then added to dictionary as well as to the language model. Since the phoneme recognizer output has no boundary between the words, the direct data-driven approach is a good candidate to extract variants where no boundary information is present. This approach is usually used in the bioinformatics field to align gene sequences.

## 5.2 Related work

There have been many studies on modeling within-word pronunciation variations for improving ASRs performance. They are divided into two approaches: Knowledge-based and data-driven. Knowledge-based variants are derived from linguistic phonological rules, whereas data-driven variants are extracted from the pronunciation corpus. There are three levels where variants can be modeled: pronunciation dictionary, language model, and acoustic model. Helmer Strik in [103] mentioned that pronunciation variations modeling should be considered at the three mentioned levels. However, adding variants to the pronunciation dictionary is the classical approach that is usually employed, also called lexical adaptation.

Sloboda and Waibel in [104] demonstrated that having dictionaries, rich with more alternative pronunciations is a key fact in improving the performance in continuous ASRs. McAllister et al. in [98] showed that using pronunciation variations enhances the performance over the baseline system that had no variants. Another study that was performed by Fosler-Lussier et al. in [105] showed that the mismatch between the phones recognized and the word's phonetic transcription in the dictionary increases WER and degrades performance. A study was performed by Saraçlar et al. in [106] showed that the ASR performance will be highly improved if there is a closer match between the phonetic sequence recognized by the decoder and the phonetic transcription in the dictionary. Therefore, the dictionary should be carefully designed to include high quality pronunciations.

Knowledge-based approaches received great interest for modeling Arabic within-word pronunciation variations at the pronunciation dictionary level. Alghamdi et al. in

[75] developed MSA broadcast news transcription system. They used a multi pronunciations dictionary developed in [78]. Ali et al. in [78] used MSA knowledge-based method to generate Arabic multi pronunciations dictionaries for large ASRs. Al-Haj et al. in [107] demonstrated knowledge-based approach to add variants to dictionary. They worked on Iraqi-Arabic speech and focused on short vowels. Biadisy et al. in [108] showed that the use of linguistic pronunciation rules could significantly improve phone recognition and word recognition results. They developed a set of pronunciation rules that encapsulate some of MSA features for within-word variation. Billa et al. in [81] discussed a number of research issues for Arabic speech recognition, e.g., absence of short vowels in written text and the presence of compound words that are formed by the concatenation of certain conjunctions, prepositions, articles, and pronouns, as prefixes and suffixes to the word stem. While the knowledge-based ( for within-word variation) is applied in Arabic ASRs, no data-driven research work has been found.

For other languages, the knowledge-based approach for within-word variations also investigated by Tajchman et al. in [109] for ten US English phonological rules. Finke and Waibel in [110] used a set of US English phonological rules to generate pronunciation variants. Wester et al. in [100] demonstrated Dutch phonological rules to model pronunciation variations. Kessens et al. in [111] applied five optional Dutch phonological rules to the words in the baseline lexicon to generate within-word pronunciation variants. Kyong-Nim and Minhwa in [112] analyzed Korean phonological rules and implemented a rule-based pronunciation variants generator to produce a pronunciation lexicon with context-dependent multiple variants. Jeon et al. in [113] demonstrated Korean phonological rules to generate pronunciation variants. Liu and

Fung in [114] applied phonological rules to produce variants for Cantonese accented Mandarin speech. The knowledge-based approach was also implemented by Seman and Jusoff in [115] for spontaneous Standard Malay.

In spite of the advantages of using knowledge-based, Amdal and Fossler-Lussier in [101] mentioned that there is a migration from knowledge-based methods to data-driven methods due to lack of domains' expertise.

Data-driven approaches use the acoustic signal to distill pronunciation variants (direct data-driven) or the underlying rules (indirect data-driven). Amdal and Fossler-Lussier in [101] presented indirect data-driven approach for US English. Wester in [100] used the same method for Dutch. For spontaneous Standard Malay, Seman and Jusoff in [115] used decision trees as pruning method after applying the indirect data-driven approach.

With regard to the direct data-driven approaches, Sloboda and Waibel in [104] proposed a direct data-driven approach to add new German pronunciations to dictionary. They used an already existing recognizer with good performance to find new pronunciation variants by applying the recognizer to the available training speech corpus. Sloboda and Waibel [104] work is close to what we propose. However, there are two differences: we propose to extract variants using sequence alignment between reference phonemes and the observation phonemes, whereas they used speech recognizer to decode the training speech, followed by phoneme recognition to collect words with their actual pronunciations. They consider the high frequency used variants in the modeling stage. The other difference is that we generate orthographic forms of variants and represent them in the language model, instead of modeling variants in the dictionary alone.

### 5.3 Motivation

In ASRs, the canonical pronunciation is the one that is usually included in the dictionary. The ultimate goal of ASR research is to have the canonical pronunciation as close as possible to the actual pronunciation. Generally, many pronunciation variation sources cause these differences as mentioned in chapter 2. Fortunately, some of these variations can be discovered and consequently modeled to reduce its undesirable effects.

The actual pronunciation can be obtained using the Phoneme recognizer. The observed phonemes will then be compared with the reference phonemes to discover the variations. Before displaying some illustrative examples, we emphasize that our phoneme set had a thorough verification process. Therefore, the occurrence of variations in the observation phonemes as compared to the reference phonemes is unavoidable. Therefore, they are true changes that should be considered in the within-word pronunciation variation modeling. Table 5-1 shows some changes occurring in speech signals. Example 1 demonstrates a change in phoneme /L/ (ل), which was replaced by the phoneme /N/ (ن). This is an example of the phoneme substitution phenomenon. Example 2 shows that the generated variant has two changes: a new phoneme /D/ (د) is inserted, and the phoneme /UH/ (ة Damma) is switched to /IH/ (ة Kasra). Example 3 has three changes.

The orthographic form of the variant is the text form of the extracted variant. The variant's phonemes are replaced with the corresponding letters to produce the orthographic form of the variant, which is the artificially generated word that will be added to the dictionary and the language model.

Table 5-1 Within-Word Pronunciation Variations Examples

Example 1	
A vocabulary	نَسْتَلِمُ
Reference phonemes	T AE S T AE <b>L</b> IH M UH
Observed phonemes	T AE S T AE <b>N</b> IH M UH
Orthographic form	نَسْتَلِمُ
Example 2	
A vocabulary	تَقَدِّمُ
Reference phonemes	T AE Q AA D UH M AE
Observed phonemes	T AE Q AA D <b>D IH</b> M AE
Orthographic form	تَقَدِّمُ
Example 3	
A vocabulary	تَخْفِیضَاتٍ
Reference phonemes	T AE KH F IY DD AH: T IH N
Observation phonemes	T AE KH <b>TT W</b> IY DD AH: T <b>UH</b> N
Orthographic form	تَخْطُوِيضَاتِن

The Levenshtein Distance (LD) is a metric for measuring the difference between two sequences. In our case, the difference is between the observation phonemes and the reference phonemes. In Table 5-1, Example 1 has one difference and example 3 has three differences. The LD is used as a metric to accept or reject the distilled variants. If we set the LD threshold to 3, no variant with more than 3 changes, as compared to the reference phonemes, will be taken as an accepted variant.

In the proposed approach, the extracted variants will be added to the language model. One reason for adding the variants to the language model is the Viterbi limitation. Jurafsky and Martin in [94] illustrated that the Viterbi algorithm is an approximation algorithm. It actually computes an approximation of the most probable word sequence, instead of computing the most probable word sequence given the acoustic of the speech signal. In multiple pronunciations dictionaries, the Viterbi decoder finds the best phone string rather than the best word string. This means that the Viterbi algorithm is biased against words with many pronunciations. The reason for this is that the probabilities' mass is split up among different pronunciations. Thus, because the Viterbi decoder can only follow one of these pronunciation paths, it may ignore the correct word that has many-pronunciations and favor an incorrect word with only one pronunciation path. Table 5-2 illustrates the method that is usually used when modeling pronunciation variants in ASRs dictionaries including Sphinx.

Table 5-2 Pronunciation Variation Modeling Techniques

Word 1	Default pronunciation Variant 1: $vi$ Variant 2: $vi+1$ Variant 3: $vi+2$ ...
Word 2	Default pronunciation Variant 1: $vi$ Variant 2: $vi+1$ Variant 3: $vi+2$ ...
...	...
Word n	Default pronunciation Variant 1: $vi$ Variant 2: $vi+1$ Variant 3: $vi+2$ ...

Table 5-3 illustrates our proposed method. It shows that instead of having a word with many pronunciations, each variant will be considered as a single word, where we will have  $m$  words corresponding to the  $n$  words and their variants. Hence, the Viterbi approximation will not penalize any word, since all variants are considered as independent words, each with its own pronunciation.

Table 5-3 Proposed pronunciation variation technique

Word 1	Word 1 Pronunciation
Word 2	Word 2 Pronunciation
...	...
Word $m$	Word $m$ Pronunciation

#### 5.4 Dynamic Programming

Dynamic programming (DP) is a technique to design a powerful algorithm that is used to solve combinatorial optimization problems, Alsuwaiyel in [117]. The problems include: sequence alignment, traveling salesman, all-pairs shortest path, etc. In our method, we used the sequence alignment method to find the maximum similarity between two input sequences: (the reference phonemes and the observation phonemes). In order to find the maximum similarity, three scores are required: a match score, a mismatch score, and a gap score. Table 5-4 shows two sequences, the alignment between these two sequences shows 6 matches, 1 mismatch, and 2 gaps.

Table 5-4 An alignment between two sequences

Sequence 1	A	T	-	C	G	A	T	C	G
:match									
Null :gap								X	
X :mismatch									
Sequence 2	A	T	A	C	G	-	T	G	G

These scores are used to calculate the total alignment score for all possible alignments to choose the optimal score. Dynamic programming usually consists of three components: Recursive relation, Tabular computation, Traceback. The recursive relation is as follows[116]:

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(xi, yj) & (\text{match/mismatch}) \\ F(i-1, j) - d & (\text{gap in } y) \\ F(i, j-1) - d & (\text{gap in } x) \end{cases}$$

Where F is scoring matrix, d is the gap penalty, and s is the score function.

## 5.5 The Proposed Method

Obtaining variants by applying the direct data-driven approach is performed using a sequence alignment process between the observation phonemes and the reference phonemes. The sequence alignment itself is performed using a dynamic programming algorithm. The following are the steps to distill the variants directly from the training pronunciation corpus:

**Step 1:**

Observation phonemes are generated using the phoneme recognizer that generates the phonemes as they are actually pronounced without any restriction. Figure 5-2 shows the transcription of a speech file with its corresponding phonemes:

```
...  
تَبْلُغُ قِيَمَتَهَا مِليَارًا وَسَبْعَ مِئَةِ مِليُونِ دُولَارِ  
SIL T AE DH UH L UW GH UX Q IX IX...  
...
```

Figure 5-2 The baseline corpus transcription and its phonemes

Note that each observation phonemes string starts and ends with silences as it is intentionally added at the beginning and at the end of each speech file in our pronunciation corpus.

**Step 2:**

Sequence alignment is usually used to align characters without gaps. As some of our phonemes have two character representations, we convert all of these two character representations into one character representation. Therefore, we convert all observation phonemes generated in step 1 into single character representations. For example, /T/ is left as /T/, whereas we assigned /#/ to represent /DH/, as an example. The reason for this representation is that we need each phoneme to be represented as a single character. Otherwise, the sequence alignment may take part of the phoneme and leave the other, resulting in a non-phoneme character. We also remove spaces between phonemes of the

observation phonemes. The same action is taken with the dictionary reference phonemes in order to have a single character representation without gaps as illustrates in Figure 5-3.

The mapping table is found in the Appendix 4.

• • •	قِيَمَتُهُ	Q>MATAHO
• • •	قِيَمَتُهَا	Q>MATOHC
• • •	قِيَمَتُهُ	Q>MATOHO

Figure 5-3 baseline dictionary after transforming the phonemes

**Step 3:**

For all dictionary words, perform a sequence alignment between the reference phonemes and the observation phonemes. The alignment is performed only in the sentence containing the related word. For example, if I want to find the variants of (التَّنْمِيَّة); the alignment is exclusively carried out in the sentences containing this word (التَّنْمِيَّة). Therefore, we do not search for variants blindly in all observation phonemes.

Figure 5-4 shows an illustrative example.

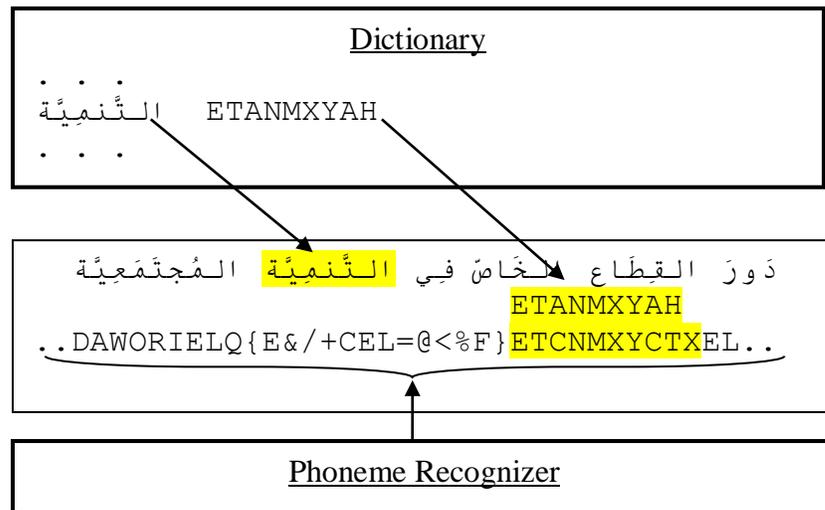


Figure 5-4 Finding variants process

#### Step 4:

For all variants collected in the previous step (step3), remove duplicates and switch phonemes back to their original forms (i.e. their two character representation, if any).

#### Step 5:

For all unique variants, generate the orthographic forms. That is, produce a new artificial word that represents the phonemes in terms of letters. For example: we have a variant for the word (التَّئِمِّيَّة) which is (E T AE: N M IH Y AE: T IH). The orthographic representation is (ءتائميئات). This new generated word will be added to the dictionary and transcription corpus in step 6.

#### Step 6:

Add the new artificially generated words to the corpus transcription by replacing each variant with its corresponding regular form. The original sentences are also added to the new transcription corpus. For example: the variant (ءتائميئات) is replaced with (التَّئِمِّيَّة) wherever it appears in the transcription. Some cases are as follows:

1 < مَعَ صُنْدُوقِ ءتائميئاتِ العَقَارِيِّ

2 < لِلْمُشَارَكَةِ فِي خُطَطِ ءتائميئاتِ

3 < سَتُرَكَّزُ فِيمَا يَبْدُو عَلَى إِسْتِثْمَارِ قَدْرِ أَكْبَرَ مِنَ الثَّرْوَةِ النَّفْطِيَّةِ فِي ءتائميئاتِ الدَّاخلِيَّةِ

4 < كَمَا يَهْدَفُ إِلَى خِدْمَةِ أَهْدَافِ ءتائميئاتِ الإِقْتِصَادِيَّةِ وَالإِجْتِمَاعِيَّةِ

5 < مِنْ عُلُومِ وَمَعَارِفِ فِي مَشَارِعِ ءتائميئاتِ فِدَاعِ الشَّمْرِيِّ وَالْمَزِيدِ مِنَ التَّفَاصِيلِ فِي سِيَاقِ التَّقْرِيرِ التَّالِيِ

Once all orthographic variants are added to the transcription corpus, we build the enhanced language model.

**Step 7:**

After decoding and before testing, we transform the variants into their regular word form, as the following example shows:

سَيَقَابِلَانِ وَجْهًا لَوْجِهِ فِي الْمُبَارَاةِ عِنْدَ عَاقِبَتِهَا

سَيَقَابِلَانِ وَجْهًا لَوْجِهِ فِي الْمُبَارَاةِ النَّهَائِيَّةِ

## 5.6 Testing and Evaluation

Initially, the following are a number of assumptions applied during testing phase: First, The sequence alignment method is good option to find variants for long words, so we performed our experiments on word lengths (WL) starting from 7 characters (including diacritics). Small words such as (في) are avoided as short sequences may introduce errors in the alignment process. Therefore, finding variants of long words such as (المُبَارَاةِ) is better than finding variants of (في). Second, We do not use the same LD threshold for all words length. We use a small LD threshold for small words and larger LD thresholds for long words. Third, We use the following sequence alignment scores: Match score=10, Mismatch score=-7, Gap score=-4.

Table 5-5 shows the recognition output achieved for different choices of LD threshold. We performed eight experiments with different specifications. The highest accuracy was found in Experiment 6 with the following specifications: the WL starts at 12 characters. For WL with 12 or 13 characters, LD = 1 or 2. This means that once a variant is found, LD should be 1 or 2 to be an accepted variant. For the other LWs in

Experiment 6, LDs are also applied in the same way. We used Experiment 6 as a representation of our enhanced system.

Table 5-5 Recognition outputs for different specifications

Experiment	1	2	3	4
	WL	WL	WL	WL
LD=1-2	7-8	8-9	9-10	10-11
LD=1-3	9-12	10-13	11-14	12-15
LD=1-4	>=13	>=14	>=15	>=16
Accuracy %	89.1	89.25	89.45	89.42
Enhancement %	1.31	1.46	1.66	1.63
Used variants	298	248	181	140
Experiment	5	6	7	8
	WL	WL	WL	WL
LD=1-2	11-12	12-13	13-14	14-15
LD=1-3	13-16	14-17	15-18	16-19
LD=1-4	>=17	>=18	>=19	>=20
Accuracy %	89.54	89.61	89.31	88.48
Enhancement %	1.75	1.82	1.52	0.69
Used variants	97	60	34	15

In Table 5-5, the used variants are the total number of variants transformed into their original forms after the decoding process. In Experiment 1, we replaced 298 variants, as an example. It should be clear that the performance is not correlated with the total number of variants used in the decoding process. Experiment 1 has the highest variants used; however, Experiment 6 has the highest accuracy achieved (1.82% reduction in WER).

Figure 5-5 shows the achieved accuracy in the eight experiments. Figure 5-5 is produced according to the data provided in Table 5-5.

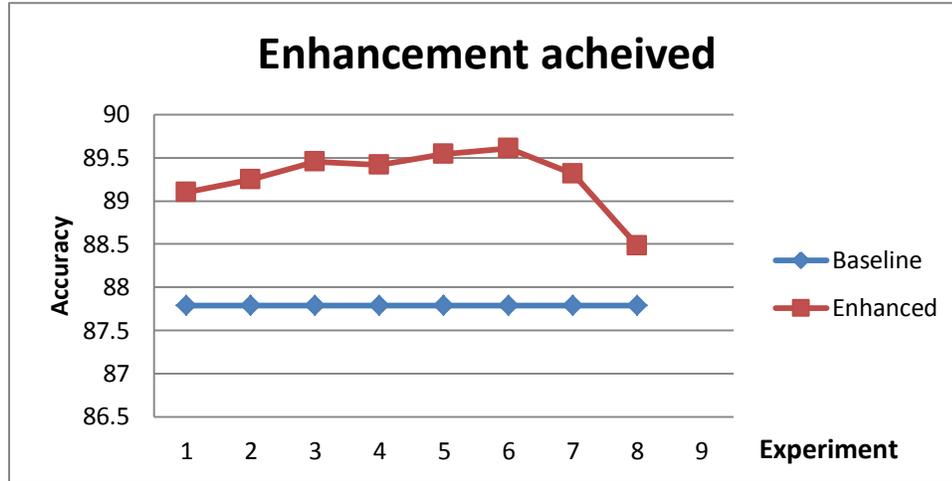


Figure 5-5 Accuracy achieved using pure data-driven variants

The maximum accuracy achieved (experiment 6) using direct-data driven approach for within-word variation is summarized in Table 5-6.

Baseline system accuracy (%)	Enhanced system accuracy (%)	WER reduction (%)
87.79	89.61	1.82

Table 5-6 the accuracy achieved using within-word modeling

Table 5-7 provides statistical information about the variants. It shows the total variants found using the proposed method. It also shows how many variants (among the total) are already found in the dictionary, alleviating the need to be accepted. After discarding the found variants, we will be left with the candidate variants that will be considered in the modeling process. After discarding the repetitions, we end up with what

we called unique variants, which will be used for modeling process. The column on right in Table 5-7 shows how many variants used (i.e. replaced back) after decoding process.

Table 5-7 Statistical information about the variants collected

Experiment	Total variants	Variants found in dictionary	Candidate variants	Unique variants	Variants used
1	7120	2965	4155	3793	298
2	5118	1901	3217	2959	248
3	3660	1224	2436	2259	181
4	2412	771	1641	1513	140
5	1533	446	1087	994	97
6	854	241	613	569	60
7	455	119	336	313	34
8	217	56	161	150	15

Table 5-7 shows that 26%-42% among suggested variants are already known to the dictionary. This metric could be used as an indicator of the selection process. In general, it should be as low as possible in order to introduce new variants. Table 5-7 also shows that 8% of the variants are discarded due to the repetitions. This repetition is an important issue in pronunciation variation modeling as it may use the highest frequency variants in the modeling process. We considered this point and collected information about the variants' frequencies as shown in Table 5-8.

Table 5-8 Variants' frequencies.

