EXPLOITING PHONOLOGICAL CONSTRAINTS AND AUTOMATIC IDENTIFICATION OF SPEAKER CLASSES FOR ARABIC SPEECH RECOGNITION

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences

2014

By
Iman Alsharhan
School of Computer Science
# Contents

Abstract 16

Acknowledgements 17

Declaration 19

Copyright 20

1 Introduction 21

1.1 Research related problems .................................. 23
1.2 Research objectives ......................................... 25
1.3 Structure of the thesis ...................................... 28

2 Automatic speech recognition: an overview 31

2.1 Types of ASR systems ....................................... 32
2.2 Challenges in developing ASR systems ....................... 33
  2.2.1 Human understanding of speech compared to ASR ...... 33
  2.2.2 Speaker variability ...................................... 34
  2.2.3 Language complexity .................................... 40
  2.2.4 Environment variation .................................. 45
2.3 Architecture of ASR ........................................ 46
  2.3.1 The Front-End ........................................... 47
  2.3.2 Probabilistic models .................................... 49
  2.3.3 Searching and decoding ................................. 53
  2.3.4 Recognition evaluation .................................. 53
2.4 Open source speech recognition systems ....................... 54

3 Modern standard Arabic from the perspective of ASR 56

3.1 Properties of Arabic ....................................... 56
3.1.1 Arabic forms ................................................. 57
3.1.2 Arabic writing system ...................................... 59
3.1.3 Phonetic system of MSA .................................... 62
3.1.4 Phonology and morphology of MSA ...................... 65
3.1.5 Arabic syntax ............................................... 67
3.1.6 Mapping between spoken and written text .............. 69

3.2 Review of the work done in this field .................... 71
3.2.1 Works dedicated to the recognition of MSA ........... 71
3.2.2 Works dedicated to the recognition of dialectal Arabic 74
3.2.3 Works done on Arabic digits recognition .............. 75
3.2.4 Works done on Arabic alphabet recognition .......... 76

4 Experimental set-up ............................................. 78
4.1 Designing an HMM-based system ............................ 79
4.1.1 Main issues in using HMMs: problems and solutions ... 79
4.2 The HTK ......................................................... 81
4.2.1 The HTK library modules ................................. 81
4.2.2 The HTK processing stages .............................. 82
4.2.3 Data settings .............................................. 97
4.2.4 Experimental scripts ..................................... 99

4.3 Training and testing corpora: collection and preparation 101
4.3.1 Pilot dataset .............................................. 102
4.3.2 Main dataset ............................................ 104

5 Acquiring a context-sensitive phonetic transcription for a given MSA text ............................................. 109
5.1 The process of deriving a phonetic transcription ........ 110
5.1.1 Pre-processing the text ................................... 111
5.1.2 Grapheme-to-phoneme rules ............................. 114
5.1.3 Lexicon of the exceptional words ...................... 121
5.2 Phonological rules ......................................... 123
5.2.1 Assimilation ............................................. 123
5.2.2 Neutralisation .......................................... 135
5.2.3 The pharyngealisation of /l/ and /r/ .................. 136
5.2.4 Aligning stress to syllables ............................ 137
5.3 Testing and evaluating of the grapheme-to-allophone system ... 140
List of Tables

1  Arabic letters as presented in Buckwalter, SAMPA, and IPA schemes 14
3.1 V ariations in digits pronunciation within MSA and five major dia-
lects. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
3.2 T h e Arabic consonants phonemes. . . . . . . . . . . . . . . . . . . 63
3.3 T h e Arabic vowels system. . . . . . . . . . . . . . . . . . . . . . . 64
5.1 T h e Arabic sound’s features as used in writing the pronunciation
rules. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
5.2 Solar and lunar letters. . . . . . . . . . . . . . . . . . . . . . . . . 124
5.3 Examples of words with solar letters. . . . . . . . . . . . . . . . . . 126
5.4 Examples of assimilating the sounds /t/ and /d/. . . . . . . . . . . 129
5.5 Examples of assimilating the dental sounds. . . . . . . . . . . . . 129
5.6 Examples of the selected test words with single and multiple pho-
netic transcription. T he table shows the phonetic transcription of
the word, how many times it appears in the text, and the sur-
rounding context. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 141
5.7 Resources for the mismatches between the generated transcription
and the annotator’s transcription. . . . . . . . . . . . . . . . . . 146
6.1 Results of using different pronunciation resources in training the
speech recogniser with two different data sets. 5-fold cross-validation
approach is used to test the system. T he word recognition accuracy
is provided as a common speech recognition accuracy metric. . . . 154
6.2 Results of testing individual phonological rules. T he table shows
the type of the eliminated phonological rule, the number of oc-
currences, and the effect of eliminating this rule on the system’s
performance. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
6.3 Results of adding the stress value in the phonetic transcription. The table provides a comparison between the baseline system (no stress) and different levels of adding the stress marker. 161

6.4 Experiments’ results using the main and pilot datasets. The table provides a comparison between the use of phonemes and words as the HMM units. Two systems are reported: one using a fixed pronunciation dictionary and another used the generated phonetic transcription. 5-fold cross-validation approach is used to test the reported systems. 164

6.5 Phoneme-level and word-level experiments’ results with sentence accuracy. 165

6.6 Underspecification experiments’ results using two different datasets. The table reports the results of different systems using different underspecification levels with both fixed dictionary and generated phonetic transcription. 5-fold cross-validation approach is used to test the system. 170

6.7 Gender-dependent experiments’ results. The first system uses 20 female speakers for the training while the second training system uses 20 male speakers. A total of 56 speakers were used in testing. 175

6.8 Results of speakers’ group experiments. The first system is trained on female and pseudo-female speakers, while the second one is trained on male and pseudo-male speakers. All of the 56 speakers are used in testing both systems. 178

6.9 Results of testing Gulf-accented speakers against the accent-specific speakers’ group models. 180

6.10 WER for increasing quantity of training data as reported by [LGA00]. 183

6.11 WER for varying amounts of training data in supervised, unsupervised, and unsupervised with reduced language model training acoustic models as shown in [LGA02]. 183

6.12 Accuracy rate for varying amounts of training data with a fixed dictionary-based system and a generated phonetic transcription-based system. 185

6.13 Accuracy rate for varying amount of training data in the filtered database. 187
B.1 Details of the speakers participated in recording the pilot corpus.  
Showing their gender, regional accent, age, total number of recordings, number of repetitions, and the average pitch. . . . . . . . . . 219

D.1 Details of the speakers participated in recording the main corpus.  
Showing their gender, regional accent, age, total number of recordings, number of repetitions, and the average pitch. . . . . . . . . . 233

G.1 Gender-specific and accent-specific experiments’ results. . . . . . . 242

H.1 Results of testing on a mixed-gender model. . . . . . . . . . . . . 244
List of Figures

1.1 Basic structure of the ASU system. ............................... 24
2.1 The distribution of $F_0$ over 326 male speakers and 136 female
speakers as shown in [Ho01]. .................................. 36
2.2 Speech recognition process. ...................................... 47
2.3 The front-end process. .......................................... 48
2.4 Probabilistic models needed for decoding a sentence. ............. 49
2.5 Three-states HMMs. ........................................... 52
3.1 Examples for the ambiguity caused by the lack of diacritics. ..... 61
3.2 Arabic derivational morphology. .................................. 66
3.3 Ambiguity resulted from pro-dropping. ............................ 68
4.1 Two major phases of running the HTK. ............................ 82
4.2 Configuration file “config”. ..................................... 82
4.3 The HTK tools used in each processing stage [YEG+06]. ........... 83
4.4 Generating a phone level transcription. ........................... 85
4.5 The contents of the script file “codelr”. ........................... 85
4.6 Coding the training data. ....................................... 86
4.7 Data preparation process. ...................................... 86
4.8 The training process. .......................................... 87
4.9 Detailed training process. ...................................... 87
4.10 “Proto” file. ................................................ 88
4.11 flat start initialisation. ......................................... 89
4.12 Modifying silence models [YEG+06]. ............................ 90
4.13 Realigning the training data. .................................. 90
4.14 Generating triphones. .......................................... 91
4.15 The structure of the decision tree file (tree.hed). ............... 92
4.16 An example of the structure of the recognition network. ......... 94
4.17 The contents of “testprompts” file. 94
4.18 The contents of testref.mlf file. 95
4.19 The structure of the “codeTest” file. 95
4.20 Recognising the test data process. 96
4.21 HTK workflow, areas written in red are the places where changes might be introduced in the coming experiments. 97
4.22 The contents of the confusion matrix file. 99
4.23 The contents of “prompts_res” file. 100
4.24 A screen shot for the web-based recording tool. 102

5.1 The architecture of the developed grapheme-to-allophone system. 110
5.2 Rules for labelling the characters with consonant and vowel value. 116
5.3 Syllabication of the phrase (’uHîb∼’u AiblîsAmatak). 119
5.4 Grapheme-to-phoneme conversion rules. 122
5.5 Arabic consonants’ point of articulation. 125
5.6 Total assimilation of “IAm” in saying /?ajfams/. 127
5.7 The pronunciation of (l) when not assimilated, as in saying /?aljams/ 127
5.8 The spectrogram of the word sAriq (the long vowel /a:/ is highlighted). 133
5.9 The spectrogram of the word SAdiq (the long vowel /a:/ is highlighted). 133
5.10 Syllabication rules. 139

6.1 HTK workflow, the area written in red indicates where changes are introduced in this kind of experiments. 152
6.2 Snapshots of the contents of the generated phonetic transcription file, fixed pronunciation dictionary file, and the multi-pronunciation dictionary file. 155
6.3 HTK workflow, areas written in red are the places where changes are introduced in this kind of experiments. 163
6.4 A snapshot of the source files contents: prompts, dictionary, and grammar for a phoneme level system. 164
6.5 HTK workflow, areas written in red are the places where changes are introduced in the underspecification experiments. 169
6.6 The average pitch rate for our male and female speakers. 176
6.7 Average WER percentage for speech recognition system with different-sized training materials as reported by [EM99] 182
6.8 Average WER percentage for speech recognition system with different-sized training materials as reported by [Pov03] 184
6.9 Accuracy rate for speech recognition system with different-sized training materials 185
6.10 Accuracy rate for speech recognition system with different-sized filtered training materials 187
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANNs</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
</tr>
<tr>
<td>ASU</td>
<td>Automatic Speech Understanding</td>
</tr>
<tr>
<td>C</td>
<td>Consonant</td>
</tr>
<tr>
<td>CA</td>
<td>Classical Arabic</td>
</tr>
<tr>
<td>CES</td>
<td>Connectionist Expert System</td>
</tr>
<tr>
<td>CFG</td>
<td>Context-Free Grammar</td>
</tr>
<tr>
<td>CH</td>
<td>Call-Home</td>
</tr>
<tr>
<td>CMU</td>
<td>Carnegie Mellon University's</td>
</tr>
<tr>
<td>CSLU</td>
<td>Center for Spoken Language Understanding</td>
</tr>
<tr>
<td>CUED</td>
<td>Cambridge University Engineering Department</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DSR</td>
<td>Distributed Speech Recognition</td>
</tr>
<tr>
<td>DTW</td>
<td>Dynamic Time Warping</td>
</tr>
<tr>
<td>ECA</td>
<td>Egyptian Colloquial Arabic</td>
</tr>
<tr>
<td>HMMs</td>
<td>Hidden Markov Models</td>
</tr>
<tr>
<td>HSPG</td>
<td>Head-Driven Phrase Structure Grammar</td>
</tr>
<tr>
<td>HTK</td>
<td>Hidden Markov Model Toolkit</td>
</tr>
<tr>
<td>IPA</td>
<td>International Phonetic Alphabet</td>
</tr>
<tr>
<td>LDC</td>
<td>Linguistic Data Consortium</td>
</tr>
<tr>
<td>LIN</td>
<td>Linear Input Network</td>
</tr>
<tr>
<td>LVCSR</td>
<td>Large Vocabulary Continuous Speech Recognition</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori</td>
</tr>
<tr>
<td>MFCCs</td>
<td>Mel-Frequency Cepstral Coefficients</td>
</tr>
<tr>
<td>MLLR</td>
<td>Maximum Likelihood Linear Regression</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi Layer Perceptron</td>
</tr>
<tr>
<td>MLPs</td>
<td>Multi-Layer Perceptrons</td>
</tr>
<tr>
<td>MRENN</td>
<td>Modular Recurrent Elman Neural Networks</td>
</tr>
<tr>
<td>MSA</td>
<td>Modern Standard Arabic</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NLTK</td>
<td>Natural Language Toolkit</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>NNs</td>
<td>Neural Networks</td>
</tr>
<tr>
<td>OOV</td>
<td>Out-of-Vocabulary</td>
</tr>
<tr>
<td>OVS</td>
<td>Object Verb Subject</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>SAAVB</td>
<td>Saudi Accented Arabic Voice Bank</td>
</tr>
<tr>
<td>SVO</td>
<td>Subject Verb Object</td>
</tr>
<tr>
<td>V</td>
<td>Vowel</td>
</tr>
<tr>
<td>VOS</td>
<td>Verb Object Subject</td>
</tr>
<tr>
<td>VSO</td>
<td>Verb Subject Object</td>
</tr>
<tr>
<td>VTLN</td>
<td>Vocal Tract Length Normalisation</td>
</tr>
<tr>
<td>WER</td>
<td>Word Error Rate</td>
</tr>
</tbody>
</table>
Arabic alphabet vs. transliteration and transcription

Throughout this thesis, we use Buckwalter scheme to romanise Arabic examples and the IPA scheme to represent the phonetic notation of these examples. An amended version of SAMPA phonetic scheme is used in writing the pronunciation rules and running the ASR experiments with HTK. Table 1 lists the characters’ representation used in the research.

Table 1: Arabic letters as presented in Buckwalter, SAMPA, and IPA schemes

<table>
<thead>
<tr>
<th>Arabic letter</th>
<th>Buckwalter</th>
<th>Amended SAMPA (original)</th>
<th>IPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>أ</td>
<td>A</td>
<td>aa</td>
<td>a:</td>
</tr>
<tr>
<td>ب</td>
<td>b</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>ت</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>ث</td>
<td>v</td>
<td>v</td>
<td>θ</td>
</tr>
<tr>
<td>ج</td>
<td>j</td>
<td>j (Z)</td>
<td>q</td>
</tr>
<tr>
<td>ح</td>
<td>H</td>
<td>h (X)</td>
<td>h</td>
</tr>
<tr>
<td>خ</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>د</td>
<td>d</td>
<td>d</td>
<td>δ</td>
</tr>
<tr>
<td>ذ</td>
<td>*</td>
<td>D</td>
<td>X</td>
</tr>
<tr>
<td>ر</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>ز</td>
<td>z</td>
<td>z</td>
<td>z</td>
</tr>
<tr>
<td>س</td>
<td>s</td>
<td>s</td>
<td>s</td>
</tr>
<tr>
<td>ﺶ</td>
<td>$</td>
<td>s^* (S)</td>
<td>ß</td>
</tr>
<tr>
<td>ﺶ</td>
<td>S</td>
<td>S^* (Ss.)</td>
<td>s^3</td>
</tr>
<tr>
<td>ض</td>
<td>D</td>
<td>D^* (d.)</td>
<td>d^3</td>
</tr>
<tr>
<td>ط</td>
<td>T</td>
<td>T^* (t.)</td>
<td>ð^3</td>
</tr>
<tr>
<td>ز</td>
<td>Z</td>
<td>Z^* (z.)</td>
<td>ð^3</td>
</tr>
<tr>
<td>ع</td>
<td>E</td>
<td>E (H)</td>
<td>ñ</td>
</tr>
<tr>
<td>غ</td>
<td>g</td>
<td>g (G)</td>
<td>x</td>
</tr>
<tr>
<td>ف</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>ق</td>
<td>q</td>
<td>q</td>
<td>q</td>
</tr>
<tr>
<td>ك</td>
<td>k</td>
<td>k</td>
<td>k</td>
</tr>
<tr>
<td>ل</td>
<td>l</td>
<td>l</td>
<td>l</td>
</tr>
<tr>
<td>م</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
</tbody>
</table>
Table 1: Arabic letters as presented in Buckwalter, SAMPA, and IPA schemes

<table>
<thead>
<tr>
<th>Arabic letter</th>
<th>Buckwalter</th>
<th>Amended SAMPA (original)</th>
<th>IPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ﺪ</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>ﺪ</td>
<td>n</td>
<td>M (not provided)</td>
<td>η</td>
</tr>
<tr>
<td>ﺪ</td>
<td>n</td>
<td>c (not provided)</td>
<td>η</td>
</tr>
<tr>
<td>ﺪ</td>
<td>n</td>
<td>e (not provided)</td>
<td>ι</td>
</tr>
<tr>
<td>ﺪ</td>
<td>h</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>ﺪ</td>
<td>w</td>
<td>w or uu</td>
<td>w or u:</td>
</tr>
<tr>
<td>ﺪ</td>
<td>y</td>
<td>y or ii (j)</td>
<td>j or i:</td>
</tr>
<tr>
<td>ﺪ</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>ﺪ</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>ﺪ</td>
<td>i</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>ﺪ</td>
<td>F</td>
<td>an</td>
<td>an</td>
</tr>
<tr>
<td>ﺪ</td>
<td>N</td>
<td>un</td>
<td>un</td>
</tr>
<tr>
<td>ﺪ</td>
<td>K</td>
<td>in</td>
<td>in</td>
</tr>
</tbody>
</table>
Abstract

Exploiting Phonological Constraints and Automatic Identification of Speaker Classes for Arabic Speech Recognition

Iman Alsharhan

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy, 2014

The aim of this thesis is to investigate a number of factors that could affect the performance of an Arabic automatic speech understanding (ASU) system. The work described in this thesis belongs to the speech recognition (ASR) phase, but the fact that it is part of an ASU project rather than a stand-alone piece of work on ASR influences the way in which it will be carried out. Our main concern in this work is to determine the best way to exploit the phonological properties of the Arabic language in order to improve the performance of the speech recogniser. One of the main challenges facing the processing of Arabic is the effect of the local context, which induces changes in the phonetic representation of a given text, thereby causing the recognition engine to misclassify it. The proposed solution is to develop a set of language-dependent grapheme-to-allophone rules that can predict such allophonic variations and eventually provide a phonetic transcription that is sensitive to the local context for the ASR system. The novel aspect of this method is that the pronunciation of each word is extracted directly from a context-sensitive phonetic transcription rather than a predefined dictionary that typically does not reflect the actual pronunciation of the word. Besides investigating the boundary effect on pronunciation, the research also seeks to address the problem of Arabic’s complex morphology. Two solutions are proposed to tackle this problem, namely, using underspecified phonetic transcription to build the system, and using phonemes instead of words to build the Hidden Markov Models (HMMs). The research also seeks to investigate several technical settings that might have an effect on the system’s performance. These include training on the sub-population to minimise the variation caused by training on the main undifferentiated population, as well as investigating the correlation between training size and performance of the ASR system.
Acknowledgements

After God Almighty, who inspired and blessed this effort all the way to its completion, I would like to express my heartfelt sense of gratitude to a number of people who provided me with guidance and support throughout this journey.

Above all, I would like to express my sincere thanks to Professor Allan Ramsay for his infinite patience, continuous guidance, and scholarly advice. His invaluable suggestions and guidance have shaped the main structure of this project even while helping me overcome the challenges that popped up along the way.

I would also like to thank Kuwait University for their generous sponsorship and the staff of the University of Manchester for their help in providing for me an academically enriching experience and a friendly environment.

To the members of my research group, especially Majid, Maytham, and Sardar who generously shared with me their knowledge and experience - Thank you.

The results reported in this thesis would not have been possible without the cooperation of those who participated in the collection of data. To all of them, I would like to extend my heartfelt gratitude for their time spent in doing this boring activity.

I will forever be thankful to my former supervisor, Dr. Salah Alnajem, for believing that I can go beyond the Master’s level and for encouraging me to pursue a career in research.

To my dear parents and loving brothers and sister, thank you for always being there for me. Thank you for being supportive and understanding towards my self-imposed isolation even during my holidays in Kuwait. Whenever things ran smoothly, I knew it was because you were all praying for me.

Last but not least, may I from the bottom of my heart thank my beloved husband Khalid. You have literally lived everything with me, the ups and downs - the latter for the most part, and you’ve always knew how to make me feel better. I really appreciate all the sacrifices you’ve made to follow me all the way while
I undertake this PhD. And to my two lovely boys, Abdulrahman and Hamad, thank you for always making me smile and for putting up with me when I had to work on my thesis instead of playing with you. Only God knows how much I love all three of you. Thank you.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Chapter 1

Introduction

Automatic speech understanding (ASU) is the process of giving the computer the ability to respond to speech in such a way that it produces the appropriate responses and actions. This is usually done by converting speech to a textual form using automatic speech recognition (ASR) techniques, which work on converting an acoustic signal to a set of words, and then constructing a meaning representation. ASU and ASR belong to the field of natural language processing (NLP) which also includes speech synthesis (text-to-speech). Generally speaking, the ultimate goal of these fields is to enhance human-computer interaction by designing machines that are indistinguishable from humans in their ability to hear, recognise, and understand speech as well as vocalising a spoken language. This technology can be useful for deployment in a number of applications such as:

- Language-to-language translation of a spoken sentence
- Voice-activated systems
- Dictation software
- Automatic transcription and subtitling
- Speech assistant language learning
- Interactive voice response
- Computer control for people with physical limitations
- Information retrieval, etc.

This work is part of a larger project aimed at automated speech understanding in terms of the Arabic language. The overall project will use a standard speech recognition toolkit to acquire a transcription of the input speech before making
use of linguistic analysis to extract the utterance’s underlying messages from the
given transcription. The work described in this thesis is at the speech recognition
phase, but the fact that it is part of an ASU project rather than a stand-alone
piece of work on ASR influences the way in which it will be carried out. Hence,
even though the major focus of the rest of the thesis will be on ASR-related
problems and solutions, we will point out areas where the intended application
influences the design or implementation.

Having a highly reliable speech recognition system is an absolute requirement
for developing any of the previously mentioned applications. Yet, there is still
a gap to bridge in order to develop accurate and robust ASR systems, notwith-
standing the effort and great progress made in this field during the last three
decades. The accuracy of the ASR system is achieved when the system is capable
of producing texts with no or little word error rate (WER). Robustness of the
ASR system is achieved when the system maintains its performance even when
exposed to different conditions.

A wide range of challenges have to be confronted when processing speech so
as to maintain the system’s accuracy and robustness. Among these challenges
is the limited sources of information given to the recognition engine compared
to human knowledge. Humans, for instance, use their knowledge of the speaker,
the topic and language system to predict the intended meaning. The variability
arising from differences among speakers (age, gender, regional accent, etc.) and
the emotional status of a single speaker can also increase speech variability and
hence affect the system’s performance. The recognition task is more challenging
with certain languages such as morphologically complex languages which tend to
have a very rich vocabulary. Furthermore, the quality of the recording channel
and the existence of any kind of noise in the speech can be obstacles towards
achieving a high performance recognition system.

Speech recognition systems can have different tasks of varying difficulty. The
complexity of the system varies greatly depending on the assigned task. For in-
stance, the following vary from low complexity to high complexity systems:

- Isolated word recognition, small vocabulary (tens of words), command-and-
control tasks.

- Continuous speech recognition (CSR), large vocabulary (thousands of words),
specified domain task.
• CSR, very large vocabulary (tens of thousands of words), spontaneous speech

The process of converting speech into a word sequence starts with the feature extraction stage where speech signals are compressed into a stream of relevant acoustic feature vectors, referred to as observations. Given the observed feature vectors, which are assumed to be well-compacted and carry sufficient information, three kinds of knowledge are required to infer the most likely word string. The sources of knowledge required by the recognition system are: acoustic model, lexicon (or pronunciation model), and language model. Briefly, the lexicon provides the sub-word units for every word present in the vocabulary and the language model. The acoustic model is constructed by mapping the extracted acoustic observations to the sub-word units (phonemes) found in the lexicon. The language model represents the local syntactic constraints of the sentences. This process is followed by a complete estimate of the sentence with the highest probability to output the most likely hypothesis of the uttered words. In ASU systems, a linguistic machinery is introduced for understanding the message that a speaker wants to convey in a speech using linguistic as well as background and contextual information. Figure 1.1 shows the basic structure of a typical ASU system.

1.1 Research related problems

Arabic is the most commonly used Semitic language in terms of the number of native speakers [Lew09]. However, far too little attention has been paid to developing an Arabic ASR system, so that the existing speech recognisers for Arabic are either limited (speaker-dependent or small vocabulary systems) or unreliable.

On top of the general challenges that may affect any language for which an ASU system is developed, the Arabic language poses the following problems:

i. Arabic is a collection of different forms and dialects, each differs in every aspect of linguistics. There is Modern Standard Arabic (MSA) which is used in formal communications, and dialectal Arabic which is used in daily communications and varies according to the speakers’ region. Arabic’s different forms should be considered as distinctive languages when developing speech recognition systems. Furthermore, the dialectal background can even influence the way MSA is spoken, mainly by replacing some phonemes, which also
have the effect of increasing the variability when developing an ASR system.

ii. Due to the complex derivational nature of Arabic morphology, a single stem can give rise to a large number of related words, either by affixation of derivational prefixes or by variation in the internal vowels. The complex morphological system also leads to the existence of multiple forms for a single word, which means that different forms of the same word can sound quite different. In other words, having two forms that sound the same does not mean that
they are the same, and having two forms that sound different does not mean that they are different. These morpho-phonological aspects of MSA can have direct consequences for the development of an Arabic ASU system. Firstly, the set of utterances required for training a system would have to be big enough in order to cover all the different forms of a single word (which may be more than thirty). These training sets are much larger than those typically used for languages with a less complex morphology. Secondly, in order to overcome the great lexical ambiguity that Arabic displays, some effective disambiguation mechanisms are needed.

iii. The local context of a phoneme in Arabic is likely to induce changes in its phonetic properties, thereby leading the recognition engine to misclassify it. This means that a considerable amount of work should be devoted to analysing the boundary effects on pronunciation, since this variation can have substantial repercussions for speech recognisers. To give a simple example, the pronunciation of the definite article in Arabic varies depending on the type of phoneme to which it is attached. For instance, if the definite article (Alاء /?al/) is attached to a word which starts with (n), such as (Alnuerǎ /?alnu:r/) “the illumination”, it will be pronounced /?annu:r/.

iv. The performance of the speech recognition system can be greatly influenced by the variability resulting from having speakers from different gender, age range, and dialectal region. That is mainly because Hidden Markov Models (HMMs)-based speech recognition systems are sensitive to any variability in the speech signals, and that makes the performance of speaker-independent recognition systems much worse compared to speaker-dependent systems.