Experiment	Variants frequency							
	1	2	3	4	5	6	7	8
5	1034	38	7	3	0	1	1	3
	95%	3.5%	≈0	≈0	=0	≈0	≈0	≈0
6	584	23	4	0	0	0	1	1
	95%	3.7%	≈0	=0	=0	=0	≈0	≈0

Table 5-8 lists information from two experiments (5 and 6), which have the highest accuracy. The table shows that most variants have a one-time repetition. The table also shows that the repetition could reach 8 times for some variants. In Table 5-8, we found that three variants had repeated 8 times in Experiment 5 and 1 variant had repeated 8 times in Experiment 6. This information highlights our inability to pick only the high frequency variants, instead of taking all variants. In fact, almost all variants are repeated one time.

In order to compare our method that is based on modeling variants in the dictionary and the language model, to the method of modeling the collected variants only in the dictionary, we performed 2 experiments, 9 and 10 as shown in Table 5-9. In this case, the language model was not involved, and the baseline language model was used. We used the variants of two experiments (3 and 6) to check the performance after adding the variants as multi pronunciations words. This option is provided by Sphinx 3 such as:

وَسَبْعِينَ W AE S AE B AI IY N AE  
 وَسَبْعِينَ(1) W AE S AE B AE IY N AE  
 وَسَبْعِينَ(2) W AE S AE B IH AY N AE  
 وَسَبْعِينَ(3) W AE: S AE B AI IY N AE

Table 5-9 shows that instead of achieving improvement, the performance was less than the baseline system. This result can be justified by the notice mentioned by Helmer Strik in [103] as he stated that pronunciation variations modeling should be considered at the three ASR levels: acoustic model, the pronunciation dictionary, and the language model.

Table 5-9 Pronunciation variation modeling without language model.

Experiment	Total variants	Accuracy %	Enhancement
9	2259	86.50	No enhancement
10	569	86.55	No enhancement

We used the performance detection method suggested by Plötz in [97] to investigate the significance of the achieved enhancement. Since the enhanced method achieved a WER of (10.39%) which is out of the confidence interval [11.53,12.89] ( see chapter 4, the baseline system), it is concluded that the achieved enhancement is statistically significant.

The OOV was also measured for both systems. It was found that the baseline system has an OOV equal to 3.53%, which was reduced to 3.39% in the enhanced system. Our ASR system is based on a closed vocabulary, so we assume that there are no unknown words. The OOV was calculated as the percentage of recognized words that are not belonging to the testing set, but to the Training Set. So OOV (baseline system) = (none Testing set words) / (total words in the testing set) =  $328/9288*100= 3.53\%$ . For the enhanced system,  $OOV=315/9288*100= 3.39\%$ . Clearly, the lower OOV is better which was achieved in the enhanced system.

One common way to evaluate the N-gram language model is perplexity. It is a way to measure the quality of the language model independent of any ASR system. The perplexity for both the baseline and the enhanced language models (experiment 6) are 34.08 and 6.73, respectively. The measurement was performed on the testing set, which contains 9288 words. Therefore, the enhanced system is clearly better since lower perplexity is better. The reason why both perplexities are low is due to the specific domains that we used in our corpus(economics and sports).

The great impact on the perplexity could be understood in two ways: First, the robustness occurred in the language model increases the probability of the testing set  $W=w_1, w_2, \dots, w_N$ , therefore reducing the perplexity according the perplexity formula:

$$PP(W) = N \sqrt{\frac{1}{P(w_1, w_2, \dots, w_N)}}$$

Second, the perplexity is defined as the average number of words that can follow a given word, [96]. Accordingly, the 569 variants (in the experiment 6) added to transcription as new words have extremely low perplexities, which reduce the overall perplexities.

Figure 5-6 and Figure 5-7 provide a sample of the recognition results of the baseline and the enhanced systems. The sample contains a deletion and insertion cases, respectively.

An original speech signal to be tested	إِنَّ أَغْلَبَ الْبُنُوكِ الْقَطْرِيَّةَ قَدْ تَمَكَّنَتْ مِنْ <u>الانتهاء من الجزء</u> الأكبر من تَبْنِي
As recognized by the baseline system	إِنَّ أَغْلَبَ الْبُنُوكِ الْقَطْرِيَّةَ قَدْ تَمَكَّنَتْ مِنْ <u>الانتهاء من الجزء</u> الأكبر من تَبْنِي
As recognized by the enhanced system	إِنَّ أَغْلَبَ الْبُنُوكِ الْقَطْرِيَّةَ قَدْ تَمَكَّنَتْ مِنْ <u>لِانتهاء من الجزء</u> الأكبر من تَبْنِي
Final output after replacing the variant	إِنَّ أَغْلَبَ الْبُنُوكِ الْقَطْرِيَّةَ قَدْ تَمَكَّنَتْ مِنْ <u>الانتهاء من الجزء</u> الأكبر من تَبْنِي

Figure 5-6 A deletion case problem fixed in the enhanced system

An original speech signal to be tested	تَوَامًا لِمِصْفَاةِ الْجُبَيْلِ حَيْثُ رَاعَتْ شَرِكَةُ <u>أَرَامْكَو السُّعُودِيَّةَ</u>
As recognized by the baseline system	تَوَامًا لِمِصْفَاةِ الْجُبَيْلِ حَيْثُ رَاعَتْ شَرِكَةُ <u>أَرَامْكَو أَنْ</u> <u>السُّعُودِيَّةَ</u>
As recognized by the enhanced system	تَوَامًا لِمِصْفَاةِ الْجُبَيْلِ حَيْثُ رَاعَتْ شَرِكَةُ <u>أَرَامْكَو</u> <u>السُّعُودِيَّةَ</u>
Final output after replacing the variant	تَوَامًا لِمِصْفَاةِ الْجُبَيْلِ حَيْثُ رَاعَتْ شَرِكَةُ <u>أَرَامْكَو</u> <u>السُّعُودِيَّةَ</u>

Figure 5-7 An insertion case problem fixed in the enhanced system

Since our method artificially creates new words and adds them to the dictionary as well as to the language model, it introduces a major change in the n-grams (in term of their total and probabilities). Table 5-10 shows the differences between the baseline and the enhanced systems (experiment 6) in terms of n-grams. The enrichment that affects the language mode will lead (most likely) to a better word recognition, which in turn will

lead to another better recognition in the 2-grams and 3-grams. In contrast, error recognition of a word may lead to another error in the word sequence and so on.

Table 5-10 N-grams in the baseline and the enhanced systems

experiment	System	1-grams	2-grams	3-grams
	baseline	14234	32813	37771
6	enhanced	14803	38680	48082

Figure 5-8 provides an example of enhancement occurring in the testing speech that has no variants (indirectly positive effect of modeling pronunciation variation).

An original speech signal to be tested	و فَضْلاً عَنِ بَحْثِهِ مَشْرُوعًا لِتَحْزِينِ النَّفْطِ السُّعُودِيِّ فِي <u>الصَّيْنِ</u>
As recognized by the baseline system	و فَضْلاً عَنِ بَحْثِهِ مَشْرُوعًا لِتَحْزِينِ النَّفْطِ السُّعُودِيِّ فَاصِلِ
As recognized by the enhanced system	و فَضْلاً عَنِ بَحْثِهِ مَشْرُوعًا لِتَحْزِينِ النَّفْطِ السُّعُودِيِّ فِي <u>الصَّيْنِ</u>
No variants to be replaced	و فَضْلاً عَنِ بَحْثِهِ مَشْرُوعًا لِتَحْزِينِ النَّفْطِ السُّعُودِيِّ فِي <u>الصَّيْنِ</u>

Figure 5-8 Indirect enhancement in the enhanced system.

However, some ambiguity has been introduced in the language model. The language model is like a pool of probabilities, when new words are introduced in the language mode, it will increase some probabilities and reduce others. This is why some

correctly recognized speech in the baseline system became incorrectly recognized in the enhanced system as shown in Figure 5-9.

An original speech signal to be tested	مَوْضُوعٌ يُمَكِّنُ التَّنَطُّرُقُ إِلَيْهِ مِنْ نَوَاحٍ عَدِيدَةٍ
As recognized by the baseline system	مَوْضُوعٌ يُمَكِّنُ التَّنَطُّرُقُ إِلَيْهِ مِنْ نَوَاحٍ عَدِيدَةٍ
As recognized by the enhanced system	مَوْضُوعٌ يُمَكِّنُ تَطْوِيرَهُ مِنْ نَوَاحٍ عَدِيدَةٍ
No variants to be replaced	مَوْضُوعٌ يُمَكِّنُ تَطْوِيرَهُ مِنْ نَوَاحٍ عَدِيدَةٍ

Figure 5-9 The negative effect of recalculating n-grams

## 5.7 Execution time

The recognition time is compared with the baseline. The comparison includes the testing set, which include 1144 speech files. The specification of the machine where we conduct the experiment is as follows : a desktop computer which contains a single processing chip of 3.2GHz and 2.0 GB of RAM. We found that the recognition time for the enhanced method is larger than the recognition time of the baseline system as shown in Table 5-11. This means that the time complexity of the proposed method is a little higher than the baseline system.

Table 5-11 Recognition time of the baseline and the enhanced systems

Execution time (in minutes) for the whole testing set	
The baseline system	The enhanced system
34.14	37.06

# CHAPTER 6

## CROSS-WORD PRONUNCIATION

### VARIATION MODELING

#### 6.1 Introduction

This chapter presents the cross-word problem of the Arabic language. It also includes the main sources of this problem: Idgham (merging), Iqlaab (changing), Hamzat Al-Wasl deleting, and merging of two consecutive unvoiced letters. The chapter also presents three methods to model the cross-word problem, the methods include: phonological rules, tags merging, and small-word merging. The proposed methods are used to capture the variations occurring at words' junctures. The proposed method is illustrated in Figure 6-1. In the figure, the underlined bold text (i.e. cross-word variations) shows the subject research areas of this chapter. Figure 6-1 also distinguishes between the types of variations and the modeling techniques by a dashed line. The variation types are above the dashed line whereas the modeling techniques are under the line.

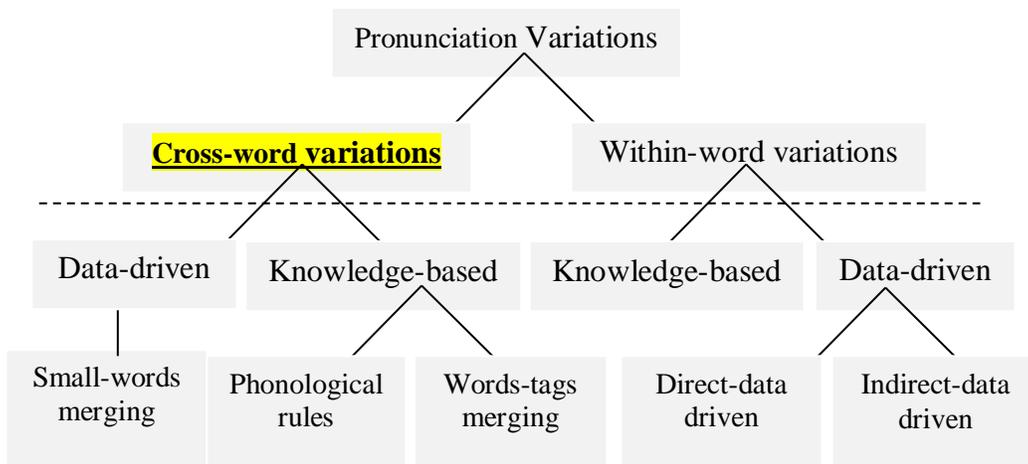


Figure 6-1 Pronunciation variations and modeling techniques

## 6.2 Effectiveness of compound-word on performance

It has been often noticed that short words are more frequently misrecognized in speech recognition system. In general, errors resulting from small words are much more than errors resulting from long words [96]. Therefore, compounding some words (small or long) to produce longer words is welcome by speech recognition decoders. Figure 6-2 shows an example. The first sentence represents the sentence to be tested, while the other sentence represents some of hypotheses that were considered during decoding process. This example shows that small words have many options, while long words are almost constant. Figure 6-2 shows that this relatively long words (دراسة فنية تنتهي) have no choices as the small words (لن), as an example. In figure 6-2, the diacritics are intentionally removed for explanation purpose. Otherwise, so many hypotheses will be displayed with no differences at words level.

لن يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الآن  
 لن يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الأمن  
 لن يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الأول  
 لن يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الأمر  
 لن يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من العام  
 لم يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الأمن  
 من يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الأول  
 بل يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من الأمر  
 لن يتم قبل إجراء دراسة فنية تنتهي بعد ستة أشهر من العام  
 ...

Figure 6-2 The difference between small and long words during decoding

The effect of compounding word was investigated by Saon and Padmanabhan in [96]. They mathematically demonstrated that compound words enhance the language model performance, therefore, enhancing the overall recognition output. They demonstrated that the compound words have the effect of incorporating a trigram in dependency in a bigram language model, as an example. In general, the compound words are most likely to be correctly recognized more than separated words. Consequently, correct recognition of a word might lead to another correct word through the enhanced n-grams language model. In contrast, error recognition of a word may lead to another error in the word sequence and so on.

### 6.3 Cross-word modeling using phonological rules

In the acoustic model, the triphones concept has been introduced to capture the phonological effects in continuous speech. Therefore, instead of training a single HMM for each phoneme, several models are trained according to the context of the phoneme. That is, each model will be trained using one preceding and following phoneme context [118]. Hazen et al. in [119] examine the advantages and disadvantages of accounting for general phonological variation explicitly with phonological rules using distinct allophonic models versus implicitly within context-dependent models.

However, this chapter attempts to model Arabic phonological rules at two ASR levels: the dictionary and the language model. In fact, we need to measure the effect of phonological rules using the same acoustic model for a baseline and an enhanced system.

Figure 6-3 shows the levels where we want to add the variants.

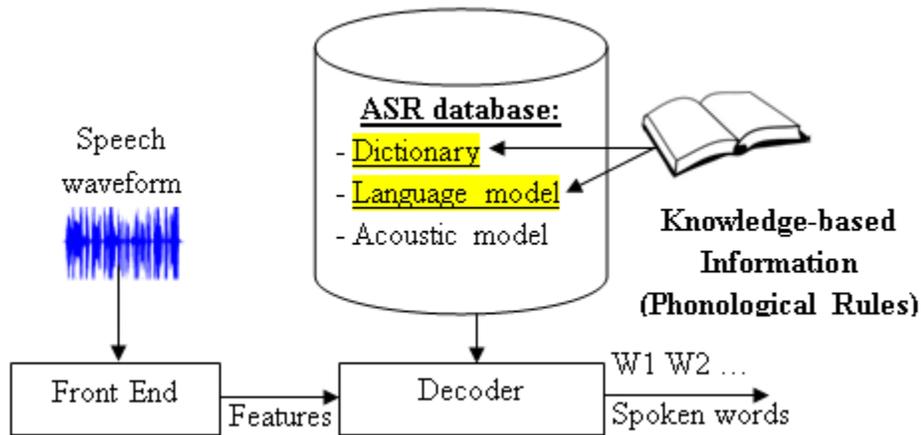


Figure 6-3 Cross-word adaptation levels

Most speech recognition systems rely on the pronunciation dictionaries that usually contain a few alternate pronunciations for most words. Additionally, the words' pronunciations in the dictionary are phonemically transcribed as if it will be uttered in isolation, which, consequently, leads to the cross-word problem. In fact, the utterance of a word in isolation is different from the same word utterance in continuous speech. The cross-word problem occurs at word junctures and is represented by coarticulation of word boundary phonemes. Figure 6-4 shows the cross-word problem that occurs at the juncture between two adjacent words (w2 and w3). The merging between w2 and w3 forms a new phoneme sequence, which the recognizer cannot match to any single word in the pronunciation dictionary. Notice that the Arabic text is read from right to left. However, we provide this example to be read as English from left to right for simplicity.

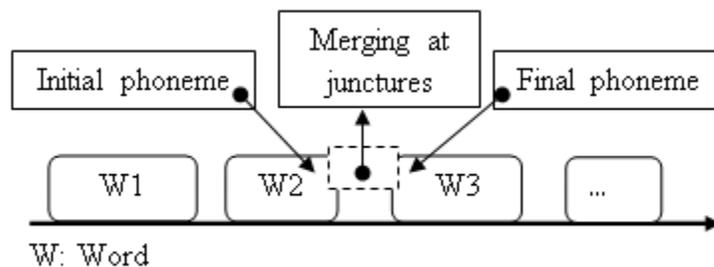


Figure 6-4 Cross-word problem

Figure 6-4 also shows that the continuous speech recognition systems face a discrimination problem when two consequent words are merged. Consequently, if the merged word is not available in the dictionary, errors may be presented in the recognition output.

With the successful use of context-dependent triphone to capture within-word and cross-word variations, the linguistic information can also be used for further enhancement

for both variation types (i.e., cross-word and within-word). The language phonological rules could predict the variation at word's junctures. Consequently, knowing the potential variations may lead to having them correctly represented in the dictionary, language model, and/or acoustic model. Certainly, four well-known Arabic phonological rules can be applied: Idgham (merging), Iqlaab (changing), Hamzat Al-Wasl deleting, and merging of two unvoiced letters. Idgham is also called assimilation, is the merging of two consecutive phonemes. Iqlaab is the replacement of one phoneme into a completely different phoneme. Even though we studied Idgham and Iqlaab of two separated words, both phenomena might occur within words as discussed by Ali et al. in [78]. Hamzat Al-Wasl is an extra Hamza that helps to start pronouncing an unvoiced letter in continuous speech. Hamzat Al-Wasl can be omitted to merge the adjacent words. To avoid the problem of meeting two unvoiced (Saakin) letters, one of them can be omitted or vowelised. In our method to model the cross-word problem, we used the Qur'an Tajweed rules as the basis of the implemented phonological rules.

### **6.3.1 Sources of cross-word problem**

The pronunciation dictionary is designed to be used with a particular set of words. However, an ASR decoder will not always be able to find a perfect match between the phonemic transcription in the dictionary and the phonetic transcription of a recognizer. This ambiguity increases the OOV, which is undesirable. OOV is a words' set of unsatisfied requests among all queries to the dictionary. In the case of unsatisfied request, another dictionary word with a nearest match pronunciation will be chosen, consequently increasing errors and reducing performance. Intuitively, to ameliorate the ASR performance, OOV should be reduced as much as possible. This reduction in OOV will

alleviate the difficulties that may rise during the decoding process. OOV problem is partially solved by extending the dictionary with some possible variants. This technique is used in modern ASRs such as Sphinx, which provide an option to add some variants such as:

أَدْنِبْرَة E AE D IH M B R AA H (default)

أَدْنِبْرَة(1) E AE D IH M B R AA T

أَدْنِبْرَة(2) E AE D IH N B R AA H

أَدْنِبْرَة(3) E AE D IH N B R AA T

Cross-word variation occurs between two separated words to produce a new compound word that, of course, is not listed in the dictionary. For example, “مِرْفَعِيهَا” is a new merged word of “مِنْ رَفَعِيهَا”, “عَمَلَايِب” is a contraction of “عَنْ مَلَايِب” and “مِمْبَبِيْن” is a coarticulation of “مِنْ بَيْن”. In general, merging, contraction, coarticulation, and compounding are alternatives. There are four main sources of cross-word pronunciation variations problem, Idgham, Iqlaab, Hamzat Al-Wasl deletion, and merging of two unvowelled letters. Idgham has three types as shown in Figure 6-5 Next chapter has more elaboration of these Arabic speech pronunciation variation phenomena. Figure 6-5 shows four reasons for cross-word merging, however, only two of them were proposed and implemented in this thesis: (Idgham and Iqlaab).

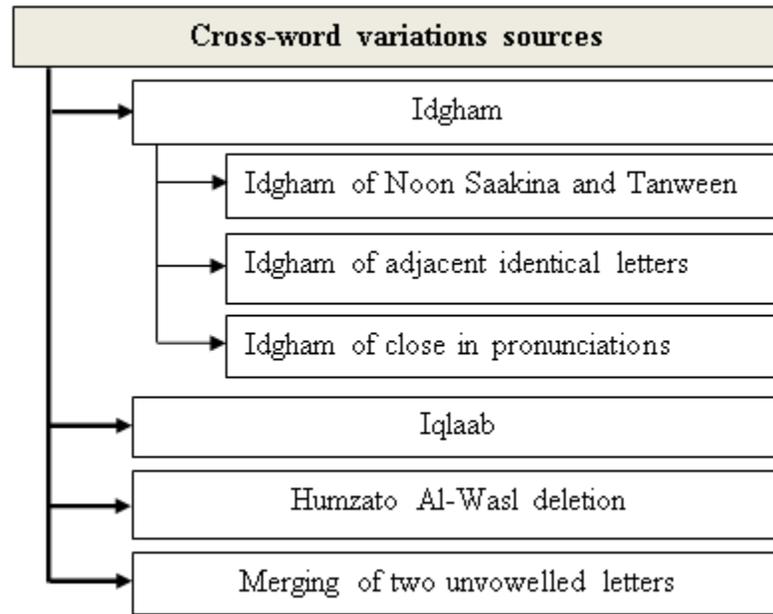


Figure 6-5 Cross-word variations sources

### 6.3.2 Arabic cross-word variations examples

In this section, we present some illustrative examples to show the effect of these variation sources. The explanation is performed with the help of the phoneme set described in chapter 4. The examples aim to disclose the phonemes variations at the word junctures. Three illustrative cases will be presented: an Idgham case (Nuun Saakina or Tanween), an Iqlaab case, and an Idgham case (close-in-pronunciation letters case). The actual speech pronunciation can be obtained using a phoneme recognizer. The phoneme recognizer output will then be compared with the canonical pronunciation to discover the resulting variations. So, a phoneme recognizer is used to produce the actual phoneme pronunciation, also called observation phonemes. Figure 6-6 shows that the phoneme /N/ (the phonemes were presented in chapter 4) is converted to phoneme /AY/. This is an Idgham case where two letters are merging to generate a double letter of the second type (i.e., /AY/).

Rule Name	Idgham ( Nuun and Yaa)
Rule Description	In Arabic, an unvowelled consonant N (نْ) at the end of a word can be merged with a vowelled consonant Y (ي) at the beginning of the next word to produce a new word with double consonant AY (يْ) at the connecting words junctures.
A speech signal with its transcription	وَأَنْ يَحْمِي الْمُسْتَهْلِك... wa'n yaHmiya almustaHlik
canonical pronunciation (Dictionary)	W AE E AE N Y AE HH M IH Y AE E L M UH S T AE H L IH K
Actual pronunciation (Phoneme recognizer)	W AE E AY Y AE HH M IH Y AE E L M UH S T AE H L IH K

Figure 6-6 The effect of Idgham in Arabic speech

Figure 6-7 shows that the phoneme /N/ is converted to /M/. This is an Iqlaab case in which one of two consequent letters is replaced while the other /B/ remains the same.

Rule Name	Iqlaab ( Nuun and Baa)
Rule Description	In Arabic, an unvowelled consonant N (نْ) at the end of a word can be merged with a vowelled consonant B (ب) at the beginning of the next word to produce a new unvowelled consonant M (مْ) at the connecting words junctures.
A speech signal with its transcription	مِنْ بَيْنِهَا سِلْتِلِ النَّابِغَةِ... min bayniha siltil altabi'a
Canonical pronunciation (Dictionary)	M IH N B AY N IH H AE: S IH L T IH L E T AE: B IH AI AE H
Actual pronunciation (Phoneme recognizer)	M IH M B AY N IH H AE: S AE L S TT R IX E L E AE T E AE B IH AI AE:

Figure 6-7 The effect of Iqlaab in Arabic speech

Figure 6-8 shows that the phoneme /T/ is converted to /D/. This is an Idgham case of two close-in-pronunciation letters.

Rule Name	Idgham two close in pronunciation letters ( Taa and Dal)
Rule Description	In Arabic, an unvoiced consonant at the end of a word Taa' (ت) can be merged with a close in pronunciation vowelled consonant Daal (د) at the beginning of the next word to produce a double consonant of the second type.
A speech signal with its transcription (A wav file)	أظهرت دراسته أعددها مجلس ... aZharat dirasatun 'a'daha majlisu
canonical pronunciation (Dictionary)	E AE DH2 H AE R AA T D IH R AA: S AE T UH N E AE AI AE D AE H AE: M AE JH L ...
Actual pronunciation (Phoneme recognizer)	E AE DH2 UH H AE: R AA D D IH R AE SS AE TT UH E N E AE AI D AE: H AE: M B ...

Figure 6-8 Idgham of two close in pronunciation case

Therefore, the one-to-one mapping that is usually used between the corpus transcription words and the dictionary entries cannot resolve the cross-word cases. As such, a technique for handling continuous speech cross-word merging is needed to achieve better performance. In the next section, we introduce the Arabic phonological rules that were considered to model the cross-word phenomenon for Arabic speech.

### 6.3.3 Arabic Phonological Rules

Arabic is a morphologically rich language in which many utterance changes can be captured by MSA phonological rules. The MSA phonological rules explained in this thesis include Idgham and Iqlaab.

In order to generate a compound word of two consecutive words, two letters are required: the final letter of the first word, and the initial letter of the second word. Modeling cross-word problem starts with the corpus transcription by searching for all cases that satisfy the modeled phonological rules. In Figure 6-9, when words w3 and w4 satisfy the constraint of a particular phonological rule, such as Idgham or Iqlaab, the two words are merged.

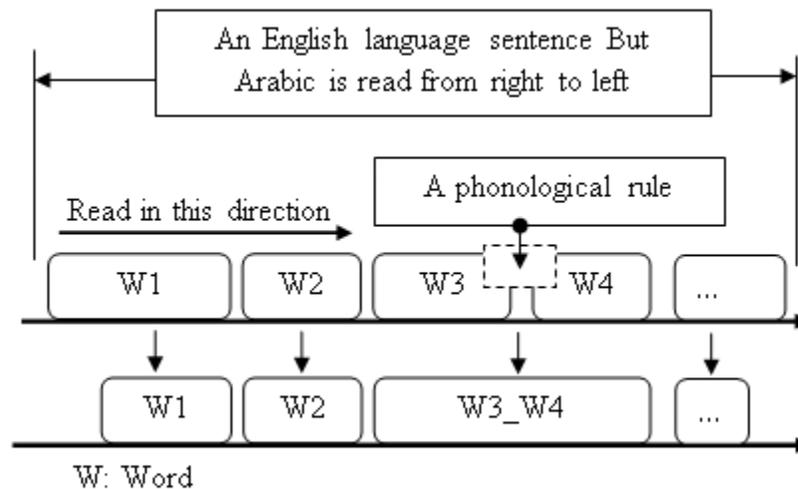


Figure 6-9 Generating a compound word

The following subsections describe the MSA phonological rules that produce the cross-word problem.

### 6.3.4 Idgham

Idgham is a merging of two consecutive letters (could be in one word or in two separated words) to produce a single geminated letter. Idgham has three different forms: Idgham of Nuun Saakina and Tanween, Idgham of two consecutive identical letters, and Idgham of two letters close in pronunciation.

#### 6.3.4.1 Idgham of Nuun Saakina and Tanween:

It is a merging between unvowelled nuun (Nuun Saakinah: نْ) or Tanween ( ُ ، ِ ، ٍ ) and one of the following consonants ( ن ، و ، ل ، م ، ر ، ي ). Table 6-1 shows examples of unvowelled nuun followed by the letters of Idgham { ن ، و ، ل ، م ، ر ، ي }. For each case in Table 6-1, the first sentence is the original sentence as it is in the corpus transcription, while the second one is the sentence after merging process. Table 6-2 provides examples only for Nuun Saakina. Tanween ( ُ ، ِ ، ٍ ) is similar.

#### 6.3.4.2 Idgham of two consecutive identical letters ( Idgham almutmathlan <> إدغام المتماثلين):

It is a merging between two consecutive identical letters shown in the following list { ن ، ل ، ك ، ق ، ف ، غ ، ع ، ظ ، ط ، ض ، ص ، ش ، س ، ز ، ر ، ذ ، د ، خ ، ح ، ج ، ث ، ت ، ب }. The rule means that any unvowelled Arabic letter followed by the same Arabic vowelled letter will be doubled in a single merged word. Note that { ا ، و ، ي } are not included in the list. (i.e. this rule is not applicable for these Arabic letters). Table 6-2 shows merging cases of consecutive identical letters.

#### 6.3.4.3 Idgham of two close in pronunciation letters (Idgham almutajanisan <> إدغام المتجانسين)

It is a merging between two consecutive different letters that are close in pronunciation. Among of these letters, we applied the following : { taa' / تْ and daal / د , taa' / تْ and Taa' / ط , daal / دْ and taa' / ت , dhaal / ذْ and Zaa / ظ , qaaf / قْ and kaaf / ك , thaa' / ثْ and dhaal / ذ , laam / لْ and raa' / ر }. Table 6-3 shows these rules with examples.

Table 6-1 Idgham cases of Nuun Saakina

The final letter Of the first word (unvowelled)	Boundary	The initial letter Of the second word (Vowelled)
نُ / nuun	space	ي / yaa'
<p>وَمِنَ الْمُتَوَقَّعِ أَنْ يَسْتَضِيْفَ أَكْثَرَ مِنْ wamina almutawaq' an yastaDiifa 'kthar min</p> <p>وَمِنَ الْمُتَوَقَّعِ أَيَسْتَضِيْفَ أَكْثَرَ مِنْ wamina almutawaq' ayyastaDiifa 'kthar min</p>		
نُ / nuun	space	ر / raa'
<p>بَعْدَ شَهْرٍ وَاحِدٍ مِنْ رَفْعِهَا لِلْحَظَرِ b'd shahrin waHidin min raf'iha lilhazr</p> <p>بَعْدَ شَهْرٍ وَاحِدٍ مِنْ رَفْعِهَا لِلْحَظَرِ b'd shahrin waHidin mirraf'iha lilhazr</p>		
نُ / nuun	space	م / miim
<p>تُجْبِرُهَا عَلَى الْإِبْتِعَادِ عَنِ مَلَاعِبِ النَّيْسِ tujbiruha 'ala al'bt'i'adi 'an mala'ibi altanis</p> <p>تُجْبِرُهَا عَلَى الْإِبْتِعَادِ عَمَّا عِبِ النَّيْسِ tujbiruha 'ala al'bt'i'adi 'ammala'ibi altanis</p>		
نُ / nuun	space	ل / laam
<p>مُؤَكَّدًا إِسْتِعَادَتُهُ بَعْضًا مِنْ لِيَاقَتِهِ الْبَدَنِيَّةِ mu'kidan 'sti 'adatahu b'dan min layaqatihi 'lbadaniya</p> <p>مُؤَكَّدًا إِسْتِعَادَتُهُ بَعْضًا مَلِيَاقَتِهِ الْبَدَنِيَّةِ mu'kidan 'sti 'adatahu b'dan milayaqatihi 'lbadaniya</p>		
نُ / nuun	space	و / waaw
<p>أَكْثَرَ مِنْ وَاحِدٍ وَسِتِّينَ مَلِيُونِ شَخْصٍ akthara min waHid wasitiin milyon shakhS</p> <p>أَكْثَرَ مَوَاحِدٍ وَسِتِّينَ مَلِيُونِ شَخْصٍ akthara miwwaHid wasitiin milyon shakhS</p>		
نُ / nuun	space	ن / nuun
<p>مَنْعَ الْجَمَاهِيرِ مِنْ نَزُولِ أَرْضِ الْمَلْعَبِ man 'a aljamahiir min nuzwl 'rd almal'ab</p> <p>مَنْعَ الْجَمَاهِيرِ مِنْزُولِ أَرْضِ الْمَلْعَبِ man 'a aljamahiir minnuzwl 'rd almal'ab</p>		

Table 6-2 Idgham of two consecutive identical letters

The final letter Of the first word (Unvowelled)	Boundary	The initial letter Of the second word (Vowelled)
س / Siin	space	س / Siin
<p>إِنَّ هَذَا الْمَجْلِسَ سَيُشْرَفُ عَلَى الثَّرْوَةِ النَّفْطِيَّةِ 'na hadha 'Imajlis sayushrifu 'ala 'ltharwati 'lnifTiya</p> <p>إِنَّ هَذَا الْمَجْلِسَ سَيُشْرَفُ عَلَى الثَّرْوَةِ النَّفْطِيَّةِ 'na hadha 'Imajlissayushrifu 'ala 'ltharwati 'lnifTiya</p>		
ع / 'ayn	space	ع / 'ayn
<p>خَاصَّةً مَعَ عَدَمِ تَوْفُرِ أَمَاكِنِ لِلْبِنَاءِ khaSatan ma' 'adam tawafur 'makin liblba'</p> <p>خَاصَّةً مَعَ عَدَمِ تَوْفُرِ أَمَاكِنِ لِلْبِنَاءِ khaSatan ma' 'adam tawafur 'makin liblba'</p>		
ل / laam	space	ل / laam
<p>التَّقْرِيرِ التَّالِيِ لِلزَّمِيلِ لُطْفِي الْمَسْعُودِي 'ltaqriir 'ltaly lilzamy lutfy almas'wdy</p> <p>التَّقْرِيرِ التَّالِيِ لِلزَّمِيلِ لُطْفِي الْمَسْعُودِي 'ltaqriir 'ltaly lilzamy llutfy almas'wdy</p>		
ت / taa'	space	ت / taa'
<p>وَبَلَّغَتْ تَكْلِفَةَ اسْتِحْوَاذِ شَرِكَةِ الْمَمْلَكَةِ wabalaghat taklifatu 'stihwadhi sharikati 'lmamlakati</p> <p>وَبَلَّغَتْ تَكْلِفَةَ اسْتِحْوَاذِ شَرِكَةِ الْمَمْلَكَةِ wabalaghattaklifatu 'stihwadhi sharikati 'lmamlakati</p>		
ف / Faa'	space	ف / Faa'
<p>الْمُتَوَقِّعِ لِلوِطَائِفِ فِي الإِقْتِصَادِ الأَمْرِيكِيِّ 'lmutawaqa ' lilwaZa'f fy 'l'iqtiSadi 'l'mryky</p> <p>الْمُتَوَقِّعِ لِلوِطَائِفِ فِي الإِقْتِصَادِ الأَمْرِيكِيِّ 'lmutawaqa ' lilwaZa'ffy 'l'iqtiSadi 'l'mryky</p>		

Table 6-3 Idgham of two close in pronunciation letters

Rule	Initial letter Of first word (Unvowelled)	Final letter Of second word (Vowelled)	Connecting letter (Double)
1	taa' / تْ	daal / د	daal / دّ
	<p>كَشَفَتْ دِرَاسَةً حَدِيثَةً أَنْ بَرِيطَانِيَا  <u>kashafat dirasatun</u> Hadythatun 'na brytanya  كَشَفَدْرَاسَةً حَدِيثَةً أَنْ بَرِيطَانِيَا  <u>kashafaddirasatun</u> Hadythatun 'na brytanya</p>		
2	taa' / تْ	Taa' / ط	Taa' / ط
	<p>تَعْتَزِمُ شَرِكَةَ طَيَّرَانَ الْإِمَارَاتِ طَلَبُ  t'tazim sharikatu Tayaran 'l'marat Talab  تَعْتَزِمُ شَرِكَةَ طَيَّرَانَ الْإِمَارَاتِ طَلَبُ  t'tazim sharikatu Tayaran 'l'maraTalab</p>		
3	daal / دّ	taa' / ت	taa' / تّ
	<p>يَقُولُ مُنْتَقِدُو هَا إِنَّهَا قَدْ تَوَجَّجَ التَّضَخْمُ  yaqwlu muntaqidwha 'naha qad tu'jiju 'ltaDakhum  يَقُولُ مُنْتَقِدُو هَا إِنَّهَا قَتَّجَ التَّضَخْمُ  yaqwlu muntaqidwha 'naha qattu'jiju 'ltaDakhum</p>		
4	dhaal / ذّ	Zaa / ظ	Zaa / ظّ
	<p>وَلَوْ أَنَّهُمْ إِذْ ظَلَمُوا أَنفُسَهُمْ  walaw 'nahum 'Z Zalamw 'nfusahm  وَلَوْ أَنَّهُمْ إِظْلَمُوا أَنفُسَهُمْ  walaw 'nahum 'ZZalamw 'nfusahm</p>		
5	qaaf / قّ	kaaf / ك	kaaf / كّ
	<p>أَعْلَنَ وَزِيرُ الْإِتِّصَالَاتِ الْمِصْرِيِّ طَارِقُ كَمَالِ  'lana wazyru 'l'tiSalat 'lmaSry Tariq kamal  أَعْلَنَ وَزِيرُ الْإِتِّصَالَاتِ الْمِصْرِيِّ طَارِكَمَالِ عَنْ طَرْحِ  'lana wazyru 'l'tiSalat 'lmaSry Tarikkamal</p>		
6	thaa' / ثّ	dhaal / ذ	dhaal / ذّ
	<p>أَوْ تَتْرُكُهُ يَلْهَثُ ذَلِكَ مِثْلَ الْقَوْمِ  'w tatruxhu yalhath dhalk mathalu 'lqawm  أَوْ تَتْرُكُهُ يَلْهَثُ ذَلِكَ مِثْلَ الْقَوْمِ  'w tatruxhu yalhathdhalk mathalu 'lqawm</p>		
7	laam / لّ	raa' / ر	raa' / رّ
	<p>التَّقْرِيرَ لِلزَّمِيلِ رَامِي إِبْرَاهِيمِ  'ltaqyr 'llzamy'ramy 'brahym  التَّقْرِيرَ لِلزَّمِيرِ رَامِي إِبْرَاهِيمِ  'ltaqyr 'llzamyrramy 'brahym</p>		

### 6.3.5 Iqlaab

Iqlaab is a replacement of Nuun Saakinah (نْ) or Tanween that comes before voweled Baa (ب) by Meem Saakinah (مْ). The following are examples of Iqlaab. Note that instead of geminating the connecting letter, it is unvoiced (مْ). Figure 6-10 shows some examples.

<p>لِلْشِتْرَاكِ فِي الْمَزَادِ الْعَالَمِيِّ مِنْ بَيْنِ سَبْعَةٍ</p> <p>lil'shtiraki fy 'Imazadi 'l'alamyi <u>min bayni</u> sab'ati</p> <p>لِلْشِتْرَاكِ فِي الْمَزَادِ الْعَالَمِيِّ مَمْبَيْنِ سَبْعَةٍ</p> <p>lil'shtiraki fy 'Imazadi 'l'alamyi <u>mimbayni</u> sab'ati</p>
<p>الْجَوْلَةُ الثَّانِيَّةُ مِنْ بَطُولَةِ الْعَالَمِ</p> <p>'ljawlati 'lthaniya <u>min buTwlati</u> 'l'alam</p> <p>الْجَوْلَةُ الثَّانِيَّةُ مَمْبِ طُولَةِ الْعَالَمِ</p> <p>'ljawlati 'lthaniya <u>mimbuTwlati</u> 'l'alam</p>
<p>تُجْبِرُهَا عَلَى الْإِبْتِعَادِ عَنِ مَلَاعِبِ التَّنِيسِ</p> <p>tujbiruha 'ala al'bt'i'adi 'an mala'ibi altanis</p> <p>تُجْبِرُهَا عَلَى الْإِبْتِعَادِ عَمَّا عِبِ التَّنِيسِ</p> <p>tujbiruha 'ala al'bt'i'adi 'ammala'ibi altanis</p>
<p>أَهْلًا بِكُمْ إِلَى النَّشْرَةِ الْاِقْتِصَادِيَّةِ</p> <p>'hlan bikum 'la 'Inashrat 'l'qtiSadiya</p> <p>أَهْلًا مَمْبِ كُمْ إِلَى النَّشْرَةِ الْاِقْتِصَادِيَّةِ</p> <p>'hlambikum 'la 'Inashrat 'l'qtiSadiya</p>

Figure 6-10 Iqlaab examples

### 6.3.6 Proposed method

In this section, we present our proposed method to model cross-word problem. The method is based on knowledge-based approach, certainly, two well-known MSA phonological rules are applied, merging (Idgham) and changing (Iqlaab). The used phonological rules were obtained from a Tajweed book written by Abdullah Heloz (2008). The modeling process includes two ASR level, the dictionary and the language mode. Therefore, the dictionary and the language model are both expanded according to the cross-word cases found in the corpus transcription. The following are the steps required in our method, the steps from 1 to 6 are offline steps ( i.e. conducting one time before recognition process), while step 7 is online step, which has to be run whenever a test file is in recognition process.

**Step 1:** Extracting the cross-word starts from the corpus transcription. Figure 6-11 shows a part of the baseline corpus transcription. In Figure 6-11, we chose small sentences for illustration purpose.

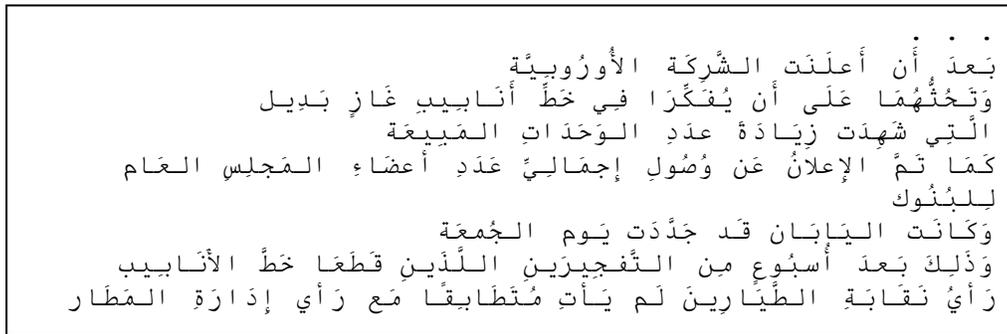


Figure 6-11 A sample of the transcription corpus used

**Step 2:** Specify the phonological rules to be applied.. In this thesis, we are interested in Idgham and Iqlaab.

**Step 3:** Using an appropriate programming language, a tool is developed to extract the compound rules from the baseline corpus transcription. In thesis, we use C as a programming language to apply our methods.

**Step 4:** After extracting the compound words using the developed C program, the compound words are then added to the corpus transcription within their sentences. Figure 6-12 shows some sentences which include compound words. Note that the original sentences (i.e., without merging) remain in the enhanced corpus transcription. In fact, we need our method to maintain both cases, merged and separated words.