1.2 Research objectives

The main project is aimed at developing an ASU system that targets MSA. Developing such a system for Arabic is difficult partly because of the problems
previously mentioned. My own part of the project is focused on the factors that make the speech signal difficult to process and presents solutions to overcome these problems within the ASR framework. The following are the main objectives of the research:

- One of the main sources of mismatch between speech and text is the boundary effect on pronunciation. The issue here is that the local context can have a considerable impact on what a phoneme may sound like. This can have substantial repercussions for speech recognisers, for it means that phonetic transcription which is required for training any speech recogniser needs to be sensitive to the local context. Our hypothesis is that these allophonic variations can be predicted and controlled by developing a set of phonological rules. Our first objective thus is to find common Arabic pronunciation rules and also use our phonological knowledge in analysing real Arabic speech in order to develop a substantial set of grapheme-to-allophone rules. Testing the robustness of the transcription system and the validity of the generated phonetic transcription will be carried out before making use of it in developing an Arabic ASR system. The grapheme-to-allophone system will be consulted while training the ASR system with a given text in order to provide a context-sensitive phonetic transcription for that text. The effectiveness of this method in improving the recognition system’s performance is investigated by comparing it with a system that uses a typical fixed dictionary.

The novelty of this method lies in integrating the grapheme-to-allophone system within the speech recogniser and canceling the role of the predefined dictionary in building the system’s pronunciation model. Instead of providing a fixed dictionary or even expanding the pronunciation dictionary with multiple pronunciations of each word, the generated phonetic transcription provides only one possible pronunciation for each word according to its context.

- Beside the problem of pronunciation variation caused by the boundary effect, developing an Arabic ASR system also faces the problem of complex morphology. The richness of Arabic morphology results in having a huge number of vocabulary items which requires training on large training sets. Our hypothesis is that this problem can be tackled using an underspecified phonetic representation. This can be done by blurring the difference
between words with similar, though non-identical, phonetic forms. For instance, to overcome the problem of having too many different forms as in (kutub) kutub, (kataba) kutaba, (kutiba) kutiba, (kut~iba) kut~iba, (kut~aba) kut~aba, etc.), where some arise from several different words and others from different forms of the same word, the recogniser will be allowed to respond to a speaker who produces any of these forms by saying that it has recognised the form (k?t?b). This makes the task of the recogniser considerably easier because there are fewer distinctions to make, and thus less chance of making a mistake. What is more, it requires a smaller training set, since any instance from any of the underlying forms will count as a training instance for the underspecified version. The output of the recogniser will thus be extremely similar to a non-diacriticised written text which can be used in many ASR applications, such as dictation and subtitling. The second objective is to investigate whether using the underspecified output to understand the text improves the performance of the recogniser.

• The third objective is to test another way of tackling the problem of complex morphology. Considering the fact that the accuracy of the speech recogniser depends greatly on the accuracy of the HMMs which link the acoustic features to the phonemes, we will work on finding the best HMM unit for training and testing the speech recogniser.

• Training on an undifferentiated population of speakers, each of whom carries distinctive characteristics, can have a considerable impact on the ASR system’s performance due to the significant variation found among speakers. This motivates us to automatically define speech communities firstly by considering the predefined categories such as gender, age, and dialectal region in order to sort out our speakers. The results of grouping speakers according to their predefined categories will then be used as seeds to grow new speakers’ communities. The effectiveness of training on speakers with similarities will be tested and evaluated by running different Arabic ASR
In addition to the objectives previously mentioned, the research will also seek to address the influence of the training size on the recognition performance.

This work will fit discretely into the main project and will use the Hidden Markov Model Toolkit (HTK) as an externally supplied speech recognition toolkit for implementing the signal processing and basic machine learning activities. The rules provided for describing the allophonic variation will be incorporated within Professor Ramsay’s PARASITE\footnote{PARASITE is an HPSG-based linguistic engine which has been designed to be particularly well-suited for languages with free phrase-order.} as the general engine for linguistic analysis.

1.3 Structure of the thesis

After presenting the main lines of the proposed project in this chapter, this section gives a brief overview of the remaining chapters’ contents.

Chapter 2: This chapter gives an overview of what speech recognition systems are about and of the various types of speech recognisers discussed in the literature. It then discusses the main challenges that have hindered the development of a high performance ASR system, highlighting the suggested solutions found in the literature to tackle each challenge. The chapter will also illustrate the recognition process for determining how recognisers actually work. Finally, the chapter briefly introduces the available open-source speech recognition toolkits.

Chapter 3: This chapter investigates the issue of developing an Arabic ASU system by looking at the Arabic linguistic features that make the language hard to process. The main Arabic-related problems will be investigated, beginning with a layout of the properties of the Arabic language, and then introducing the attempts found in the literature to overcome these challenges. The chapter concludes by reviewing a number of interesting studies carried out in the field of developing Arabic ASR systems.

Chapter 4: This chapter overviews the experimental dimensions of the developed ASR systems, mainly by introducing the tools and algorithms used.
The chapter starts by describing the main component of the speech recogniser, i.e. HMMs. The main concept of the HMM algorithm will be defined, pointing out the strengths and drawbacks in applying it to build speech recognisers. The chapter then introduces the typical processing steps in designing an ASR system using the HTK, which is the toolkit used throughout the research, defining the set of tools and library modules used in each step. Afterwards, the chapter identifies the source files required for running the toolkit in order to conduct standard experiments. This is followed by explaining the amendments to the HTK processing stages in order to use the generated phonetic transcription instead of the predefined dictionary. All the steps that involve using the predefined dictionary during the training phase have been replaced or modified in order to achieve the main objective of the research which is to test the effectiveness of using the context-sensitive phonetic transcription in training the ASR system. Finally, the chapter gives an overview of the text and speech corpora collected for the aim of conducting the experiments carried out in this thesis.

Chapter 5: This chapter describes the process of developing a comprehensive system for grapheme-to-allophone conversion of the MSA which will be used to enhance the performance of the ASR system. The first section in this chapter describes the initial stages required before applying the phonological rules. The second section, which constitutes the principal part, investigates the process of creating a substantial set of rules which can predict within-word and cross-word pronunciation variations in MSA speech. The developed system is tested and evaluated in the third section to ensure its robustness and to verify that it captures the actual phonological processes. The fourth section introduces a survey of similar works found in the literature, and then concludes by highlighting the main advantages of our hypothesised method over the other presented solutions.

Chapter 6: In this chapter, we investigate certain factors that affect the robustness of the Arabic ASR system. This is partly done by exploring some properties of phonological features that can influence the way the recogniser performs. We start by evaluating the use of the generated phonetic transcription which is sensitive to the local context and then comparing it with the use of both a fixed and multi-pronunciation dictionary. Then, we
investigate the impact of using different units in building HMMs, phoneme and word-level HMMs. Within the phonological investigations, we attempt to measure the effectiveness of using an underspecified phonetic representation by blurring the diacritical marks to improve the system's performance. The chapter also seeks to investigate some interesting technical issues that can affect the performance of the ASR system. For instance, we investigate the use of the predefined classes such as gender and accent to categorise the speakers and build a training model that is specific to each category. Understanding the outcomes of these experiments will help group speakers according to their best-matching speech communities. Furthermore, we examine how increasing the training data size influences the recognition system performance. In addressing each question, we review the problem and its suggested solutions in the literature; we lay out the experimental dimensions, followed by reporting and analysing the results, and finally a conclusion with the main findings will be given at the end of each section.

Chapter 7: The chapter provides our conclusion in terms of the research outcomes as well as suggestions for future work.
Chapter 2

Automatic speech recognition: an overview

ASR is the process of converting an acoustic waveform to a string of words using computational methods. In other words, it is a technology that allows a computer to detect the words that a person speaks through a microphone. This field of science has its roots in the idea of human-machine interaction where speech is used in a wide range of applications as the most natural way of communication. By using this kind of interaction, the computer is not expected to understand the uttered words as the understanding of the intended meaning of speech is regarded as a separate field called ASU. In ASR, the computer is expected to transcribe the acoustic waveforms into symbols. As we shall see, there is a distinction to be made between ASR and ASU. The major processes in ASU depend on ASR which has received the greatest attention in the literature. Therefore, we will concentrate on ASR in this review, although the research described in the remainder of the thesis is concerned with ASU and we shall point out any areas of divergence whenever applicable. The ultimate goal of any research in this field is to design machines that are indistinguishable from humans in their ability to communicate.

This technology has made life much easier after having been employed in several applications, such as hands-free operation and control (as in cars and airplanes) and interactive voice response [AEAG+08]. Moreover, it can also be used to help people with physical limitations in dealing with technology, let alone being an ideal way for language teaching and testing e.g. pronunciation support systems. Dictation and translation systems are another set of advanced applications for this technology, such as broadcast news transcription and translation.
from one language to another [GME11].

Implementing a speech interface in the aforementioned applications brings with it certain advantages. For instance, speech input is easier to perform as it requires no special training. In addition, inputting information via speech is much faster than handwriting and typewriting. Speech is also flexible as it leaves the hands and eyes free, not to mention the fact that speech interfaces are ideal in cases where only telephones are available.

2.1 Types of ASR systems

Speech recognition systems vary in terms of applicability and robustness. They can thus be classified according to a set of dimensions that is responsible for such a variety of systems.

Isolated word vs. Continuous speech

This category is concerned with the style of speech that the system can recognise. Isolated word speech recognition systems are trained to recognise single uttered words, whereas continuous speech recognition systems are able to identify continuous speech as in every day conversations. Continuous speech recognition systems are, not surprisingly, harder to construct as the sequential uttered words need to be segmented first of all. The goals of these continuous word systems also vary greatly in terms of the degree of difficulty. For instance, human to machine speech is much easier to recognise than human to human speech. This is due basically to the fact that speakers are normally aware that the other partner in this conversation is a machine, so they would simplify their speech by talking more slowly and clearly [JMK09].

Speaker-dependent vs. Speaker-independent systems

A speaker-dependent system requires user training and can only recognise speech by a single speaker. An example of this system is the one used in Windows XP where the user is asked to go through a training phase before using the system. Speaker-dependent systems are more accurate, but the training is not feasible for many applications such as telephone-based services. This kind of system must deal with anyone who makes contact via telephone, and must allow the user not to undergo training before using the service. In such cases, using a speaker-independent system is essential. Such a system is able to recognise speech from people who have not previously been identified by the system.
Small vs. Large vocabulary systems

This category is concerned with the size of the system's vocabulary, from small vocabulary to large vocabulary. Small vocabulary systems typically have less than 100 words. Obviously, the speech recognition task would be easier and the result would be more accurate if the number of words needed to be recognised is small. The digits recognition engine is an example of a small vocabulary system, with an approximately eleven-word vocabulary. Large vocabulary systems, on the other hand, have roughly 20,000 to 60,000 words [JMK09].

Different recognition performances may be attained depending on the type of speech recogniser being proposed. For instance, isolated-words, small-vocabulary recognition systems, like the ones used in digits or alphabet recognition, can achieve a high performance level. In contrast, increasing the complexity by adopting a large-vocabulary or conversational speech, the accuracy of the speaker-independent system will drop significantly.

2.2 Challenges in developing ASR systems

There have been several research studies aimed at delivering a robust and high-recognition accuracy system (close to human-like performance) in order to enhance human interaction with machines. Studies in ASR have achieved a significant progress over the past three decades. This progress is a result of far more advanced computing power in terms of speed and storage capability, development of successful algorithms, and access to large quantities of data [DH04]. In spite of this progress, the performance of the speech recogniser is still inferior to human recognition. This is due mainly to the existence of many challenges that have hindered the effort by speech recognisers. These challenges need to be addressed first in some way if progress is to be made in any future studies as many challenging problems remain unresolved. Below is a brief description of the main challenges.

2.2.1 Human understanding of speech compared to ASR

One of the major challenges is concerned with the difference between human-human interaction and human-computer interaction. The key distinction may have to do with the nature of human communication; a human utterance is usually influenced by the speaker's emotional state and the type of the utterance
(statement, question, or command). These emotions are conveyed by prosody (e.g. by using stress and intonation). The emotive aspects of prosody are not encoded by grammar or by the choice of vocabulary, so despite being potent in human-human interaction, it may be disruptive in human-computer interaction [Shn00].

Humans, on the other hand, use their knowledge of the speaker, the subject and language system to predict the intended meaning while in ASR one only has speech signals [For03]. Therefore, paying attention to the differences between human-human interaction and human-computer interaction will help researchers choose the suitable application for human-computer communication.

2.2.2 Speaker variability

It is clear from the previous description of the speech recognition types that speaker-dependent systems achieve better recognition performance when compared with speaker-independent systems. Research shows that the performance of speaker-independent recognition systems are generally 2-3 times worse compared to speaker-dependent ones in terms of WER [HCC01]. That is mainly because HMM-based speech recognition systems (for more details on HMMs refer to 2.3.2.1) are very sensitive to the variability in the speech signals [CJ08]. The speech signals can vary even when the same word is pronounced by a speaker and then the same speaker tries to utter the same word again in the same manner. This variability in the acoustic waves will negatively influence the performance of the speech recognition system. In addition to the variability that arises within the speaker-dependent system (intra-speaker variability), the speech recognition system can also be influenced by the variability resulting from having different speakers (inter-speaker variability). Inter-speaker variations are often large and difficult to account for as it is commonly believed that the uniqueness of a speaker’s voice results from a complex combination of physiological and cultural aspects [BDMD+06b]. The acoustic variations found among speakers can be sorted in two categories:

i. The physiological characteristics associated with the speaker gender, age, voice characteristics, etc.

ii. The psycho-sociological characteristics, e.g. emotional state, speaking style, regional accent, and education, etc.
Inter-speaker variabilities have motivated many researchers to develop machines that can identify a person from a spoken phrase depending on the variability caused by each speaker's characteristics (speaker recognition or identification) [CJ97]. The main techniques proposed for modelling the variation resulted from the speakers are speaker normalisation and adaptation which are used to improve speaker independent automatic speech recognition system.

The following is a description of these variations, what impact they have on speech recognition performance, and how they can be overcome:

2.2.2.1 Gender

Researchers have argued that the two principal components that influence the performance of a speech recognition system correspond to the gender and accent [HCL+01]. The gender variation effect is related to anatomical differences which give male and female distinctive speech characteristics. This prime factor is caused mainly by the vocal fold characteristics and the vocal tract size. Male speakers have relatively longer and thicker vocal folds than female speakers do, which results in low pitch speech. The size of the vocal tract is another factor that contributes to the gender effect. Female speakers, who have shorter vocal tract compared to male, have higher formant frequencies [CW91]. The gender effect on the speech characteristics was demonstrated by the results of a study conducted on the TIMIT American English database. TIMIT database consists of 3,696 utterances from 462 American-English speakers (326 male and 136 female covering 8 major dialects of American-English [PWtB96]). The gender effect is measured by the distribution of the speaker's fundamental frequency in addition to the phonemes' average power spectrum. The bar chart shown in Figure 2.1 displays the results of the study where the X-axis is \( F_0 \) and the Y-axis is the occurrence frequency. The mean value is 117 and 199 for male and female speakers respectively, whilst the standard deviation is 16.0 Hz for male speakers and 20.7 Hz for female speakers [Ho01].

The differences between male and female speech is not only caused by the physiological differences. Speech habits that can vary according to gender also make a difference. [Byr94] summarises the findings regarding the gender effect in the TIMIT database in the following points:

- Males speak faster than females.
Male speakers release sentence-final stops less often than female speakers do.

- Males use schwa more than females.
- Males produce glottal stops more often than females.
- Males produce the syllabic [n] more than females.
- Males tend to produce more [h]'s than females.

2.2.2.2 Variability in language results from accents, dialects, and foreign accented language

The language used in the recognition process may have different features according to the people who speak it:

Regional accents  As mentioned earlier, accent is one of the fundamental components of inter-speaker variability with a direct impact on the recognition performance. This variation is caused by the different ways in which a group of people from the same language community pronounce words in a manner influenced by the geographical area each speaker comes from. The accent variability is a serious hurdle for the recognition process as researchers suggest the error rate to increase by 30% if the speaker's accent is not covered in the training corpus [HCC01].

Dialect  Dialect refers to the difference which is not only in the pronunciation but also in the use of vocabulary and grammar as well [Cry03]. In many cases
the gap between the dialects is so huge that researchers in ASR have found themselves forced to consider the dialect as another language [For03]. This issue is particularly complicated for Arabic. MSA is supposed to be the standard use in formal settings, nonetheless, because Arabic is spoken in a wide range of countries, there are major dialectal variations even within MSA. This issue will be discussed in more detail in Chapter 3.

Foreign accented language  It is widely believed that ASR systems perform poorly when confronted with foreign accented language [Mor04] [BFI07]. These systems are not designed to identify non-native speech and the databases used in the training do not contain foreign accents. This is generally explained by the fact that non-native speakers have different approaches in adapting to the new language, like omitting some unfamiliar phonemes or replacing them with the closest phoneme in their mother language. This omission/replacing behaviour cannot be handled by the usual native language modelling [BDMD+06a]. As a consequence, the recognition performance drops significantly with non-native speakers due to the wide variation in speech signals. However, the acoustic confusability varies depending on the nature of the language and on the level of proficiency of the speaker. Many researchers have proposed solutions to limit this variability, basically by introducing multiple phonetic transcriptions in the lexical model that deal with alterations produced by non-native speakers using techniques such as Artificial Neural Networks (ANNs) [CKAJ05]. Other studies suggested using phonetic confusion rules extracted from a non-native speech database to modify the acoustic models (HMMs). Implementation of this technique by [BFIH06] [BFI07] achieved noticeable improvements in recognition accuracy. Another study shows that by involving non-native speakers in both training and testing stages one can get a better recognition performance. Furthermore, it suggested concentrating on an analysis of the major problematic issues in the phonetic system of the targeted language in order to define the common errors and work on recovering them [ASO08].

2.2.2.3 Age

Many studies have found a strong correlation between age and recognition performance. The accuracy rate in children’s speech when using speech recognition system trained with adult’s speech, for instance, is too low to be practically useful
Elenius and Blomberg demonstrated this fact by conducting some experiments trained and tested among adults and children. They confirmed in their study that the speech recogniser worked very well when trained and tested by adults with a 97% accuracy. This rate dropped significantly when tested by children with only a 51% accuracy rate [EB04]. This drop in accuracy rate resulted from the fact that the acoustic and linguistic features of children’s speech are extremely different from those of adults [GG03]. Adults, for instance, have a longer vocal tract and heavier vocal folds compared to children. This physiological difference results in a non-linear rise in the formant frequencies [BDMD+06b]. Furthermore, children have their own language system with a different vocabulary, grammar, and pronunciation.

However, the recognition performance for children improves when the system is trained on the correct class of speakers where it rises to 87%, although still lower than the adults’ performance [EB04]. This is mainly because children vary quite considerably in their speaking abilities. Moreover, children’s speech characteristics change rapidly as they grow up due to physiological changes occurring during a child’s growth [GG03].

Several approaches have been proposed in order to develop a speech recogniser that is suitable for children. Those approaches include vocal tract length normalisation (VTLN) [DNP98], training with age-dependent speech data [WJ96] [EB04], and the use of customised dictionaries [LR02].

2.2.2.4 Emotional state

The emotional state of the speaker is another component of variation in the speech. Similar to the previously mentioned variation components, emotional state is found to have a considerable impact on the speech spectrum [BDMD+06b]. As pointed out by [Han96] speakers affected by their emotional states certainly do not speak like people with neutral emotional states.

In attempting to overcome the variability in the speech signals caused by the emotional states and to achieve a better recognition accuracy, many techniques have been proposed. An interesting training procedure called multi-style training has been investigated by Lippmann et al. [LMP87]. This technique focuses on improving the recognition performance when the speaker is under emotional pressure and the system cannot be trained in these conditions. Speakers in this system are asked to produce different talking styles (5 normally spoken tokens)
with multi-style training (one token each time from normal, clear, fast, loud, and question-pitch talking style). With the multi-style training technique, the error rate decreased by 50% compared to 25% under conditions sampled during training [LMP87].

After exploring the speaker variability aspects, it is clear that they can have considerable impact on the speech recognition performance. This impact is considered to be one of the most fundamental problems of ASR. However, many researchers have recently been more concerned with developing new approaches to limit the large variability in the acoustic realisation among different speakers. The smaller the variability on the speech signals, the better is the recognition performance being achieved. One way of handling variability is to use a multiple modelling approach instead of using a single model to cover all the variability values. This can be done by building several models, each covering only a subset of variances, such as gender-dependent model or accent-dependent model. The next step is to use the corresponding model selection scheme for decoding the utterance [CJ08] [HCC01]. Another method to deal with the speaker variability problem is to adapt the speech recognition system to the user by re-training the system using the appropriate corpora [HCC01]. The motivation behind these methods is to improve the performance of “large vocabulary continuous speech recognition” (LVCSR) system while minimising the amount of training data needed to cover for speaker variability.

Several adaptation techniques have been presented, two of them have been widely used: Maximum A Posteriori (MAP) algorithms [GL94] and Maximum Likelihood Linear Regression (MLLR) algorithms [LW95].

MAP algorithm is a popular and reliable method for the estimation of HMMs [SVL+06]. The basic idea behind this method is to enable prior knowledge about HMM parameters to be integrated into parameter estimation, hence reducing the data needed to attain a good speaker-dependent acoustic model [LH05].

MLLR is another well-known adaptation model. In MLLR, the speaker specific information is represented by applying linear regression transformation to the model parameters [Ham]. The advantage of this method over MAP is that the speaker adaptation performs better than using a relatively small quantity of speaker specific enrolment data [SBJ08] [LH05].
2.2.3 Language complexity

In ASR systems, we take the spoken language as input; the input speech will undergo the recognition process to produce the intended written form. Deriving the written form from the speech is quite challenging as natural languages tend to exhibit all kinds of complexity. We identify below some aspects of this language complexity.

2.2.3.1 Non-verbal communication

Speech is not the only source of communication. Humans also use non-verbal language to deliver their message, e.g. body language, posture, gestures, facial expressions or eye movements [KH09]. This kind of language is completely neglected by ASR systems. This problem has been addressed by researchers working within the area of multimodality. The aim of multimodal approaches is to develop emotionally intelligent systems which can recognise the emotions that integrate information from body language and speech [CKC07]. Incorporating the non-verbal language is vital for computers to successfully interact with humans.

2.2.3.2 Out of coverage utterances

One major obstacle against achieving a high performance ASR engine is the presence of out-of-coverage utterances. That is, the words or phrases that are not covered or not allowable in the rules of the system’s grammar, or not incorporated within the system’s lexicon.

Ungrammaticality Grammar is one of the main components of a typical ASR system. In order to avoid misrecognition by the system, speakers should obey the system’s grammar rules. However, it is certain that the application will receive a number of ill-formed sentences. This is mainly because speakers do make grammatical mistakes while they speak. Although native speakers tend not to frequently make grammatical mistakes, non-native speakers normally produce ungrammatical sentences while speaking such as errors of agreement (sub-verb, number, etc.) or using constituents in unusual orders [KS79] [War89]. This violation of the constraints used by speech recognisers may result in a complete recognition failure for an utterance.
**Extra-grammaticality**  The ASR system’s grammar is supposed to specify the linguistic structure of the words and phrases expressed as a set of rules. It is very common that these sets of grammar may not be comprehensive due to the variety of grammatical utterances. In such cases, the utterance will be misrecognised by the system as it is not covered within the allowable words and phrases generated by the rules of grammar. One source of extra-grammaticality comes from the primary difference between spontaneous speech and read speech. Spontaneous speech, for instance, may be influenced by some speech disfluencies like using non-lexical vocabulary including “um, ah, uh, etc.” to fill in for pauses or background events such as coughing. Speakers also tend to cut off sentences in mid-utterances then repeat the repaired utterance (false start) as for example: “The best thing about this house is, well, the best thing about this house is the location”. This sort of utterance is considered to be a big hurdle for speech recognisers. Another source of extra-grammaticality is the evolving grammar of natural language over time, as there are always new words or grammatical constructions that come into use [Beu07]. An utterance may be acceptable from the grammatical standpoint, but could well be beyond the syntactic coverage of most systems.

Extra-grammaticality is thus defined on the basis of the capabilities of a system’s grammar model [LKS95] [CH84]. So, utterances may be rejected by the speech recogniser not because they are ungrammatical but because these kinds of utterances are not readily included in the current grammar model.

Many researchers have investigated the ungrammatical and extra-grammatical phenomena and suggested several techniques for making the systems capable of dealing with a wide range of ungrammatical and extra-grammatical inputs. For example, Hipp proposed a new algorithm for minimum-distance parsing to be used for decoding the structure of the user’s utterance [Hip92]. This algorithm is based on letting the grammar rules be modified by insertions, deletion and substitution. Another approach presented by Worm and Rupp aims at treating these problems after parsing by selecting the right sequence of partial analyses [WR98]. In a more recent study, Ailomaa et al. presented two techniques which mainly concentrate on dealing with extra-grammaticality in NLP [AKCR05]. The first approach is based on selecting the most probable optimal maximum coverage with partial analyses, while the second is focused on extending the preliminary grammar by adding relaxation rules to the grammar in a controlled manner.
Out-of-vocabulary (OOV) Recognising the OOV words is one of the key problems for ASR. These words cannot be recognised by speech recognisers as they are not contained within the system’s language model. In most cases, speech recognition systems are incapable of returning the uncovered word; alternatively, these words may be mistaken for other words in the system.

This problem is prominent in many speech recognition applications especially with large vocabulary systems. To overcome this problem, several approaches have been introduced to work within the ASR system. Bazzi and Glass [BG00] have presented an approach for detecting the OOV words problem. Their study is based on supporting the recognisers with an OOV word model so that the language model in the recogniser will be able to predict the OOV word. This study has later been extended by adding several OOV model, one for each word class [BG02]. In a similar way, Schaaf [Sch01] has described an approach to handle this problem by using a language model based on semantic categories besides the new type of generalised word models with a mixture of specific and general acoustic units. He showed that by using this approach, the expected number of words errors following an OOV word can be reduced considerably by 37%. Other researchers have proposed an approach to deal with the OOV words. The proposal used a hybrid language model combining lexical and sub-lexical units [YS04] [BN05]. The research showed that using this approach will enhance the OOV word detection by over 10% [YS04].

2.2.3.3 Language ambiguity

The complexity of the human language is perhaps the most significant challenge for speech recognition. Languages have, for instance, various aspects of ambiguity. That is, of course, a problem for any computer-related language application. Many kinds of ambiguities arise within the ASR such as syntactic ambiguity, lexical ambiguity, pronunciation ambiguity, the ambiguity resulting from a lack of punctuation or from employing intonation within the speech. These ambiguity aspects will now be introduced briefly.