<p style="text-align: center;">. . .          كَشَفْتُهُ دِرَاسَةً ُ حَدِيثَةً أَنْ بَرِيْطَانِيَا          كَشَفْتُهُ دِرَاسَةً ُ حَدِيثَةً أَنْ بَرِيْطَانِيَا          تَعْتَزِمُ شَرِكَةُ طَيْرَانِ الْإِمَارَاتِ طَلَبَ          تَعْتَزِمُ شَرِكَةُ طَيْرَانِ الْإِمَارَاتِ طَلَبَ          أَعْلَنَ وَزِيرُ الْإِتِّصَالَاتِ الْمِصْرِيِّ طَارِقُ كَمَالٍ عَنْ طَرَحِ          أَعْلَنَ وَزِيرُ الْإِتِّصَالَاتِ الْمِصْرِيِّ طَارِقُ كَمَالٍ عَنْ طَرَحِ          . . .</p>
---

Figure 6-12 A sample of the enhanced corpus transcription

**Step 5:** We use the enhanced corpus transcription generated in Step 4 to build the enhanced dictionary. Figure 6-13 shows some entries of the enhanced dictionary. The figure shows some cross-word entries, even though it contains all words of the enhanced corpus transcription (i.e., merged and non-merged words).

Partial Pronunciation Dictionary	
...	
مَرْفَعَهَا	M IH R AA F AI IH H AE:
عَمَلَابِ	AI AE M AE L AI IH B IH
مِمْبَيْنِ	M IH M B AE AY N IH
...	

Figure 6-13 A sample of the dictionary entries

**Step 6:** Build the language model according to the enhanced corpus transcription. This means that the compound words in the enhanced corpus transcription will be involved in the unigrams, bigrams, and trigrams of the language model.

**Step 7:** During recognition process, the recognition result is scanned for decomposing compound words to their original state (two separated words). This process is done using a lookup table such as:

مِرْفَعِهَا (mirraf'iha) → مِنْ رَفْعِهَا (min raf'iha)  
 عَمَلَايِب (‘ammala‘ib) → عَنْ مَلَايِب (‘an mala‘ib)  
 مِمْبَايِن (mimbayn) → مِنْ بَايِن (min bayn)

It is worth noting that each transformation case is represented in a separate sentence. For example, the following sentence:

سَتَصْرِفُ خِلَالَ أَيَّامِ رَاتِبِ شَهْرِ وَاحِدٍ لِرُبْعِ مُوظَّفِي  
 satasrifu khilala 'yamin ratiba shahrin waHidin lirub'i muwaZafyi

has been modeled using four separated sentences (the original one plus three transformation cases), as shown below.

(1) سَتَصْرِفُ خِلَالَ أَيَّامِ رَاتِبِ شَهْرِ وَاحِدٍ لِرُبْعِ مُوظَّفِي

satasrifu khilala 'yamin ratiba shahrin waHidin lirub'i muwaZafyi

(2) سَتَصْرِفُ خِلَالَ أَيَّامِ رَاتِبِ شَهْرِ وَاحِدٍ لِرُبْعِ مُوظَّفِي

satasrifu khilala **'yamratiba** shahrin waHidin lirub'i muwaZafyi

(3) سَتَصْرِفُ خِلَالَ أَيَّامِ رَاتِبِ شَهْرِ وَاحِدٍ لِرُبْعِ مُوظَّفِي

satasrifu khilala 'yamin ratiba shahrin **waHidlirub'i** muwaZafyi

(4) سَتَصْرِفُ خِلَالَ أَيَّامِ رَاتِبِ شَهْرِ وَاحِدٍ لِرُبْعِ مُوظَّفِي

satasrifu khilala 'yamin ratiba **shahrwaHidin** lirub'i muwaZafyi

the steps for modeling cross-word phenomenon can be described in the algorithm shown in Figure 6-14.

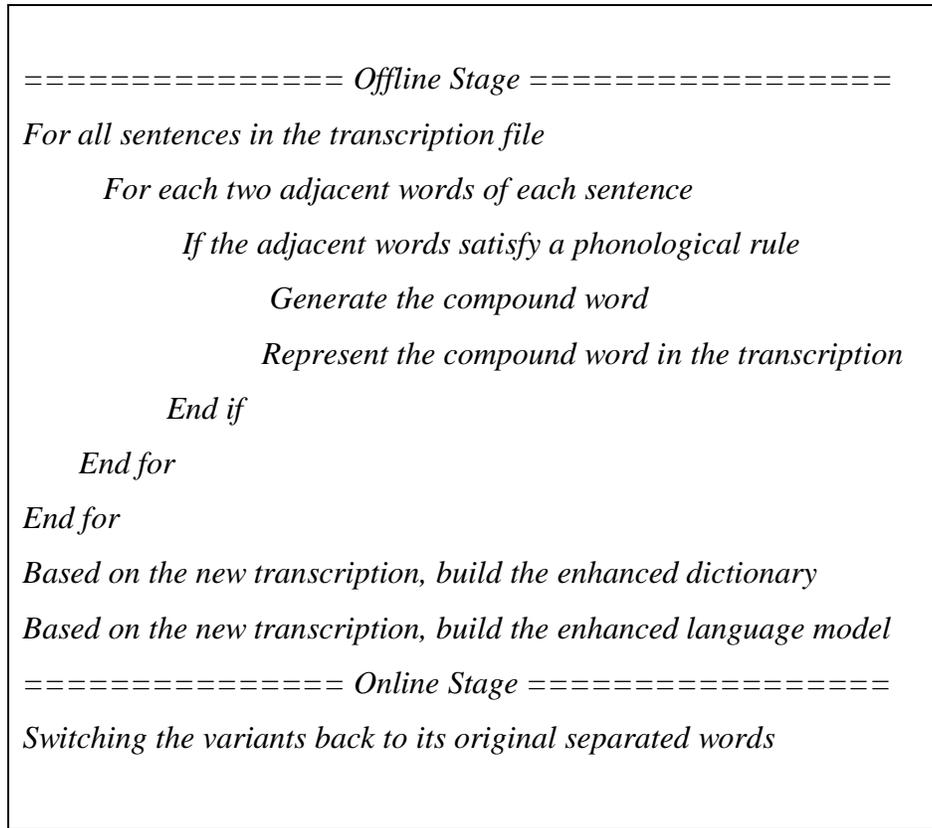


Figure 6-14 Cross-word modeling using phonological rules

### 6.3.7 Testing and evaluation

This section presents the results achieved by modeling cross-word pronunciation variation problem of MSA. We investigated two MSA phonological rules (Idgham and Iqlaab) which significantly enhanced the recognition accuracy. Three ASR's metrics were measured: word error rate (WER), out of vocabulary (OOV), and perplexity (PP).

The metrics (WER, OOV, and perplexity) explained in the previous section were measured. The enhanced system achieved a WER of 9.91% on the testing set. The WER

significant decreased by 2.3% compared to the WER of the baseline system which was 12.21%, as summarized in Table 6-4.

Table 6-4 Performance improvement using phonological rules

System	WER %
baseline	12.21
enhanced	9.91
enhancement →	2.30

The OOV was also measured for both systems. It was found that the baseline system has an OOV equal to 3.53%. The OOV was then reduced to 2.89% in the enhanced system. The OOV of both the systems (baseline and enhanced) was measured by dividing none testing set words over the total words in the testing set as follows:

$$\text{OOV (baseline system)} = \frac{\text{none testing set words}}{\text{total words in the testing set}} * 100$$

$$\text{OOV (baseline system)} = \frac{328}{9288} * 100 = 3.53\%$$

$$\text{OOV (enhanced system)} = \frac{269}{9288} * 100 = 2.89\%$$

Clearly, the enhanced system is better.

Regarding perplexity, it was measured for both systems (baseline and enhanced) and found to be 34.08 and 4.00, respectively. The measurement was performed on the testing set, which contains 9,288 words. Therefore, the enhanced system is clearly better as the lower perplexity is better. The reason why both perplexities are low is that the

specific domains of our corpus are limited to the economics and sports news. For more information about our corpus, please refer chapter 4.

The three metrics used to measure the performance clearly show that our method achieved a certain enhancement. To check whether this enhancement is statically significant, we used the performance detection method suggested by Plötz in [97]. Since the enhanced method achieved a WER of (9.91%) which is out of the confidence interval [11.53,12.89] ( see chapter 4, the baseline system), it is concluded that the achieved enhancement is statistically significant.

Appendix 5 shows some statistical information collected during the testing stages. It shows that the total cases of Idgham are 1,818 and the total cases of Iqlaab are 200. The Idgham of Nuun Saakina and Tanween is the highest to occur among all Idgham forms. This shows that Idgham occurred more than Iqlaab in MSA. Appendix 5 also shows that Lam (ﻝ) followed by Lam (ﻝ) is the highest frequency to occur in Idgham of identical latter. It has showed up 49 times in the corpus transcription. Other statistical information collected during testing stage is available in Appendix 5.

### **6.3.8 Execution time**

The recognition time is compared with the baseline. The comparison includes the testing set, which include 1144 speech files. The specification of the machine where we conduct the experiment is as follows: a desktop computer which contains a single processing chip of 3.2GHz and 2.0 GB of RAM.

We found that the recognition time for the enhanced method is less than the recognition time of the baseline system as shown in Table 6-5. This means that the proposed method is better than baseline system in term of time complexity. From decoder

point of view, it is much better to use one compound word instead of using two separated words. therefore, discarding half of the overhead needed when using one long word.

Table 6-5 Execution time comparison of the enhanced and the baseline systems

Execution time (in minutes) for the entire testing set	
The baseline system	The enhanced system
34.14	33.49

Even though 2,018 compound words have been found in the corpus, only 1,639 compound words have been actually added to the dictionary after excluding the repetition. Figure 6-15 to 6-17 provide samples of the recognition results of the baseline and the enhanced systems. The samples show how the added compound words help to improve the performance.

Original speech to be tested	بَعْدَ شَهْرٍ وَاحِدٍ مِنْ رَفْعِهَا لِلْحَظْرِ b'd shahrin wahidin min raf'iha lilhazr
As recognized by the baseline system	بَعْدَ شَهْرٍ وَاحِدٍ رَفْعِهَا لِلْحَظْرِ b'd shahrin wahidin raf'iha lilhazr
As recognized by the enhanced system	بَعْدَ شَهْرٍ وَاحِدٍ مِرْفَعِهَا لِلْحَظْرِ b'd shahrin wahidin mirraf'iha lilhazr
Final output after decomposing the merging	بَعْدَ شَهْرٍ وَاحِدٍ مِنْ رَفْعِهَا لِلْحَظْرِ b'd shahrin wahidin min raf'iha lilhazr

Figure 6-15 Idgham case: unvowelled nuun (nuun Saakinah) followed by raa'

Original speech to be tested	تُجْبِرُهَا عَلَى الْإِبْتِعَادِ عَنْ مَلَاعِبِ tujbiruha 'ala al'bti'adi 'an mula'ib
As recognized by the baseline system	تُجْبِرُهَا الْإِبْتِعَادِ عَنْ اللَّاعِبِ tujbiruha al'bti'adi 'an 'lla'ib
As recognized by the enhanced system	تُجْبِرُهَا عَنِ الْإِبْتِعَادِ عَمَّا لَاعِبِ tujbiruha 'an al'bti'adi 'ammula'ibi
Final output after decomposing the merging	تُجْبِرُهَا عَنِ الْإِبْتِعَادِ عَنْ مَلَاعِبِ tujbiruha 'an al'bti'adi 'an mula'ibi

Figure 6-16 Idgham case: unvowelled nuun (nuun Saakinah) followed by miim

Original speech to be tested	لِلْإِشْتِرَاكِ فِي الْمَزَادِ الْعَالَمِيِّ مِنْ بَيْنِ lil'shtiraki fy 'lmazadi 'l'alamyi min bayn
As recognized by the baseline system	لِلْإِشْتِرَاكِ فِي الْمَزَادِ الْعَالَمِيِّ بَيْنِ lil'shtiraki fy 'lmazadi 'l'alamyi bayn
As recognized by the enhanced system	لِلْإِشْتِرَاكِ فِي الْمَزَادِ الْعَالَمِيِّ مِمْبَيْنِ lil'shtiraki fy 'lmazadi 'l'alamyi mimbayni
Output after decomposing the merging	لِلْإِشْتِرَاكِ فِي الْمَزَادِ الْعَالَمِيِّ مِنْ بَيْنِ lil'shtiraki fy 'lmazadi 'l'alamy min bayni

Figure 6-17 Iqlaab case: unvowelled nuun (nuun Saakinah) followed by baa'

During recognition, 117 compound words were provided by the enhanced dictionary. After recognition process, these compound words were switched back to its separated form. However, this does not mean that they were misrecognized in the baseline system. Many of them were correctly recognized in the baseline system as separated words.

For more clarification, we carefully analyzed the recognizer outputs. We measured the percentage of recognition in both systems among all tested files. Table 6-6 shows that the proposed method leads to improvement in some speech files and, however, to decrease in performance in others. Figure 6-18 demonstrates the information provided in Table 6-6 in Pie chart.

Table 6-6 A comparison between the baseline and the enhanced systems

Among the 1144 speech files	Recognized files in (baseline, enhanced)
1047 speech files ( 91.5% )	Both systems (the baseline and the enhanced) agreed upon recognition of these files, either correctly or incorrectly (We ignored light diacritic differences).
23 speech files ( 2.01% )	Recognized correctly in the baseline system but are not in the enhanced system.
74 speech files ( 6.46% )	Recognized correctly in the enhanced system but are not in the baseline system.

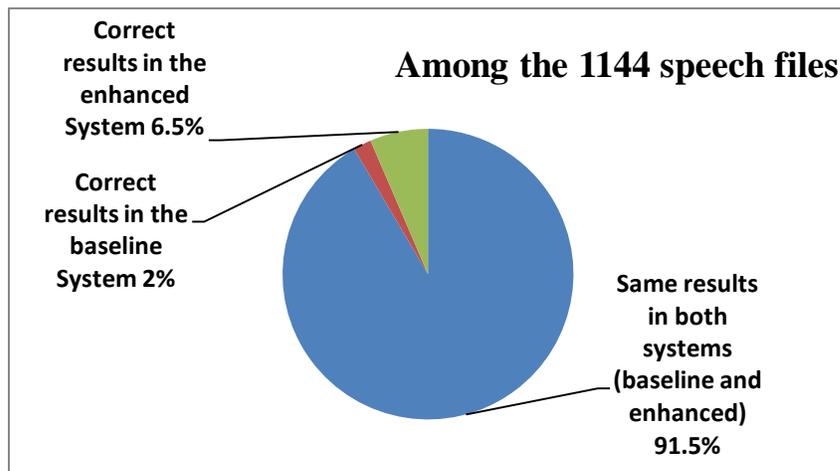


Figure 6-18 A comparison between the baseline and the enhanced systems

We mentioned in Table 6-6 that some correctly recognized words in the baseline were misrecognized in the enhanced system. The following are two illustrative examples listed in the following order: original speech to be tested, baseline system recognition results, and enhanced system recognition results, respectively.

فَسَيُتْرَكُ قَرَارُ التَّخْصِيصِ لِهَيْئَةِ سُوقِ الْمَالِ

fasayutraku qararu altakhsiisi lihy'ti swq 'lmal

فَسَيُتْرَكُ قَرَارُ التَّخْصِيصِ لِهَيْئَةِ سُوقِ الْمَالِ

fasayutraku qararu altakhsiisi lihy'ti swq 'lmal

فَسَيُتْرَكُ قَرَارُ التَّخْصِيصِ فِي لِهَيْئَةِ سُوقِ الْمَالِ

fasayutraku qararu altakhsiisi lihy'ti swq 'lmal

\* \* \* \* \*

الَّتِي لَدَيْهَا صَرَافَاتُ آلِيَّةٍ أَوْ مُصْدِرَةٌ لِلِبَطَّاقَاتِ الذَّكِّيَّةِ

'laty ladyha Sarafatun 'liya 'w muSadira lilbitaqati aldhakiya

الَّتِي لَدَيْهَا صَرَافَاتُ آلِيَّةٍ أَوْ مُصْدِرَةٌ لِلِبَطَّاقَاتِ الذَّكِّيَّةِ

'laty ladyha Sarafatun 'liya 'w muSadira lilbitaqati aldhakiya

الَّتِي لَدَيْهَا صَرَافَاتُ آلِيَّةٍ أَوْ الدَّيْنِ مُصْدِرَةٌ لِلِبَطَّاقَاتِ الذَّكِّيَّةِ

'laty ladyha Sarafatun 'liya 'w 'ldayn muSadira lilbitaqati aldhakiya

We noticed that most of the errors that occur in the enhanced system (i.e., they are correct in the baseline) have no relation with compound words. None of them made cross-word transformation process. We believe that the source of these errors is the language model as it is recalculated according to the enhanced corpus transcription. Recalculation of the language model probabilities according to the new transcription

presented a major change in the n-gram probabilities. Table 6-7 shows the total count of 1-grams, 2-grams, and 3-grams of the language model for both the baseline system and the enhanced system. So, the new language model might be biased to some word sequences on the account of others.

Table 6-7 N-grams of both systems (baseline and enhanced)

System	1-grams	2-grams	3-grams
baseline	14234	32813	37771
enhanced	15873	37852	45858

According to the data provided in Table 6-7, we found that n-grams have been increased according to the compound words. This increase in the total of n-grams will provide an opportunity for enhancement. Saon and Padmanabhan in [96] showed mathematically that compound words will enhance the performance. They demonstrated that the compound word has the effect of incorporating a trigram in dependency in a bigram language model, as an example. Generally, compound words are most likely to be correctly recognized more than separated words. Consequently, correct recognition of a word might lead to another correct word through the enhanced n-gram language model. In contrast, misrecognition of a word may lead to another error in the word sequence and so on.

Table 6-8 gives an example of the robustness we described above which leads to indirect enhancement. It shows the enhancement of a sentence that has no transformation

process, i.e., the enhancement is there while there is no cross-word phenomenon in the sentence to be tested.

Table 6-8 Samples of indirect improvements by the language model

<p>Original speech to be tested</p>	<p>من العَازِ الإيرانيِّ إلى الهند min 'lghaz 'l 'iirany 'la 'lhind وَمُمْتَلِينَ عَن عَدَدٍ مِنَ الدُّوَلِ الأوروپِيَّةِ wamumathiliina 'an 'adadin min 'lduwal 'l'wrwbiya بِمَرَضِ جُنُونِ البَقَرِ bimaraD junwn 'lbaqar</p>
<p>As recognized by the baseline system</p>	<p>من العَازِ الإيرانيِّ إلى الخَلْبَةِ min 'lghaz 'l 'iirany 'la 'lHalaba وَمُمْتَلِينَ عَن إِنَّ الدُّوَلِ الأوروپِيَّةِ wamumathiliina 'an 'na 'lduwal 'l'wrwbiya في بِمَرَضِ جُنُونِ النَّقَاعِ fy bimaraD junwn 'ltaqa 'ud</p>
<p>As recognized by the enhanced system</p>	<p>من العَازِ الإيرانيِّ إلى الهند mn alghaz alayrany ala alhnd وَمُمْتَلِينَ عَن عَدَدٍ مِنَ الدُّوَلِ الأوروپِيَّةِ wamumathiliina 'an 'adadin min 'lduwal 'l'wrwbiya في بِمَرَضِ جُنُونِ البَقَرِ fy bimaraD junwn 'lbaqar</p>

We can conclude that the new language model, generated by the expanded transcription, introduces both improvement and ambiguity. This is why 2.01% among testing files were misrecognized in the enhanced system.

Although our method enhanced the overall performance of the speech recognizer, however, we have observed a few cases in which the application of the method created misrecognition cases, which were properly recognized before. The performance enhancement together with the introduction of new errors is related to the language model's n-grams recalculation. It is clear that the more cross word cases we append to the language model, the more cross-word errors we remove from the error set, though not in a linear proportion. In the meantime, the modification in the language model may negatively change the n-gram probabilities of some words, leading to new recognition errors. This phenomenon may raise a question for further research about possible optimality of the modified language model, a language model that makes the best compromise between removing the cross-word errors, and generation of other errors

The great impact on the perplexity could be understood in two ways: first, the robustness that occurred in the language model increases the probability of the testing set  $W = w_1, w_2, \dots, w_N$ , therefore reducing the perplexity according to:

$$PP(W) = N \sqrt{\frac{1}{P(w_1, w_2, \dots, w_N)}}$$

The perplexity formula explained in chapter 4.

According to the formula, it is clear that increasing P will reduce the PP. Second, the 1,639 compound words added to the transcription as new words have an extremely low perplexity. For example, consider the two words (من) and (بعد). These two words have an average certain perplexity. When the compound word (مبعد) is represented in the language model, it will share others with its low perplexity, so reducing the overall perplexities. Finally, our method was implemented as a preprocess step to extend the

span of the dictionary and the language model. The training stage has not evolved, i.e., the acoustic models of all training utterances have not been changed during the experiment.

#### 6.4 Cross-word modeling using Part of Speech Tagging

One major source of suboptimal performance in automatic continuous speech recognition systems is misrecognition of small words. In general, errors resulting from small words are much more than errors resulting from long words. Therefore, compounding some words (small or long) to produce longer words is welcome by speech recognition decoders.

Therefore, we expect that if we compound some words as one word, better performance could be achieved. We consider two pronunciation cases: nouns followed by an adjective, and prepositions followed by any word. Our proposed method is not restricted to small words, but any word length satisfying the aforementioned two word sequences: <noun, adjective> and <preposition, any word>.

Figure 6-19 shows the merging that occurs between two words: a noun (مُنَافَسَةٌ) and an adjective (شَدِيدَةٌ). The first row shows the waveform of an Arabic sentence with its text form. The dashed line in the waveform indicates the boundary of these two words. In the second row, we enlarged the waveform of these two words for more elaboration, to show the connection spot between these two words. It is clear that the connection spot is not silence. In fact, we checked many Arabic speech waveforms and found that nouns followed by adjectives are usually pronounced together as one compound word.

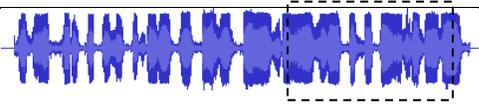
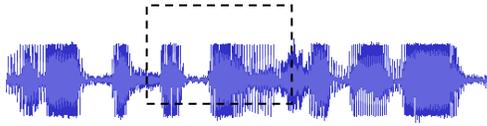
A waveform with its text form for a sentence	 <p>تَشْهَدُ حَرَكَةُ الْإِثْصَالَاتِ فِيهَا إِنْتِعَاشًا وَمُنَافَسَةً شَدِيدَةً</p>
A noun followed by an adjective waveform	 <p>وَمُنَافَسَةً شَدِيدَةً</p>
The boundary spot waveform	 <p>No silence in the boundary between the noun and the adjective.</p>

Figure 6-19 A connection spot between a noun and an adjective

#### 6.4.1 Proposed method

Our proposed method is based on the Arabic tags that are generated by the Stanford Arabic tagger, which consists of 29 tags as shown in Table 6-9. Since the scope of our work is focused on adjectives, nouns, and prepositions, only the first 13 tags listed in Table 6-9 were examined. In Table 6-9, DT is a shorthand for the determiner article ( ال ) (التعريف) that corresponds to "the" in English.

Table 6-9 also shows that nouns and adjectives have many forms, all of which were considered in our method. In this thesis, we will use the Noun-Adjective as shorthand for a compound word generated by merging a noun and an adjective. We also use preposition-word as shorthand for a compound word generated by merging a preposition with a subsequent word. The prepositions used in our method include: ( من ، )

(الى ، عن ، على ، في ، حتى ، منذ). Other prepositions were not included as they are rarely used in MSA. Table 6-10 shows the tagger output for a simple sentence.

Table 6-9 The Arabic tags of Stanford Tagger.

#	Tag	Meaning with examples
1	ADJ_NUM	Adjective, Numeric السابع،الرابعة
2	DTJJ	DT + Adjective النفطية،الجديد
3	DTJJR	Adjective, comparative الكبرى،العليا
4	DTNN	DT + Noun, singular or mass المنظمة،العاصمة
5	DTNNP	DT + Proper noun, singular العراق،القاهرة
6	DTNNS	DT + Noun, plural السيارات،الولايات
7	IN	Preposition or subordinating conjunction حرف جر مثل : في حرف مصدري مثل : أن
8	JJ	Adjective جديدة،قيادية
9	JJR	Adjective, comparative أدنى،كبرى
10	NN	Noun, singular or mass إنتاج، نجم
11	NNP	Proper noun, singular أوبك،لبنان
12	NNS	Noun, plural توقعات،طلبات
13	NOUN_QUANT	Noun, quantity الربع، ثلثي

Table 6-10 An Arabic sentence and its tags

An input sentence to the tagger	وَدَرَجَةٌ رِجَالٍ الْأَعْمَالِ فِي مَطَارِ الْكُوَيْتِ الدُّوَلِيِّ
Tagger output (read from left to right)	ودرجة/NN رجال/NN الأعمال/DTNN في/IN مطار/NN الكويت/DTNNP الدولي/DTJJ

The tagger output is used to generate compound words by searching for noun-adjective and preposition-word sequences. Table 6-10 shows two possible compound words: (الكويتالدولي) and (فيمطار) for noun-adjective case and for preposition-word case, respectively. These two compound words are, then, appended to the baseline dictionary. Additionally, these two compound words are also represented in the language model. Modeling the compound words in the language model require adding them to the baseline transcription corpus. Note that the original sentence (without compound words) also exists in the baseline transcription corpus. The following two new sentences are appended in the baseline transcription corpus to fulfill the compound words representation:

وَدَرَجَةٌ رِجَالٍ الْأَعْمَالِ فِي مَطَارِ الْكُوَيْتِ الدُّوَلِيِّ  
وَدَرَجَةٌ رِجَالٍ الْأَعْمَالِ فِيْمَطَارِ الْكُوَيْتِ الدُّوَلِيِّ

Figure 6-20 highlights the process of reading a tagged Arabic sentence, generating a compound word upon encountering a noun followed by an adjective. The preposition-word case is handled similarly. It is noteworthy to mention that our method is independent from handling pronunciation variations that may occur at words junctures.

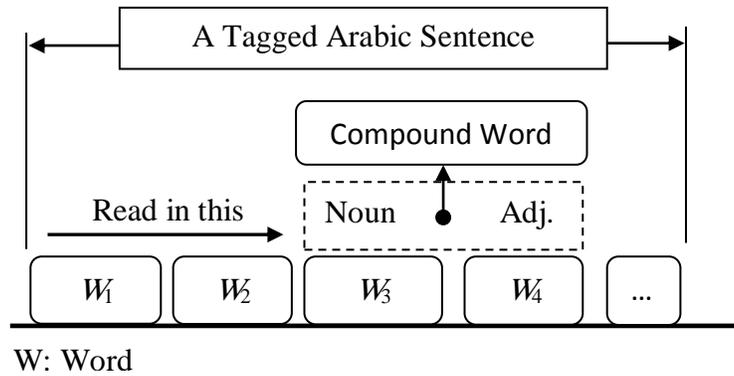


Figure 6-20 A noun-adjective compound word generation

The steps for modeling cross-word phenomenon can be described by the algorithm shown in Figure 6-21.

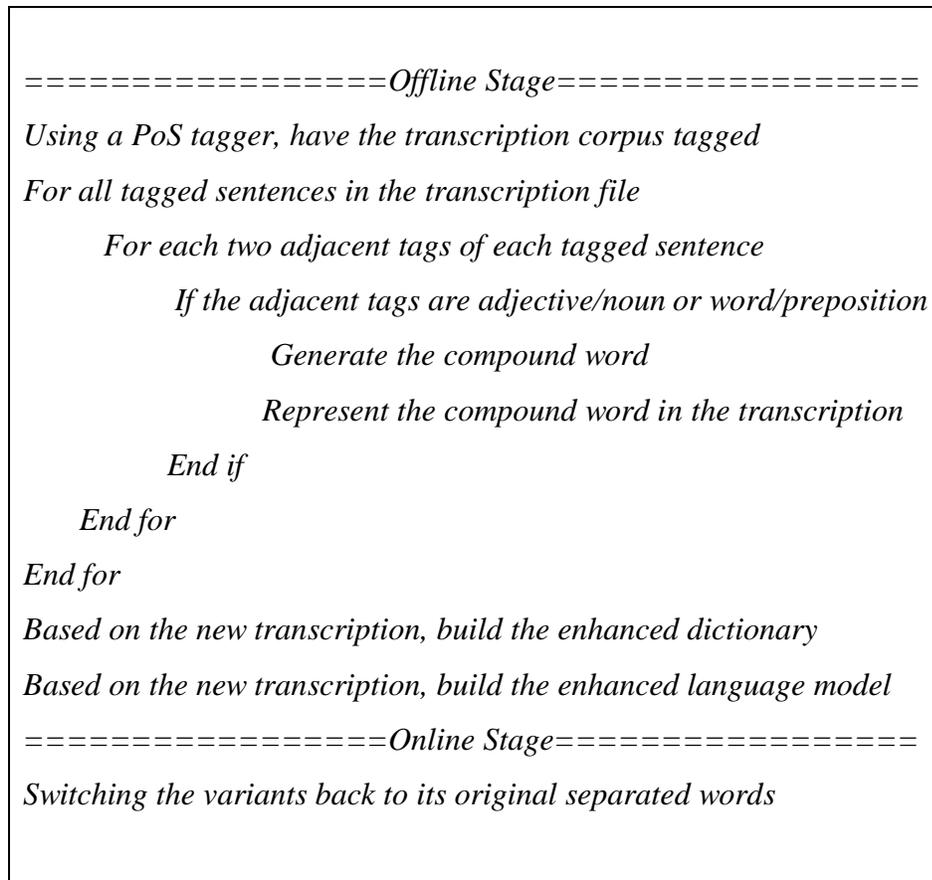


Figure 6-21 Cross-word modeling algorithm using tags merging

## 6.4.2 Testing and evaluation

Table 6-11 shows the enhancements for different experiments. To check whether the achieved enhancement is significant, we used the performance detection method suggested by Plötz in [97] to investigate the significance of the achieved enhancement. Since the enhanced method (in Noun-Adjective case) achieved a WER of (9.82%) which is out of the confidence interval [11.53,12.89] (see chapter 4, the baseline system), it is concluded that the achieved enhancement is statistically significant. The other cases are same, i.e. (Preposition-word, and Hybrid cases achieved significant improvement).

Table 6-11 Accuracy achieved

#	Experiment	Accuracy (%)
	baseline System	87.79
1	Noun-Adjective	90.18
2	Preposition-Word	90.04
3	Hybrid (1 & 2)	90.07

Table 6-11 shows that the highest accuracy achieved is in noun-adjective case. The reduction in accuracy in the hybrid case is due to the confusion introduced in the language model. For more clarification, our method depends on adding new sentences to the corpus transcription that is used to build the language model. Therefore, adding too many sentences will finally cause the language model to be biased for some n-grams (1-grams, 2-grams, and 3-grams) on the account of others.

The common way to evaluate the N-gram language model is using perplexity. The perplexity for the baseline is 34.08. For the proposed cases, the language models'

perplexities are displayed in Table 6-12. The measurements were taken based on the testing set, which contains 9288 words. The enhanced cases are clearly better as their perplexity is lower. The reason for the low perplexities is the specific domains that we used in our corpus, viz. economics and sports.

Table 6-12 Perplexities and OOV in different experiments made

#	Experiment	Perplexity	OOV (%)
	baseline System	34.08	328/9288 = 3.53%
1	Noun-Adjective	3.00	287/9288 = 3.09%
2	Preposition	3.22	299/9288 = 3.21%
3	Hybrid (1 & 2)	2.92	316/9288 = 3.40%

The OOV was also measured for the performed experiments. Our ASR system is based on a closed vocabulary, so we assume that there are no unknown words. The OOV was calculated as the percentage of recognized words that do not belong to the testing set, but to the training set. Hence,

$$\text{OOV (baseline system)} = \frac{\text{none testing set words}}{\text{total words in the testing set}} * 100$$

which is equal to  $328/9288 * 100 = 3.53\%$ . For the enhanced cases, Table 6-12 shows the resulting OOVs. Clearly, the lower the OOV the better the performance is, which was achieved in all three cases.

Table 6-13 shows some statistical information collected during experiments. The “compound words collected” is the total number of noun-adjectives found in the corpus transcription. The “unique compound words” indicates the total number of noun-

adjectives after removing duplicates. The last column, “compound words replaced” is the total number of compound words that were replaced back to their original two disjoint words after the decoding process and prior to the testing stage.

Table 6-13 Statistical information for compound words

#	Experiment	compound words collected	unique compound words	compound words replaced
1	Noun-Adjective	3328	2672	377
2	Preposition	3883	2297	409
3	Hybrid (1 & 2)	7211	4969	477

Despite the claim that the Stanford Arabic tagger accuracy is more than 96%, a comprehensive manual reviewing was performed on the tagger output in order to accurate our method based on high accurate data. It was reasonable to review the collected compound words as our transcription corpus is small (39217 words). For large corpuses, the accuracy of the tagger is crucial for the results. For example, Table 6-14 shows an error that occurred in the tagger output. The word “الأوّل” should be DTJJ instead of DTNN.

Table 6-14 An error in the tagger

An input sentence for the tagger	فِي النِّصْفِ الأوَّلِ مِنَ العَامِ الجَارِي
Tagger output (read from left to right)	فِي/IN النصف/NOUN_QUANT الأوّل/DTNN من/IN العام/DTNN الجاري/DTJJ

Figure 6-22 shows an illustrative example of the enhancement that was achieved in the enhanced system. It shows that the baseline system missed one word (من) while it appears in the enhanced system. Introducing a compound word in this sentence avoided the misrecognition that occurred in the baseline system.

A waveform of a speech sentence with its text form	فِي الْمَرْحَلَةِ السَّابِعَةِ وَالثَّلَاثِينَ مِنَ الدَّورِيِّ الإسْبَانِيِّ لِكُرَةِ الْقَدَمِ
As recognized by the baseline system	فِي الْمَرْحَلَةِ السَّابِعَةِ وَالثَّلَاثِينَ الدَّورِيِّ الإسْبَانِيِّ لِكُرَةِ الْقَدَمِ
As recognized by the enhanced system	فِي الْمَرْحَلَةِ السَّابِعَةِ وَالثَّلَاثِينَ مِنَ الدَّورِيِّ الإسْبَانِيِّ لِكُرَةِ الْقَدَمِ
Final output after decomposing the merging	فِي الْمَرْحَلَةِ السَّابِعَةِ وَالثَّلَاثِينَ مِنَ الدَّورِيِّ الإسْبَانِيِّ لِكُرَةِ الْقَدَمِ

Figure 6-22 An example of enhancement in the enhanced system

According to the algorithm, each sentence in the enhanced transcription corpus can have a maximum of one compound word, since sentences are added to the enhanced corpus once a compound word is formed.

After the decoding process, the results are scanned in order to decompose the compound words back to their original form (two separate words). This process is performed using a lookup table such as:

الكُوَيْتِ الدُّوَلِيِّ → الكُوَيْتِ الدُّوَلِيِّ  
فِي مَطَارٍ → فِي مَطَارٍ

### 6.4.3 Execution time

The recognition time was compared with the baseline. The comparison includes the testing set which includes 1144 speech files. The specification of the machine where

we conduct the experiment is as follows: a desktop computer which contains a single processing chip of 3.2GHz and 2.0 GB of RAM.

We found that the recognition time for the enhanced method is less than the recognition time of the baseline system as shown in table 6-15. This means that the proposed method is better than baseline system in term of time complexity.

Table 6-15 Execution time comparison of the enhanced and the baseline systems

Execution time (minutes)	
The baseline system	The enhanced system
34.14	33.05

## 6.5 Cross-word modeling using small words merging

Unlike isolated speech, continuous speech is known to be a source of augmenting words. This augmentation depends on many factors such as the phonology of the language and the lengths of the words. In this section, our work is focused on adjacent small words being a source of this merging of words. During our previous research work in Arabic speech recognition, it became evident that adjacent small words contribute negatively to achieving high performance. Figure 6-23 presents an example of the small-word problem.

A speech sentence to be tested	وَمُمَثِّلِينَ عَنِ الدَّوَلِ الأوروپِيَّةِ عَدَدٍ مِنْ
Recognized as (baseline):	وَمُمَثِّلِينَ عَنِ الدَّوَلِ الأوروپِيَّةِ إِنَّ

Figure 6-23 A small-word problem explanation

Figure 6-23 shows that small words were negatively affected by the concatenations. The decoder mistakenly recognized two separated small words as one word, although it recognized longer words correctly. Therefore, we expect that if we compound the small words as one word, a better performance will be achieved.

### 6.5.1 Proposed method

Modeling the small-word problem is a data-driven approach in which a compound word is distilled from the corpus transcription. The compound word length is the total length of the two adjacent small words that form the corresponding compound word. The small word's length could be 2, 3, 4 letters, or more. During training, several experiments were made to choose the best small word's length. As an illustrative example, suppose as shown in Figure 6-24 that the sentence has many words, and that w2 and w3 are small words. According to our method, w2 and w3 will be merged to generate a compound word. It is worth mentioning that no phonological rules or any kind of knowledge-based approaches are involved in this merging. Figure 6-24 also shows that the boundary appearing between word 2 and word 3 disappears after merging.

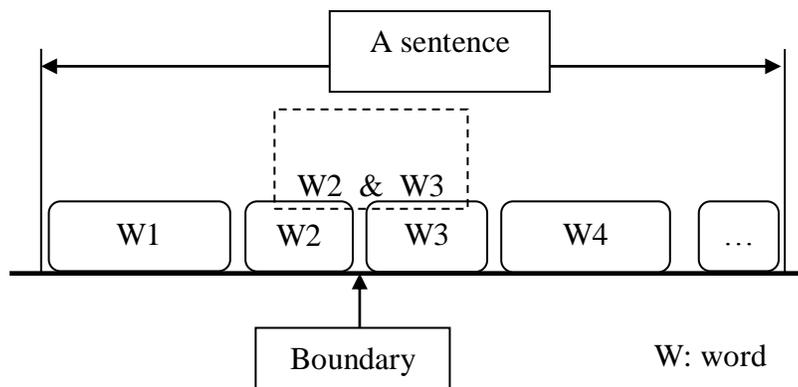


Figure 6-24 The concept of modeling small-word

The generated compound words are then filtered to remove all duplicates. Finally, the unique compound words are added to the dictionary and to the language model. The process can be explained in the following example:

أَظْهَرَ الْإِكْتِتَابُ **فِي بِنِكَ** الرَّيَّانَ الْقَطْرِيَّ

أَظْهَرَ الْإِكْتِتَابُ **فِي بِنِكَ** الرَّيَّانَ الْقَطْرِيَّ

The first sentence is from the baseline corpus transcription sentences, where the text in bold represent two words, one 2-letter word followed by one three-letter word. The second one shows that the two small words found in the first sentence were merged to generate the new compound word. In this example, the total length of the small words is 9, as the diacritics are included in computing the length. Both sentences will be appended during corpus transcription to generate the enhanced pronunciation dictionary and the enhanced language model. The expansion of the pronunciation dictionary and the language model depends on the length of small words chosen for merging. As it gets larger, the dictionary and the language model expand more. The proposed method can be described in the algorithm provided in figure 6-25.

```

=====Offline Stage=====
For all sentences in the transcription file
  For each two adjacent words of each sentence
    If the adjacent words less than a certain threshold
      Generate the compound word
      Represent the compound word in the transcription
    End if
  End for
End for
Based on the new transcription, build the enhanced dictionary
Based on the new transcription, build the enhanced language model
=====Online Stage=====
Switching the variants back to its original separated words

```

Figure 6-25 Cross-model pronunciation variation algorithm using small words

### 6.5.2 Testing and evaluation

In order to test our proposed method, we used the baseline proposed in chapter 4. In order to analyze the effect of the length of the small words on the system performance, we compare the results of our approach when applied on compound words of lengths 5,6,7,8,9,10,11,12 and 13. Table 6-16 summarizes the results of executing the 9 experiments. We use the following shorthand for the keys in Table 6-16:

TL: Total Length of the two adjacent small words.

TC: Total Compound words found in the corpus transcription.

TU: Total Unique compound words without duplicates.

TR: Total Replaced words after recognition process.

AC: Accuracy achieved.

EN: enhancement achieved. It is also the reduction in WER.

Table 6-16 Results for different small word lengths

TL	TC	TU	TR	AC (%)	EN (%)
5	8	6	25	87.80	0.01
6	103	48	41	88.23	0.44
7	235	153	51	88.53	0.74
8	794	447	132	89.42	1.63
9	1618	985	216	89.74	1.95
10	3660	2153	374	89.95	2.16
11	5805	3687	462	89.69	1.90
12	8518	5776	499	89.68	1.89
13	11785	8301	510	88.92	1.13

Table 6-16 shows that the best reduction of 2.16% in WER is achieved when the length of the compound word is 10. It also shows that performance noticeably decreases when the number of characters in the compound words exceeds 10. Figure 6-26 shows the accuracy of the system with respect to the words length.

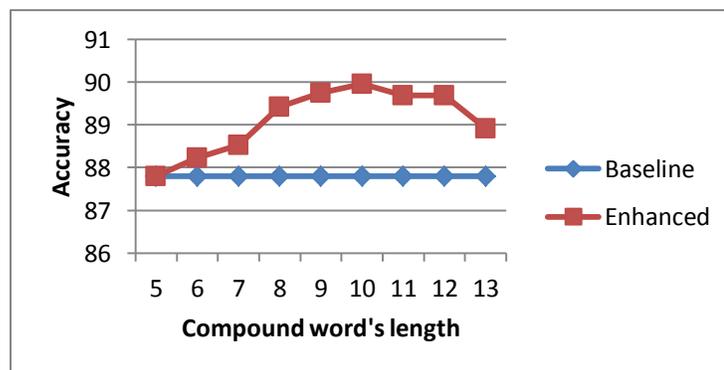


Figure 6-26 A comparison of accuracy for different compound words lengths

With 87.79% accuracy of the baseline system, Figure 6-25 shows that the accuracy of the enhanced system starts increasing until a specific compound word's length (10), and then starts decreasing. The reason of this reduction in accuracy is the confusion introduced in the language model. Figure 6-27 shows that using a high number of compound words does not unconditionally increase the performance. There is a maximum limit to utilize these compound words, after this limit the performance start decreasing due to the ambiguity occurred in the language model. Figure 6-27 shows that 510 compound words used (see Table 6-16, TL=13) do not help to maintain the performance.

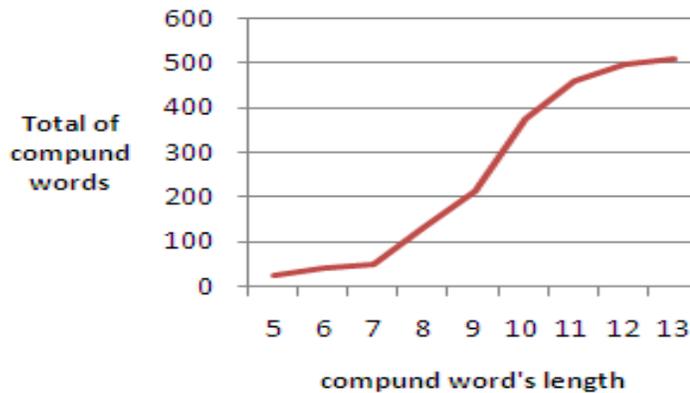


Figure 6-27 Compound words usage

The standard measure for language model quality is perplexity. The perplexity for the baseline language model is 32.88, which is based on 9288 words (testing set words) words. For the enhanced system, the perplexity is 7.14 computed based on the same testing set words (9288 words). This means that the performance of the enhanced system is better than the baseline system since it has a lower perplexity value.