Syntactic ambiguity Syntactic ambiguity is considered to be one of the major sources of ambiguity in the natural language. The weak representation of syntactical knowledge results in part from the presence of sentences that can be structurally analysed in more than one way. A classical example is this:
“I saw the man with the telescope”

The prepositional phrase “with the telescope” can be part of the object noun phrase “the man”. In that case, the meaning would be I saw the man who had a telescope. Alternatively, the prepositional phrase could be attached to the VP and the meaning would then be: I saw the man by using a telescope. To overcome this problem, semantic constraints need to be applied into the speech recognition system in order to improve its accuracy. The structural ambiguity does not matter for ASR since all interpretations of a given sentence are written in the same way. It does, however, matter for ASU which aims at understanding the meaning of the sentence as well as identifying the written form.

**Lexical ambiguity** It is well known that context has a direct role in appropriating meanings to a given word. Words have different meanings in different context, e.g. “John lies” could have two valid interpretations, either as John being a liar, or that John is lying on a sofa. Words that are pronounced the same but with different meanings are called “homophones”. The context, in such cases, is the only source for deciding the intended meaning. One way to eliminate the negative influence of lexical ambiguity is to add semantic and pragmatic knowledge into speech recognisers. A suggested dis-ambiguation technique is to introduce an interface between ASR and the linguistics engine at the textual level. This can be addressed by using underspecified common forms for the textual representation of homophones. Obviously, the lexical ambiguity problem should not affect the work of the ASR engine which aims at transcribing natural speech. However, developing an ASU system requires consideration of the lexical ambiguity problem in order to deliver the proper meaning of the transcription.

**Word-boundary ambiguity** Finding the boundaries between words is a challenging process in the continuous speech recognition system since natural speech normally does not have pauses between words. The following example illustrate this difficulty:

*Go to the Topps tiles* (Topps tiles is a name of a shop)

*Go to the top styles*

Reliably detecting word boundaries in continuous speech is crucial for delivering effective speech recognition systems. Researchers have suggested employing acoustic information such as energy [JMR94], pitch variations, intensity, and
duration [RRS96] [LCCH98] [AJPA10] to hypothesise word boundaries in continuous speech. Despite the fact that acoustic information carries useful clues for assigning word boundaries, the improvements are still limited. The limitation of this method can be observed in case the beginning of the word is co-articulated with the end of the previous word. As a way to cover this limitation, [TGGN09] focused on capturing lexical information in addition to the acoustic information to obtain more accurate word boundary estimate. The proposed method showed a considerable improvement in word boundary detection.

**Pronunciation ambiguity** Many studies confirm that a major cause of deterioration in the ASR system is the mismatch between the recognised phonetic sequence and the word’s representation in the lexicon. For instance, [FLM99] show that general-purpose lexica do not cover the pronunciation of spontaneous speech efficiently. Only 33% of the pronunciation variants found in the hand-written part of Switchboard ¹ were introduced in the Pronlex dictionary [KSM97]. [MGSN98] reported that when the lexicon pronunciation matched the acoustic pronunciation, Switchboard word error rate decreased from 40% to 8%.

Speech recognition systems need to handle a wide range of pronunciation variation to make the performance more robust. Variation in the pronunciation may result from different sources such as speaker’s dialect, non-native mother tongue, speaking style, etc..

The type of pronunciation variation we are concerned about here is the one that results from linguistic variation at the segmental level. The issue here is that in continuous speech different sorts of interactions may occur which require applying various phonological processes such as assimilation, co-articulation, deletion, insertion, and voice reduction. Such processes may take place either within the word or across the word boundary. For example, the phrase “have to” may be pronounced as /hafta/ in fast conversational speech.

Considering the difference in pronunciation between continuous speech and the associated written form can help avoid errors and frustration caused by the difference between the input and the source form. Typical lexica are not capable of capturing every kind of pronunciation variation especially cross-word variations as they may occur at the junctures of the words [SC99].

¹Switchboard is a large multispeaker corpus of conversational speech and text that contains about 2500 conversations by 500 speakers from around the U.S [GHM92].
Modelling the pronunciation variation may take place at the acoustic modelling phase or at the pronunciation modelling phase (for more about acoustic modelling and pronunciation modelling refer to 2.3.2.1 and 2.3.2.2). Building a more robust acoustic model can help in much of the variation seen in speech by using the advantages of triphone models to capture the context effect on pronunciation [JWB+01] [DWA+02]. This technique is powerful in capturing the pronunciation variation, especially for resource poor languages. However, it can only model the pronunciations that are observed sufficiently in the training phase. Furthermore, triphone modelling is not able to handle all kinds of pronunciation variation such as syllable deletion [JWB+01].

Modelling the pronunciation during the pronunciation modelling phase can be done either by adding multiple pronunciations to the lexicon as in [AEAG+08] or by using pronunciation rules to control the variation as in [BHH09]. Expanding the phonetic lexicon and adding multiple possible pronunciations per word is a widely used technique in speech recognition systems [AFL03]. However, augmenting the pronunciation lexicon with all the observed variants can cause extra confusability due to rare pronunciations [JMK09] [SNK00]. [Hai05] confirms that using single pronunciation dictionaries can perform equal or better than using dictionaries with multiple pronunciation.

Employing pronunciation rules to find the closest pronunciation to the natural speech can be delivered from using predefined language-dependent phonological rules or by generating new phonological rules according to the database [LG04]. A detailed investigation of this approach can be found in Chapter 5.

Supplying speech recognisers with the “actual” rather than the “canonical” pronunciation yields a better recognition performance as this research will prove.

**Employing intonation and speech timbre** Intonation and speech timbre can change the meaning of a word or sentence entirely, e.g. *Go!* and *Go?* or *Go.* The meaning of these expressions can clearly be recognised by humans but for a computer it is very challenging to recognise the differences and therefore to assign the proper punctuation to the textual representation of the prosody.

### 2.2.4 Environment variation

Besides the impact of the speaker and the language variability of the speech process, the robustness of the speech recogniser is greatly influenced by its ability
to cope with the presence of background noise. That is, the speech recognition system needs to be able to handle not just the speech signals as the speech is uttered in an environment of sounds. These unwanted sounds are called “noise”. There are different kinds of noises depending on the cause. It may result from environmental sounds like the disruption when speaking in an airport lobby filled with people talking, loud calls and advertisements, a radio playing somewhere, or a child crying. Another kind of noise results from the devices being used as talking from a cellular phone or using a cheap microphone which can cause a humming.

Building a robust system in noisy acoustic environments is very challenging as this unwanted information needs to be filtered out from the speech signals in order to avoid the mismatch between training and the input speech. The problem of building a noise robust speech recognition system has been addressed by many researchers over the past several years. These researchers have generally focused on the minimisation of the mismatch. [PH+00] has categorised these researches into three groups:

- Robust speech signal parameterisation: the speech signal is presented using minimally affected parameters to reduce the level of additive noise in the input speech signal [KK07].

- Compensation techniques: the additive noise is removed from the signal by adapting the parameters representing the noisy speech to a reference clean representation [DLTFH01] [LG08].

- Adaptation of the models to noise conditions: the models in the recogniser are trained to handle different kinds of noises [Aur02].

Most of the research carried out in this area are placed within a common evaluation framework called Aurora [DH04]. The main task of Aurora is to develop algorithms for distributed speech recognition (DSR) system in mobile environments [DWA+02] [Aur02].

2.3 Architecture of ASR

The main goal of a speech recognition system is to take a speech waveform as an input and produce a string of words as an output. The basic structure of a typical
ASR system is shown in Figure 2.2. As illustrated in the figure, various types of knowledge are required for the speech recognition system so as to be able to associate the input speech with the text. Briefly, the process starts by extracting the acoustic feature vectors from the speech signals, and then finding the set of phonemes that best matches the acoustic input using a combination of three kinds of probabilistic models, namely: acoustic model, pronunciation model, and language model. These models will acquire a complete estimate of the sentence with the highest probability.

This section will discuss the main components of the ASR system, providing an overview of its associated algorithms.

2.3.1 The Front-End
This initial step is concerned with converting the sound input into a form of data that the system can recognise (called “feature vectors”). This has a crucial
role in the recognition performance since better signal feature extraction leads to better recognition performance. Therefore, many feature extraction methods have been proposed for speech recognition, with the Mel-frequency cepstral coefficients (MFCCs) by far being the most common one. This method is based on the idea of cepstrum which will be described later. However, this process should be preceded by converting the analog signal to a digital one, which is done through sampling and quantisation. It is worth remembering here that speech recognisers will work better if the acoustic model was trained at the same sampling rate/bits as the input speech. After acquiring the digitised waveform, it is time to extract a sequence of N-dimensional MFCC feature vectors (the majority of current systems use 39 dimensions). Figure 5.1 shows the seven steps in this process as described in [HCC01].

![Image of the front-end process](image)

**Figure 2.3: The front-end process.**

Basically, the process starts by boosting the amount of energy in the high frequencies in order to improve the phone detection accuracy. The next step is classifying the provided spectral features in a small window that characterises a particular subphone. This windowed signal needs then to go to a tool called a Discrete Fourier Transform (DFT) in order to determine the amount of energy in the signal at different frequency bands. The frequencies output by the DFT will then be wrapped onto the mel scale by creating a bank of filters that collect
energy from each frequency band. In addition, the log is used afterwards to make the feature estimate less sensitive to variations in the input speech, as in the power variation caused by the distance change between the speaker’s mouth and the microphone. The next step is to extract the cepstrum with inverse DFT. The cepstrum is the only way to separate the glottal source, which is not that important in phone detection and for the filter. The filter is the exact position of the vocal tract and since knowing the shape of the vocal tract will help in recognising the produced phone, filters are more useful for phone detection. This information about the vocal tract filter will be represented in 12 cepstral coefficients for each frame. The last step is to add energy (the 13th feature), 12 delta cepstral coefficient (plus 1 energy coefficient), and 12 double delta cepstral coefficient (plus 1 energy coefficient) which are used to control the change in cepstral features over time. After summing up these features, we will end up with 39 MFCC features.

2.3.2 Probabilistic models

Three probabilistic models are required by the speech recognition system in order to estimate the sentence with the highest probability, namely: the acoustic model, the pronunciation model, and the language model. Figure 2.4 shows a rough block diagram of how the decoder computes the prior probability and the observation likelihood.

![Figure 2.4: Probabilistic models needed for decoding a sentence.](image-url)
The main practical challenge in developing speech recognition systems is to build accurate probabilistic models that can correctly reflect the input speech [HAH+01].

After having a sequence of feature vectors, e.g. MFCC vectors, the acoustic model works to create statistical representations to estimate the probability of these acoustic features, given the word or phonetic model \( P(X|O) \) in Figure 2.4. It is apparent from the early work done in ASR that using the acoustic model for recognition alone is not sufficient. For this reason, the pronunciation model and the language model are vital for refining and restricting the search process. The pronunciation model which is introduced in the lexicon serves as an intermediate link between the acoustic and language models by providing the words or sub-words of the targeted language with their pronunciation to link between sub-word HMMs and language model \( P(W|X) \) in Figure 2.4. The language model contains the syntactic information of the input speech presented in the grammar. The task of the grammar is to detect the connections between the words in a given sentence to guide the recognizer for the correct word sequence \( P(W) \) in Figure 2.4, so it has fewer chances to make mistakes.

Given these types of knowledge, the decoder performs the core recognition by estimating the set of phonemes with the highest probability [LH05] [JMK09].

2.3.2.1 Acoustic modelling

As mentioned earlier, the acoustic model is one of the main components of the ASR system which has been used alone for recognising the speech in the early speech recognition attempts. The task of this model is to transfer the acoustic signals to a set of observations. For a given acoustic observation, which are normally feature vectors extracted from the speech signal, the model works on finding the word sequence that has the maximum posterior probability in the light of these observations [Sii07].

Acoustic modelling has a crucial role in improving the performance as it is considered to be the core process in any speech recognition system.

Different approaches can be used for modelling the variations of speech signals such as: HMMs, Dynamic Time Warping (DTW), Neural Networks (NNs), expert systems, as well as a combination of different algorithms. However, the HMM is the most widely used algorithm for acoustic modelling.
HMMs as an acoustic model  The HMM is a commonly used algorithm in speech and language processing. It is generally considered to be the most significant advance in speech recognition technology. The implementation of HMM techniques has greatly improved ASR performance. It was introduced in the 1970s and gradually became the dominant approach to acoustic modelling [LH05]. The explanation in this section is based on [JMK09] and on the well-known publications by Rabiner [Rab89] [RJ93].

HMM-based technique is basically a statistical method for characterising the spectral features of the speech. The underlying hypothesis of the HMMs is that the acoustic signals can be well modelled as a parametric random process and that these parameters can be predicted in a precise manner.

The essential component of a Markov model is a set of states and a set of transitions between states. These states correspond to the positions of the vocal system of the speaker. As the name suggested, these states are “hidden”, in other words, not observable. The challenging task here is to determine the hidden parameters from the observable ones, in other words, to define the probabilities of the actual state sequence given the observed speech signals.

Figure 2.5 shows how HMMs represent the speech phonemes. A simple left-to-right 3 state model topology is presented. The model consists of:

i. Five states in total, including three emitting states and non-emitting input and output states.

ii. A set of transition probabilities between these states, each \( a_{ij} \) represent the probability of each state to go to the next state.

iii. The emission probabilities \( b_i \) (also known as observation likelihood).

Further details on applying this algorithm to ASR are given in Chapter 4.

2.3.2.2 Pronunciation modelling

The lexicon is a list of words with the pronunciation of each word presented in a phone sequence. The standard pronunciation of the words is represented in the dictionary, for example, the pronunciation of the word “hello” is presented in this way /h eh l oe/. Considering the fact that some words have different pronunciations, each of which differs significantly from the standard form, some dictionaries provide multiple pronunciations for some words. Besides the word’s
pronunciation, the lexicon may have a silence model /sil/ to help determining sentence start and sentence end tokens.

Providing the phonetic sequence for each word in the system’s vocabulary helps to estimate the probability of the phonetic models which are linked together to form a word HMM representing each word in the vocabulary.

2.3.2.3 Language modelling

The task of the language model is to assign a probability for each linguistic units (prior probability) $P(W)$. Unlike acoustic models which are based on phoneme-level speech structure, language models are based on word-level language structure. That is to say, language models contain a list of words and sentences with their probability for appearing in a given sequence. The grammar, which is the main component of the language model, defines the acceptable sentence structure based on the possible terminal symbols specified by the lexicon.
Many kinds of language models are introduced in the literature such as $n$-gram statistic models which calculate the probability of a word on the basis of the identity of that word and of the preceding $n-1$ words. If approximating the probability of a word depends on the preceding two words, the model is called a trigram model: $P(w_i|w_{i-2}, w_{i-1})$. If it depends only on the identity of the immediately preceding word, it is called a bigram language model: $P(w_i|w_{i-1})$. If the probability of the word depends on no other word it is a unigram model: $P(w_i)$.

Another type of grammar used in building language models is the context-free grammar (CFG) where the structure of the sentence is described regardless of the context of a non-terminal.

Many studies used a combination of both $n$-gram and CFG formalisms to benefit from the advantages that each model can provide and avoid some of the standard problems associated with each model [LST+92] [MR93] [WMH00]. The aforementioned studies found this method to be useful and practical in building a robust statistical grammar.

### 2.3.3 Searching and decoding

Finally, all these types of knowledge (extracted features, acoustic likelihood, word pronunciation, word predictability) are combined to solve the problem of decoding. The task of the decoder is to combine all the probability estimators to produce the string of words with the highest posterior probability therefore it is often referred to as just a search process [JMK09]. Several basic search algorithms are presented in the literature to serve as the basic foundation in developing ASR systems. Most HMM-based ASR systems use the Viterbi algorithm for decoding which will be introduced in Chapter 4.

### 2.3.4 Recognition evaluation

There is a standard metric for evaluating speech recognition performance based on the WER. This method is used to determine the extent to which the word string returned by the recogniser differs from the reference transcription. To compute the total errors of the recognisers the first step is to compute the minimum number of word Substitutions, word Insertions, and word Deletions in the output string being referred to the correct string. The following equation defines how to
compute the WER.

\[ WER = 100 \times \frac{Substitutions + Insertions + Deletions}{Totalwords} \] (2.1)

### 2.4 Open source speech recognition systems

Normally, to develop a new speech recogniser, an entire system needs to be built from scratch. Obviously, building a system from scratch is a very complicated process. Fortunately, several successful speech recognition toolkits have been developed to help build bespoke systems. Examples of such toolkits include Sphinx, HTK, and Center for Spoken Language Understanding toolkit (CSLU). These three systems are the most widely used speech recognition engines in the literature. They are all open source HMM-based speech recognition systems. The Sphinx system was developed at Carnegie Mellon University in 1988 [HAH+92] and can be found online \(^2\) [RJ93]. Sphinx is a collection of several Sphinx engines, each of which has unique characteristics and usage; Sphinx I, II, III, IV and pocket Sphinx. CSLU is another powerful open source speech recognition system available online for researchers \(^3\). The development of this system started at 1992 and it was aimed at developing a high performance speech recogniser to work in the real-world. However, maintenance of the system has been very sporadic and there is very little documentation.

The HTK is a general toolkit used mainly for designing and implementing ASR systems, though it does have research applications. It was developed at the Cambridge University Engineering Department (CUED) by the Machine Intelligence Laboratory (formerly known as the Speech Vision and Robotics Group), where it was used to build large vocabulary speech recognition systems \(^4\). It consists of a set of library modules and a set of more than 30 tools used for training and manipulating HMMs. The HTK tool was developed gradually, and there are many versions of it. In this research, the latest version of the HTK, 3.4.1, is used for conducting our experiments on Arabic. More details about the HTK are given in Chapter 4.

Generally speaking, those three tools do the same job in similar ways. The reason for opting for the HTK is because we believe it is more flexible to install

---

\(^2\)http://cmusphinx.sourceforge.net/
\(^3\)http://www.cslu.ogi.edu/toolkit
\(^4\)http://htk.eng.cam.ac.uk/
and is well-documented with plenty of community support.
Chapter 3

Modern standard Arabic from the perspective of ASR

With the increasing role of the computer in today’s society, there is an ever growing desire to communicate with computers in a more natural way. Communicating with computers naturally via speech can make life much easier as it provides easy access to information sources and services. ASR is one of the human-machine interfaces that aim at obtaining natural communication between humans and machines. ASR has witnessed extensive studies and efforts dedicated at various languages over the last two decades. However, Arabic, which comes in fourth in terms of the number of native speakers (estimated at 221 million) [GG05], has been comparatively less studied. A major reason for this lack of research is the challenges involved which make the language difficult to process. This chapter investigates the issue of developing an Arabic ASU system, beginning with a layout of the properties of the Arabic language, and identifying the various aspects that make the language difficult to process. Afterwards, we will review a number of interesting works carried out in relation to Arabic speech recognition.

3.1 Properties of Arabic

The development of a high performance natural language speech recognition system is challenging for several reasons discussed in Chapter 2. Those challenges relate to speaker variability, the nature of the human language, and the environmental noise. In addition to these general challenges, each language has its own characteristics. Arabic, for instance, poses a number of challenges somewhat
different from other languages for which speech recognition systems have been developed. These arise from a variety of sources—the gap between Modern Standard Arabic and dialectal Arabic, the predominance of non-diacriticised text material, the complex morphology which increases the perplexity and out-of-vocabulary rate, and the significant differences between the spoken and written forms. The following is a brief description of the distinctive properties of the Arabic language focusing on the ones that may affect the recognition or understanding process and pointing out what researchers have suggested so as overcome these problems.

3.1.1 Arabic forms

It is important to state here that Arabic is a collection of different forms and dialects. These forms vary significantly in just about every aspect of linguistics. The MSA is a descendant of Classical Arabic (CA) with minor changes in the essentials of syntax, morphology, and phonology but with greater variety in the lexicon [Hol04]. The best example of CA is the form of Arabic found in the holy Qur’an. MSA, on the other hand, is the language used in formal writing and speech (e.g. education, newspapers, broadcast news, etc.) and is considered to be the official language in all Arabic speaking countries.

In contrast, people generally speak in their own dialects in daily communication. These dialects are neither taught in schools nor even have any organised written form. Arabic dialects differ substantially from MSA in terms of phonology, morphology, vocabulary and syntax. This collection of dialects not only varies according to the geographical continuum (e.g. Gulf Arabic, Iraqi Arabic, Levantine Arabic, Egyptian Arabic, and North African Arabic) but also depends on some sociolinguistic variables such as the urban, rural, or the Bedouin dimension within the same region [CDH+06]. Saudi Arabia, for instance, has three major dialectal groups, namely Hijazi, Najdi, and Sharqi. These numerous dialects have a noticeable impact even when the speaker is speaking in MSA. Take the example of (rajul رجل) “man”, where the letter (j ح) could be pronounced /radʒul/ as the sound /dʒ/ in the English word jungle; /ragul/ as the sound /g/

The examples provided in this thesis are transliterated into Latin characters using the Buckwalter transliteration scheme [HSB07] along with the International Phonetic Alphabet (IPA) [Ass99] system to represent the sounds of the spoken language.
Table 3.1: Variations in digits pronunciation within MSA and five major dialects.

<table>
<thead>
<tr>
<th>Digits</th>
<th>MSA</th>
<th>Gulf dialect</th>
<th>Iraqi dialect</th>
<th>Levantine dialect</th>
<th>Egyptian dialect</th>
<th>North Africa dialect</th>
</tr>
</thead>
<tbody>
<tr>
<td>waHad</td>
<td>/waːhid/</td>
<td>/waːhid/</td>
<td>/waːhid/</td>
<td>/weːhad/</td>
<td>/weːhid/</td>
<td>/waːhid/</td>
</tr>
<tr>
<td>AinAn</td>
<td>/ʔɪnajn/</td>
<td>/ʔɪnajn/</td>
<td>/ʔɪnajn/</td>
<td>/ʔɪnajn/</td>
<td>/ʔɪnajn/</td>
<td>/ʔɪnajn/</td>
</tr>
<tr>
<td>valAvap</td>
<td>/θoːlaːθa/</td>
<td>/θoːlaːθa/</td>
<td>/θoːlaːθa/</td>
<td>/tlaːθa/</td>
<td>/tlaːθa/</td>
<td>/tlaːθa/</td>
</tr>
<tr>
<td>&gt;arbaEap</td>
<td>/ʔarbaʔa/</td>
<td>/ʔarbaʔa/</td>
<td>/ʔarbaʔa/</td>
<td>/ʔarbaʔa/</td>
<td>/ʔarbaʔa/</td>
<td>/ʔarbaʔa/</td>
</tr>
<tr>
<td>xamsap</td>
<td>/xamsa/</td>
<td>/xamsa/</td>
<td>/xamsa/</td>
<td>/xamsa/</td>
<td>/xamsa/</td>
<td>/xamsa/</td>
</tr>
<tr>
<td>sit~ap</td>
<td>/sitta/</td>
<td>/sitta/</td>
<td>/sitta/</td>
<td>/sitta/</td>
<td>/sitta/</td>
<td>/sitta/</td>
</tr>
<tr>
<td>sabEap</td>
<td>/sabʔa/</td>
<td>/sabʔa/</td>
<td>/sabʔa/</td>
<td>/sabʔa/</td>
<td>/sabʔa/</td>
<td>/sabʔa/</td>
</tr>
<tr>
<td>vamaniya</td>
<td>/ʔamanija/</td>
<td>/ʔɪmanja/</td>
<td>/ʔɪmanja/</td>
<td>/tmaniy/</td>
<td>/tmanja/</td>
<td>/tmanja/</td>
</tr>
<tr>
<td>tisEap</td>
<td>/tisʔa/</td>
<td>/tisʔa/</td>
<td>/tisʔa/</td>
<td>/tisʔa/</td>
<td>/tisʔa/</td>
<td>/tisʔa/</td>
</tr>
<tr>
<td>Ea$rap</td>
<td>/ʕafra/</td>
<td>/ʕafra/</td>
<td>/ʕafra/</td>
<td>/ʕafra/</td>
<td>/ʕafra/</td>
<td>/ʕafra/</td>
</tr>
</tbody>
</table>

in the English word get; or /ɾaZul/ as the sound /ɡ/ in the English word leisure.

To gain better understanding of the dialectal variation, Table 3.1 compares the pronunciation of Arabic digits in MSA with five different Arabic dialects.

Arabs can often adjust their speech in different ways when speaking to people from different regions, which makes it easier to recognise their speech. However, this is not the case with computers. This variety of Arabic language forms has negative consequences for Arabic NLP for two main reasons. Firstly, the speech recognisers that are designed for MSA do not work accurately when exposed to dialectal Arabic and vice versa, since the difference between MSA and dialectal Arabic is more like the difference between two different languages. Secondly, getting the training data for dialectal Arabic is a real challenge since there is only
one standardised speech corpus of dialectal Arabic available, the so-called “Call-
Home” corpus of Egyptian Colloquial Arabic (ECA) distributed by the Linguistic
Data Consortium (LDC). This lack of training data is a major obstacle for the
development of speech recognisers dedicated to dialectal Arabic.

One of the suggested solutions to tackle the problem of the variety across
Arabic dialects and the lack of Arabic dialectal databases has been introduced
by [KV05]. The proposed technique is based on sharing data by using the acous-
tic data from MSA to recognise ECA. [KV05] used two different corpora for
their study: the LDC Call-Home (CH) corpus of ECA and the FBIS corpus of
MSA. By using this approach, researchers show a considerable improvement in
the recognition of ECA conversational speech. In the same manner, [EGMA10]
have effectively used acoustic models trained with MSA data in recognising digits
in ECA and reported an accuracy rate of 99.34%. The introduced multilingual
approach can help to cover the lack of Arabic dialectal corpora.

3.1.2 Arabic writing system

The Arabic alphabet consists of 29 letters, 26 of which are consonants. The
remaining three letters represent the long vowels (the phonemes /i:/, /a:/, /u:/).Arabic is written in script from right to left. Each letter in the alphabet can
appear in up to four different shapes, depending on its location in the word. For
example, the letter (h َو) can appear in the following shapes (َ و َـو َـو َـو).

The Arabic writing system commonly uses nine optional orthographic symbols
(diacritics) that normally appear after the letters:

i. Three short-vowel diacritics (fatHa ِ, kasra َ:, Damma ُ) representing the
vowels /a/, /i/ and /u/ respectively.

ii. One long-vowel diacritic (Dagger Alif ً) indicating /a:/ occurs in few words,
but these words include some common ones such as (َو A َُا) which means
“this”.

59
iii. Three nunciation diacritics (/an/, /in/, /un/) which come at the end of the word as an indefiniteness marker in Arabic.

iv. One consonant gemination diacritic (called Shadda َ) which is used to duplicate the previous consonant (e.g., َ/kat~aba َ/kattaba/).

v. One diacritic (called sukun َ) indicates that the consonant to which it is attached is not followed by a vowel.

These diacritics are extremely helpful for the readability and understanding of the text. Diacritics, however, are mainly restricted to religious texts and language learning textbooks and they are absent in most other texts. Normally, there are two kinds of diacritics each with a different purpose:

- Diacritics can occur within the word to specify the pronunciation of the phonemes so as to distinguish between words of similar forms. These kinds of diacritics affect the meanings of the word (lexical disambiguation). For example, the diacritic following the letter َ in (kAtb َ/kAtb/ “writer”) changes the meaning from (kAtb َ/kAtb/) to (kAtaba َ/kAtaba/) “to correspond”.

- Diacritics can also appear at the end of the word to demonstrate the role of the word in the sentence (e.g. whether the word is subject or object of a verb). For instance, the diacritic /u/ in (kAtBu َ/kAtBu/) “a writer” indicates that the word is nominative, while the diacritic /a/ at the end of the same word indicates that the word is accusative. In addition to marking the case, these diacritics may change according to voice, mood, and definiteness.

The lack of diacritics in Arabic scripts is the reason for the existence of many similar word forms in the source script, which is consequently found to reduce predictability in both the acoustic and the language models. These words, which look identical, can be associated with different diacritic patterns to deliver different meanings. For example, the word (kr) when diacriticised can be: (akar) ذكر“male”, (ikr) ذكر“citation”, (ak~ir) ذكر“remind”, (a~kar) ذكر“he reminded”,...
(*uk~ir) 'has been reminded', (*ukir) 'has been mentioned', or (*akir) 'remembered'. These examples are presented in Figure 3.1. Arabic readers infer the appropriate diacritics from the context and their linguistic knowledge.

Figure 3.1: Examples for the ambiguity caused by the lack of diacritics.

It has been argued that using data with diacritics can improve the accuracy of speech recognition applications [ANX+05] [ZSS06] [MGL06], especially in cross dialectal speech modelling [KV05]. Hence, much work is suggested to be done in this area in order to attain a satisfactory accuracy either by obtaining vowelised resources or by developing efficient ways to use non-vowelised data. For instance, [VK04] introduced a method for automatic insertion of the missing diacritics into the targeted text toward improving Arabic ASR performance. They used different approaches and showed that lower diacriticisation error rates can be obtained when combining linguistic information (morphology and syntactic context) with the acoustics. [ZSS06] proposed a maximum entropy classifier for restoring the diacritics of each word and obtained a diacritic error rate of 5.1%.
[HR07] introduced a diacriticisation system for Arabic scripts based on a lexical resource. They improved on the results reported by [ZSS06] with a diacritic error rate of 4.8%. In the aforementioned studies, the researchers believed that using non-diacriticised texts in designing ASR systems can significantly deteriorate the performance, and that is what urged them to look for some techniques to overcome this problem. In contrast, other researchers in more recent studies [AEAM09] [AAZ+12] found that using non-diacriticised texts improved the recognition accuracy.

In Section 6.4, we will investigate the issue of missing diacritics in the text dataset and how it affects the performance of Arabic ASR systems. The main comparison is made between result of the system that uses a fully vowelised text material vs. the system that uses non-vowelised text materials.

3.1.3 Phonetic system of MSA

The phonetic system of MSA has 34 phonemes (a phoneme is the smallest unit in a sound system which can indicate differences in meaning between words [Cry03]). These phonemes are divided into two main categories: consonants and vowels. In terms of their phonetic realisation, vowels are articulated with no major airflow restriction in the vocal tract; consonants are made by closure or narrowing of the vocal tract [Cry03]. Vowels can be described in terms of the position of the lips (rounded, spread, or neutral), the part of the tongue raised (front, central, or back), and the height to which it moves (closed or open), while consonants are described in terms of their place and manner of articulation. Arabic has 28 consonants and 6 vowels as listed in Tables 3.2 and 3.3.

Arabic consonants have two distinctive classes, called emphatic and pharyngeal phonemes, which are not found in most languages. There are four emphatic consonants in Arabic: two stops: ض /dˤ/, ط /tˤ and two fricatives ص /sˤ/, and ط /ðˤ. In addition, Arabic has two pharyngeal phonemes: ع /ʔ/ and ح /h/. 
Table 3.2: The Arabic consonants phonemes.

<table>
<thead>
<tr>
<th>Place/ manner of articulation</th>
<th>Bilabial Labiodental</th>
<th>Dental Alveo-dental</th>
<th>Post- Alveolar</th>
<th>Palatal Velar</th>
<th>Uvular</th>
<th>Pharyngo-Glottal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oral stop</td>
<td>ب /b/</td>
<td>د /d/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>م /m/</td>
<td>ن /n/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasal stop</td>
<td>ف /f/</td>
<td>ث /θ/</td>
<td>س /s/</td>
<td>ج /j/</td>
<td>ع /ɣ/</td>
<td>ح /h/</td>
</tr>
<tr>
<td></td>
<td>د /d/</td>
<td>ك /k/</td>
<td>ق /q/</td>
<td></td>
<td>ئ /ʔ/</td>
<td></td>
</tr>
<tr>
<td>Fricative</td>
<td>ط /ð/</td>
<td>ز /z/</td>
<td>حج /خج/</td>
<td>حج /خج/</td>
<td>ع /ɣ/</td>
<td>ح /h/</td>
</tr>
<tr>
<td>Lateral</td>
<td>ص /s/</td>
<td>ط /ð/</td>
<td></td>
<td></td>
<td>ع /ɣ/</td>
<td>ح /h/</td>
</tr>
<tr>
<td>Trill</td>
<td>ل /l/</td>
<td>ر /r/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approximant</td>
<td>ع /w/</td>
<td>ا /ʔ/</td>
<td>و /w/</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The sound /w/ is shown in two articulation places. The reason behind this is that it is articulated with two actions: narrowing of the lip which makes it bilabial, and a raising of the back of the tongue toward the velar which makes it velar. This sound is normally described as labio-velar.*
Table 3.3: The Arabic vowels system.

<table>
<thead>
<tr>
<th>Tongue position/height</th>
<th>Front</th>
<th>Central</th>
<th>Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>High or closed</td>
<td>/i/</td>
<td>/i:/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Unrounded)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low or open</td>
<td>/a/</td>
<td>/a:/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Unrounded)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High or closed</td>
<td>/u/</td>
<td>/u:/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Rounded)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Researchers have confirmed that these distinctive Arabic consonants are a major source of difficulty for Arabic ASR [AM10]. Different problems may arise from these phonemes: they can be mispronounced by both native and non-native speakers; and even when they are pronounced correctly, they have similar counterparts with very similar acoustic features and hence are challenging for ASR systems. This can be shown in the following examples:

<table>
<thead>
<tr>
<th>Emphatic consonant</th>
<th>/d'arb/</th>
<th>non-emphatic counterpart</th>
<th>/darb/</th>
<th>In Arabic + meaning</th>
<th>/mzarib/</th>
<th>“hitting”</th>
<th>In Arabic + meaning</th>
<th>/karib/</th>
<th>“path”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emphatic consonant</td>
<td>/t'inn/</td>
<td>non-emphatic counterpart</td>
<td>/tin/</td>
<td>In Arabic + meaning</td>
<td>/tinn/</td>
<td>“clay”</td>
<td>In Arabic + meaning</td>
<td>/tinn/</td>
<td>“fig”</td>
</tr>
<tr>
<td>Emphatic consonant</td>
<td>/d'yal/</td>
<td>non-emphatic counterpart</td>
<td>/yal/</td>
<td>In Arabic + meaning</td>
<td>/yal/</td>
<td>“still”</td>
<td>In Arabic + meaning</td>
<td>/yal/</td>
<td>“cringed”</td>
</tr>
<tr>
<td>Emphatic consonant</td>
<td>/nasa'aba/</td>
<td>non-emphatic counterpart</td>
<td>/nasaba/</td>
<td>In Arabic + meaning</td>
<td>/nasaba/</td>
<td>“erected”</td>
<td>In Arabic + meaning</td>
<td>/nasaba/</td>
<td>“imputed”</td>
</tr>
</tbody>
</table>

Two studies investigating Arabic distinct consonants have been conducted by [SC98] and [AM10]. These studies focused on the pharyngeal, uvular, and emphatic consonants from the point of view of ASR. They used different techniques and showed a noticeable failure to recognise these distinct consonants when produced by both native and non-native speakers.
In a previous study [AARM12], we confirmed that the emphatic sounds pose a major source of difficulty in recognising Arabic speech. By analysing the confusion matrix resulted from the phoneme recognition experiment, we found that apart from ص /s/ and its non-emphatic counterpart س /s/, the recognition of the emphatic sounds and their non-emphatic counterparts is always below 50%. The poorest accuracy was found in the ض /d/ sound (20%) which has been confused most often with the emphatic sound ط /ð/ (16.2%).

3.1.4 Phonology and morphology of MSA

With a total of 34 phonemes, the allowed syllables in MSA are: CV, CVC, and CVCC (under certain conditions) where V indicates a vowel (long or short) and C indicates a consonant. The Arabic syllable system has the following restrictions:

- The syllable must begin with a consonant.
- The syllable never begins with two consonants

Arabic word structure depends on the idea of "root". The root is characterised as a sequence of consonants, mostly three consonants (or triconsonantal root). Different words with different meanings can be derived from one root, but these words are semantically related in meaning to the root from which they have been derived. Vowels can be combined into the root to form the word’s “stem” (this is why Arabic morphology is commonly referred to as “non-concatenative morphology”). This stem can then be attached to different affixations to form different word patterns. For example, a simple verb root such as (ktb) (meaning “to write”) can be modified to a fairly large number of stems. These stems are surrounded by affixes indicating grammatical categories such as person, number and gender. This process is accomplished in a non-concatenative manner using processes like infixation and gemination [BFI107]. Figure 3.2 shows examples of words derived from the same root in different patterns.

Arabic consists of approximately 5000 roots [BBL+08] with hundreds of patterns; consequently, the resulting number of possible word forms is enormous. On top of that, the number of word forms could increase even further by attaching some articles or prepositions to the beginning of the word (like: the, and, to, from, with, etc.) or appending possessive suffixes at the end of the word (like: ours, theirs, etc.)
This richness in morphology means that Arabic is considered as one of the most morphologically complex languages. This high vocabulary growth rate causes a high language model perplexity in ASR by producing a large number of OOV [KBD+03] and hence a potential increase in WER and a wider search space during decoding which slows down the recognition process [VKDS04].

To investigate the problem of rich morphology in the development of Arabic ASR systems, [KBD+03, VKDS04, KV05, GYMC05] proposed different techniques for better smoothing of the language model by incorporating morphological knowledge in the modelling of dialectal Arabic. So, instead of accommodating the probabilities on the few preceding words, the probabilities are accommodated on the word’s set features. These works reported minor WER reductions.

In a different approach towards solving this problem, [ASK+06] and [XNN+06] separately introduced a morphological decomposition algorithm for creating the ASR lexicon. [ASK+06] applied this algorithm to create an Iraqi Arabic ASR
lexicon, while [XNN+06] applied it to MSA. They both reported significant improvements in reducing WER using a faster system in terms of speed and needing less memory.

More recently, [MSN12] have proposed an effective technique to improve the robustness of the language modelling in building Arabic LVCSR. The idea is to use morphemes rather than the whole word in language modelling which gives a better lexical coverage and less language model perplexity. This method is combined with the use of features derived from morphemes to increase the robustness of the language modelling. The use of morpheme level features has lead to a relative decrease in WER of 3.9% compared to a word level based system.

3.1.5 Arabic syntax

Arabic has some special structural features which may cause syntactic ambiguity over and above the standard sources of syntactic ambiguity such as attachments. There are many sources of syntactic ambiguity in Arabic:

i. The pro-drop nature of the Arabic language. Arabic has a tendency to omit the subject pronoun when following a verb. Look at the following example:

\[ ra\rightarrow at \ Al b\int \]  
\[ رأَتُ الْبَنِّ \]  
saw the girl

It is not obvious in this sentence whether the NP following the verb is the subject (in which case the meaning is “the girl saw”) or the object and the subject is an omitted pronoun (meaning ‘she saw the girl’). This pro-dropping nature of Arabic can cause a great deal of difficulty for the syntactic parsing as such sentences will produce two structural analyses depending on whether there is an omitted pronoun or not [Cha04] as shown in Figure 3.3. The ambiguity arises when a verb is followed by only one NP, as many verbs in MSA can be both transitive and intransitive.

ii. The free word order in Arabic. Arabic allows VSO, VOS, SVO, and OVS constructions as shown in the following examples:
SV O with OVS sentences and VSO with VOS can get mixed up with each other, especially when the text is not diacriticised since the diacritics are the major tool used to distinguish between subjects and objects in Arabic. This means that many VSO and SVO sentences can be interpreted as VOS and OVS sentences respectively, especially when both subject and object of the verb are
expected to be human. For instance, in ($\text{šarība Ašwāl ad Aḥlāyb}$) "the boy drank the milk", it is obvious that the boy is the subject (not the milk). Meanwhile, in ($\text{yaḥtarīm Ely MuHam-ad}$) "respect Ali Muhammad" the subject can either be Muhammad or Ali.

This flexibility in the word order can get more complicated when constructing zero copula sentences. The normal order followed in these sentences is subject and then predicate as in ($\text{Al-kitāb Ašty Al-Tawīlāp}$) "the book is on the table". This order can be inverted allowing the predicate precede the subject as in ($\text{Ely Al-Tawīlāp Al-kitāb}$) "on the table is the book". This exchangeability in position must be well constrained in order to eliminate a good deal of ambiguities.

We may emphasise here that to overcome the syntactic ambiguity, the diacritics should first be restored. The diacritics play an important role in assigning the proper syntactic analysis as they help distinguish between nouns and verbs, transitive and intransitive verbs, active and passive forms, and imperative and declarative forms (see Figure 3.1). Moreover, the syntactic analysis cannot be done without a comprehensive lexical and morphological description in order to eliminate ungrammatical sentences during parsing.

[Dai01] pointed out that the problem of Arabic syntactic ambiguity had not received enough attention among researchers. These limited efforts introduced different tools and mechanism to manage Arabic syntactic ambiguity [Dai01] [Att08]. Detecting the syntactic ambiguity is not crucial for developing ASR systems whose main job is to provide a text transcription for the given speech. It is, however, crucial for ASU which is aimed at understanding the meaning as well as producing the written form of the speech.

### 3.1.6 Mapping between spoken and written text

Arabic is a phonetic language, which means that the correspondence between the pronunciations and individual letters, when all the diacritics are included, is
deterministic, unlike English or French. In English, using the rule-based transcription system alone is a formidable task [EI04]. For example, it is almost impossible to put a rule that can predict how to spell /f/ in the following English words: *rough* and *cuff*.

However, although Arabic offers a direct relationship between the isolated letters and their spellings, transcribing Arabic text is not always that easy. This is primarily caused by the previously discussed problem related to the lack of phonetic information in Arabic scripts which requires restoring the omitted diacritics first of all.

Furthermore, transcribing Arabic texts is not just a matter of converting the graphemes into phonemes but also converting them to phones represent the actual sound of the language. This task is done partly by analysing the boundary effects on pronunciation. The issue here is that the local context can have considerable impact on what a phoneme sounds like. This can have substantial consequences for speech recognisers, since it means that the phonetic transcription which is required for any speech recogniser needs to be sensitive to the local context. To see the influence of the acoustic context in the phoneme realisation, consider the example (manī *AlqAdim*) who is coming?”. Here, the phoneme (A) in the word (AlqAdim) is affected by the neighbouring vowel (i) therefore, the phoneme (A) is not pronounced /manī lqAdim/. On the other hand, if the same word was at the start of the utterance it would be pronounced /?alqadim/. The acoustic realisation of the word varies depending on its position in the sentence.

Despite the fact that Arabic phonetics and phonology have been extensively studied and documented fairly early (from 12 centuries ago), researchers emphasise that these valuable studies still need to be revised and formulated in a modern mathematical framework [EI04]. This modern framework can be formulated into algorithms suitable for computer-based processing which can be used to produce the most likely pronunciation or pronunciations for a given word.

In recent years, a limited amount of literature has focused on the influence of
the surrounding context over the pronunciation and its consequences in developing Arabic ASR systems. Researchers emphasise the importance of this area and suggest carrying out more extensive research. That is because it has been proven that capturing the pronunciation variations can considerably enhance the Arabic ASR systems performance. Since this issue is a major concern in this thesis, Chapter 5 will list the recent studies and suggested methods in the literature that tackle the problem of mismatch between the spoken and written forms.

3.2 Review of the work done in this field

Development of Arabic ASR systems requires a multidisciplinary expertise with a good understanding of Arabic phonetics, natural language processing, and Arabic speech processing techniques.

Recent efforts in developing Arabic ASR systems can be divided into two groups. The first group focused on the recognition of MSA, while the main focus of the second group was on dialectal Arabic, mainly ECA.

This section starts with a brief overview of the recent work done in developing ASR systems dedicated to MSA. It then reviews the efforts put into developing dialectal Arabic speech recognisers and finally it introduces the works done in developing small vocabulary speech recognisers such as digits and alphabet speech recognition systems. In describing each study, we try to introduce the domain of the work and the techniques used and then we provide the accuracy of the developed system (if reported). The accuracy of a speech recognition system depends on a huge number of factors—size of the training data, perplexity of the language (which in turn depends on the size of the vocabulary and the complexity of the grammar), quality of the recordings, homogeneity of the subject population, to mention just a few. It is therefore unfeasible to compare the performance of different systems merely by looking at the accuracy rate.

3.2.1 Works dedicated to the recognition of MSA

The recognition of MSA speech has been addressed by a number of researchers. They used different techniques for different recognition tasks. [AO01], for instance, addressed the problem of recognising continuous speech in MSA in his Master’s thesis. He studied different approaches to building the Arabic speech
Corpus. He reported a recognition rate for speaker dependent ASR of 93.78% using a new technique for labelling Arabic speech based on the HTK.

[BNS+02] addressed the problem of indexing Arabic broadcast news (TIDES OnTap system). The three main components of the system are ASR, speaker identification, and Arabic document tracking. The proposed system dealt with many challenges presented in Arabic such as the absence of diacritics and the presence of compound words which are formed by concatenating prefixes and suffixes to the word stem. This system is capable of achieving accuracy of 84.4%.

[ECEKA+03] described a new approach for developing an Arabic isolated-word speech recognition system. The presented work was implemented by the Modular Recurrent Elman Neural Networks (MRENN). They used a separate Elman network for each word in the vocabulary set. They argue that by using the new neural networks approach, we could have promising results compared with the traditional HMM-based speech recognition approaches. Although they obtained good results (87% accuracy), the approach they used for the isolated-word small vocabulary system may not be suitable for LVCSR for it would face many problems related to memory and performance.

A novel approach for improving the robustness of the Arabic ASR system was presented by [SRA+03]. The system is based on a hybrid set of speech features consisting of intensity contours and formant frequencies and reported 82.59% recognition accuracy.

In a different study, [BS04] proposed a Connectionist Expert System (CES) dedicated to Arabic isolated-word ASR. The motivation behind developing this system was to overcome the limitations inherit in the topologies and learning style of ANNs. To achieve this, the pattern processing of ANNs was combined in a speech recognition hybrid system with the CES. It consists of an expert system implemented throughout a Multi Layer Perceptron (MLP). In this system, the syllables were used as the phonetic unit.

[AHR+06] used speech recognition techniques in developing a computer-aided pronunciation learning system directed to non-native speakers. The speech recognition engine was used to detect the speaker’s errors. The system’s evaluation is done using a dataset that contains 6.6% wrong speech segments. The system correctly recognised the error in 62.4% of the errors and asked for “Repeat Request” for 22.4% of pronunciation errors and made false acceptance of 14.9% of total errors.
[EM07] investigated the use of neural network language models for Arabic broadcast news and broadcast conversations speech recognition. The proposed method showed an improvement over the baseline N-gram model resulting in reductions of up to 0.8% absolute and 3.8% relative in WER.

More recently, [EAMAG08] reported their on-going research towards achieving a high performance, natural language, large vocabulary, speaker-independent Arabic ASR system. This work is still in its initial stages where they utilised two different recognition tools (Sphinx and HTK) to build their system and developed an infrastructure for the research. A rule-based phonetic dictionary was also provided. They used 5.4 hours of Arabic broadcast news corpus for training and testing and reported comparable accuracy to the English ASR system for the same vocabulary size with 91% accuracy rate. They expected that further improvements would be added in the next phase, including expanding the corpus to 40 hours, enhancing the rule-based phonetic dictionary, and using a finer parameterisation of the acoustic modelling.

[AEAM09] have reported the development of an Arabic broadcast news transcription system based on the Carnegie Mellon university Sphinx tools and the HTK tools which were utilised at some testing stages. 7 hours of data was used to train the system and half an hour used for the testing phase. Key issues addressed in this paper are the generation of fully vocalised transcription, and rule-based spelling dictionary. After extensive testings and amendments in the recognition parameters they achieved an accuracy of 91.39%.

[LMG09] introduced additional improvements to an automatic transcription system for Arabic broadcast news. Researchers highlighted some techniques that were introduced to deal with Arabic-specific challenges, such as: lack of diacritics, dialectal variants, and huge lexical variety. Those improvements were found to yield significant improvements in speech-to-text transcription performance.

[PDG+09] presented various schemes for applying features derived from Multi-Layer Perceptrons (MLPs) for Arabic ASR. Three schemes have been explored: first one investigates using MLP features to include short vowels into the graphemic system. Second scheme describes a rapid training approach for use with the standard PLP feature and MLP feature adaptation. The final scheme demonstrated the use of linear input network (LIN) adaptation as an alternative to the standard HMM-based linear adaptation. The main finding of this study is that using rapid training with MLP features and their use for short-vowel incorporating and
LIN adaptation give noticeable reductions in WER.

[MKC+11] described the fifth phase of developing an Arabic broadcast transcription system fielded by IBM in the GALE project. The key advances in the system includes using Bayesian Sensing HMM acoustic models, improved neural network acoustic feature, MADA vowelised acoustic models. These advances have led to a recognition accuracy of 92.6% with absolute improvement of 0.9% in WER over the previous system.

[AEAMAK12] demonstrated the usefulness of exploiting linguistic knowledge in improving the Arabic ASR system by presenting a new design for language models. A syntax-mining approach was introduced to rescore N-Best hypotheses for Arabic ASR systems. Using CMU Pocket Sphinx speech recognition engine trained with 7.57 hours pronunciation corpus of MSA, the proposed method reported a slight and inconsistent enhancement in the performance.

[AAZ+12] reported comparable recognition results from developing a speaker-independent continuous Arabic ASR system. The work includes collecting a phonetically rich and balanced speech corpus with all diacritics included, building Arabic phonetic transcription, and creating an Arabic statistical language model

In order to compensate for the lack of reliable speech databases dedicated to Arabic, [ASAM12] designed a cross-language scheme between Arabic and English. The obtained results confirm that English acoustic models for common phonemes can be used for MSA to solve the problem of speech data sparsity.

3.2.2 Works dedicated to the recognition of dialectal Arabic

Within the second group (concerned with dialectal Arabic), [BMM+97] and [ZMM+98] demonstrated the adaptability of the Byblos English system to Spanish and ECA. They introduced an LVCSR system which uses a combination of Phonetically Tied-Mixture Gaussian HMMs and State-Clustered Tied-Mixture Gaussian HMMs. They addressed the problems arising from the dialectal differences and the differences between the written and the spoken language. The proposed work was performed within the framework of the 1996/97 NIST (national institute of standards and technology) benchmark evaluations on the CallHome. The best performance obtained in those evaluations was 39% in word recognition accuracy.

In a workshop held in 2002 at John Hopkins University, [KBD+03] worked
on a novel approach to Egyptian Arabic ASR using the LDC's CallHome Arabic Speech Corpus. They concentrated on three major problems: i. the absence of diacritics in the Arabic text; ii. handling the complex and rich morphology of Arabic; and iii. the differences between dialectal and formal Arabic. As an attempt to present solutions for the mentioned problems, an automatic vowel restoration approach was presented by using automatic Romanisation techniques which will annotate the script with diacritics and other phonetic information normally missed in Arabic texts. They showed that by having this phonetic information the recognition rate will be significantly improved. Moreover, they investigated three different types of language models designed to better exploit Arabic as a morphologically rich language: particle models, morphological stream models, and factored language models. They observed a slight but consistent reduction in WER by using morphologically based language models when combined with standard word-based language models. Finally, they found that using MSA data for the ECA language modelling did not yield any improvements as they behave like two distinct languages.

Other studies based on the same ECA CallHome corpus have been carried out by [VKDS04] and [KV05]. [VKDS04] focused on the improvement of the morphological knowledge of the language model, while [KV05] aimed at improving the recognition rate of ECA by using the acoustic data from MSA.

[HEA07] proposed a speech recognition application dedicated to the Lebanese dialect based on a phoneme-like template model. They reported 96.8% and 95.9% accuracy rate using 94 utterances and 171 utterances, respectively.

[EGM12] presented novel techniques to train a reliable speaker-independent dialectal Arabic ASR system. The proposed method presents solutions for the problems of limitation in dialectal speech corpora and that lack of dialectal scripts as there is no standard orthography for dialectal Arabic. The presented solutions are about finding ways to benefit from the existing MSA speech corpora and finding a novel approach to phonetically transcribe dialectal speech.

3.2.3 Works done on Arabic digits recognition

As we can see, most of the reviewed studies have been conducted on MSA and dialectal Arabic in a general manner, considering either a limited vocabulary, or speaker-dependent system. In addition, a few studies have been conducted with a focus on digital processing, Arabic alphabets recognition, and Arabic vowels.
more specifically. Arabic digits recognition systems were introduced in [AAK85], [HAZ06], [SHC07], [Alo08], [Alo05], [AAA], and [LKSK95].

[AAK85] developed another Arabic digits recogniser using positive-slope and zero-crossing duration as the feature extraction algorithm. He reported a 97% accuracy rate. The system in [HAZ06] used the Carnegie Mellon University’s (CMU) Sphinx-IV engine, based on the HMM. 99.21% word recognition rate was reported for about 35 minutes of training speech data and 7 minutes of testing speech data.

The system in [SHC07] was also using CMU Sphinx-IV engine based on HMM for the same task and obtained a word recognition rate of 86.66% using Moroccan Arabic digits monophone-based recognition. Alotaibi in [Alo08] and [Alo05] compared two versions of Arabic digits recognition system based on ANN and HMM. The ANN-based recognition system attained a correct digit recognition rate of 99.5% whereas the HMM-based recognition system achieved a 98.1% correct digit recognition rate for the speaker-dependent system. On the other hand, the same versions of the system were tested for multi-speakers and showed correct digit recognition rates of 94.5% and 94.8% based on ANN and HMM, respectively.

In [AAA] a telephony Saudi accented Arabic corpus was used as the speech data for a digits recognition system. The system used the HTK tools based on HMMs and reported a correct digit recognition rate of 93.67%.