To check whether the achieved enhancement is significant, We used the performance detection method suggested by Plötz in [97] to investigate the significance of the achieved enhancement. Since the enhanced method ( at TL=10, see Table 6-16) achieved a WER of (10.05%) which is out of the confidence interval [11.53,12.89] ( see chapter 4, the baseline system), it is concluded that the achieved enhancement is statistically significant.

### 6.5.3 Execution time

The recognition time is compared with the baseline. The comparison includes the testing set which include 1144 speech files. The specification of the machine where we conduct the experiment is as follows: a desktop computer which contains a single processing chip of 3.2GHz and 2.0 GB of RAM.

We found that the recognition time for the enhanced method is almost the same as the recognition time of the baseline system as shown in Table 6-17. This means that the proposed method is almost equal to the baseline system in term of time complexity.

Table 6-17 Execution time comparison of the enhanced and the baseline systems

Execution time (minutes)	
The baseline system	The enhanced system
34.14	34.31 (for the experiment with highest recognition accuracy, experiment 10)

## 6.6 A comparison between cross-word modeling approaches

Table 6-18 shows a results comparison of the suggested methods for cross-word modeling. It shows that part of speech tagging approach outperform the other methods ( i.e. the phonological rules and small word merging). However, more research should be conducted for more confidence. This conclusion, however, is subject to change as more cases need to be investigated for both techniques. Cross-word modeling used two rules of the Arabic phonological rules, while only two compounding schemes were applied in part of speech tagging approach.

Table 6-18 A comparison between combined proposed techniques

#	System	Accuracy (%)	Execution Time (minutes)
	baseline	87.79	34.14
1	Phonological rules	90.09	33.49
2	PoS tagging	90.18	33.05
3	Small word merging	89.95	34.31
4	Combined system (1,2,and3)	88.48	30.31

## 6.7 Combining of within-word and cross-word methods

Table 6-19 shows the accuracy and the execution time of a combined system. The PoS tagging compounding method was selected (as it has the highest accuracy among cross-word modeling techniques) to be combined with the within-word technique explored in chapter 5. The results show no enhancement. this means that the PoS tagging method achieved the highest accuracy among within-word and cross-word pronunciation variations. two reason to justify that no noticeable enhancement: the increase in the total number of words, and the also increase the n-grams in the langue model.

Table 6-19 A comparison between compound words techniques

Combined method	Accuracy (%)	Execution Time (minutes)
Within-word and merging based on PoS tagging	90.15	32.17

# CHAPTER 7

## RESCORING N-BEST HYPOTHESES

### 7.1 Introduction

Improving speech recognition accuracy through linguistic knowledge is a major research area in automatic speech recognition systems. In this chapter, we present a syntax-mining approach to rescore N-best hypotheses for Arabic speech recognition systems. The proposed method depends on a machine learning tool (weka-3-6-5) to extract the N-best syntactic rules of the baseline tagged transcription corpus, which was tagged using Stanford Arabic tagger. The chapter presents the modeling technique of syntactically incorrect structure of the baseline output. The syntactically incorrect output structure problem appears in the form of different orders of words, out of the Arabic correct syntactic structure.

Figure 7-1 demonstrates an example of one baseline output sentence with its corresponding hypotheses. In this figure, the output sentence (to be released to the user) is the first hypothesis, while the correct sentence is the second one, the highlighted sentence. The sentences in Figure 7-1 are called N-best hypotheses (also called N-best list), where N is chosen to be 6.

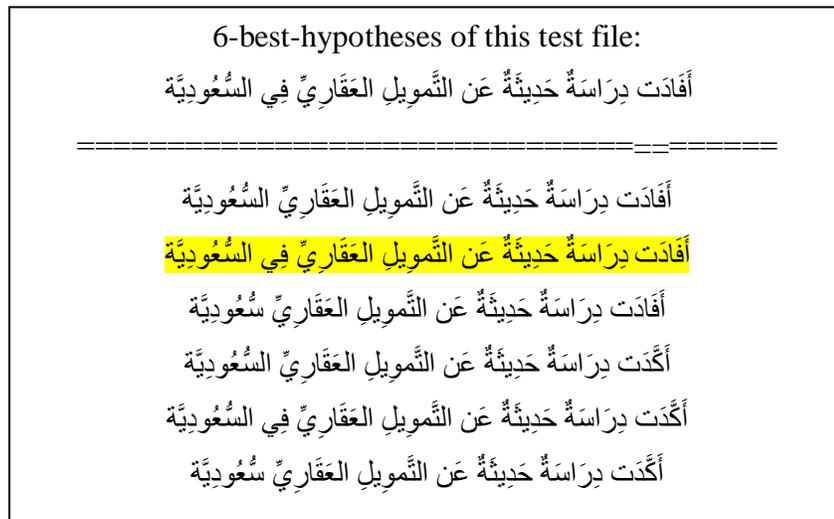


Figure 7-1 An example of 6-best hypotheses of a sentence

To model this problem (i.e. out of language syntactic structure results), the tags of the words were used as a criterion for rescoring and sorting the N-best list. The tags use the word’s properties instead of the word itself. We used “language syntax rules” to indicate for the most frequently tags relationships appearing in the Arabic language. The rescored hypotheses are then sorted to pick the top score hypothesis. Figure 7-2 shows the idea behind the proposed rescoring model.

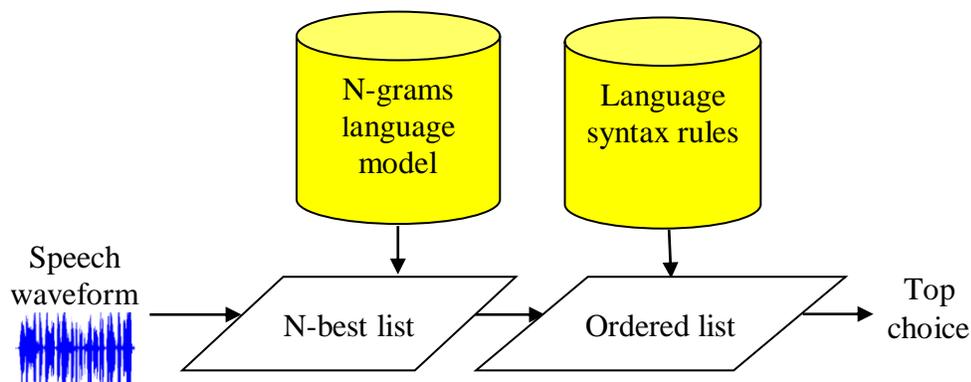


Figure 7-2 Illustration of rescoring N-best list

## 7.2 Related work

Using linguistic knowledge to improve speech recognition systems was used by many researchers. Salgado-Garza et al. in [39] demonstrated the usefulness of syntactic trigrams in improving the performance of a speech recognizer for Spanish. They achieved a significant enhancement. Wang et al. in [123] compared the efficacy of a variety of language models for rescored word graphs and N-best lists generated by a large vocabulary continuous speech recognizer. These language models differ based on the level of knowledge used (word, lexical features, syntax) and the type of integration of that knowledge. Xiang et al. in [124] presented advanced techniques that improved the performance of IBM Malay-English speech translation system significantly. They generated linguistics-driven hierarchical rules to enhance the formal syntax-based translation model. In [133], Jeon et al. integrated prosodic information for ASR using an n-best rescored scheme. Their rescored method achieved a WER reduction of 3.64% and 2.07% using two different ASR systems. Ganapathiraju et al. in [134] addressed the use of a support vector machine as a classifier in a continuous speech recognition system. A hybrid SVM/HMM system has been developed that uses SVMs to rescore an N-best list hypotheses generated by a conventional HMM system. Birkenes et al. in [135] used logistic regression to rescore N-best list for continuous speech recognition systems. Jang [136] proposed an unsupervised learning algorithm that learns hierarchical patterns of word sequences in spoken language utterances. It extracts cluster rules from training data based on high n-gram probabilities to cluster words or segment a sentence. The learned cluster rules were used to improve the n-best utterance hypothesis list.

As Arabic Part of speech (PoS) tagging is an essential component in our method, we performed the following literature review. The stochastic method dominates PoS tagging models. Diab et al. in [125] presented an SVM based approach to automatically tag Arabic text. Al-Shamsi and Guessoum in [126] presented a PoS Tagger for Arabic using a HMM approach. El-Hadj et al. in [127] presented an Arabic PoS tagger that uses an HMM model to represent the internal linguistic structure of the Arabic sentence. A corpus composed of old texts extracted from books written in the ninth century AD was created. They presented the characteristics of the Arabic language and the set of tags used. Albared et al. in [128] presented an HMM approach to tackle the PoS tagging problem in Arabic. Finally, the Stanford Natural Language Processing Group developed an Arabic tagger [129] with an accuracy range between 80% and 96%.

According to the literature review, and to the best of our knowledge, we have not found any research work that employs a machine learning algorithm to distill N-best syntactic rules to be used for rescoring N-best hypotheses for large vocabulary continuous speech recognition systems.

### **7.3 Data-Mining Approach (WEKA tool)**

Weka is a collection of machine learning algorithms for data mining tasks which represents a process developed to examine large amounts of data routinely collected. Extracting N-best syntactic rules using weka tool was described by Tobias Scheffer in [130]. He presented a fast algorithm that finds the  $n$  best rules which maximize the resulting criterion. The strength of this tool is the ability to find the relationships between tags with no consecutive constraint. For example, if we have a tagged sentence, then it is

possible to describe the relations between its tags as follows: if the first word's tag is noun and the sixth word's tag is adjective, then the ninth word's tag is adverb with certain accuracy. This also could be used for words, i.e. an extracted rule could have  $n$  words with its relationships and accuracy. Data mining is used in most areas where data are collected such as health, marketing, communications, etc. it is worth noting that data mining algorithms require high performance computing machines. For more information about weka tool, Please refer to Machine Learning Group at University of Waikato in [131].

#### 7.4 The Proposed Method

Rescoring N-best hypotheses is the basis of our method. The rescoring process was performed for each hypothesis to find the new score. A hypothesis new score is the total number of the hypothesis' rules that are already found in the language syntax rules (extracted from the tagged transcription corpus). The hypothesis with the maximum matched rules is considered as the best one. Our method can be described using Figure 7-3.

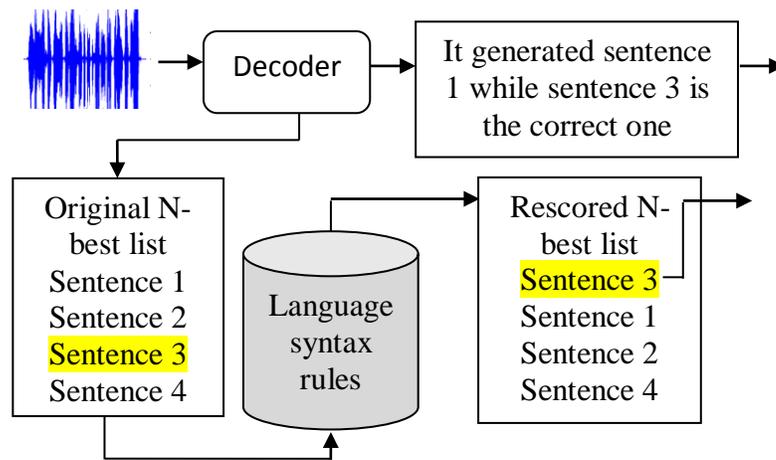


Figure 7-3 Generation of rescored N-best list

In Figure 7-3, suppose that the third sentence is the correct sentence that should be returned by the decoder. If the N-best hypotheses list was rescored using language syntax rules, we expect, hopefully, to get a better result since the final output will be syntactically evaluated. In this case, the hypothesis with maximum number of rules is chosen since the other hypotheses are less likely to be the best one. Hence instead of returning the previously top choice (sentence 1) of N-best list, it will return the top choice of Rescored N-best list (sentence 3) as shown in Figure 7-3. For more clarification, suppose that the two hypotheses of a tested file are as follows:

(1) VBD NN NNP DTNNP NN NNP NNP DTJJ DTNN

(2) VBD NN NNS DTNNP JJ NNP NN DTJJ DTNNS

Each hypothesis is evaluated by finding the total number of the hypothesis' rules already found in the language syntax rules. Suppose that hypothesis number (2) has 4 matching rules while hypothesis number (1) has only 3. In this case, hypothesis number (2) will be chosen as output since it has the maximum matching rules. Since the N-best hypotheses are sorted according to the acoustic score, if two hypotheses have the same matching rules, the first one will be chosen as it has the highest acoustic score. Therefore, two factors are contributed to decide which among hypothesis in N-best list would be the best one: *acoustic score and the total number of language syntax rules belong the hypothesis.*

Before using weka tool, the transcription corpus was tagged using Stanford Arabic tagger which contains 29 tags as shown in Appendix 6.

Finding language syntax rules was performed using a machine learning tool (weka-3-6-5). This tool was called to find N-best syntactic rules. In our method, we choose to find the best 3000 syntactic rules. For more elaboration, Table 7-1 shows the first best five rules.

Table 7-1 First 5-Best syntactic rules of 3000 extracted rules

Rule	Syntactic relations
1	TAG4=CD TAG6=DTNN ==> TAG5=IN acc: (0.95635)
2	TAG1=VBD TAG3=DTJJ TAG7=DTNN ==> TAG2=DTNN acc: (0.95635)
3	TAG7=CD TAG8=IN ==> TAG9=DTNN acc: (0.95222)
4	TAG7=CD TAG9=DTNN ==> TAG8=IN acc: (0.95222)
5	TAG2=DTNN TAG3=IN TAG5=DTNN ==> TAG4=NN acc: (0.94985)

Our transcription corpus contains sentences that include up to 30 words. Therefore, our rules have the relationships between tags in the range from 1 to 30. The first rule in Table 7-1 shows that if the fourth word's tag is a number and the sixth word's tag is a noun, then the fifth word's tag will be preposition with rule accuracy of 95.635%. Rule 2 in Table 7-1 shows the relationships between not neighboring tags (tag1, tag3, tag7, tag2). That is, Weka tool can be used to find the relationships between long-distance tags. As example, the following rule provides the relationships between 6 not-consecutive tags.

TAG1=VBD TAG3=DTNN TAG4=DTJJ TAG5=NN TAG12=NN ==> TAG2=NN  
acc: (0.92298)

As we mentioned in section 7-3 that extracting association rules in a large data require a high performance computing (HPC) environment. In our experiments, we found that a desktop computer which contains a single processing chip of 3.2GHz and 2.0 GB of RAM could obtain no more than 530 rules. Therefore, extracting high number of rules in a large corpus requires HPC. Extracting 3000 rules using HPC took around 4 hours while it had taken around 24 hours in the desktop.

HPC is the application of "supercomputers" to computational problems that are either too large for standard computers or would take too long. HPC environment consists of a network of nodes, each of which contains one or more processing chips, as well as its own memory. In our method, we choose to extract 3000 rules, so we used the HPC at KUPM which has the following hardware characteristics, [120]:

- 128 compute-node e1350 IBM eServer cluster.
- The cluster has 128 compute nodes. Each compute node of the cluster is dual-processor having two 2.0 GHz x3550 Xeon Quad-core E5405 – processors.
- The total number of cores in the cluster is 1024.
- Each master node has 45 GB of RAM.
- Each compute node has 4 GB of RAM.

Our method can be described in the following algorithm:

---

---

## N-best Hypothesis Rescoring Algorithm

---

*Have the transcription corpus tagged*

*Using the tagged corpus, extract N-best rules*

*Generate the N-best hypotheses for each tested file*

*Have the N-best hypotheses tagged for tested files*

*For each tested file*

*For each hypothesis in the tested files*

*Count the total number of matched rules\**

*Return the hypothesis of the maximum matched rules*

*End for*

*End for*

*\* Matched rules: Hypothesis rules that are also found in the language syntax rules*

---

We used the CMU PocketSphinx to generate the 50-Best hypotheses for each utterance in the test set. After intensive investigation of our method, we did not find significant enhancement. However, we found enhancements in some tested files as well as new errors introduced in others. Figure 7-4 and Figure 7-5 show enhancement in some tested files.

A waveform of a speech sentence with its text form	هَذَا وَقَدْ بَلَغَتْ مَبِيعَاتُ شَرِكَةِ فُورْدِ مُوتُورزِ فِي الصِّينِ جِلالَ عَامِ أَلْفَيْنِ وَخَمْسَةَ
As recognized by the baseline system	هَذَا وَقَدْ بَلَغَتْ مَبِيعَاتُ شَرِكَةِ فُورْدِ مُوتُورزِ النَّسَعِينَ جِلالَ عَامِ أَلْفَيْنِ وَخَمْسَةَ
Found at →	Hypothesis # 36
As recognized by the enhanced system	هَذَا وَقَدْ بَلَغَتْ مَبِيعَاتُ شَرِكَةِ فُورْدِ مُوتُورزِ فِي الصِّينِ جِلالَ عَامِ أَلْفَيْنِ وَخَمْسَةَ

Figure 7-4 A perfect enhancement in a tested file

A waveform of a speech sentence with its text form	حَدَرَ البَنْكُ الدَّوْلِي دُولَ الخَلِيجِ العَرَبِيَّةِ مِنْ صَنَحِ المَزِيدِ مِنْ عَائِدَاتِهَا النَّفْطِيَّةِ فِي مَشْرُوعَاتِ
As recognized by the baseline system	حَدَرَ البَنْكُ الدَّوْلِي دُولَ الخَلِيجِ العَرَبِيَّةِ مِنْ صَنَحِ المَزِيدِ مِنْ عَائِدَاتِهَا النَّفْطِيَّةِ فِي مَشْرُوعَاتِ
Found at →	Hypothesis # 50
As recognized by the enhanced system	حَدَرَ البَنْكُ الدَّوْلِي دُولَ الخَلِيجِ العَرَبِيَّةِ مِنْ صَنَحِ المَزِيدِ مِنْ عَائِدَاتِهَا النَّفْطِيَّةِ فِي مَشْرُوعَاتِ

Figure 7-5 A perfect enhancement in a tested file

For the tested file in Figure 7-4, the best hypothesis was found at position #36, while the hypothesis #50 was found to be best one in Figure 7-5. The previous two examples show a perfect enhancement where a wrong word was switched to a correct one. The following are two other examples to show partial enhancements in the tested files. Figure 7-6 found the best choice to be the hypothesis #8, while the hypothesis #4 was found to the best one in Figure 7-7.

A waveform of a speech sentence with its text form	وَأَكَّدَ التَّقْرِيرَ أَنَّ مُتَوَسِّطَ سِعْرِ السَّلَّةِ فِي شَهْرِ دَيْسَمْبَرِ بَلَغَ ثَمَانِيَةَ وَخَمْسِينَ دُولَارًا وَعَشْرَةَ سِنَنَاتٍ
As recognized by the baseline system	وَأَكَّدَ التَّقْرِيرَ أَنَّ مُتَوَسِّطَ سِعْرِ السَّلَّةِ فِي شَهْرِ السَّنْبُورَةِ بَلَغَ ثَمَانِيَةَ وَخَمْسِينَ دُولَارًا وَعَشْرَةَ سِنَنَاتٍ
Found at →	Hypothesis # 8
As recognized by the enhanced system	وَأَكَّدَ التَّقْرِيرَ أَنَّ مُتَوَسِّطَ سِعْرِ السَّلَّةِ فِي شَهْرِ دَيْسَمْبَرِ اللهُ بَلَغَ ثَمَانِيَةَ وَخَمْسِينَ دُولَارًا وَعَشْرَةَ سِنَنَاتٍ

Figure 7-6 A partial enhancement in a tested file

A waveform of a speech sentence with its text form	إِنَّ فَرْقَ الْإِنْفَادِ
As recognized by the baseline system	إِنَّ فَرْقَ الْإِنْتَرِنِتِ
Found at →	Hypothesis # 4
As recognized by the enhanced system	إِنَّ فَرْقَ الْإِنْفَادِ اللهُ

Figure 7-7 A partial enhancement in a tested file

The previous examples show that our method is a promising method to enhance speech recognition accuracy. However, with enhancements in some tested files, we found new errors (i.e. previously correct recognized words) introduced in some tested files as shown in Figure 7-8.

A waveform of a speech sentence with its text form	وَذَلِكَ بِمُشَارَكَةِ عَدَدٍ مِنْ رِجَالِ أَعْمَالٍ وَمُسْتَثْمِرِينَ سُعُودِيِّينَ
As recognized by the baseline system	وَذَلِكَ بِمُشَارَكَةِ عَدَدٍ مِنْ رِجَالِ أَعْمَالٍ وَمُسْتَثْمِرِينَ سُعُودِيِّينَ
Found at →	Hypothesis # 9
As recognized by the enhanced system	وَذَلِكَ بِمُشَارَكَةِ عَدَدٍ لِرِجَالِ أَعْمَالٍ وَمُسْتَثْمِرِينَ سُعُودِيِّينَ

Figure 7-8 A wrong hypothesis selection example

We also would like to present a case where the N-best hypotheses already has the correct choice but was not selected after the rescoring process. Figure 7-9 shows as example.