[LKSK95] reported satisfactory results for recognising Arabic spoken digits. The system is based on HMMs and Tree distribution approximation model and uses second-order derivatives of MFCC in recognising Arabic digits. Compared to an MFCC based system, the developed system reported an increase in recognition accuracy by 4.6% which lead to an overall recognition accuracy of 98.41%.

3.2.4 Works done on Arabic alphabet recognition

In the domain of recognising individual Arabic phonemes, two recent studies were conducted for recognising Arabic alphabets, [AAA10b] investigated the spoken Arabic alphabets from the point of view of speech recognition problems. The proposed system used the HTK to implement the isolated word recogniser with phoneme-based HMM models. The experiments were carried out with the aid of SAAVB (Saudi Accented Arabic Voice Bank) as the speech corpus. The recognition system achieved a 64.06% overall correct alphabets recognition.

[AEA11] proposed an approach to recognise Arabic spoken alphabets from
multi-speakers based on the Principal Component Analysis (PCA) technique for letters featuring extraction and ANNs for the recognition process. They reported 95.5% recognition accuracy over a large dataset.

A few other studies have focused on investigating isolated phonemes such as vowels and Arabic unique consonants from the point of view of ASR.

[Alg98] conducted an interesting spectrographic analysis of Arabic vowels based on a cross-dialect study. He studied the intensity, frequency, and bandwidth of the first five formants of the six Arabic vowels. The data was collected from speakers of different Arabic dialects, including Saudi, Sudanese, and Egyptian dialects. The author found that the phonetic implementation of the standard Arabic vowel system differs according to dialects. He pointed out that the results of this study are useful for further research on speech processing.

[AH10] developed an automatic vowel recognition system for MSA. They studied differences and similarities between the vowels depending on the syllabic nature of the Arabic language in terms of syllable types and structure, in addition to some primary stress rules. This recogniser is based on HMMs and it is built to analyse the similarities and dissimilarities between the phonetic features of vowels. They stated that the goal of this study is to enable future speech recognisers to classify the vowels.
Chapter 4

Experimental set-up

This chapter describes the experimental dimensions of the developed ASR systems, mainly by introducing the tools and algorithms used. We use version 3.4.1 of the HTK developed at Cambridge University. The HTK has a wide range of applications, but is primarily used for speech recognition. The HTK consists of a set of tools and library modules used for constructing the components of the speech recogniser. The main component manipulated by the HTK is a set of acoustic speech HMMs. Other components used by the recogniser include a pronunciation dictionary and language models.

This chapter is organised as follows: firstly we will talk about the main component of the HTK, namely HMMs. The HMM methodology can be employed in different areas of applications to provide a highly reliable way of recovering a data sequence. In this section, our focus will be on HMMs as applied to ASR. We will define the main concept of HMMs, pointing out the strengths and drawbacks of applying the HMM methodology in building speech recognisers. The second section of this chapter describes the typical stages of building an ASR system using the HTK. We will define the HTK tools and libraries then go through the processing steps of designing an ASR system using the HTK. Then, we will introduce the standard way of running the HTK and identify the source files needed before running the toolkit. Finally, we will explain the amendments to the HTK processing stages in order to use the generated phonetic transcription instead of the dictionary. In the third section, we describe the text and speech corpora used in conducting the experiments carried out in this thesis.
4.1 Designing an HMM-based system

Looking at the predominance of speech recognition studies based on HMMs, we can confirm that HMMs are currently the most successful modelling approach for speech recognition. An HMM is typically described as a stochastic process that uses probability measures to model sets of sequential data represented by a series of observation vectors [MG00]. In HMM-based speech recognition systems, the model is used for characterising the spectral features corresponding to each word in a parametric random process and the parameters used in the stochastic process are predictable in a precise manner [Alo08]. An example of three simplified states of HMMs was seen in Figure 2.5. According to [JR91], the reasons behind the popularity of using HMMs as the core in the speech recognition field is due to the following advantages:

i. The rich statistical framework of the HMM methodology: HMM methodology provides a rich mathematical framework for building ASR systems and this is considered to be one of its main advantages. As a result, it is able to characterise speech signals in a statistically tractable way. This statistical theory allows for the use of a combination of short-time spectral characteristics (related to the individual states) and the temporal relationship between the processes (the Markov chain) using a single consistent statistical framework. Consequently, the utterance is automatically segmented during the recognition process such that it is not required to have pre-segmented data.

ii. Flexibility and the ease of implementation: one of the strengths of using the HMM methodology in ASR systems is the inherent flexibility and the ease of implementing the training algorithms. The HMM presents a flexible topology for statistical phonology and syntax so it can easily accommodate various levels of phonological and syntactical constraints. Furthermore, the ease of implementing the HMM algorithms makes it reasonably straightforward to build systems with large corpora in a short time and with limited complexity of the recognition process.

4.1.1 Main issues in using HMMs: problems and solutions

In the development of the HMM methodology, three commonly recognised problems are encountered [JR91] [RJ86] [Rab89] [Pic90] [RJ93] [JMK+00]. Below
is a summary of the problems in association with the proposed solutions and algorithms.

i. Likelihood problem
The first challenge in implementing the HMM technique is to compute the probability of a particular observation sequence. For instance, given the HMM \((\lambda = (A, B, \pi))\) and the observation sequence \(O = (O_1, O_2, O_3...O_T)\) the question is how to efficiently evaluate \(P(O|\lambda)\). The most straightforward way to overcome this problem is by selecting a model that closely matches the observations. The forward algorithm is a dynamic algorithm which is used for efficient calculations.

ii. Decoding problem
This problem concerns the difficulty of calculating the most likely hidden state sequence \((Q= q_1, q_2, q_3...q_T)\) given an HMM model \((\lambda = (A, B, \pi))\) which led to observation sequence \(O = (O_1, O_2, O_3...O_T)\). The solution to this problem has to do with finding the right state sequence associated with the given observation sequence. The way of deciding the most probable state sequence is by using a dynamic programming algorithm called the Viterbi algorithm. This algorithm is similar to the forward algorithm but instead of summing up procedure in forward calculation, the Viterbi algorithm maximises the previous state.

Forward algorithm:
\[
\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1})
\]  

Viterbi algorithm:
\[
\delta_{t+1}(j) = [\max_{i} \delta(t) a_{ij}] b_j(O_{t+1})
\]

iii. Learning problem
This is the third and perhaps the most crucial issue. The learning problem involves in finding how to adapt the HMM parameters \((\lambda = (A, B, \pi))\) to observed training sequences (training data) and maximising \(P(O|\lambda)\).

The solution to this problem lies in determining how to adjust the HMM parameters to maximise the probability of the observation sequence using the observation sequence as training data. There is no algorithm known to
especially find an optimal parameter value. However, the best way is the use of iterative procedure such as the Baum-Welch re-estimation method which combines both forward and backward algorithms.

We can conclude that the development of the HMM methodology involves evaluating the probability of an observation sequence produced by the source model in order to decode the most likely hidden state sequence and learn the HMM parameters that give the greatest probability.

4.2 The HTK

The HTK is a software toolkit for handling and manipulating HMMs. It consists of a comprehensive set of library modules and tools written in C which can be used together to build and test HMM-based recognisers. The HTK was developed gradually, many versions of HTK have been introduced, the latest one is 3.4.1 which is used in our study. The tool runs mainly under the UNIX operating system but can also run under other operating system environments [YEG+06] [You93].

The HTK toolkit is designed to ease building systems using continuous density Gaussian mixture HMMs [YWB94]. The main structure of the HTK is composed of the training phase which involves using a set of training tools to estimate the parameters of a set of HMMs. The second phase is the recognition phase, which involves producing a transcription for the speech using a set of recognition tools. The tools work on training and decoding the HMMs based on the forward-backward algorithm i.e. Baum-Welch, which is used for the training, and the forward and Viterbi methods, which are used for decoding. Figure 4.1 visualises the basic phases of the HTK. The description presented in this section is based on the documentation provided by the HTK team.

4.2.1 The HTK library modules

The HTK is composed of a number of library modules and tools that allow for sophisticated analysis of the speech. A command-line style interface is used to execute each tool. There are more than 20 tools available in version 3.4.1, each is designed to perform specific tasks in certain phases. Each tool is called with a number of required arguments plus optional arguments which are marked by
prefixing the “minus” sign. The operation of a tool can be controlled using parameters saved in a configuration file besides the command-line. For example, when invoking the command

```
HTool -C config -f 65.6 -a -s myfile Xfile Yfile
```

the tool will load the coding parameters stored in the file “config” to complete the task. Figure 4.2 explores the parameters stored in the “config” file.

Figure 4.1: Two major phases of running the HTK.

Figure 4.2: Configuration file “config”.

4.2.2 The HTK processing stages

The best way to introduce the HTK is by exploring the processing stages of designing a speech recogniser and explaining the task of the tools involved in each stage. As mentioned previously, there are two main stages in building a speech recogniser: training and testing (the steps that are carried out during testing
are also used when the final system is deployed). Each stage is preceded by a number of data preparation steps in order to make the data ready for processing. Once the testing is done, the data analysis tool is executed to determine the performance of the system. A number of tools are required in establishing each stage, as shown in Figure 4.3. The following is a brief overview of each processing stage and a description of the tools involved.

4.2.2.1 Training

This is the first stage in developing a speech recogniser. Before training the recogniser a number of data preparation steps need to be done. The aim of these steps is to prepare the datasets to be processed by the HTK tools during training through converting them to an appropriate format. This requires, first of all, providing a set of speech data files and their associated transcriptions before executing any tool. The required data are:

Wave files: the speech files can be recorded using the HTK tool HSLAB or any appropriate recording software. The wave files must be labelled with the corresponding word.

Text files: the transcription of the wave files must be given in a file called prompts which provides the training sentences. It is necessary to create a word list from the provided sample sentences and store it in a file called wlist using
the script prompts2wlist provided by the HTK. In addition, the pronunciation dictionary is required to be used in conjunction with the prompts to obtain the initial phone level transcription which is needed in the training process. The dictionary contains a sorted list of the required words. It can be created by hand or for more complex tasks it can be extracted from a lexicon such as the British English BEEP pronunciation dictionary using the HDMan tool. This tool needs to have the desired word list stored in wlist, which has been previously extracted from the prompts, and a standard source pronunciation dictionary. The HDMan tool will create two outputs which will be needed for training and testing the system. The first one is a pronunciation dictionary called dict obtained by searching through the provided standard source dictionary to find the pronunciation(s) for each word in wlist. The second output of the HDMan is a list of phones used in the dictionary stored in the file monophones1. Optionally, the HDMan tool can provide statistics about the constructed dictionary in a log file called dlog.

This is one of the places where we introduce changes to use the generated phonetic transcription instead of the dictionary as will be described in Section 4.2.4.

After having those requirements, two steps must be taken to prepare the data for training:

i. Transcribing the data: this process involves converting the text data file into an appropriate format that is required to train a set of HMMs. This includes generating word level and phone level transcriptions as a Master Label File (MLF). An MLF is a file that contains a complete set of transcriptions of words to be processed. A script called prompts2mlf is used to create the word level transcription MLF file words.mlf. After the creation of this file, the phone level MLF file can be generated by matching the words in words.mlf with the associated transcription found in the dictionary file dict using the label editor HLEd.

Figure 4.4 shows the process of generating the phone level transcription needed to start the HMM training process.

The HLEd tool relies mainly on using the dictionary to generate phone level and word level transcription. This tool is not used in our modified version of the system which uses the generated phonetic transcription as will be discussed in Section 4.2.4.
ii. Parameterisation of the data: it is necessary to convert the wave files into sequences of feature vectors to represent the speech and their transcription. Each feature vector has 39 dimensions: 13 features (12 cepstral features plus energy), 13 deltas, and 13 double deltas. Converting the input wave file into MFCC vectors is performed using the HCopy tool which uses the conversion parameters stored in the file config (Figure 4.2). In addition, a file containing a list of each source file, and its corresponding output file is required. This file is referred to as a “script file” and given the extension “scp”. Figure 4.5 illustrates the contents of the script file codetr.scp and Figure 4.6 summarises this step.

![Figure 4.4: Generating a phone level transcription.](image)

![Figure 4.5: The contents of the script file “codetr”.](image)
By the end of this step we should have the appropriate acoustic feature format that the HTK requires for the wave files associated with the transcriptions of the speech in phone and word labels. It is time now to train the system and construct a set of HMMs using the provided data. Figure 4.7 summarises the data preparation steps that were undertaken.

The training phase is concerned with initialising and estimating the maximum likelihood of the HMM model parameters. It requires having the speech data supplemented by its phonetic transcription in a special formatting. The simple structure of this process is presented in Figure 4.8.

A set of the HTK tools are used for the purpose of building a well-trained set of HMMs. Figure 4.9 gives a detailed representation of this step which is described below.

1. Creating monophone HMMs: the aim of this step is to create a set of well-trained single Gaussian monophone HMMs, beginning by creating identical monophones in mean and variance. These identical monophones are then
retrained after short pauses silence models are added. This is followed by realigning the training data and creating new phone level MLF for the words with multiple pronunciations. The following is a brief description of each step.
i. Creating flat start monophones: the starting point in the training stage is creating identical monophones HMMs in mean and variance. Firstly, defining the model topology required for each HMM by writing a prototype definition. The vector is of size 39 and the mean is initialised to 0.0 while the variance is diagonal to 1.0 as shown in Figure 4.10. Then the HTK initialisation tool HCopmV is carried out to calculate the global mean and variance and set all the Gaussians in a given HMM to an identical mean and variance. The HCopmV tool specifies the initial parameters by scanning the set of data files stored in train.scp, and then creating a new version of proto in the hmm0 directory where the 0.0 means and 1.0 variances are replaced by the global means and variances as illustrated in Figure 4.11.

![Figure 4.10: Proto file.](image)

Given the generated prototype model stored in the file proto in the hmm0 directory, an MLF file called hmmdefs is created by copying the prototype of each of the required monophone HMMs (including “sil”). The process of creating flat start monophones stored in the directory hmm0 is followed by a re-estimation process that invokes the HERest tool. The HERest is the core embedded training tool. This tool is used to refine the parameters of existing HMMs using Baum-Welch Re-estimation. It works by loading all the HMMs in hmm0 which are listed in monophones0
and re-estimating them using the data provided in train.scp before the new model set is stored in a new directory hmm1. The HERest tool should be executed twice more in order to change the name of the input and output directories until we have the directory hmm3 which contains the final set of the initialised monophone HMMs.

ii. Creating silence models: after having a 3-state HMM for each phone and an HMM for the silence model (sil) generated from the previous step, we need to include the short pause (sp) silence model. The sp silence model refers to the type of short pauses that occur between words in continuous speech. The sp model is tied to the centre sil state. Two steps are required to fix the silence models: firstly, duplicating the centre state of the silence model stored in hmmdefs inside hmm3 directory to the hmm4 directory and including the new sp model. Secondly, by running the HMM editing tool HHED which applies a set of commands in a script file called sil.hed to modify the HMMs by introducing the extra transition from states 2 to 4 and from states 4 to 2 in the silence model and tying the sp state to the centre sil state. This step is intended to improve the robustness of the context-dependent model by allowing transition skipping over all output states. The re-estimation tool HERest is executed twice following this step to re-estimate the parameters of the modified set and having it stored it in the directory hmm7. This step is illustrated in Figure 4.12.

iii. Realigning the Training Data: in this step, the HVite tool is applied to take into account all pronunciations for each word in the dictionary (for words with multiple pronunciations) and output the word that best
matches the acoustic data. It transforms the word level transcription words.mlf to the newly created phone level transcription aligned.mlf as explained in Figure 4.13.

The alignment tool (HVite) is cancelled in the modified version of the system in order to use the generated phonetic transcription to obtain the correct pronunciation of the word. Details will be given in Section 4.2.4. Again, the HERest tool is executed using the created file aligned.mlf to produce hmm8 and hmm9 files.

II. Creating tied-state triphones: the second step in building the training model is to create a context-dependent triphone HMMs. This step improves the recognition accuracy as it looks to match a specific sequence of 3 sounds at the same time rather than a single sound. This step is carried out in two
stages described below:

i. Generating triphones from monophones: the creation of context-dependent triphone transcriptions is realised by executing the HLEd tool. The HLEd tool converts the monophone transcription in aligned.mlf to a set of triphone transcription and have it stored in the file wintri.mlf. In addition, it writes a list of all the triphones in the file triphone1. This is done with the aid of the edit script mktri.led. Figure 4.14 shows the process of generating triphones, giving an example of how to generate triphones for the Arabic sentence (yaktubu rajul) “a man writes”.

![Figure 4.14: Generating triphones.](image)

It can be noted from Figure 4.14 that the words boundary symbols are not converted and some biphones will also be generated in case the context at word boundaries includes only two phones. The next step is to use the HMM editor HHEd to clone the models. Again, this step is followed by two passes of re-estimation so the resultant HMMs are ultimately stored in hmm12.

ii. Tying triphones states: this process uses the set of triphone HMMs obtained from the previous step to tie the states within triphone sets
in order to deliver more robust parameter estimates. The HTK uses a “decision tree” to allow states to be clustered and then tied together. The decision tree is made of three parts, the first part contains a set of linguistic questions that defines the left and right context of the phones. The second part contains statistics for the training corpus which determine the possible questions of each phoneme created by executing the HHeD tool. The third part of the decision tree contains a set of three commands described in the comments on Figure 4.15.

Figure 4.15: The structure of the decision tree file (tree.hed).

Finally, after tying the HMMs, the resulting tied-state triphones are re-estimated using two rounds of the Baum-Welch algorithm, so the trained models are ultimately contained in the file hmmdefs inside hmm15 directory.

4.2.2.2 Testing and recognition

With the building of the speech recogniser completed, it is time to test its performance. The main tool used in this stage is HVite which is a general purpose Viterbi word recogniser used to match the speech signals with the HMMs network and to return a transcription for each speech signal. The HVite tool uses a number of test data files as a reference for the matching. Before explaining the way the recognition tool works, we will first describe the files that we have not come across so far. The files are:
i. Test transcription files

Before doing the recording and testing, the HTK needs to be provided with a task grammar which consists of a set of variables described by regular expressions to define the legal word sequences. An example of a simple task grammar is:

\[
\begin{align*}
\$n &= \text{banAt} \mid \text{mudaris} \mid \text{TAlib} \mid \text{bintu} \mid \text{rajul} \ldots; \\
\$v &= \text{taktub} \mid \text{tadrus} \mid \text{takAtab} \mid \text{katab} \mid \ldots; \\
\$p &= \text{min} \mid \text{fiy} \mid \text{Ean}; \\
\$np &= \$n \ [\$n]; \\
\$pp &= \$p \ \$np; \\
\$s &= \$v \ \$np \ [\$np] \mid \$np \ \$v \ [\$np] \mid \$np \ \$pp \ldots; \\
( & \text{SENT-START} \ (\$s) \ \text{SENT-END} )
\end{align*}
\]

The dollar symbols here define the variable names, the vertical bars express possible alternatives, and the square brackets denote optional items.

The grammar is used to extract a word network \texttt{wnet} which explicitly lists word-to-word transition networks. That is essential to the recognition stage.

The word network is built by invoking the \texttt{HParse} command.

The \texttt{wnet} file is defined using the HTK Standard Lattice Format (SLF), which provides each word instance and the transitions from word to word. Figure 4.16 gives a simple example of a word-based network for one of the experiments.

Also, the prompts for testing sentences must be provided. The \texttt{testprompts} file contains the sentences that will be recorded and tested on the recogniser. As shown in Figure 4.17, the structure of the file is similar to the \texttt{prompts} file which contains the training sentences.

The \texttt{testprompts} file can be created manually or by using the HTK tool \texttt{HSGen} to generate a random list of test utterances derived from the \texttt{wnet} and \texttt{dict} files. After generating the test transcription, an \texttt{MLF} file that contains a word level transcription of the test files should be created. The \texttt{testref.mlf} file will be used later in analysing the results as a reference file. The \texttt{testref.mlf} file is generated by running the \texttt{Perl} script file \texttt{prompts2mlf}. The effect of
ii. Test speech files

The HTK needs to have the recording files in MFC format rather than WAV format. After providing the test utterances and recording them, coding from WAV to MFC format should be made using the HTK tool HCopy. HCopy tool needs a configuration file config which specifies all of the conversion parameters (as seen in Figure 4.2) and then generates the codetest.scp file which lists the source and output files as seen in Figure 4.19. The list of the coded test files is then stored in the test.scp file.

Assuming that all the files needed for the recognition are provided, the HTK tool HVite is performed so that each test file specified in test.scp will go
The contents of the `testref.mlf` file are shown in Figure 4.18. The file contains transcription data for each test file. Similarly, the structure of the `codeTest` file is depicted in Figure 4.19. This file contains a list of files to be processed by the HTK workflow.

HVite recognises the entire list of HMMs by matching it against a network of models and then writes the recognised words in the `recout.mlf` file. The process of recognising the test data is visualised in Figure 4.20.

Figure 4.21 provides a conceptual description of the HTK workflow.

### 4.2.2.3 Recognition evaluation

After the completion of the recognition process and assuming that the word level transcription for each test file is provided in `testref.mlf`, the performance of the recogniser can be evaluated by running the `HResults` tool. This tool computes various statistics related to the recogniser performance by conducting a comparison between the file that contains the recognition results `recout.mlf` with the corresponding reference transcriptions `testref.mlf`. `HResults` will print out the analysis of the recogniser’s results as follows:
The summary of the results contains two main parts; the first part provides information about the date and time of the experiment with the names and locations of the used files, the second part provides statistics in sentence and word levels. The line starting with \texttt{SENT} gives the sentence level statistics and shows that the total number of utterances is 149, 136 (91.28\%) which were recognised correctly, while 13 sentences were substituted. The next line starting with \texttt{WORD} indicates that 734 words were tested, and the number of correctly recognised words is 718 (97.82\%). The results indicate that there were 0 deletion errors (D), 16 substitution errors (S), and 2 insertion errors (I). The \texttt{Acc} percentage of 97.55\% is lower than the \texttt{WORD Correct} percentage because it takes into account the insertion and deletion errors (if there are any) which are ignored by the latter.

With these statistics by the evaluation tool, we have a comprehensive assessment of the recogniser’s performance.
4.2.3 Data settings

It is obvious from the description given in the previous section that in order to build a recognition system and test its efficiency, a large number of commands will have to be executed and many tools must be generated. Hence, in the experiments carried out in this thesis, all the steps have been written in order as a Python script which includes the commands mentioned in the HTK manual book with some additions and modifications to overcome a few technical problems. In this script, all the tools needed to run the HTK have been generated and then
executed within the script such as perl, hed, and led scripts. The main advantage of using the script over executing each command individually is its superiority in saving time and organising the relevant processes, especially when we need to run hundreds of experiments repeatedly with different settings.

Furthermore, in order to handle the large amount of data and to manipulate the training data in different ways, we developed various programs that can help in coding the train and test data, extracting subsets of the data to be used for testing, separating the dataset in 5 folds, and other programs with a similar purpose. All of the programs are written in Python and executed prior to running the HTK.

The HTK evaluation tool outputs an abstract of the system’s performance and a file containing the recognised inputs. Because in certain cases we need to see what words (or phonemes) were most misrecognised and with what were they substituted, we have developed a program that lays out a table as a confusion matrix to provide such information. The information provided in the confusion matrix is crucial to understanding how the HTK works and what kind of improvements are needed to make it perform better. Obtaining the confusion matrix starts by creating a file that copies the information found in the recout.mlf file but in a format that is similar to the testref.mlf file which lists a word (or phoneme) level representation for the test sentences. Then, a comparison is made between the two files, and a table with three columns is given. The first column in the table lists all the language units as they appear in the recognition output file recout.mlf, the second holds a list of the reference transcription, and the third indicates the relationship between the recognised and the reference transcription. The recognised output can be the same as the reference, different, or wrongly inserted, and in some cases the reference unit is deleted from the recognition output. At the end, a summary of statistics is given, which particularly indicates the number of occurrences for each unit, correctly recognised times, deletions, and insertions. Figure 4.22 gives a snapshot of the contents of the generated confusion matrix.

In order to build phonetic transcription-based systems, we need to consult the grapheme-to-allophone system at a certain point of training the recogniser to acquire the context-sensitive phonetic transcription for the given text, and the basic script was extended for this purpose as described below.
4.2.4 Experimental scripts

The proposed research suggested using a generated phonetic transcription rather than fixed dictionaries which are normally used in typical ASR systems. In order to generate the phonetic transcription for the text files and use it in the training, several modifications have been made in the data preparation and training stages. This section introduces the changes made to these two steps and the adopted solutions to bridge the gap that might be caused by these modifications.

i. Changes made in the data preparation steps:

As previously stated, the HTK requires a number of source files in order to prepare the data and transfer it to a format that can be processed during training and testing. In addition to the prompts, a pronunciation dictionary is normally provided prior to invoking the data preparation tools. Because we want to generate a context-sensitive phonetic transcription for the training text, we cancelled executing the dictionary management tool HDMan whose
main task is to construct and edit the dictionary, and we do not provide any predefined dictionary. Instead, the grapheme-to-allophone system is consulted for the training text. The output of this step is stored in a text file called “prompts_res” as shown in Figure 4.23. This file will be used as a replacement for the dictionary in any step where the dictionary is needed. For instance, the HLEd tool which is used to create phone level MLFs transcription is cancelled because it is based on the information given in the dictionary. This cancelation is compensated for by generating the phone level MLFs directly from the “prompts_res” file. The output of this step is stored in the file phones.mlf with a list of phones extracted from the generated phonetic transcription. On top of that, we also generate a list of the phones which are normally created with the aid of HDMan tool and stored in monophones1. The contents of the generated monophones1 file are extracted directly from the “prompts_res”.

ii. Changes made during the training stage:

The training stage consists of two main steps: the first one is creating monophones HMMs while the second is concerned with creating tied state triphones. At the end of the first step, the HTK recognition tool HVite is invoked in order to transform the word level transcription stored in words.mlf to a phone level transcription aligned.mlf using the pronunciations given by the dictionary. The difference between the two phone level transcriptions phones.mlf (which have been created during the data preparation stage)
and `aligned.mlf` is that in the latter all the pronunciations provided by the dictionary are considered and the output pronunciation is the one that best matches the acoustic data. In contrast, in `phones.mlf` the first pronunciation found for the word is picked. In the case of using a generated phonetic transcription, we do not have a dictionary and we provide only one pronunciation for each word according to the context and the morphological features of the word. So, the HVite tool is cancelled and we provided a special command that creates the `aligned.mlf` file by extracting the word’s pronunciation directly from “prompts_res” bearing in mind to include short pauses at word boundaries.

### 4.3 Training and testing corpora: collection and preparation

Building an ASR system requires providing the recognition tool with wave files associated with their textual transcription. This section describes the corpora used in training and testing the speech recognition systems built throughout this thesis. Because of the difficulty of having access to shared MSA datasets and because we want to put constraints on what is going to be said, we have created our own corpora to be used in this study. We mainly used two corpora in running most of the experiments: pilot study corpus and main study corpus. The speech corpus was collected with the aid of developed web-based recording tool. Different advantages makes the use of the developed recording tool better that using the HTK recording tool HSLAB. For instance, having the recording tool on the web means that it can be reached by speaker from different places without the need of installing the HTK toolkit into their machines. In addition, the web-based tool presented one sentence at a time and the participants had to listen to their recordings when recording the first 10 sentences before it can be saved. After the 10th sentence this feature becomes optional. This feature was added for quality monitoring purpose, partly to make sure that the recordings are audible to reduce the possibility of having noisy recordings and partly to get the participants to use to the tool so they know what is the best time to start talking and partly because we know that they may make mistakes and want to self correct as normal speakers do, so we wanted to give them the chance to do that. Figure 4.24 provides a screen shot for the web-based recording tool.
4.3.1 Pilot dataset

Prior to collecting the main study corpus, we worked on collecting a small scale dataset to be used in running a number of preliminary studies. These preliminary studies are conducted for three purposes:

- To test the work of the web-based recording tool and consider any feedback or suggestions given by the participants to improve the tool.
- To test the robustness of the grapheme-to-allophone system and verify the existence of the proposed phonological variations in real speech.
- To evaluate the feasibility of using the automatically generated phonetic transcription in building Arabic ASR systems.

4.3.1.1 Text corpus

In order to carry out the pilot study experiments, we have manually created 20 sentences. The sentences are created manually mainly because we wanted to use these sentences to test and evaluate our developed grapheme-to-allophone system. Hence, we wanted to include all the phonological variation aspects with the minimum number of sentences, which is hard to achieve automatically.
Despite the shortness of the manually transcribed corpus, the following methodology was adopted to ensure the richness and comprehensiveness of the provided sentences:

- The sentences contain all Arabic phonemes.
- Every kind of phonological variation is covered in the sentences with special attention paid to the problematic phonemes such as “*hamzatu AlwaSl*” and emphatic phonemes.
- Use as few words as possible and ensure that every word has distinctive features that cannot be seen in the other words within the same dataset.
- Every word occurs only once (except prepositions).
- The sentences’ length is between 3 to 5 words and vary structurally (nominal or verbal) where verbal sentences have different moods (perfective, imperfective, and imperative moods).

A list of the created sentences is provided in Appendix A.

### 4.3.1.2 Speech corpus

29 native Arabic speakers from the major dialectal regions (Gulf, Iraqi, Egyptian, Levantine, North African) participated in reading and recording the created sentences. The recording process was completely supervised using a web-based recording tool from one machine. Recording was performed in a quiet room. Speakers were asked to read the prompted sentences naturally and re-record utterances where hesitations or mistakes were observed. After recording each sentence, we used the “play” feature provided by the recording tool to ensure that the sentence starts with a proper silence and no noise was to be found in the recording like the one caused by having mobile phones nearby. The total number of utterances collected from the 29 speakers was 2000 which accounts for about 2.3 hours of recording.

A table listing the personal details of the speakers who participated in the collection of the pilot dataset is given in Appendix B.
4.3.2 Main dataset

After testing the proposed transcription system and the recording tool and gathering information about different ways to improve matters, a large-scale data collection phase then took place. The following is a description of the text and speech corpus.

4.3.2.1 Text corpus

In the main dataset, we wanted to automatically generate Arabic sentences with the aid of PARASITE which contains well-defined morphological and grammatical constraints. The generator was fed with a number of Arabic words to be used in constructing the required sentences. We specified the domain of the vocabulary to include words that can take part in a travel query system.

Using the provided vocabulary, the generator outputs the required number of grammatical sentences. However, we found that many sentences were meaningless or even semantically wrong. Participants found the meaningless sentences hard to pronounce naturally.

This motivated us to add extra constraints to the vocabulary to only use certain vocabularies in a certain context. Semantic types were assigned to nouns and restrictions were attached to verbs’ arguments. For instance, the provided nouns were semantically sorted in the following way:

nonliving >> [place, number, meal, belongings, office, centre, documents, flight_types, directions, time, action].
airport >> [maTAr].
rest_place >> [‘SAlap’, qAEap, ‘maHa~Tap’, maTEam, findiql].
number >>[‘vAniy’, ‘sAbiE’, ‘tAsiE’, ‘xAmis’].
meal >>[‘<ifTAr’, ‘gadA’’, ‘Ea$A’’’].
belongings >> [‘amtiEap’, ‘HaqA}ib’].
tanzAnya, bagdAd, ‘xarTuwm’, ‘AldawHap’].
centre >>[‘jamArik’, ‘taftiy$', ‘AistiElAm’].
documents >> [jawAz, biTAqap, ticket].
ticket >> [‘ta*karap’].
flight_types >> [‘mutawa~jh’, ‘mutawa~qif’, ‘dawli~y’, ‘mubA$ir’,}
Furthermore, for each verb, we specified the type of the verb, the type of possible subject, how many objects it can take (if any), and each kind of object. For example, the following rule is used to describe the verb “tanAwal” (means “ate”):

```
word('tanAwal', X) :-
  X <> [t2verb(OBJ, OBJ2)],
  OBJ <> np,
  kind@OBJ << (edible),
  OBJ2 <> pp,
  kind@OBJ2 <<('rest_place'),
  kind@subject@X << (human).
```

By constraining syntax as well as meaning, the generator was consulted to output 300 generated sentences that vary in length from 2 to 6 words. The following methodology has been adopted in generating the sentences:

- The sentences are relatively short, so speakers do not have to pause for breath while recording.
- The sentences are phonetically rich and balanced in order to produce a robust Arabic speech recogniser. In this way, we selected a number of
phonetically rich words to include all Arabic phonemes. Also, we aimed at creating sentences that are phonetically balanced by covering most phoneme clusters in Arabic.

- Producing both declarative and interrogative sentences.
- Producing the two main sentence structures in Arabic (nominal and verbal sentences).
- The sentences are structurally simple in order to ease readability for the speakers.

A list of the generated sentences is given in Appendix C.

4.3.2.2 Speech corpus

After the creation of the text corpus, recording this dataset is required. Participants from all Arab regions were asked to help in collecting the speech dataset. Volunteer participants were asked to record any number of sentences but no less than 50 utterances, whilst paid participants were asked to record a minimum of 900 sentences which is equivalent to one hour of recording. Typically, in collecting speech corpora, recordings are performed in a well-equipped and quiet room using high standard recording technology. Due to the limitation in time and effort and since the aim of collecting the data is to test how different factors can affect the performance of the speech recogniser regardless of the maximum accuracy we can achieve, the data collection process was unsupervised and conducted using ordinary machines.

Our aim was to collect 30 hours of recordings, taking into consideration the following points:

- Launching the recording tool on the web to facilitate accessibility for participants.
- Using speakers from different Arab regions to cover all the main accents.
- Using speakers from different age groups (15-65).
- Using a balanced set of male and female speakers.
- Excluding speakers with major speech disorders.
Before recording the data, participants were given the following instructions to bear in mind during the recording process:

- Record in a quiet room avoiding any kind of external noises.
- Use a reasonably good quality microphone in recording.
- Make sure to pronounce the sentences with all the provided diacritics.
- Speak as naturally as possible without pausing between the words.
- Listen to each sentence after recording before pressing “save” button, if any mistake was found please record the sentence again.

Speakers were asked to register the following personal details:

- Name
- Age
- Gender
- Regional dialect

The recorded sentences have an average length of 4 seconds. By the end of the speech data collection phase, we had a total of 26,866 utterances (about 29.8 hours in duration) recorded by 69 speakers. Because speakers were unsupervised in recording the sentences and due to some technical issues, three problems were found in the recordings:

- Very noisy wave files (noise in the background or poor quality microphones).
- Sound files without any initial silence (speakers started speaking before the recorder was on).
- Damaged files.

Those problems require post-processing the speech corpus to avoid having them in training or testing the recogniser. For this reason, a noise detection algorithm was developed in order to automatically search for the previously mentioned problems within the collected recordings and exclude all the unwanted
files. We run PRAAT\(^1\) to identify unwanted files by setting the acceptable range of intensity and then excluding the files that do not follow the allowable thresholds from the dataset. This algorithm should tackle the three main problems simultaneously. For instance, noisy recordings exceed the acceptable rate of intensity, recordings with no proper silence do not start with 0 intensity as it is supposed to do, and the intensity of the damaged files is always 0.

By the end of post-processing the recorded wave files, a total of 23,060 wave files were recovered (approximately 25.6 hours in duration) from 56 native Arabic speakers.

Because the recording process in the pilot dataset was completely supervised, we did not need to carry out any post-processing procedures over the wave files. Recording in similar condition with one recording machine in a supervised manner explains the superiority of this dataset in building a speech recogniser with a high accuracy as we will see in Section 6.1.

Appendix D contains a table of the participants' personal details after filtering the speech dataset.

Despite the efforts made to collect a balanced set of genders, ages, and dialectal regions, imbalances were found in the collected dataset. For instance, the majority of the speakers were females with a total of 36 female speakers vs. 20 male speakers. Most of the participants were between 20 and 30 years old with a few speakers above or beyond this age scale. Moreover, due to the fact that I belong to the Gulf community and most of my relatives and friends are based in the Gulf area, the majority of speakers come from the Arabic Gulf region. The total number of speakers from the Gulf is 39, the other 17 speakers have Levantine, Egyptian, and Iraqi accents. The only two speakers we have from North Africa were excluded from the dataset due to the bad quality of their recordings.

\(^1\)A software meant for analysing human speech.
Chapter 5

Acquiring a context-sensitive phonetic transcription for a given MSA text

Phonetic transcription is an essential source for training the speech recogniser. A typical speech recogniser needs to have the sounds associated with their textual and phonetic transcription. For languages with complex letter-to-sound mappings (like English and French), mapping between the textual and phonetic transcription is usually done using pronunciation dictionaries. In Arabic, the relationship between the fully diacritised individual letters and their associated pronunciations is relatively deterministic and it is therefore tempting to carry out this mapping using a set of rules. Nevertheless, deriving a context-sensitive phonetic transcription for a given text is normally faced with the problem of the boundary effects on pronunciation. In this chapter, we will go through the development of a comprehensive system for grapheme-to-allophone conversion of the MSA which will be used to enhance the performance of the ASR system. The first section in this chapter describes the initial stages required before applying the phonological rules. The second section, which forms the greatest part, investigates the process of creating a substantial set of rules which can predict within-word and cross-word pronunciation variations in MSA speech. The developed system is tested and evaluated in the third section to ensure the robustness of the developed system and to verify that it captures the actual phonological processes. The fourth section introduces a survey of similar works found in the literature, and concludes by highlighting the main advantages of our hypothesised method over
the presented solutions.

5.1 The process of deriving a phonetic transcription

This section reviews the outlines of our transcription algorithm for MSA. This system utilises a set of comprehensive pronunciation rules which includes the grapheme-to-phoneme mapping and phoneme-to-phone conversion so as to produce the actual sounds of the language. This process must be preceded by preprocessing texts. This is an essential front-end for any system that deals with transcribing text. Basically, it manipulates the information from textual input and prepares the text to undergo further processing by the system. Besides, a lexicon for exceptional words, numbers, abbreviations, symbols, and acronyms is introduced. Figure 5.1 describes the architecture of the developed grapheme-to-allophone system.

![Figure 5.1: The architecture of the developed grapheme-to-allophone system.](image)

A rule-based method is introduced in this section which can be used to process a given text and generate its context-sensitive phonetic transcription. The
pronunciation rules, which are implemented as a set of two-level finite state automata, are formalised in the following way:

\[ [A] \rightarrow [B]: [X]##[Y]. \]

The rule has two main parts separated by a colon. The first part describes the conversion process by indicating what sound(s) is changed and what the sound changes to (the abstract representation of the actual sound). The second part of the rule indicates in what environment the pronunciation rule is applied. X is the backward context and Y is the forward context. The double hashes represent the location of the sound(s) that is to be changed. Square brackets are used to represent the individual sounds or features that the sounds have in common (represented in curly brackets) or null.

To illustrate the format of these rules, consider the well-known English pronunciation rule for converting the /n/ sound to /m/ whenever it precedes a bilabial consonant, e.g. the word impossible which was originally “in-possible” but because it was followed by a bilabial sound /p/, the /n/ sound has been converted to /m/.

\[ [n] \rightarrow [m]: [[??]?##[{+bilabial}]]. \]

This rule applies to all sounds that share the feature bilabial (in English /b/, /p/, and /m/). The triple inquiries [??] in the backward context means it can be anything.

Table 5.1 lists the features of the Arabic sounds as used in writing the conversion rules. Consonants are described by their place and manner of articulation, while vowels are described in terms of the position of the lips, the part of the tongue raised, and the height to which it moves.

The remaining of this section describes the task allocated to each step in the phonetic transcription extraction process.

5.1.1 Pre-processing the text

Text normalisation or pre-processing includes the following:

- Restoring the diacritics of the text, as the absence of the diacritics means
Table 5.1: The Arabic sound’s features as used in writing the pronunciation rules.

<table>
<thead>
<tr>
<th>Class</th>
<th>Sound</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consonants</td>
<td>/ʔ/</td>
<td>glottal stop</td>
</tr>
<tr>
<td></td>
<td>/b/</td>
<td>bilabial stop</td>
</tr>
<tr>
<td></td>
<td>/t/</td>
<td>alveo-dental stop</td>
</tr>
<tr>
<td></td>
<td>/θ/</td>
<td>dental fricative</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>post-alveolar fricative</td>
</tr>
<tr>
<td></td>
<td>/h/</td>
<td>pharyngeal fricative</td>
</tr>
<tr>
<td></td>
<td>/x/</td>
<td>uvular fricative</td>
</tr>
<tr>
<td></td>
<td>/d/</td>
<td>alveo-dental stop</td>
</tr>
<tr>
<td></td>
<td>/ð/</td>
<td>dental fricative</td>
</tr>
<tr>
<td></td>
<td>/r/</td>
<td>alveo-dental trill</td>
</tr>
<tr>
<td></td>
<td>/z/</td>
<td>alveo-dental fricative</td>
</tr>
<tr>
<td></td>
<td>/s/</td>
<td>alveo-dental fricative</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>post-alveolar fricative</td>
</tr>
<tr>
<td></td>
<td>/s/</td>
<td>alveo-dental fricative</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>alveo-dental fricative (emphatic)</td>
</tr>
<tr>
<td></td>
<td>/d/</td>
<td>alveo-dental stop (emphatic)</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>alveo-dental fricative (emphatic)</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>dental fricative (emphatic)</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>pharyngeal fricative</td>
</tr>
<tr>
<td></td>
<td>/ʃ/</td>
<td>uvular fricative</td>
</tr>
<tr>
<td></td>
<td>/f/</td>
<td>labio-dental fricative</td>
</tr>
<tr>
<td></td>
<td>/q/</td>
<td>uvular stop</td>
</tr>
<tr>
<td></td>
<td>/k/</td>
<td>velar stop</td>
</tr>
<tr>
<td></td>
<td>/l/</td>
<td>alveo-dental lateral</td>
</tr>
<tr>
<td></td>
<td>/m/</td>
<td>bilabial stop</td>
</tr>
<tr>
<td></td>
<td>/n/</td>
<td>alveo-dental stop</td>
</tr>
<tr>
<td></td>
<td>/h/</td>
<td>glottal fricative</td>
</tr>
<tr>
<td></td>
<td>/w/</td>
<td>bilabial approximant</td>
</tr>
<tr>
<td></td>
<td>/j/</td>
<td>palatal approximant</td>
</tr>
<tr>
<td>Vowels</td>
<td>/a/, /a:/</td>
<td>unrounded, open, front</td>
</tr>
<tr>
<td></td>
<td>/u/, /u:/</td>
<td>rounded, close, back</td>
</tr>
<tr>
<td></td>
<td>/i/, /i:/</td>
<td>unrounded, closed, front</td>
</tr>
</tbody>
</table>
losing a large amount of phonetic information, which leads to a considerable ambiguity when processing the text.

- Delete unnecessary symbols like (sukon ـ). The function of this symbol is to indicate that the consonant to which it is attached is not followed by a vowel. Because this symbol does not correspond to any sound it can be deleted.

- Text segmentation into words, syllables, and phoneme boundaries. This process is crucial to enhancing the quality of the conversion rules since many rules are restricted to special boundaries. For instance, the insertion of the short vowel /i/ only occurs between two words when the first one ends with a consonant and the second one starts with “hamzatu Alwast”. Furthermore, the syllable plays an essential role in writing the rules as many phonetic phenomena seen in Arabic are syllable-internal, e.g. the spread of pharyngealisation and the pronunciation of the semi-consonants (w) و and (y) ي, which is dependent on their position within the syllable.

- Exceptional words or words of irregular spellings. Most languages have words that do not follow the general rules for pronunciation. In English, for instance, some particular words have silent letters that do not correspond to any sound in the word’s pronunciation, such as (w) in answer, (s) in island, and (t) in ballet. In the case of Arabic, there are silent letters and missing letters. The silent letters have just been described, the missing letters are the sounds that are not included in the word but are pronounced, for example, the sound /a:/ in words like (hُA) هَنَا /ha:*a:/ or (ka*lik) كِذِيْكَ /ka*alik/. These types of words do not follow a specific pronunciation rule. For such cases, there should be a set of rules that associate the graphemic form of the exceptional words to their phonemic form.

- Remove the letters that do not correspond to any sound in the word’s pronunciation. Such letters may appear in certain words, in which case the best way to deal with them is to introduce and map them with their corresponding pronunciation in the lexicon. However, they may appear in special contexts like the silent “Alif” א which follows “the group waw” وَأُوِلَى indicating masculine plural morpheme in verbs. For example: the verb (katabuwA) كَتَبَوْا “they wrote” is pronounced /katabu:/.

Another silent
“Alif” appears after the “fatH nunation”. This step also includes removing short vowels that precede a long vowel. These short vowels usually appear in automatically diacritised texts before their long vowel counterparts like (taquwul) /taqu:1/ “she said” and (turiyd) /turixd/ “she wants”. Because these short vowels do not have any phonetic function, they need to be removed from the text as they are merely used in the text to help distinguish between cases where semi-consonants are realised as consonants and ones where they are realised as vowels.

• Remove the gemination marker (shadda) and duplicate the previous consonant. Whenever (shadda) appears after a character, that character is geminated. For example, the word (say~id) master is pronounced /sayyid/.

5.1.2 Grapheme-to-phoneme rules
This section is concerned about the sound-spelling correspondences. The outcome of this step will be used as an input to the phonological analysis. Generally speaking, sound-spelling correspondence is a straightforward process in Arabic with one way of pronunciation for each grapheme. However, certain irregularities can be noticed with some graphemes in Arabic that have multiple pronunciations. Three of them are the semi-consonants جرفُ أَلفَة namely: “Alif” (A) \, “wAw” (w) \, and “yA’” (y). The fourth one is the feminine marker “ta’ marbuTa” ﺖاءَ مَرْبوطَة. Besides, some graphemes may appear in the text but be deleted during the pronunciation and some phonemes are pronounced although they do not have an orthographic representation. These issues will be investigated in this section before going through the process of direct one-to-one grapheme-to-phoneme mapping.

• The pronunciation of “wAw” (w) \ and “yA’” (y)

The Arabic writing system has one grapheme to represent two different sounds from various phonetic classes. There is (w) \ to present both the long vowel /u:/ and the consonant /w/, and there is also (y) \ to present the long vowel /i:/ and the consonant /j/ at the same time. This variety in the phonetic
representation of the two mentioned letters causes a great deal of confusability in analysing the language, since it affects not only the pronunciation of the word, but also the syllabic structure of the word as the two representations come from different phonetic classes. Consider the examples: (yadEu) يذَعُو “he prays” which is pronounced /jadˈuː/ and the dual form of the same verb (yadEuwAn) يذَعُوْانَ which is pronounced /jadˈuːwən/. The (w) character is pronounced as a long vowel in the first example and as a consonant in the second example. To give an example of the (y) character, consider the form: (yarmy) يُزَمَي “he throws” pronounced /jarmiː/, and compare it with the dual form of the same verb (yarmiyAn) يُزَمَيْانَ which is pronounced /jarmiːjən/.

Many researchers have recognised this problem and tried to offer solutions. The most widely used method is to identify the sounds of (w) and (y) by looking at the preceding vowel; if (w) or (y) is preceded by its counterpart short vowel /u/ or /i/, it will then be pronounced as a long vowel. In contrast, they will be pronounced as a consonant when preceded by “fatHa” /a/ [SC99] [Wes03]. We argue that the presented solution is not based on a real understanding of the Arabic phonetic and phonological system for two reasons. (i) By saying that a long vowel (w or y) is preceded by a short vowel, we are violating a fundamental phonetic rule that does not allow a vowel cluster in a single syllable. (ii) A quick glance over Arabic texts would show that this rule does not always apply. For instance, despite the fact that (w) and (y) in the words: (yarjuwAn) يَرَجُوْانَ “they hope” and (yabniyAn) يَبْنُيْانَ “they construct” both have a short vowel predecessor (/u/ and /i/), they are pronounced as consonants.

The best way then to determine whether it is a consonant or vowel is to look at the context in which the (y) or (w) occurred. The Arabic phonetic system, for instance, does not allow a syllable to start with a vowel and does not allow for adjacent consonants within the same syllable (except when it occurs in the final syllable of the word). Therefore, the (w) and (y) are considered vowels in two cases: when they are preceded and followed by consonants and when
they come at the end of the word and are preceded by a consonant. Otherwise, they are consonants. Figure 5.2 lists the rules which are designed to assign the phonetic class to semi-consonants:

```
fixSC([_]).

fixSC([S | T]):-
    +final:pos@word:char@S,
    !,
    fixSC(T).

fixSC([CO, Cl, V | L0]):-
    CO <> consonant,
    Cl <> consonant,
    V <> vowel,
    fixSC([Cl, V | L0]).

fixSC([C, V | L0]):-
    C <> consonant,
    V <> vowel,
    !,
    fixSC([V | L0]).

fixSC([V, C | L0]):-
    C <> consonant,
    V <> vowel,
    !,
    fixSC([C | L0]).

fixSC([C | L0]):-
    C <> consonant,
    fixSC(L0).
```

Figure 5.2: Rules for labelling the characters with consonant and vowel value.
The pronunciation of “Alif” (A) 

The “Alif” letter can be pronounced as “hamza” /ʔ/, or long “Alif” /aː/. It is pronounced as “hamza” when it comes after a pause or at the beginning of an utterance, i.e. it is either a part of the definite article (Al) ال as in (A lmadr asap) المدرسة “the school” or a verb or noun initial as in (A*hab) اذهب “go” and (AntiSAr) انطلاق “a victory”. The mentioned examples are pronounced as follows: /ʔalmadr asa/, /ʔ*hab/, and /ʔintiSa:/ respectively. Otherwise, it will follow “hamzatu AlwaSl” rules which will be covered in the following section. On the other hand, the (A) letter is pronounced as the long vowel /aː/ if the above conditions do not apply. In other words, when it occurs in words medially or finally like: (kitAbuHumA) كيتابهم “their book” which is pronounced /kitAbuhuma:/.

The rules developed to cover the phonetic variants described above are:

```
["A", +initial@utterance]] ===
["", ""]
[???] ## [c0, c1, ??]].

["A"] ===
[aa]:
[???] ## [??]].
```

According to these rules, the ‘A’ is pronounced as “hamza” whenever it occurs initially in the utterance and pronounced as long vowel /aː/ when occurs word’s medially or finally. The notation (c0) means any sound that has the feature (+consonant).

The pronunciation of the feminine marker “la’ marbuTa”

In order to form a feminine noun from the masculine in Arabic, a morpheme called “la’ marbuTa” is added to the word’s final syllable and looks like (ٍ) or (٠) depending on the type of character to which it is connected. The
feminine marker in Arabic has two different pronunciations; it is silent when not followed by a vowel, otherwise, it is pronounced as /t/. The following phonological rule explains this transformation:

\[
[p] \Rightarrow [t];
\]

\[
[\text{????}] \text{## } [\text{v0}].
\]

\[
[p] \Rightarrow [ ];
\]

\[
[\text{????}] \text{## } [ ].
\]

The first part of the rule says that the feminine marker (p) is pronounced as (t) when followed by a vowel and is deleted when it occurs at the end of the utterance.

- Deletion and insertion

The deletion phenomenon, sometimes referred to as “elision”, is mainly about deleting graphemes where the pronounced form has no corresponding sound. Insertion, sometimes referred to as “epenthesis”, is the process of inserting a phonetic element into a string without having an orthographic representation. The main deletion process that can be observed in MSA is the deletion of “hamzatu AlwaSl”. Arabic phonetic structure does not allow the syllable to begin with two consonants (CCV). However, because Arabic is a morphologically driven language, some words may begin with a two-consonant cluster. Therefore, “hamzatu AlwaSl” is added as an initial hamza to many morphological forms to avoid violating this dominant phonetic rule. The pronunciation of “hamzatu AlwaSl” is totally dependent on the context. For instance, it is not pronounced when it is preceded by a vowel because if that happens the first consonant after “hamzatu AlwaSl” will be the coda of the last syllable of the preceding word and in such cases the “beginning with two consonants” restriction does not apply. For example: a phrase like (>uHib~~u AibtisAmatak) أَحْبَيْتُكَ أَيُّهُمَا مَانِدَةُ “I love your smile” is pronounced as /?uhibu btis:matak/. The syllabification analysis for this phrase is presented in Figure 5.3. It is worth remembering here that while “hamzatu Alwastl” can be found in Arabic nouns
and verbs, it can also be a part of the definite article (Al) আল.

Figure 5.3: Syllabification of the phrase (‘uHib close u AibtisAmatak).

Insertion is another phenomenon that can be seen in Arabic. The general rule in pronouncing two successive words in MSA when the last sound of the first word is a consonant and the first sound of the second word is “hamzatu AlwaSl” is to introduce a short vowel /i/ after the first word and to delete the “hamzatu AlwaSl” accordingly as discussed previously. This can be seen in phrases like man AlqAdim مَنْ الْقَادِمَ /manil qa:dim/ “who is coming?”. The short vowel /i/ is inserted in line with the basic phonetic rule in MSA that does not allow having two successive consonants except at the end of the utterance. After inserting the short vowel, the hamza deletion process takes place.