A waveform of a speech sentence with its text form	أَفَادَتِ دِرَاسَةٌ حَدِيثَةٌ عَنِ التَّمْوِيلِ الْعَقَارِيِّ فِي السُّعُودِيَّةِ
As recognized by the baseline system	أَفَادَتِ دِرَاسَةٌ حَدِيثَةٌ عَنِ التَّمْوِيلِ الْعَقَارِيِّ السُّعُودِيَّةِ
The chosen →	Hypothesis # 4
As recognized by the enhanced system	أَفَادَتِ دِرَاسَةٌ حَدِيثَةٌ عَنِ التَّمْوِيلِ الْعَقَارِيِّ سُعُودِيَّةِ
The correct →	Hypothesis # 3
Neither baseline nor enhanced	أَفَادَتِ دِرَاسَةٌ حَدِيثَةٌ عَنِ التَّمْوِيلِ الْعَقَارِيِّ فِي السُّعُودِيَّةِ

Figure 7-9 Not-selected correct hypothesis example

In our method, part of speech tagging was crucial to support the correctness of the method used. Even though the Stanford tagger which was used in our method has many

correct tagged sentences, however, there are many mistakenly tagged sentences. We provide two examples of a correct tagged sentence and a wrong tagged one as shown in Table 7-2.

Table 7-2 Two examples of tagged sentences

A correct tagged sentence
دال/NNP وشركة/NN السعودية/DTNNP أرامكو/NNP شركة/NN قالت/VBD اليوم/DTNN الأمريكية/DTJJ كيميكلز/NNP
A wrong tagged sentence
مصممة/VN الإسلامية/DTJJ الجمهورية/DTNN إن/NN متقي/JJ وقال/NN بالثقة/JJ وجديرا/NN فعلا/NN للنفط/VN مزودا/VB تكون/VBP أن/NN على/IN

In Table 7-2, the highlighted texts were wrongly tagged. Therefore, extracting the language syntax rules using many errors will not be strong enough for rescoring the N-best hypotheses. This is our justification of our result, enhancement in some tested files and new errors in others.

In addition to the tagger problem, we finalize this section by explaining the effect of diacritics in this research work. Not like English, Arabic sentences are diacritized. Accordingly, the N-best hypotheses will also be diacritized.

9106-	السُّعُودِيَّة	الغَاز	فِي	عَلَى	تَعْتَمِدُ	الَّتِي
9179-	السُّعُودِيَّة	الغَاز	فِي	عَلَى	تَعْتَمِدُ	الَّتِي
9320-	السُّعُودِيَّة	الغَاز	فِي	عَلَى	تَعْتَمِدُ	الَّتِي
9130-	السُّعُودِيَّة	الغَاز	فِي	عَلَى	تَعْتَمِدُ	الَّتِي
9203-	السُّعُودِيَّة	الغَاز	فِي	عَلَى	تَعْتَمِدُ	الَّتِي
9344-	السُّعُودِيَّة	الغَاز	فِي	عَلَى	تَعْتَمِدُ	الَّتِي
9564-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9588-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9609-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9633-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9655-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9679-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9756-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9780-	السُّعُودِيَّة	الغَاز	عَلَى	تَعْتَمِدُ	الَّتِي	تَعْتَمِدُ
9909-	السُّعُودِيَّة	الغَاز	لِصَفَى	عَلَى	تَعْتَمِدُ	الَّتِي

Figure 7-10 10-Best list of a tested file.

The problem is the gap between diacritized hypothesis and non-diacritized tagger used. Therefore, the highlighted hypothesis in Figure 7-10 are considered the same from tagger point of view. This same-tags case prevents the diversity that should be presented in the N-best hypotheses. One case, among 300-best hypotheses, we found 16 distinct hypotheses, (i.e. at words level). As the acoustic scores are sorted in decreasing order, the problem showed up when, as example, finding the first 50 hypotheses with same words and different diacritics. So, instead of searching among first different hypotheses like English, the search will be deep (in diacritized Arabic) which in the same time moving away from the best hypotheses group, i.e. the beginning of hypotheses which have high acoustic scores.

# CLOSING REMARKS

**Within-word:** Extracting pronunciation variants directly from training pronunciation corpus and have it represented in the dictionary and the language model shows significant enhancement for MSA ASRs. The sequence alignment method was used to extract a number of variants to model them in the dictionary and the language model. The experiments show that as we move away from the small words, the system gives better performance. The enhancement we achieved has not only come from the pronunciation variation modeling in the dictionary, but was an indirect result of the recalculated bigrams and trigrams probabilities in the language model.

As future work, we propose to try the indirect data-driven approach to mine the transformation rules that can be used to generate the variants. Then a comparison could be made between both approaches. Other sequence alignment scores and LD measures can also be investigated.

**Cross-word:** The proposed knowledge-based approach achieved feasible improvement for cross-word variation modeling. Mainly, two MSA phonological rules were applied, the Idgham and Iqlaab. The experiment results clearly showed that the Idgham occurred more than Iqlaab. The Idgham rules dominate the generation of the cross-word variants. The significant enhancement we achieved has not only come from the cross-word pronunciation modeling in the dictionary, but also indirectly from the recalculated n-grams probabilities in the language model.

We conclude that Viterbi algorithm works better with long words. Speech recognition research should consider this fact when designing dictionaries. We found that merging words based on their types, viz. the tag, leads to significant improvement in Arabic ASRs. The third approach we implemented in merging words was small words merging which also gives a significant enhancement. We also found that adding compound words to the dictionary as well as to the language model reduces the perplexity and enhances the performance as compared to the baseline system.

As future work, we propose to check more phonological rules more than just two cases as we did, Arabic has more rules to be investigated. We also propose investigating more word-combination cases for merging using PoS tagging. In particular, we expect that the construct phrases (الإضافة) make a good candidate. Examples include : (مدينة القدس، ( مطار بيروت، سلسلة جبال ( واو )). Another suggested candidate is the Arabic "and" connective (يتعلق بقضايا العراق والسودان، مواد أدبية ولغوية) (العطف), such as:

**N-best rescoring**: We conclude that N-best rescoring for Arabic speech recognition (using Arabic data-driven syntax) does not provide significant enhancement. However, more investigation can be performed with a high accurate part of speech tagging model.

As future work, we recommend to utilize linguistic knowledge at the decoder level, i.e. before releasing the decoder output. We also recommend to do further research on Arabic part of speech tagging, especially for diacritized text. we also propose to review Arabic phoneme set to be extracted using data-driven technique as an alternative method of the currently used linguistic method. additionally, the high frequently syntactic rules appearing in the language could be used in the modeling, instead of using all rules.

# APPENDICES

## Appendix 1 : Arabic Terminologies

**Al-Alta'rif** The determiner ( الـ ).

**Damma** An Arabic short vowel ( ُ ), pronounced like (u).

**Dammatan** Two Damma (or doubling of Damma), pronounced like (n). Also called Tanween of Damma.

**Fatha** An Arabic short vowel ( ا ), pronounced like (a).

**Fathatan** Two Fatha (doubling of Fatha), pronounced like (n). Also called Tanween of Fatha.

**Hamzat Al-Wasl** It is an extra Hamza that helps to start pronouncing an unvowelled letter in Arabic continuous speech.

**Idgham** Also called geminating or assimilation, it is a merging of two consecutive letters of the second type letter.

**Idgham almutajanisan** It is a merging between two consecutive different letters that are close in pronunciation. Some of these cases include: taa' / تْ and daal / د , taa' / تْ and Taa' / ط , dhaal / ذْ and Zaa / ظ , qaaf / قْ and kaaf / ك , laam / ل and raa' / ر .

**Idgham almutmathlan** It is a merging between two consecutive identical letters shown in the following list { ب , ت , ث , ج , ح , خ , د , ذ , ر , ز , س , ش , ص , ض , ط , ظ , ع , غ , ف , ق , ك , ل , ن } . The rule means that any unvowelled Arabic letter followed by the same Arabic vowelled letter will be doubled in a single merged word. Note that { ا , و , ي } are not included in the list.

**Iqlaab** it is a replacement of unvowelled nuun (Nuun Saakinah <> نْ) or Tanween ( ُ , ِ , ٍ ) that comes before vowelled baa' ( ب ) by unvowelled miim (Miim Saakinah <> مْ).

**Kasra** An Arabic short vowel ( ا ), pronounced like (i).

**Kasratan** Two Damma (doubling of Kasra), preannounced like (n). Also is called Tanween of Kasra.

**Nuun Saakina** An unvowelled nuun symbolized as ( ُ )

**Shadda** It is a doubling of consonant and symbolized as ( ّ )

**Shamsi group** Arabic letters include (taa', thaa', daal, dhaal, raa', zaay, siin, shiin, Saad, Daad, Taa', Zaa', laam, and nuun).

**Sukun** Absence of vowel, symbolized by ( ْ )

**Ta'al marbouta** It is an Arabic letter symbolized as ( ة ) and shown at the end of the words.

**Tanween** Includes any one of Dammatan, Fathatan, or Kasratan. It is symbolized as ( َ , ِ , ُ ).

**Appendix 2 : Arabic–Roman letters mapping table**

Arabic	Roman	Arabic	Roman	Arabic	Roman	Arabic	Roman
ء (hamza)	’	د (daal)	d	ض (Daad)	D	ك (kaaf)	k
ب (baa’)	b	ذ (dhaal)	dh	ط (Taa’)	T	ل (laam)	l
ت (taa’)	t	ر (raa’)	r	ظ (Zaa’)	Z	م (miim)	m
ث (thaa’)	th	ز (zaay)	z	ع (‘ayn)	‘	ن (nuun)	n
ج (jiim)	j	س (siin)	s	غ (ghayn)	gh	ه (haa’)	h
ح (Haa’)	H	ش (shiin)	sh	ف (faa’)	f	و (waaw)	w or u
خ (khaa’)	kh	ص (Saad)	S	ق (qaaf)	q	ي (yaa’)	y or ii

**Appendix 3 : The phonemes set used in the baseline system (IPA )**

<b>Phoneme And IPA</b>	<b>Letter and Examples</b>	<b>Phoneme And IPA</b>	<b>Letter</b>
/AE/ <u>æ</u>	بَ ◀ - Fatha	/DH/ <u>ð</u>	ذ (Thal)
/AE:/ <u>æː</u>	بَاب ◀	/R/ <u>r</u>	ر (Raa)
/AA/ <u>ɑ</u>	حَ ◀ - Hard Fatah	/Z/ <u>z</u>	ز (Zain)
/AH/ <u>ɑː</u>	حَ ◀ - Soft Fatah	/S/ <u>s</u>	س (Seen)
/UH/ <u>u</u>	بُ ◀ - Damma	/SH/ <u>ʃ</u>	ش (Sheen)
/UW/ <u>uː</u>	و ◀ - Damma	/SS/ <u>sˤ</u>	ص (Sad)
/UX/ <u>o</u>	غ ◀ - Kasra	/DD/ <u>dˤ</u>	ض (Dad)
/IH/ <u>e</u>	بِنْت ◀ - Kasra	/TT/ <u>tˤ</u>	ط (Taa)
/IY/ <u>iː</u>	ي ◀ - Kasra	/DH2/ <u>ðˤ</u>	ظ (Thaa)
/IX/ <u>i</u>	ي ◀ - Kasra	/AI/ <u>ʔ</u>	ع (Ain)
/AW/ <u>u</u>	و ◀ - Damma	/GH/ <u>ɣ</u>	غ (Ghain)
/AY/ <u>eː</u>	ي ◀ - Kasra	/F/ <u>f</u>	ف (Faa)
/E/ <u>ʔ</u>	ء (Hamza)	/Q/ <u>q</u>	ق (Qaf)
/B/ <u>b</u>	ب (Baa)	/K/ <u>k</u>	ك (Kaf)
/T/ <u>t</u>	ت (Taa)	/L/ <u>l</u>	ل (Lam)
/TH/ <u>θ</u>	ث (Thaa)	/M/ <u>m</u>	م (Meem)
/JH/ <u>dʒ</u>	جيم فصحي (Jeem)	/N/ <u>n</u>	ن (Noon)
/HH/ <u>h</u>	ح (Haa)	/H/ <u>h</u>	ه (Haa)
/KH/ <u>χ</u>	خ (Khah)	/W/ <u>w</u>	و (Waw)
/D/ <u>d</u>	د (Dal)	/Y/ <u>j</u>	ي (Yaa)

#### Appendix 4: Phoneme-Character mapping

#	Unique character representation	Phoneme	Arabic representation
1	A	AE	اَ
2	C	AE:	اِ
3	I	AA	اِ
4	J	AH	اَ
5	O	UH	اُ
6	P	UW	وُ
7	U	UX	اُ
8	X	IH	اَ
9	}	IY	يَ
10	{	IX	اَ
11	]	AW	وُ
12	[	AY	يَ
13	.	TH	ثَ
14	,	JH	جَ
15	!	HH	حَ
16	@	KH	خَ
17	#	DH	ذَ
18	\$	SH	شَ
19	%	SS	صَ
20	^	DD	ضَ
21	&	TT	طَ
22	*	DH2	ظَ
23	+	AI	عَ
24	=	GH	غَ
25	E	E	ءَ
26	B	B	بَ
27	T	T	تَ

28	D	D	د
29	R	R	ر
30	Z	Z	ز
31	S	S	س
32	F	F	ف
33	Q	Q	ق
34	K	K	ك
35	L	L	ل
36	M	M	م
37	N	N	ن
38	H	H	ه
39	W	W	و
40	Y	Y	ي

**Appendix 5: Rules usage in the entire transcription corpus**

Rule	Final letter of first word (unvowelled)	Initial letter of second word (vowelled)	Usage times
1	A letter	Identical with the previous letter	
	baa' / ب	baa' / ب	17
	taa' / ت	taa' / ت	38
	thaa' / ث	thaa' / ث	0
	jiim / ج	jiim / ج	0
	Haa' / ح	Haa' / ح	0
	khaa' / خ	khaa' / خ	0
	daal / د	daal / د	2
	dhaal / ذ	dhaal / ذ	0
	raa' / ر	raa' / ر	16
	zaay / ز	zaay / ز	1
	siin / س	siin / س	7
	shiin / ش	shiin / ش	0
	Saad / ص	Saad / ص	0
	Daad / ض	Daad / ض	0
	Taa' / ط	Taa' / ط	0
	Zaa' / ظ	Zaa' / ظ	0
	'ayn / ع	'ayn / ع	18
	ghayn / غ	ghayn / غ	0
	faa' / ف	faa' / ف	12
	qaaf / ق	qaaf / ق	3
	kaaf / ك	kaaf / ك	0
	laam / ل	laam / ل	49
	miim / م	miim / م	42
	nuun / ن	nuun / ن	0
	haa' / ه	haa' / ه	0

			===== 205
2	Nuun Saakinah and Tanween	yaa' / ي raa' / ر miim / م laam / ل waaw / و nuun / ن	1531
3	Nuun Saakinah and Tanween	baa' / ب	200
	A letter	A close in pronunciation letter	
4	taa' / ت	daal / د	25
5	taa' / ت	Taa' / ط	4
6	daal / د	taa' / ت	32
7	baa' / ب	miim / م	14
8	dhaal / ذ	zaay / ظ	0
9	kaaf / ق	kaaf / ك	1
10	laam / ل	raa' / ر	6
Total			2018

### Appendix 6: Stanford tagging set

#	Tag	Meaning with examples
1	ADJ_NUM	Adjective, Numeric السابع،الرابعة
2	DTJJ	DT + Adjective النفطية،الجديد
3	DTJJR	Adjective, comparative الكبرى،العليا
4	DTNN	DT + Noun, singular or mass المنظمة،العاصمة
5	DTNNP	DT + Proper noun, singular العراق،القاهرة
6	DTNNS	DT + Noun, plural السيارات، الولايات
7	IN	Preposition or subordinating conjunction حرف جر مثل : في حرف مصدري مثل : أن
8	JJ	Adjective جديدة،قيادية
9	JJR	Adjective, comparative أدنى،كبرى
10	NN	Noun, singular or mass إنتاج، نجم
11	NNP	Proper noun, singular أوبك،لبنان
12	NNS	Noun, plural توقعات،طلبات
13	NOUN_QUANT	Noun, quantity الربع، ثلثي
14	CC	Coordinating conjunction ثم ، و
15	CD	Cardinal number مئة، ألفين
16	DT	Demonstrative pronouns هذه،ذلك
17	PRP	Personal pronoun هي، هو
18	PRP\$	Possessive pronoun هم
19	RB	Adverb هناك، حيث
20	RP	Particle لم، لا
21	VB	Verb, base form

22	VBD	Verb, past tense أعلن، قالت
23	VBG	Verb, gerund or present participle نية، اعتبار
24	VBN	Verb, past participle يقام، يعد
25	VBP	Verb, non3rd person singular present تتزايد، يعمل
26	VN	Verb, 3rd person singular present مسجلة، مدعومة
27	WP	Whpronoun الذين
28	WRB	Whadverb حيث
29	UNK	Unknown word

# NOMENCLATURE

ASR	Automatic speech recognition
ANN	Artificial neural networks
CD	Untied context-dependent phase
CHMM	Continuous HMM
CI	Context-independent phase
CMU	Carnegie Mellon University
CRF	Conditional random fields
DARPA	Defense Advanced Research Projects Agency
DP	Dynamic programming
DTW	Dynamic time warping
FPGA	Field programmable gate array
GALE	Global autonomous language exploitation
GMM	Gaussian mixture models
HMM	Hidden Markov Model
HPC	High performance computing
HTK	Hidden Markov Model Toolkit
IT	Information technology
LD	Levenshtein Distance
LIN	linear input networks
LM	Language Model
LPCC	Linear Predictive Cepstral Coefficients
LVCSR	Large vocabulary continuous speech recognition
MFCC	Mel Frequency Cepstrum Coefficients
ML	Maximum-likelihood
MMSE	Minimum mean-square-error
MSA	Modern standard Arabic
MLP	multilayer perceptron

NLP	Natural language processing
NNLMs	Neural Network Based Language Modeling
OOV	Out Of Vocabulary
PLP	perceptual linear predictive
PoS	Part of Speech
PP	Perplexity (PP)
SCHMM	Semi-continuous HMM
SGMM	Subspace Gaussian mixture models
SGMM	subspace Gaussian mixture models
SVM	Support vector machine
WER	Word Error Rate
WNN	Wavelet Neural Network

# REFERENCES

1. Farghaly A, Shaalan K (2009) Arabic natural language processing: challenges and solutions. *ACM Trans Asian Lang Inform Process* 8(4):1–22
2. Ryding KC (2005) A reference grammar of modern standard Arabic (reference grammars). Cambridge University Press, Cambridge
3. Lamel L, Messaoudi A et al (2009) Automatic speech-to-text transcription in Arabic. *ACM Trans Asian Lang Inform Process* 8(4):1–1822 2 Arabic Speech Recognition Systems
4. Khasawneh M, Assaleh K et al (2004) The application of polynomial discriminant function classifiers to isolated Arabic speech recognition. In: Proceedings of the IEEE international joint conference on neural networks, 2004
5. Benzeghiba M, De Mori R et al (2007) Automatic speech recognition and speech variability: a review. *Speech Commun* 49(10–11):763–786
6. Rabiner L, Juang B (1993) Fundamentals of speech recognition. Prentice Hall, Upper Saddle River, NJ
7. Jelinek F (1998) Statistical methods for speech recognition. MIT, Cambridge, MA
8. Baker JK (1975) Stochastic modeling for automatic speech understanding. In: Reddy R (ed) *Speech recognition*. Academic, New York, pp 521–542
9. Morgan N, Bourlard H (1995) Continuous speech recognition. *IEEE Signal Process Mag* 12(3):25–42
10. Young S (1996) A review of large-vocabulary continuous-speech recognition. *IEEE Signal Process Mag* 13(5):45–57
11. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 77(2):257–286
12. Young SJ, Evermann G, Gales MJF, Hain T, Kershaw D, Moore GL, Odell JJ, Ollason D, Povey D, Valtchev V, Woodland PC (2004) *The HTK Book*
13. HTK (2011) <http://htk.eng.cam.ac.uk/>. Accessed 1 Sep 2011
14. Lee KF (1988) Large vocabulary speaker independent continuous speech recognition: the Sphinx system. Doctoral dissertation, Carnegie Mellon University

15. CMU Sphinx Downloads (2011)  
<http://cmuSphinx.sourceforge.net/wiki/download>, Accessed 1 Sep 2011
16. Open Source Toolkit for Speech Recognition (2011)  
<http://cmuSphinx.sourceforge.net/wiki/download/>, Accessed 1 Sep 2011
17. Rabiner, L. R. and Juang, B. H., Statistical Methods for the Recognition and Understanding of Speech, Encyclopedia of Language and Linguistics, 2004
18. The Perception Processor (2011)  
<http://www.siliconintelligence.com/people/binu/perception/node22.html>,  
 Accessed 1 Dec 2011
19. The CMU Pronunciation Dictionary (2011), <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>, Accessed 1 September 2011.
20. Hwang M-H (1993) Subphonetic acoustic modeling for speaker-independent continuous speech recognition, Ph.D. thesis, School of Computer Science, Carnegie Mellon University
21. Lamere P, Kwok P, Walker W, Gouvea E, Singh R, Raj B, Wolf P (2003) Design of the CMU Sphinx-4 decoder. In: Proceedings of the 8th European conference on speech communication and technology, Geneva, Switzerland, pp 1181–1184
22. Huang XD (1992) Phoneme classification using semicontinuous hidden Markov models. IEEE Trans Signal Process 40(5):1062–1067
23. Forney GD (1973) The Viterbi algorithm. Proc IEEE 61(3):268–278
24. Clarkson P, Rosenfeld R (1997) Statistical language modeling using the CMU-Cambridge toolkit. In: Proceedings of the 5th European conference on speech communication and technology, Rhodes, Greece
25. Bilmes J (2006) What HMMs can do. IEICE Trans Inf Syst E89-D(3):869–891
26. Baker J, Deng L, Glass J, Khudanpur S, Lee C, Morgan N (2007) Historical development and future directions in speech recognition and understanding, MINDS report. <http://www-nlpir.nist.gov/MINDS/FINAL/speech.web.pdf>
27. Deng L, Huang X (2004) Challenges in adopting speech recognition. Commun ACM 47(1):69–75
28. Gales, M. and S. Young (2007). "The application of hidden Markov models in speech recognition." Found. Trends Signal Process. 1(3): 195-304.