The rules below express the short vowel insertion process before “hamzatu AlwaSl” and the process of deleting the phonetic representation of the letter A:

```plaintext
{"A", +initial@word}] =>
[i, "A"],

[???, c0] # [c1, c2, v1, ??].
```
A short vowel /i/ is inserted before the letter A according to the first rule. The second rule says that "hamzatu AlwaSl" which occurs at the start of the word, must be deleted whenever it is preceded by a vowel and followed by two adjacent consonants. Also, the vowel that preceded it must be shortened if it was long. For example, fy Almadrasap في المدرسية “in the school” is pronounced as /fi Imadrasa/.

Another case where the insertion takes place is when “hamzatu AlwaSl” is in the position (+initial) in the utterance. In this case, “hamzatu AlwaSl” will be pronounced as a glottal stop, as we discussed previously, which needs a following vowel. Two kinds of short vowels may be inserted depending on the type of the first vowel in the word. If it has the feature (-rounded), an /i/ is inserted e.g. the word AinfataH أفتتح “opened” is pronounced /?infataH/ when it comes at the beginning of the utterance. If the first vowel has the feature (+rounded), then a short /u/ is inserted e.g. At~uhim أتهم “he has been accused” is pronounced /?uttuhima/.

The process of mapping grapheme-to-phoneme also includes a set of one-to-one rules that inspect the graphemes and convert them into phonemes. This type
of rule is straightforward; the task of these rules is to map the Arabic letters to their matching phonemes.

All Arabic letters have only one form except “hamza” (glottal stop) \( \text{HAMZA} \), “alif” \( /a:/ \), and the long “alif” \( /a:/ \). “hamza” has multiple forms depending primarily on its vocalic context: \( \text{HAMZA} \), \( \text{HAMZA} \), \( \text{HAMZA} \), \( \text{HAMZA} \), \( \text{HAMZA} \). There are special orthographic rules for using each form. For example, \( \text{HAMZA} \) is used when “hamza” is followed by the short vowel \( /i/ \), while “hamza” form \( \text{HAMZA} \) is used in a word medially and finally when preceded or followed by an \( /u/ \) vowel. Similarly, the long “alif” has three different forms in Arabic. Besides the main form \( \text{ALIF} \), it can be presented as a diacritic in the middle of words (dagger “alif” \( \text{ALIF} \) or at the end \( \text{ALIF} \)). These multiple forms of writing “hamza” and long “alif” do not affect how they are pronounced, so they are not a problem for deriving pronunciation transcription.

We have provided a set of rules for grapheme-to-phoneme mapping as an initial step before applying the phonological rules (Figure 5.4). In the provided rules, we use the Buckwalter transliteration scheme and an extended version of the Speech Assessment Methods Phonetic Alphabet (SAMPA) phonetisation scheme to present the letters and sounds of the language, respectively. The reason we use the SAMPA notation rather than the IPA in writing the program is because the IPA is not an ASCII-friendly scheme. There are a number of alternative ASCII-oriented versions of the IPA. We use SAMPA in particular because in other parts of our work we use the natural language toolkit (NLTK), which makes use of SAMPA.

These rules convert the listed graphemes into another representation form. Any grapheme not mentioned should remain the same.

5.1.3 Lexicon of the exceptional words

Most languages have words that do not follow the general rules for pronunciation. In Arabic, for instance, there are silent letters and missing letters. The silent letters can be seen in the example (Emrw) \( مرم \) where the \( (w) \) is silent and the word is pronounced as \( /γαμρ/ \). The missing letters are the sounds that are not included in the word but are pronounced. For example, the sound \( /u:/ \) in the
Figure 5.4: Grapheme-to-phoneme conversion rules.

words like (dAwd) /da:wud/ or (TAwus) /tʰa:wus/. These types of words do not follow a specific pronunciation rule. For such cases, there needs to be a dictionary that associates the graphemic form of the exceptional words to their phonemic form.

This also includes conversions of numbers, symbols, acronyms, abbreviations, and non-alphanumeric characters into appropriate word or phrase descriptors. Researchers confirm that these kinds of words are a major source of errors in phonetic transcription [SC99]. Irregular forms of this kind must be identified prior to executing the pronunciation rules.
5.2 Phonological rules

This section describes the development of a set of phonological rules that convert phonemes into phones. These rules are intended to control the allophonic variation. In other words, these are context-sensitive rules that operate on phonemes to convert them to phones in specific contexts. In writing these rules we used common Arabic pronunciation rules along with some of the tajweed rules (the tradition of the Holy Qur’an’s recitation) which found to comply with the MSA speech, and a number of novel rules obtained by analysing spoken Arabic by speakers from different regions.

These rules are ordered to ensure that the most appropriate rule is chosen first. Only the first applicable rule is used. The system works by scanning the word letter by letter and whenever the conditions for applying the rule are met then a conversion takes place.

The phonological rules introduced in this section include assimilation (total and partial), neutralisation, and emphasis.

5.2.1 Assimilation

Assimilation refers to the influence of one sound upon the articulation of another so that the two sounds become alike if not identical [Cry03]. From this prospective, the change in sound may either be total or partial. The assimilation is total in the English phrase *ten men* /tem men/, where the sound /n/ is influenced by the succeeding sound /m/ so that they become identical. The assimilation is partial in *ten books* /tem boks/ where the sound /n/ is influenced by the following sound /b/ and has adopted its bilabiality, but not its plosiveness. This process can also be described depending on the direction of assimilation; it is progressive when the phoneme is changed to match the preceding phoneme, and regressive when the phoneme is changed to match the following phoneme.

The prime cause of assimilation is phonological conditioning and smoothing, so that “assimilations are not compulsory in many languages, including English. A speaker may, if he chooses, avoid making them” [Abe67]. Arabic presents different cases of total and partial assimilation that can either be obligatory or optional. Assimilation can be observed in spontaneous speech. However, in careful speech the assimilation may not occur as by speaking slowly a short pause will separate the consonants from close points of articulation. This section will introduce the
Table 5.2: Solar and lunar letters.

<table>
<thead>
<tr>
<th>solar consonants</th>
<th>lunar consonants</th>
</tr>
</thead>
<tbody>
<tr>
<td>/t/, /ð/, /ð/, /θ/, /ð/</td>
<td>/ب/ , /ث/ , /ج/ , /ح/ , /ح/</td>
</tr>
<tr>
<td>/r/, /ð/, /ð/, /θ/, /ð/</td>
<td>/غ/ , /خ/ , /خ/</td>
</tr>
<tr>
<td>/s/, /ð/, /ð/, /θ/, /ð/</td>
<td>/غ/ , /غ/ , /خ/ , /خ/</td>
</tr>
<tr>
<td>/t/, /ð/, /ð/, /θ/, /ð/</td>
<td>/غ/ , /غ/ , /خ/ , /خ/</td>
</tr>
<tr>
<td>/l/, /l/, /l/, /l/</td>
<td>/م/ , /م/ , /م/ , /م/</td>
</tr>
</tbody>
</table>

aspects of assimilation in MSA, providing phonological rules that control this kind of variation.

5.2.1.1 Total assimilation in MSA

Speakers tend to assimilate certain sounds where the places of articulation are close together. This process is aimed at making it easier to pronounce since the transition from one consonant to another with a place of articulation nearby is awkward. This section will address: (i) the assimilation of the “lAm” consonant حرف الة in the definite article (Al) ال; (ii) the assimilation of the nasal consonant /n/; (iii) the assimilation of the stop alveolar consonants /t/ and /d/; (iv) the assimilation of the dental consonants /θ/, /ð/, and /ð/.

i. The total regressive assimilation of /l/

The major case of total assimilation in Arabic is that of the “lAm” consonant /ل/ of the definite article (Al) ال. The lateral of the Arabic definite prefix has different phonetic realisations depending on the type of initial consonant of the noun to which it is prefixed. The “lAm” consonant, for instance, can be totally assimilated when followed by any one of a group consisting of fourteen consonants called “the solar letters” الأحرف الشمسية. On the other hand, the pronunciation of the “lAm” is not affected when attached to a word which begins with a consonant of a group called “the lunar letters” الأحرف القمرية (both listed in Table 5.2).

For example, consider the word (Al$ams) الشمس /؟افس/ “the sun”. The
The definite article /ʔal/ becomes /ʔaʃ/ when it occurs in combination with ش /ʃ/. More examples of the “lAm” assimilation to the successor sound can be found in Table 5.3.

![Arabic consonants' point of articulation.](image)

To imagine the impact that the “lAm” assimilation has on the speech processing applications, Figures 5.7 and 5.6 show a spectrogram of the word (Al$ams) when the “lAm” is assimilated to the adjacent consonant /ʔaʃams/ and when mistakenly pronounced without assimilation /ʔalʃams/.

This phenomenon is described in Arabic resources as a morphological inflection process since it is concerned with prefixing the definite article (Al) to nouns in order to get the definite forms and it does not affect any /l/+coronal sequence [KFDA01]. However, it can be explained phonetically to the close articulation area of /l/ and the solar letters as they are all originate from between the teeth to the lower part of the palate [Aur02]. The following rule is used to specify the solar letters:

```
solarLetters(X):-
    X <> consonant,
    (X <> [place=alveo(_)] or
    X <> [interfric]).
```

After specifying the solar letters, a conversion rule is provided so as to convert the orthographic form of the definite article (Al) to its pronunciation form.
Table 5.3: Examples of words with solar letters.

<table>
<thead>
<tr>
<th>Lexical form (input)</th>
<th>Pronunciation form (output)</th>
<th>Arabic word + Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altamr</td>
<td>/?attamr/</td>
<td>النمر “the dates”</td>
</tr>
<tr>
<td>Alvawb</td>
<td>/?a00awb/</td>
<td>الثوب “the dress”</td>
</tr>
<tr>
<td>Aldarb</td>
<td>/?addarb/</td>
<td>الدرب “the path”</td>
</tr>
<tr>
<td>Al*urap</td>
<td>/?a00ura/</td>
<td>النورة “the corn”</td>
</tr>
<tr>
<td>Alraswl</td>
<td>/?arraswI/</td>
<td>الرسول “the messenger”</td>
</tr>
<tr>
<td>AlzarAfap</td>
<td>/?azzara?a/</td>
<td>الزرافة “the giraffe”</td>
</tr>
<tr>
<td>Alsabt</td>
<td>/?assabt/</td>
<td>السبت “Saturday”</td>
</tr>
<tr>
<td>Al$ams</td>
<td>/?affams/</td>
<td>الشمس “the sun”</td>
</tr>
<tr>
<td>AlSadyq</td>
<td>/?as?$adi?q/</td>
<td>الصديق “the friend”</td>
</tr>
<tr>
<td>ALDaw’</td>
<td>/?d’d?aw’/</td>
<td>الضوء “the light”</td>
</tr>
<tr>
<td>AITawilap</td>
<td>/?at?$awila/</td>
<td>الطاولة “the table”</td>
</tr>
<tr>
<td>AlZarf</td>
<td>/?a00arf/</td>
<td>الطرف “the envelop”</td>
</tr>
<tr>
<td>Allawn</td>
<td>/?allawn/</td>
<td>اللون “the color”</td>
</tr>
<tr>
<td>AlnAr</td>
<td>/?anna:r/</td>
<td>النار “the fire”</td>
</tr>
</tbody>
</table>

The first of these rules says that whenever /l/ is preceded by ‘A’ and that ‘A’ is the first sound in the word, the /l/ should be assimilated with the following solar consonant. The notation (c0) means any sound that has the feature (+consonant). The same process is applied in the second rule except for a change in the backward context to cover the case where the prepositional (li) introduces the defined word.

ii. The total regressive assimilation of /n/

An Arabic speaking person would find it difficult to utter a phrase like (man ra’aY) من رأى “who saw” or (mín liqA’) من لقاء “from meeting” without
assimilating the sound /n/ with the liquid lateral /l/ or the liquid trill /r/, to be pronounced in this way /marraʔa:/ and /millīqaʔa:/.

The /n/ shares with the liquid consonants the same place of articulation since they are all alveo-dental consonants.

The /n/ consonant can also be totally assimilated when followed by /m/ which shares nasality with /n/. Take for example: (ṮAlibN muhaʔ*ab) ئاليب مهدب “a polite student” which is pronounced /ṯ̂alibummuhadabaːb/.

This type of assimilation can only occur in this order. In other words, the /n/ can be assimilated with the /l/, /r/, and /m/ but not vice versa. The
assimilation takes place either within the word or at a word boundary. Nasal assimilation is mandatory in Arabic if it occurs within a word, but optional between two words [Wat07].

The following rule expresses this type of assimilation:

\[
\begin{align*}
\text{[n]} & \Rightarrow \\
\{c0, \text{feature=Y}\} : \\
\{??\} & \Rightarrow \{c0, \text{feature=Y}, \text{??}\}
\end{align*}
\]

if (Y=nasal or Y=liquid).

By applying this rule, the /n/ consonant is assimilated with the following consonant if this consonant belongs to one of two categories (nasal or liquid).

iii. The total regressive assimilation of /t/ and /d/

The /t/ consonant is assimilated when followed by (\(\text{[t]} /\theta/, \text{ج}/\delta/, \text{د}/\delta/, \\
\text{ذ}/\delta/\text{ز}/\varepsilon/, \text{ص}/\delta\varepsilon/, \text{ض}/\text{ت}/\delta\varepsilon/, \text{ط}/\delta\varepsilon/, \text{ظ}/\delta\varepsilon/). These consonants are articulated at close articulatory points. In the same manner, the /d/ consonant can be assimilated when followed by the mentioned consonants (with /t/ instead of /d/). This assimilation takes place at word and phrase boundaries. This type of assimilation is stated in tajweed rules. However, according to tajweed rules, the /t/ is totally assimilated only when followed by /d/ and /t\varepsilon/ whilst the /d/ is totally assimilated only when followed by /t/ [Cse00] [Sid03]. This assimilation takes place at word and phrase boundaries. Table 5.4 shows examples of this type of assimilation:

\[
\begin{align*}
\text{[c0, alveodental_stop, -emphatic]} & \Rightarrow \\
\{c1, \text{feature=Y}\} : \\
\{??\} & \Rightarrow \{c1, \text{feature=Y}, \text{??}\}
\end{align*}
\]

if (Y=dental_fricative or Y=alveodental_stop).

The rule presented here is applied to any consonant that has the features alveo-stop and non-emphatic to assimilate it with the following consonant which has the feature dental-fricative or alveo-stop regardless of the preceding letter.

iv. The total regressive assimilation of the dental consonants /\theta/, /\delta/, and /\delta\varepsilon/
Table 5.4: Examples of assimilating the sounds /t/ and /d/.

<table>
<thead>
<tr>
<th>Assimilated sound</th>
<th>Text</th>
<th>Pronunciation</th>
<th>In English</th>
</tr>
</thead>
<tbody>
<tr>
<td>ت /t/</td>
<td>(At$tarat vaubaF)</td>
<td>/ʔiftaraʔawban/</td>
<td>“she bought a dress”</td>
</tr>
<tr>
<td></td>
<td>(&gt;axa*at daftar)</td>
<td>/ʔaxaad daftar/</td>
<td>“she took the notebook”</td>
</tr>
<tr>
<td>د /d/</td>
<td>(Eudt)</td>
<td>/ʃutt/</td>
<td>“I came back”</td>
</tr>
<tr>
<td></td>
<td>(lam yazid ZarfAF)</td>
<td>/lam jaʃjoðfarfan/</td>
<td>“he did not find an envelope”</td>
</tr>
</tbody>
</table>

There are three dental fricative consonants in the Arabic phonetic system, namely: ت /t/, د /d/, and ط /ð/. These consonants are normally assimilated when preceded or followed by a consonant from the same point of articulation. Table 5.5 gives examples of this process, subdivided on the basis of the type of assimilated consonant.

Table 5.5: Examples of assimilating the dental sounds.

<table>
<thead>
<tr>
<th>Assimilated sound</th>
<th>Text</th>
<th>Pronunciation</th>
<th>In English</th>
</tr>
</thead>
<tbody>
<tr>
<td>د /ð/</td>
<td>(xu* vawbak)</td>
<td>/ xuθawbak/</td>
<td>“take your dress”</td>
</tr>
<tr>
<td></td>
<td>(junqi* ZabyAF)</td>
<td>/junqiθabyan/</td>
<td>“he saves a deer”</td>
</tr>
<tr>
<td>ت /θ/</td>
<td>(yabHav *lik)</td>
<td>/jabaθalik/</td>
<td>“search about this”</td>
</tr>
<tr>
<td></td>
<td>(yaHdwv ZuhrAF)</td>
<td>/jahduðuhran/</td>
<td>“occurs at noon”</td>
</tr>
<tr>
<td>ط /ð/</td>
<td>(tastayqiZ *ikrY)</td>
<td>/tastajjo ikra:/</td>
<td>“Thikra wakes up”</td>
</tr>
<tr>
<td></td>
<td>(yat~aEiZ *Amir)</td>
<td>/jattaθamir/</td>
<td>“Thamir learns”</td>
</tr>
</tbody>
</table>

The rule for assimilating dental consonants has the following form:
5.2.1.2 Partial assimilation

Partial assimilation is a linguistic process by which a sound becomes similar to the adjacent sound but not identical. MSA presents two major partial assimilation phenomena, namely: the nasal assimilation and the spread of pharyngealisation.

i. The nasal assimilation

MSA has two nasal consonants: one alveo-dental nasal consonant /n/ and one bilabial nasal consonant /m/. Different kinds of assimilation can be observed in the MSA for the /n/ consonant. Total assimilation was discussed earlier in terms of assimilating /n/ with /l/, /r/, and /m/, and partial assimilation where the /n/ is converted to another sound that shares some features with the adjacent consonant.

The /n/ is assimilated to /m/ when followed by the bilabial consonant /b/ such as the Arabic word (منبر) "platform" which is pronounced /mimbar/. The /m/ consonant shares the nasality with /n/ and the bilabiality with /b/.

Moreover, the /n/ consonant adopts the labiality of the consonant /f/ and is assimilated to /m/ every time it is followed by /f/ either within the same word or at word boundary. For example, the word (يُمْتَفِقْ "to run out" is pronounced /yanfadh/. Assimilating the /n/ consonant when followed by /f/ has been introduced in the tajweed references which state that the /n/ should be hidden while keeping its nasalisation whenever followed by /f/ [Cse00] [Sid03]. However, by analysing the native speakers’ speech and by comparing this with other languages like English, we believe that the /n/
consonant is assimilated into /ŋ/ rather than being hidden. Take as an example the English word *infer* /ɪnˈfɜr/.

The /n/ consonant also adopts the velarity of the sounds َل /k/ and ق /q/ when followed by one of them. This can be seen in words like: *(bank)* /bɑŋk/ and *(munqAd)* /muŋqad/ “submissive”.

Furthermore, the /n/ is assimilated to the palatal nasal consonant /ɲ/ when followed by one of the sounds articulated from the dental and post-alveolar points, both fricatives and stops, namely: ث /θ/, ذ /ð/, ظ /tʃ/, ط /tʃ/, ض /dʒ/, د /d/, س /s/, ز /z/, ح /ʃ/, and خ /ʃ/. Take these examples: *(manvur)* /maŋµur/ “scattered”, *(manZar)* /maŋɔ zar/ “view”, and *(yantabih)* /yaŋtabih/ “to pay attention”.

An interesting point can be observed here; these consonants are exactly what we have seen in the solar letter group (if we added the /ŋ/ consonant) excluding /l/ and /r/ which are totally assimilated with /n/. This might be due to the fact that the consonant /n/ is the closest sound to /l/ in terms of place of articulation. This proximity in the two mentioned consonants makes them share certain changes when they occur in close conditions. However, this process is optional and varies in use among Arabic speakers from different dialects.

Partial nasal assimilation involves regressive transfer of features within a consonant cluster. So, assimilation occurs whenever /n/ is followed by one of the mentioned set of consonants. No assimilation occurs in the case of mirror-image clusters. The following rules are introduced to account for these types of assimilation: ¹

(a) [n] => [”M”]: [???] # [f, ??].

¹In these rules we used the symbols /M/, /c/, and /e/ to stand for the IPA phones /ŋ/, /ŋ/, and /n/, respectively.
(b) \[n]\Rightarrow [c]:

\[
\text{[[c0, grapheme=X], ???]} \quad \text{if } (X=q \text{ or } X=k).
\]

(c) \[n]\Rightarrow [e]:

\[
\text{[[c0, feature=Y], ???]} \quad \text{if } (Y=\text{dental_fricative} \\
\text{or } Y=\text{alveodental_stop} \\
\text{or } Y=\text{alveodental_fricative} \\
\text{or } Y=\text{post_alveolar}).
\]

ii. The spread of pharyngealisation

The Arabic phonetic system possesses pharyngealised coronal consonants called emphatics, namely (\(/s^\emptyset\), \(/d^\emptyset\), \(/t^\emptyset\), \(/\theta^\emptyset\)). These consonants are produced with a primary articulation at the dental/alveolar region and with a secondary articulation that involves the constriction of the upper pharynx" [Dav95]. It has been found that where one of these emphatic consonants occurs in a word, emphasis (pharyngealisation) may spread to the neighbouring sounds in the same syllable or word [AO09]. By this process, the influenced sounds (both consonants and vowels) tend to be pronounced much like the emphatics. Two questions arise here; what is the domain of the pharyngealisation? In other words, how far is the spread likely to go? The second question is: in which direction does the process work? backwards or forwards?

Different studies have tried to address these questions [Dav95] [Wat99]. They all emphasised that the domain of pharyngealisation spread and the directions of this spread vary greatly from dialect to dialect. However, since this process is quite complicated, we will need to draw the frame of this phenomenon mainly by analysing native speakers' speech. By doing so, we would argue that the emphatic consonants affect mainly the vowels that occur in its syllable. So, the vowel is pharyngealised when preceded by an emphatic consonant (the onset of the syllable) and when followed by an emphatic consonant within the same syllable. e.g. (SA'di) صادق
Compare the vowel pronunciation in the words (sAriq) /saːriːq/ “thief” and (SAdiq) /sˤɑːdiːq/ “honest”. It can be seen in the spectrogram given in Figure 5.8 and 5.9 that the formants are clearer in 5.8 than in 5.9, marking the fact that the vocal tract is more open and therefore there is more resonance.

Figure 5.8: The spectrogram of the word sAriq (the long vowel /aː/ is highlighted).

Figure 5.9: The spectrogram of the word SAdiq (the long vowel /aː/ is highlighted).

The vowels pharyngealisation rules take the form:

[v0] =>
[{{v0, +pharyngealised}}]:
[???, {c0, +emphatic, +coda@syllable}, ??].

[v0] =>
[{{v0, +pharyngealised}}]:
[???, {c0, +emphatic, +onset@syllable}] # # [??].

133
According to these rules, the vowel is pharyngealised whenever it is followed by an emphatic consonant and that emphatic has the position coda in the syllable, or preceded by an emphatic consonant which is the onset of the syllable.

The emphatic consonants may also affect the neighbouring consonants. This can be seen in Arabic when the consonant س/s/ is replaced by its emphatic counterpart س/sˤ/ when neighbouring the emphatic consonant ط/tˤ/. This conversion takes place in two cases:

(a) When the consonant directly following /s/ is /tˤ/, either separated by a vowel like (sATiE) ساطع or not like (yasTabir) يسطبر.

(b) When the /s/ and /tˤ/ are separated by a consonant which must be any consonant except /t/, e.g. (musayTir) مسيطر and (sawT) سوط.

This is a regressive assimilation that occurs only within the word. The rules covering this kind of assimilation are:

\[
\begin{align*}
[s] & \Rightarrow ["S"] \\
[???] & \Rightarrow [(v0), "T", ??].
\end{align*}
\]

\[
\begin{align*}
[s] & \Rightarrow ["S"] \\
[???] & \Rightarrow [v0, \{c0, \text{grapheme} = t\}, "T", ??].
\end{align*}
\]

\[
\begin{align*}
[s] & \Rightarrow ["S"] \\
[???] & \Rightarrow [(v0), \{c0, \text{grapheme} = t\}, (v1), "T", ??].
\end{align*}
\]

The vowel in the second part of the rule is in brackets to indicate that it is optional. Besides the emphatic consonants, the pharyngealisation effect can be seen in the vowels neighbouring back consonants like uvular consonants which are: ق/q/, خ/x/, and غ/g/. This effect can be noticed on the front vowels only (/a/ or /a:/) which are pharyngealised by becoming more open and more back when followed or preceded by an uvular consonant within the same syllable. Take the examples:
5.2.2 Neutralisation

In MSA, there is a contrast in sound length in some contexts. The sounds’ neutralisation can be observed in various positions:

- Short vowel neutralisation: a dominant Arabic pronunciation rule says that you cannot start an utterance with a non-vowelised consonant or end it with a vowel. This means that a terminal unstressed short vowel, e.g. a case or mood marker, must not be pronounced when occurring at the end of the utterance. The resulting vowel from this process is transcribed as schwa /ə/. For example, a word like *(taktub)* “she writes” is transcribed as /taktubə/ when it occurs at the end of the utterance.

```
[v0, +short, +final@utterance] ==>
[+schwa]:
[??] ## []
```

- Long vowel neutralisation: long vowels are shortened in certain situations in MSA, namely when they occur at the end of the word and are followed by “*hamzatu Alwasl*”. For example, the phrase *(AiftaHuWAlKitAb)* “open the book” is pronounced /iftahu ikita:b/.

```
[v0, +long, artic=X]} ==>
[v1, +short, artic=X]:
[??] ## ["A", ??].
```

- Double consonant neutralisation: it is noticed that double consonants are shortened and pronounced as one consonant when they occur at the end of an utterance as in *(Haj~)* “pilgrimage” which is pronounced /haḍ̣/. In order to apply this conversion, the word should not be followed by a vowel like ُُٰ where it is pronounced /haḍ̣a/. The rule responsible for this kind of variation is:
5.2.3 The pharyngealisation of /l/ and /r/

Tajweed rules state that there are two characteristics of /l/ and /r/ sounds; they both can be described as heavy "taryq" or light "taryq". The "lAm" /l/ is always "taryq" except in the pronunciation of the name (Allâh) where it is pharyngealised when it comes at the beginning of the utterance or when its predecessor vowel is either "fatHa" /a/ or "Damma" /u/ e.g. (All'humma) and (yA All'h) /yâllâh/. On the other hand, the /l/ is not pharyngealised when preceded by a "kasra" /i/ e.g. (lill'h) /lilla:h/. The pharyngealisation of the sound does not only affect the pronunciation of the word but may also change the meaning. Compare the following utterances:

- (waAllah) /wallâh/ which means "and Allah"
- (wallah) /wallâh/ which means "he makes him responsible"

The only difference between the two utterances is the pharyngealisation of the sound /l/.

[1, 1] =>
[{1, +pharyngealised}, {1, +pharyngealised}]:
["A"] #[[ ' , h, ??]}.
According to these rules, the /l/ is pharyngealised in the word when it comes at the beginning of the utterance and when it is preceded by a vowel that is not (-open) and not (-rounded) which means it cannot be “kasra”.