29. Ye-Yi W, Dong Y et al (2008) An introduction to voice search. *IEEE Signal Process Mag* 25 (3):28–38
30. Sainath, T. N., B. Ramabhadran, et al. (2009). An exploration of large vocabulary tools for small vocabulary phonetic recognition. *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*.
31. Zweig G, Nguyen P (2009) A segmental CRF approach to large vocabulary continuous speech recognition. *IEEE workshop on automatic speech recognition and understanding, 2009. ASRU 2009*
32. Luo X (2011) Chinese speech recognition based on a hybrid SVM and HMM architecture advances in neural networks. In: Liu D, Zhang H, Polycarpou M, Alippi C, He H (eds) *ISNN 2011, LNCS 6677*. Springer, Berlin, pp 629–635
33. Xi X, Lin K, Zhou C, Cai J (2005) A new hybrid HMM/ANN model for speech recognition. In: *Proceedings of the second IFIP conference on artificial intelligence applications and innovations (AIAI 2005)*, pp 223–230
34. Xiao Y, Qin A (2010) Noise robust speech recognition based on improved hidden Markov model and wavelet neural network. *Comput Eng Appl* 46(22): pp 162–164, 235
35. Sloin A, Burshtein D (2008) Support vector machine training for improved hidden Markov modeling. *IEEE Trans Signal Process* 56(1):172–188
36. Xian T (2009) Hybrid Hidden Markov Model and artificial neural network for automatic speech recognition. *Pacific-Asia conference on circuits, communications and systems, 2009. PACCS'09*
37. Middag C, Martens J-P et al (2009) Automated intelligibility assessment of pathological speech using phonological features. *EURASIP J Adv Signal Process* 2009:1–9
38. Clarkson, P., & Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of the 5<sup>th</sup> European conference on speech communication and technology, Rhodes, Greece, Sept. 1997*.
39. Salgado-Garza, L., R. Stern, et al. (2004). N-Best List Rescoring Using Syntactic Trigrams *MICAI 2004: Advances in Artificial Intelligence*. R. Monroy, G. Arroyo-Figueroa, L. Sucar and H. Sossa, Springer Berlin / Heidelberg. **2972**: 79-88.
40. Beutler R (2007) Improving speech recognition through linguistic knowledge. *Doctoral dissertation, ETH Zurich*

41. Schwenk H (2007) Continuous space language models. *Comput Speech Lang* 21(3):492–518
42. Yuecheng Z, Mnih A, Hinton G (2008) Improving a statistical language model by modulating the effects of context words, in: *ESANN*, 2008
43. Elshafei MA (1991) Toward an Arabic text-to-speech system. *Arab J Sci Eng* 16(4B):565–583
44. Alghamdi M (2000) Arabic phonetics. Attaoobah, Riyadh
45. Al-Ghamdi M, Elshafei M, Al-Muhtaseb H (2003) An experimental Arabic text-to-speech system. Final report, King Abudaziz City of Science and Technology
46. Elshafei M, Almuhtasib H, Alghamdi M(2002) Techniques for high quality text-to-speech. *Inform Sci* 140(3–4):255–267
47. Elshafei M, Ali M, Al-Muhtaseb H, Al-Ghamdi M (2007) Automatic segmentation of Arabic speech. Workshop on information technology and Islamic sciences, Imam Mohammad Ben Saud University, Riyadh, March 2007
48. Elshafei M, Al-Muhtaseb H, Alghamdi M (2006) Statistical methods for automatic diacritization of Arabic text. In: *Proceedings of 18th national computer conference NCC'18*, Riyadh, March 26–29, 2006
49. Al-Otaibi F (2001) speaker-dependant continuous Arabic speech recognition. M.Sc. thesis, King Saud University
50. Hyassat H, Abu Zitar R (2008) Arabic speech recognition using Sphinx engine. *Int J Speech Tech* 9(3–4):133–150
51. Kirchhoff K, Bilmes J, Das S, Duta N, Egan M, Ji G, He F, Henderson J, Liu D, Noamany M, Schoner P, Schwartz R, Vergyri D (2003) Novel approaches to Arabic speech recognition: report from the 2002 John-Hopkins summer workshop, *ICASSP 2003*, pp I344–I347
52. Soltau H, Saon G et al (2007) The IBM 2006 Gale Arabic ASR system. *IEEE international conference on acoustics, speech and signal processing*, 2007. *ICASSP 2007*
53. AzmiM, Tolba H, Mahdy S, FashalM(2008) Syllable-based automatic Arabic speech recognition in noisy-telephone channel. In: *WSEAS transactions on signal processing proceedings*, World Scientific and Engineering Academy and Society (WSEAS), vol 4, issue 4, pp 211–220

54. Abdou SM, Hamid SE, Rashwan M, Samir A, Abd-Elhamid O, Shahin M, Naz W (2006) Computer aided pronunciation learning system using speech recognition techniques, NTERSPPEECH 2006, ICSLP, pp 249–252
55. Choi F, Tsakalidis S et al (2008) Recent improvements in BBN’s English/Iraqi speech-to-speech translation system. IEEE Spoken language technology workshop, 2008. SLT 2008
56. Rambow O et al (2006) Parsing Arabic dialects, final report version 1, Johns Hopkins summer workshop 2005
57. Nofal M, Abdel Reheem E et al (2004) The development of acoustic models for command and control Arabic speech recognition system. 2004 international conference on electrical, electronic and computer engineering, 2004. ICEEC’04
58. Park J, Diehl F et al (2009) Training and adapting MLP features for Arabic speech recognition. IEEE international conference on acoustics, speech and signal processing, 2009. ICASSP 2009
59. Shoaib M, Rasheed F, Akhtar J, Awais M, Masud S, Shamail S (2003) A novel approach to increase the robustness of speaker independent Arabic speech recognition. 7th international multi topic conference, 2003. INMIC 2003. 8–9 Dec 2003, pp 371–376
60. Imai T, Ando A et al (1995) A new method for automatic generation of speaker-dependent phonological rules. 1995 international conference on acoustics, speech, and signal processing, 1995. ICASSP-95
61. Choueiter G, Povey D et al (2006) Morpheme-based language modeling for Arabic LVCSR. 2006 IEEE international conference on acoustics, speech and signal processing. ICASSP 2006 proceedings
62. Bourouba H, Djemili R et al (2006) New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition. 2nd Information and Communication Technologies, 2006. ICTTA’06
63. Sagheer A, Tsuruta N et al (2005) Hyper column model vs. fast DCT for feature extraction in visual Arabic speech recognition. In: Proceedings of the fifth IEEE international symposium on signal processing and information technology, 2005
64. Taha M, Helmy T et al (2007) Multi-agent based Arabic speech recognition. 2007 IEEE/WIC/ ACM international conferences on web intelligence and intelligent agent technology workshops

65. Elmisery FA, Khalil AH et al (2003) A FPGA-based HMM for a discrete Arabic speech recognition system. In: Proceedings of the 15th international conference on microelectronics, 2003. ICM 2003
66. Mokhtar MA, El-Abddin AZ (1996) A model for the acoustic phonetic structure of Arabic language using a single ergodic hidden Markov model. In: Proceedings of the fourth international conference on spoken language, 1996. ICSLP 96
67. Gales MJF, Diehl F et al (2007) Development of a phonetic system for large vocabulary Arabic speech recognition. IEEE workshop on automatic speech recognition and understanding, 2007. ASRU
68. Bahi H, Sellami M (2001) Combination of vector quantization and hidden Markov models for Arabic speech recognition. ACS/IEEE international conference on computer systems and applications, 2001
69. El-Ramly SH, Abdel-Kader NS, El-Adawi R (2002) Neural networks used for speech recognition. In: Proceedings of the nineteenth national radio science conference (NRSC 2002), March 2002, pp 200–207
70. Bahi H, Sellami M (2003) A hybrid approach for Arabic speech recognition. ACS/IEEE international conference on computer systems and applications, 14–18 July 2003
71. Alimi AM, Ben Jemaa M (2002) Beta fuzzy neural network application in recognition of spoken isolated Arabic words. Int J Contr Intell Syst 30(2), Special issue on speech processing techniques and applications
72. Alotaibi YA (2004) Spoken Arabic digits recognizer using recurrent neural networks. In: Proceedings of the fourth IEEE international symposium on signal processing and information technology, pp 195–199
73. Essa EM, Tolba AS et al (2008) A comparison of combined classifier architectures for Arabic speech recognition. International conference on computer engineering and systems, 2008. ICCES 2008
74. Emami A, Mangu L (2007) Empirical study of neural network language models for Arabic speech recognition. IEEE workshop on automatic speech recognition and understanding, 2007. ASRU
75. Alghamdi M, Elshafei M, Almuhtasib H (2009) Arabic broadcast news transcription system. Int J Speech Tech 10:183–195
76. Alghamdi M., Almuhtasib H., Elshafei M., "Arabic Phonological Rules", King Saud University Journal: Computer Sciences and Information. Vol. 16, pp. 1-25, 2004.

77. The CMU Pronouncing Dictionary (2011) <http://www.speech.cs.cmu.edu/cgi-bin/cmudict> , Accessed 1 Sep 2011
78. Ali M, Elshafei M, Alghamdi M, Almuhtaseb H, Alnajjar A (2009) Arabic phonetic dictionaries 236 for speech recognition. *J Inform Tech Res* 2(4):67–80
79. Satori H, Harti M, Chenfour N (2007) Introduction to Arabic speech recognition using CMU Sphinx system. *Information and communication technologies international symposium proceeding ICTIS07, 2007*
80. Afify M, Nguyen L, Xiang B, Abdou S, Makhoul J. Recent progress in Arabic broadcast news transcription at BBN. In: *Proceedings of INTERSPEECH. 2005*, pp 1637–1640
81. Billa J, Noamany M et al (2002) Audio indexing of Arabic broadcast news. 2002 IEEE international conference on acoustics, speech, and signal processing (ICASSP)
82. Messaoudi A, Gauvain JL et al (2006) Arabic broadcast news transcription using a one million word vocalized vocabulary. 2006 IEEE international conference on acoustics, speech and signal processing, 2006. *ICASSP 2006 proceedings*
83. Elmahdy M, Gruhn R et al (2009) Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition. In: *Eighth international symposium on natural language processing, 2009. SNLP'09*
84. Vergyri D, Kirchhoff K, Duh K, Stolcke A (2004) Morphology-based language modeling for Arabic speech recognition. *International conference on speech and language processing. Jeju Island*, pp 1252–1255
85. Xiang B, Nguyen K, Nguyen L, Schwartz R, Makhoul J (2006) Morphological decomposition for Arabic broadcast news transcription. In: *Proceedings of ICASSP, vol I. Toulouse*, pp 1089–1092
86. Selouani S-A, Alotaibi YA (2011) Adaptation of foreign accented speakers in native Arabic ASR systems. *Appl Comput Informat* 9(1):1–10
87. Saon G, Soltau H et al (2010) The IBM 2008 GALE Arabic speech transcription system. 2010 IEEE international conference on acoustics speech and signal processing (ICASSP)
88. Kuo HJ, Mangu L et al (2010) Morphological and syntactic features for Arabic speech recognition. 2010 IEEE international conference on acoustics speech and signal processing (ICASSP)

89. Alotaibi Y, Selouani S, O'Shaughnessy D (2008) Experiments on automatic recognition of nonnative Arabic speech. *EURASIP J Audio Speech Music Process*: 9 pages. doi:10.1155/2008/679831, Article ID 679831
90. Abushariah MAM, Ainon RN et al (2010) Natural speaker-independent Arabic speech recognition system based on Hidden Markov Models using Sphinx tools. 2010 international conference on computer and communication engineering (ICCCE)
91. Huang X, Acero A, Hon H (2001) *Spoken language processing*. Prentice Hall PTR, Upper Saddle River, NJ
92. Muhammad G, AlMalki K et al (2011) Automatic Arabic digit speech recognition and formant analysis for voicing disordered people. 2011 IEEE symposium on computers and informatics (ISCI)
93. Gallwitz F, Noth E, et al (1996) A category based approach for recognition of out-of-vocabulary words. In: *Proceedings of fourth international conference on spoken language, 1996. ICSLP 96*
94. Jurafsky D, Martin J (2009) *Speech and language processing, 2nd edn*. Pearson, NJ
95. Jelinek F (1999) *Statistical methods for speech recognition, Language, speech and communication series*. MIT, Cambridge, MA
96. Saon G, Padmanabhan M (2001) Data-driven approach to designing compound words for continuous speech recognition. *IEEE Trans Speech Audio Process* 9(4):327–332
97. Plötz T (2005) *Advanced stochastic protein sequence analysis, Ph.D. thesis, Bielefeld University* Saon G, Padmanabhan M (2001) Data-driven approach to designing compound words for continuous speech recognition. *IEEE Trans Speech Audio Process* 9(4):327–332
98. McAllester D, Gillick L, Scattone F, Newman M (1998) Fabricating conversational speech data with acoustic models: a program to examine model-data mismatch. In: *Proceedings of ICSLP, Sydney, Australia, December 1998*
99. Hofmann H, Sakti S et al (2010) Improving spontaneous English ASR using a joint-sequence pronunciation model. 2010 4th international universal communication symposium (IUCS)

100. Wester M (2003) Pronunciation modeling for ASR—knowledge-based and data-derived methods. *Comput Speech Lang* 17:69–85
101. Amdal I, Fosler-Lussier E (2003) Pronunciation variation modeling in automatic speech recognition. *Teletronikk*, 2.2003, pp 70–82
102. Wester M, Fosler-Lussier E (2000) A comparison of data-derived and knowledge-based modeling of pronunciation variation, ICSLP, Beijing, China, 2000
103. Helmer Strik , Pronunciation Adaptation At the Lexical Level ,Proceedings ISCA ITRW Workshop Adaptation Methods for Speech Recognition , Sophia Antipolis, France, 2001.
104. Sloboda T, Waibel A (1996) Dictionary learning for spontaneous speech recognition. In: *Proceedings of ICSLP-96*, Philadelphia, PA, USA, pp 2328–2331
105. Fosler-Lussier E, Greenberg S, Morgan N (1999) Incorporating contextual phonetics into automatic speech recognition. In: *Proceedings of the international congress on phonetic sciences*, pp 611–614
106. Saraçlar M, Nock H, Khudanpur S (2000) Pronunciation modeling by sharing Gaussian densities across phonetic models. *Comput Speech Lang* 14:137–160
107. Al-Haj H, Hsiao R, Lane I, Black A, Waibel A (2009) Pronunciation modeling for dialectal Arabic speech recognition, ASRU 2009: IEEE workshop, Italy
108. Biadys F, Habash N, Hirschberg J (2009) Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. The 2009 annual conference of the North American chapter of the ACL, Colorado, pp 397–405
109. Tajchman G, Foster E, Jurafsky D (1995) Building multiple pronunciation models for novel words using exploratory computational phonology. In *EUROSPEECH-1995*, pp 2247–2250
110. Finke M, Waibel A (1997) Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition. In: *Proceedings of EuroSpeech-97*, Rhodes, pp 2379–2382
111. Kessens JM, Wester M et al (1999) Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation. *Speech Comm* 29(2–4):193–207

112. Kyong-Nim L., Minhwa C., "Morpheme-Based Modeling of Pronunciation Variation for Large Vocabulary Continuous Speech Recognition in Korean," IEICE - Transactions on Information and Systems, v.E90-D n.7, p.1063-1072, July 2007.
113. Jeon, J., Cha, S., Chung, M., Park, J., & Hwang, K. (1998). Automatic generation of Korean pronunciation variants by multistage applications of phonological rules. In ICSLP-1998 (paper 0675).
114. Liu, Y., & Fung, P. (2003). Modeling partial pronunciation variations for spontaneous Mandarin speech recognition. *Computer Speech and Language*.
115. Seman N, Kamaruzaman J (2008) Acoustic pronunciation variations modeling for standard Malay speech recognition. *Comput Inform Sci* 1(4):112–120, ISSN 1913-8989
116. Marina Alexandersson, Sequence analysis – Pairwise sequence alignment, introduction in bioinformatics, 2005  
[http://bio.lundberg.gu.se/courses/ht05/bio1/L3\\_PairAlign.pdf](http://bio.lundberg.gu.se/courses/ht05/bio1/L3_PairAlign.pdf)
117. Alsuwaiyel, M. H. (2003). *Algorithms: design techniques and analysis*. Singapore: World Scientific.
118. Hirsimaki T (2003) A review: decision trees in speech recognition. Helsinki University of Technology, Finland
119. Hazen TJ, Hetherington IL, Shu H, Livescu K (2005) Pronunciation modeling using a finite-state transducer representation. *Speech Comm* 46(2):189–203
120. High Performance Computing (HPC) Center, 2011.  
<http://hpc.kfupm.edu.sa/Home.htm>
121. Cao, G., J.-Y. Nie, et al. (2005). Integrating word relationships into language models. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. Salvador, Brazil, ACM: 298-305.
122. Ruiz-Casado et al., 2007 M. Ruiz-Casado, E. Alfonseca and P. Castells, Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data Knowledge and Engineering*, 61 3 (2007), pp. 484–499.
123. Wang, W., Y. Liu, et al. (2002). Rescoring effectiveness of language models using different levels of knowledge and their integration. *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*.

124. Xiang, B., B. Zhou, et al. (2009). Advances in syntax-based Malay-English speech translation. Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE Computer Society: 4801-4804.
125. Diab M., Hacioglu K., Jurafsky D., "Automatic tagging of Arabic text: from raw text to base phrase chunks", 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference, 2004.
126. Al-Shamsi F. and Guessoum A., "A Hidden Markov Model –Based PS Tagger for Arabic", 2006, CiteSeerX.
127. El-Hadj Y., Al-Sughayeir I. and Al-Ansari A., Arabic Part-Of-Speech Tagging using the Sentence Structure, Proceedings of the Second International Conference on Arabic Language Resources and Tools, The MEDAR Consortium, 2009.
128. Albared M., Omar N., Ab Aziz M., Zakree M., Nazri A., Automatic part of speech tagging for Arabic: an experiment using Bigram hidden Markov model, RSKT'10 Proceedings of the 5th international conference on Rough set and knowledge technology, 2010.
129. Stanford Log-linear Part-Of-Speech Tagger, 2011.  
<http://nlp.stanford.edu/software/tagger.shtml>
130. Tobias Scheffer, T. (2005). "Finding association rules that trade support optimally against confidence." *Intell. Data Anal.* 9(4): 381-3
131. Machine Learning Group at University of Waikato, 2011.  
<http://www.cs.waikato.ac.nz/ml/weka/>
132. High Performance Computing (HPC) Center, 2011.  
<http://hpc.kfupm.edu.sa/Home.htm>
133. Jeon, J.H., Wang, W., and Liu, Y. N-Best Rescoring Based on Pitch-accent Patterns. In Proceedings of ACL. 2011, 732-741.
134. Ganapathiraju, A., J. E. Hamaker, et al. (2004). "Applications of support vector machines to speech recognition." *Signal Processing, IEEE Transactions on* **52**(8): 2348-2355.
135. Birkenes O., Matsui, T. Tanabe, K. and André T. (2008). Automatic Speech Recognition via N-Best Rescoring using Logistic Regression, *Speech Recognition*, France Mihelic and Janez Zibert (Ed.), ISBN: 978-953-7619-29-9, InTech.

136. Jang, P.J. and Hauptmann, A.G. Hierarchical cluster language modeling with statistical rule extraction for rescoring n-best hypotheses during speech decoding. In Proceedings of ICSLP. 1998.

# VITA

## **Personal Information:**

Name: Dia Eddin M. AbuZeina

Nationality: Palestinian – Jordanian Passport

Resident of: Alzahran – Eastern Provenance – Saudi Arabia

Birth date: 22 April, 1976

Marital status: Married

Cell Phone: 00966555786176

Email: [abuzeina@kfupm.edu.sa](mailto:abuzeina@kfupm.edu.sa)

## **Education:**

June 2001                      Palestine Polytechnic University, Hebron, Palestine.

Bachelor's degree, Computer System Engineering.

August 2005                  Southern New Hampshire University, Manchester, USA

Master of Science, Information Technology.

December 2011              King Fahd University of Petroleum and Minerals, Saudi Arabia.

PhD in Computer Science and Engineering.

## **Research Interest:**

Arabic Speech Recognition, data-mining, and hidden Markov models

**Permanent Address:** Hebron – Palestine, [abuzeina@hotmail.com](mailto:abuzeina@hotmail.com), [abuzeina@ppu.edu](mailto:abuzeina@ppu.edu)