Furthermore, the /r/ sound could be pharyngealised or not depending on the context. It is not pharyngealised whenever it is preceded or followed by “kasra” /i/ within the same syllable, e.g. (riHlap) رحّل /rihla/ “a trip” and (sir) رئ /sir/ “a secret”. In contrast, it is pharyngealised in all other cases e.g. (rajul) رجل /rajul/ “a man” and (wurud) رزق /wurud/ “flowers”. This can be described in the following rule:

\[
\begin{align*}
[\{r, +\text{coda}@\text{syl}\}] & = > \\
[\{r, +\text{pharyngealised}, +\text{coda}@\text{syl}\}] & \\
[??? , \{v0, open=0, rounded=R\}] & # # [???] \\
\text{if not(} & \text{(} O = - \text{ and } R = - \text{).)
\end{align*}
\]

\[
\begin{align*}
[\{r, +\text{onset}@\text{syl}\}] & = > \\
[\{r, +\text{pharyngealised}, +\text{onset}@\text{syl}\}] & \\
[???] & # # [\{v0, open=0, rounded=R\}, ???] \\
\text{if not(} & \text{(} O = - \text{ and } R = - \text{).)
\end{align*}
\]

The pharyngealisation of the consonant /r/ is applied when it is preceded or followed by a vowel that does not have the features (-open) and (-rounded) within the same syllable. The pharyngealisation of /l/ and /r/ normally spreads to the neighbouring vowels within the same syllable, and as is the case with the emphatic consonants, the vowel preceding or following the pharyngealised /l/ or /r/ within the same syllable will also be pharyngealised.

5.2.4 Aligning stress to syllables

In this section the stress rules are presented based on research in the literature and observation of how spoken MSA is actually enunciated [AA70] [McC79] [DJZ99]. This process comes at the end because it depends mainly on the internal structure of the syllables that make up the word, which may substantially change after applying the phonological rules. Although word stress in MSA is non-phonemic, which means it cannot be used to distinguish the meaning, researchers confirm that well-defined stress rules can help enhance Arabic speech processing applications [EI04]. The stress rules presented here are based on the standard pronunciation of MSA. However, the dialect may cause a shifting in the
stress position, e.g. Egyptian speakers pronounce the word (maktaba) مكتبة “library” as /mak-ta-ba/ with stress on /ta/ rather than the standard /mak-ta-ba/.

Stress Rules:

1. If the word contains a super heavy syllable (CVVC² or CVCC), this syllable must be stressed. Consider the word (sijil) سجل “register” /si-jil/ and (rah~Al) راحال “traveler” /rah-ha:l/. This kind of syllable can only occur once in a word and should be in the word’s final syllable as previously mentioned in Section 5.1.1.

2. The stress is placed on the last open heavy syllable (CVV) if there is one. Consider for instance, (dArīs) دارش “student” /da:-ris/ and (dārsuːn) دارسون “students” /da:-ri-suː-na/.

3. In case the word has no super heavy syllable or open heavy syllable, the position of the stressed syllable depends on the number of syllables in the word:

   • In disyllabic words, the stress falls on the first syllable. e.g. (huwa) هو “he” /hu-wa/ and (maktab) مكتبة “office” /mak-tab/.

   • In polysyllabic words, the stress is placed on the antepenultimate syllable. For example (rasamat) رسمت “she draws” /ra-sa-mat/ and (maktabatK) مكتبة “a library” /mak-ta-ba-tun/.

Before the process of assigning stress into syllables, a set of of rules to syllabify the words must be applied (as in Figure 5.10).

After the application of syllabification rules, a set of rules is provided to scan the word’s syllables and assign the stress value to the stressed syllable (given in Appendix F).

³The VV notation stands for long vowels, not two single vowels.
for heavy syllable CVCC at the word’s final and long closed syllable CVC in the mid of the word
markSyllables([]).
markSyllables([C0, V, C1, C2 | REST]):-
  -vowel:char@C0,
  +vowel:char@V,
  -vowel:char@C1,
  -vowel:char@C2,
  syllable:char@C0 <- S,
  syllable:char@V <- S,
  syllable:char@C1 <- S,
  -open:syllS,
  heavy:syllS <- final:pos@syllpos:char@C2,
  !,
(+final:pos@word:char@C2 ->
(syllable:char@C2 <- S);
markSyllables([C2 | REST])).

for heavy syllable CVCC at the word’s final
markSyllables([C0, V0, C1]):-
  -vowel:char@C0,
  +vowel:char@V0,
  -vowel:char@C1,
  syllable:char@C0 <- S,
  syllable:char@V0 <- S,
  syllable:char@C1 <- S,
  -open:syllS,
  +long:char@V0 <- +heavy:syllS,
  +final:pos@word:char@C1,
  !.

for short syllable CV and long open syllable CVV
markSyllables([C0, V0 | REST]):-
  -vowel:char@C0,
  +vowel:char@V0,
  syllable:char@C0 <- S,
  syllable:char@V0 <- S,
  +open:syllS,
  -heavy:syllS,
markSyllables(REST).

Figure 5.10: Syllabification rules.
5.3 Testing and evaluating of the grapheme-to-allophone system

A great deal of attention in the literature has been paid to the evaluation of NLP applications such as speech recognition systems and speech synthesis systems. However, researchers have confirmed that there is still a lack of studies in the objective evaluation of the performance of the system’s components like the grapheme-to-allophone conversion system individually [MGSN98]. As a result, there is no standard methodology to follow in evaluating our system.

Testing and evaluating the proposed system involved two stages: in the first stage we verify that the developed set of rules produces the phonetic transcriptions that correspond to our expectations. The issue here is that these are complex rules which apply in combination and they may interact in unexpected ways especially across word boundaries. In the second stage we investigate how accurately the rules capture the actual phonological realisation.

5.3.1 Testing the system’s performance

In the first testing stage, the PENN Arabic treebank text corpus is used [AAKEAM11]. The corpus sentences were used as an input to the transcription system to test how words are pronounced in a freely occurrence text. Then, we extracted a reasonable amount of transcribed words making sure that they illustrate every possible boundary condition in order to cover all aspects of variability in speech. The rules were tested on 500 words that have a single phonetic transcription and another 500 words with multiple phonetic transcriptions. Words with multiple phonetic transcriptions are particularly important for this work, since the aim of the rules is to make sure that we obtain the correct pronunciation in the various contexts in which the word can occur. For each word that we have chosen in our sample, we have tested occurrences of that word in a variety of contexts in the PATB to ensure that we have a complete test of the rule set. Table 5.6 gives examples of words that have a single phonetic transcription and the ones that have multiple phonetic transcriptions. A word correct metric is used here as a scoring scheme to evaluate the system performance.

Generally speaking, the system produced few word errors (3.4%) when compared to a manually pre-annotated text. The overall score of the system is 98.8%
Table 5.6: Examples of the selected test words with single and multiple phonetic transcription. The table shows the phonetic transcription of the word, how many times it appears in the text, and the surrounding context.

<table>
<thead>
<tr>
<th>Single phonetic transcription</th>
<th>Word</th>
<th>Transcription</th>
<th>#</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ahAbAF</em></td>
<td>D ahaaban</td>
<td>2</td>
<td>taEAdaA *ahAbAF</td>
<td></td>
</tr>
<tr>
<td>biAln~isobapi</td>
<td>binnisbatî</td>
<td>25</td>
<td>biAln~isobapi ilaY</td>
<td></td>
</tr>
<tr>
<td>wa&gt;axiyrAF</td>
<td>waQaxiyr a n</td>
<td>6</td>
<td>wa&gt;axiyrAF a$Arat wa&gt;axiyrAF fiy wa&gt;axiyrAF tu$Ariku</td>
<td></td>
</tr>
<tr>
<td>Multiple phonetic transcription</td>
<td>Almusota$ofayAti</td>
<td>5</td>
<td>aHadi Almusota$ofayAti</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lmusta$ fayaati</td>
<td>2</td>
<td>fiy Almusota$ofayAti</td>
<td></td>
</tr>
</tbody>
</table>

for words with a single transcription and 96.5% for words with multiple transcriptions. The main source of errors was the existence of foreign names such as lut$yAnw, ryg$dl, dnfr, and lkHand and encountering symbols or abbreviations which were missed in the lexicon such as <lx, mlm, and tg. Besides, some errors were caused by encountering mis-spelled items such as mSdq. Such items do not follow the phonetic patterns of Arabic, and hence should not be considered when looking at the system’s performance. When they are removed, the systems output exactly matches our expectations. The aim of this testing phase was to ensure that the rules behaved as we intended them to, in particular to ensure that there were no unexpected interactions between rules. As such, comparing the output with a manually pre-annotated text is an appropriate test. The aim here is to test the integrity of the implementation, not to investigate the accuracy of the rules as a description of Arabic phonology.

5.3.2 Evaluating the developed pronunciation rules

The second evaluation stage was aimed at testing the effectiveness of the rules for predicting how native speakers pronounce words in context. The overall goal of the research reported here is to obtain an accurate description of how people actually say things, to be used, among other things, for providing accurate phonetic transcription for the training phase of a speech recogniser. In this second stage we therefore wanted to verify that these phonological effects have audible
consequences in natural speech.

An independent human annotator with experience in Arabic phonetics was enlisted to transcribe a total of 100 sentences using a narrow phonetic transcription without knowing anything about the expected results.

We manually created 20 sentences with a total of 560 phonemes (listed in Appendix A). The created sentences cover all Arabic phonemes and all the phonological contexts that we looked at in our study. The sentences were recorded by five native speakers from different dialectal regions (Gulf, Egypt, Iraq, Levantine). An independent human annotator with experience in Arabic phonetics was asked to transcribe the total of 100 sentences using a narrow phonetic transcription without knowing anything about the expected results. Given that the annotator had no information about what we expected to find, her transcription provides a fair picture of the sounds that the speakers produced. If she heard one sound being assimilated to another, or if her transcription had a phoneme missing, then that is evidence that these sounds are indeed assimilated or deleted. Thus if her independent description of the sounds that a native speaker produced when reading a text aloud corresponds to the transcription produced by our rules then this provides evidence in support of our rules.

The output of the system matched the annotator’s transcription at 98.3%. The mismatches fall into five categories which are listed in Table 5.7 and discussed below.

1. Dialect

The largest set of mismatches arise from dialectal variations. For instance, the MSA phoneme ح /d̪/ as in the word حجرة /tuğra:/ is generally preserved in the Gulf and Iraqi dialects, whilst it is realised as /g/ in the Egyptian dialect /tugra:/ and /ʒ/ in Levantine Arabic /tuğra:/.

We also found that Egyptian speakers may relax the pronunciation of the emphatic consonant ط /tˤ/ by making it sound like /t/. For example, the word الخطب /?alxatˤib/ has been transcribed as /?alxatib/.

Another aspect of pronunciation variation resulting from dialectal influence is the emphasis spread. Typically, the emphasis spreads to the neighbouring vowels and in some cases the neighbouring consonant. However, looking at the transcribed sentences of the Egyptian and Levantine speakers we found that in some cases the effect of the emphatic consonant spreads to the whole
word. For example, the word /îtatzar/ is pronounced /îtatzar/. In addition, we found that Iraqi accents include an extra letter in the definite article assimilation, which is the palatal stop consonant (j) ح. For example, the word /aljamiya/ is pronounced /aljamiya/ though the /ð/ consonant is not considered as one of the solar consonants where the assimilation normally takes place. Phonetically, adding the ح /ð/ consonant to the solar letter group makes sense because they all have a common area of articulation (including dentals, alveo-dentals, and post-alveolars, which are all made with the front part of the tongue) as shown in Figure 5.5.

The final aspect of pronunciation variation resulted from the way that names are pronounced in the dialect which affects the way they are pronounced in MSA. In the Egyptian dialect, for instance, people say /alparba/ in their daily communication “Wednesday”, as a result, they pronounce this word /alparba:/ instead of /alparbi:/ when speaking MSA.

2. Misreading

Part of the mismatches between the generated phonetic transcription and the one obtained by the annotator resulted from people misreading the words. This is a consequence of the fact that assigning the word’s final vowel depends on the word’s function in the sentence. Speakers vary in their grammar skills, so many less well-educated people drop the vowel at the end of the word, pronouncing it as a schwa, or sometimes they produce the wrong vowel. For example, one of the speakers said /addawri/ instead of /addawri/.

3. Ignoring assimilation

Some people make an extra effort when speaking to machines, which means that they do not speak naturally. As a result, they avoid some sorts of
assimilation aimed at clarifying their speech, as saying /bank/ instead of /bank/. Another reason for ignoring assimilation is pausing between words, which changes the syllabic structure of the word and consequently affects the pronunciation by changing the assimilation context.

4. Individual speaking styles

Speaking styles differ among individuals. For instance, some people tend not to pronounce the final sound in the sentence like saying /muḥtaḥidu:/ when pronouncing the word حيّد /muḥtaḥidun/ and some of them simplify the “ḥamza”, like the one in الأصلة /Iyasyila/ which was pronounced by one speaker as /Iyasyila/.

5. New phenomena

By analysing the annotator’s transcription we found new kinds of regular assimilations:

- Converting the phoneme /d/ to /t̪/ when it is at the end of a syllable which starts with an emphatic consonant and is followed by /q/ like saying /ʾlsātq/ instead of /ʾlsidqu/.
- Simplifying the consonant /z/ to its voiceless counterpart /s/ when followed by the /q/ phoneme, like pronouncing the word الرق /ʾlrisq/ instead of /ʾlrisq/.
- Moving from back vowels to front vowels and vice versa is a difficult task for some speakers especially in fast speech. We found that words like انتموا /ʾintabihu:/ are sometimes pronounced as /ʾintabahu:/ to avoid the tongue movement.
- Some speakers have difficulties in moving between phonemes which are close in articulation points. For instance, we found one speaker pronounced the phrase الثاني لجنس /ʾlān:li: limaqlis/ as /ʾlān:li: nimaqlis/.
Given that the last two kinds of assimilation are not universal, even for speakers from a single dialect, including them is not likely to be useful for our target applications. On the other hand, we have added the first two phenomena to our phonological rules to control such variations.

5.4 Relation to previous work

The problem of pronunciation variation is more significant in Arabic than in other languages due to the great influence of the context in pronouncing Arabic letters. Despite the fact that this problem was studied and highlighted by ancient Arabic scholars in the early centuries, efforts put into algorithmic formulation for modelling pronunciation variation in MSA speech are rare. The only linguistic study we found that investigated transcribing standard Arabic written text into sounds is the one presented in [EI04]. The purpose of this study is to formalise the pronunciation rules of standard Arabic into a framework that can be used by researchers to develop computer-based implementations. The study presents a large set of grapheme-to-phoneme and phoneme-to-phone rules, besides a dictionary for exceptional words. These hand-constructed rules were assessed by testing them on a set of examples to see if they match the researcher’s expectations. However, the developed rules were not validated in terms of their existence in real speech.

Researchers have realised the importance of modelling the pronunciation variation in building Arabic ASR systems and how good pronunciation models contribute greatly to the robustness of the recognition system. Such models can help improve performance by shrinking the mismatch between the speech and text used in designing the acoustic model. There are two main methods used in the literature for modelling the pronunciation variation [AKS00] [FLW99]:

- Knowledge-based approach, which uses phonetic and linguistic knowledge to write phonological rules that can generate variants in pronunciation.

- Data-driven approach, which uses a corpus of real speech to derive the variation in speech.

Each method has its advantages and disadvantages and choosing which method to use depends mainly on the type of variation that needs to be specified and the purpose of modelling this variation [AKS00].
Table 5.7: Resources for the mismatches between the generated transcription and the annotator’s transcription.

<table>
<thead>
<tr>
<th>Mismatch source</th>
<th>Types</th>
<th>Example</th>
<th>#</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialect</td>
<td>converting /v/ to /s/</td>
<td>/Alvulu:/ → /Alulu:/</td>
<td>2</td>
<td>0.06%</td>
</tr>
<tr>
<td></td>
<td>converting /xi/ to /g/</td>
<td>/tuqra:/ → /tqra:/</td>
<td>7</td>
<td>0.24%</td>
</tr>
<tr>
<td></td>
<td>converting /xi/ to /j/</td>
<td>/qam'iyyAt/ → /qam'iyyAt/</td>
<td>4</td>
<td>0.14%</td>
</tr>
<tr>
<td></td>
<td>converting /d/ to /z/</td>
<td>/Al'ahab/ → /Alzahab/</td>
<td>3</td>
<td>0.10%</td>
</tr>
<tr>
<td></td>
<td>assimilation of /xi/</td>
<td>/A'xm'iyya/ → /A'xm'iyya/</td>
<td>1</td>
<td>0.03%</td>
</tr>
<tr>
<td></td>
<td>relaxing of /t'/</td>
<td>/Alat'i:b/ → /Alatib/</td>
<td>1</td>
<td>0.03%</td>
</tr>
<tr>
<td></td>
<td>emphasis spread</td>
<td>/Aintas'arat/ → /Aintas'arat/</td>
<td>4</td>
<td>0.14%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>0.74%</strong></td>
</tr>
<tr>
<td>Misreading</td>
<td>/?addawri:/ → /?addawru/</td>
<td></td>
<td>5</td>
<td>0.17%</td>
</tr>
<tr>
<td>Ignoring assimilation</td>
<td>not speaking naturally</td>
<td>/ban'k/ → /bank/</td>
<td>4</td>
<td>0.14%</td>
</tr>
<tr>
<td></td>
<td>pausing between words</td>
<td>/bad?u limtihaːn'at/ →</td>
<td>2</td>
<td>0.06%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/bad?u ?al?limtihaːn'at/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>0.20%</strong></td>
</tr>
<tr>
<td>Individual speaking styles</td>
<td>relaxing “hamza”</td>
<td>/'as'ila/ → /'asyila/</td>
<td>5</td>
<td>0.17%</td>
</tr>
<tr>
<td>New phenomena</td>
<td>converting /d/ to /t'/</td>
<td>/Als'idq/ → /Als'it'q/</td>
<td>5</td>
<td>0.17%</td>
</tr>
<tr>
<td></td>
<td>converting /z/ to /s/</td>
<td>/Alrizq/ → /Alrisq/</td>
<td>2</td>
<td>0.06%</td>
</tr>
<tr>
<td></td>
<td>moving from back to front vowels</td>
<td>/yabsit'u/ → /yabsut'u/</td>
<td>3</td>
<td>0.10%</td>
</tr>
<tr>
<td></td>
<td>moving between sounds with close</td>
<td>/'I'h'aniː lima:'atis/ →</td>
<td>3</td>
<td>0.10%</td>
</tr>
<tr>
<td></td>
<td>articulation points</td>
<td>/'I'h'aniː nima:'atis/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>0.43%</strong></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td><strong>1.71%</strong></td>
</tr>
</tbody>
</table>
Given these two approaches, we can sort out the works done in modelling Arabic pronunciation accordingly. The only work we found for modelling the Arabic pronunciation variation using a data-driven approach is in [AAKEAM12]. The researchers attempted to distil the pronunciation variants from the training speech corpus. The proposed method aims at modelling within-word pronunciation variation by performing phoneme recognition followed by a sequence alignment task between the observed and reference phonemes. The generated pronunciation variants are then added to the system’s dictionary as well as the language model. Results of applying this method show no improvement when expanding the pronunciation dictionary alone; however, the WER was reduced by 2.22% when representing those pronunciation variants in the language modelling.

The knowledge-based approach has received great interest among researchers for modelling Arabic pronunciation variation. The classical approach involves generating an Arabic multi-pronunciation dictionary. For instance, Hiyassat used SPHINX-IV engine for building an isolated speech ASR system and building an automatic pronunciation dictionary toolkit for both the Qura’an and Arabic digits [Hiy07]. Biadsy et al. generated a multi-pronunciation dictionary using pronunciation rules and then the MADA [HRR09], as a morphological disambiguation tool, to determine the most likely pronunciation of a given word in its context [BHH09]. The proposed method reported a significant improvement of 4.1% in accuracy compared to the baseline system. [AEAG+08] provided a limited set of phonetic rules for automatic generation of an Arabic phonetic dictionary. The rules were mainly direct grapheme-to-phoneme with a few rules for the assimilation of “lAm” with solar letters, the conversion of (n) to (m) when followed by (b), and emphatics with pharyngeal vowels. The effectiveness of using the generated dictionary was tested using a large-vocabulary speaker-independent Arabic ASR system and achieved a comparable accuracy with the same vocabulary-size English ASR system. The work of [AEAG+08] was then implemented in many other publications such as in [AEAM09], [AAKEAM11], and [AAKEAM12]. In [AAKEAM11], the researchers tackle the problem of cross-word pronunciation variation by expanding the pronunciation dictionary. Besides utilising the dictionary generating tool developed by [AEAG+08], the researchers considered compounding words with “Idgham” (merging) and “Iqlaab” (changing) and including these compound words in the dictionary. Using this technique, an interesting improvement in cross-word variation was achieved. Another way of
modelling cross-word pronunciation variation in a knowledge-based framework is using part of speech tagging as found in [AAME12]. The adopted method focuses on finding a way to merge two words and introduce the long merged words in the pronunciation dictionary. The word’s tag such as noun, pronoun, verb, adverb, prepositional, etc. is used in order to find words that can be merged. Using this technique has led to noticeable improvement in the ASR performance, especially when compounding nouns with adjectives.

We can gather from the studies reviewed above that the predominant methodology in tackling the problem of cross-word and within-word pronunciation variation is generating an expanded version of the system’s dictionary. The generated dictionary contains multiple pronunciations for each word either by using phonological rules or forming compound words. The choice is then left to the speech recogniser to pick the closest pronunciation for the word, given the acoustic evidence. This method does not help the system use information about the local context to determine which pronunciation was used.

In our work, we provide a comprehensive set of conversion rules that can handle cross-word and within-word pronunciation variation in continuous speech. Instead of expanding the pronunciation dictionary with multiple pronunciations of each word, the generated phonetic transcription provides only one possible pronunciation for each word according to its context. The recognition tool is then forced to use the generated pronunciation of the word rather than doing complicated alignment tasks to choose the most applicable pronunciation (as discussed in Section 4.2.4). The experimental evidence confirms that this method outperforms the use of a fixed pronunciation dictionary as well as a multi-pronunciation dictionary as we will see in Section 6.1.
Chapter 6

Experiments

In this chapter, we investigate certain factors that affect the performance of the Arabic ASR system. We start by exploring some properties of phonological features that can influence the way the recogniser performs. Following these phonological investigations, the chapter describes some interesting technical issues that can affect the performance of the ASR system. Within the phonological investigation, we aim at addressing the following questions:

• What kind of impact can we get by providing the recogniser with a context-sensitive phonetic transcription?

• Can we use stress information to improve speech recognition performance?

• Can we use underspecified phonetic representation by blurring the diacritical marks to improve the system’s performance?

• Is phoneme level better than word level in building HMM models?

The chapter then seeks to address the following technical questions:

• Can we improve the robustness of the system by categorising speakers according to their predefined classes such as gender and accent?

• Would increasing the training data size lead to an improvement in the recognition performance?

The chapter has been organised in the following way. In the first section, we investigate the problem of pronunciation variation that resulted from the coarticulation effect. In the second section we investigate the influence of locating
the word’s stress in the acoustic modelling. The third section presents a study on the effectiveness of using phonemes rather than words in modelling speech. In the fourth section, we compare the performance of the ASR system when using fully diacriticised text materials and when using non-diacriticised text materials. The effectiveness of grouping speakers according to their best-matching speech communities is tested in the fifth section. Finally, the correlation between the training data size and the system’s performance is studied in the sixth section. In each section, we review the problem and its suggested solutions in the literature; we lay out the experimental dimensions, followed by reporting and analysing the results, and finally a conclusion with the main findings will be given at the end of each section.

6.1 Experiments on the effectiveness of using a context-sensitive phonetic transcription

It is widely believed that pronunciation variation is one of the major factors that lead to a deterioration in the performance of ASR systems, especially in continuous speech systems [MGSN98] [SC99] [Wes03]. Not surprisingly, the effort put into modelling pronunciation variation for ASR has recently increased.

As mentioned in Chapter 5, two main techniques are used in the literature to account for the variation in pronunciation: data-driven approaches and knowledge-based approaches. The two approaches differ in how the information on pronunciation variation is obtained. In data-driven approaches, the acoustic data is used solely to find the pronunciation variants. On the other hand, knowledge-based approaches use information derived from linguistic sources such as pronunciation dictionaries and linguistic findings on pronunciation variation. Controlling the pronunciation variation using a data-driven framework is frequently used, as in [AKS00] [FLW99] [AAKEAM12] [LWS09]. A possible drawback with this method is the need for huge training materials to avoid the problem arising from undertraining less frequent acoustic events [Kir99]. Another disadvantage of data-driven studies is the process of deriving information on pronunciation variation which needs to be resumed for every ASR system [SC99]. The knowledge-based approach is also found in many studies [AD95] [TFJ+95] [BGMM98] [HHS05] [OYK07]. The main difficulty with this method is to find
a linguistic resource that covers all possible variations besides the possible mismatch between information found in linguistics literature and used data [SC99].

The problem of pronunciation variation is more significant in Arabic than in other languages due to the great influence of the context in pronouncing letters. Different studies with different methodologies and scopes have been proposed in the literature to deal with the problem of pronunciation variation in Arabic (introduced in Section 5.4). Most of the solutions presented to account for the coarticulation effects in Arabic have tended to focus on providing the speech recogniser with multiple pronunciations in the dictionary. With multi-pronunciation dictionaries, the HMM can estimate the likelihood of the possible pronunciation at different routes throughout the network to insure it has the chance of finding the right route.

In the experiments introduced here, we will draw a comparison between three different ASR systems. The first system is based on a fixed dictionary, the second on a multi-pronunciation dictionary, and the third on a context-sensitive phonetic transcription. The aim of this comparison is to measure the effectiveness of using each method. The context-sensitive phonetic transcription is generated with the aid of the grapheme-to-allophone system described in Chapter 5. The process of integrating the grapheme-to-allophone system within the HTK to use the generated phonetic transcription and cancel the role of the lexicon was given in Chapter 4. Figure 6.1 summarises the changes made during the training stage. We provide this linguistic knowledge to the HTK, which automatically infers likely pronunciation alternatives at the stage of doing tying. So these phonological rules are used in addition to the built in machine learning provided by the HTK. We presented this technique so as to overcome the problems related to solely using data-driven algorithms. Those problems are mainly related to the need for an enormous amount of training data to cover all possible triphones in the language, besides the fact that with a data-driven approach, we cannot learn the kind of rules we are working with. Details of the experiments are given below.

6.1.1 Experimental design

Two datasets are used in these experiments to ensure the correctness of the results from different systems. The main dataset is based on 300 automatically generated sentences varying in length from 2 to 6 words. The sentences are written in a phonetically rich and balanced manner in order to produce a robust Arabic speech
Figure 6.1: HTK workflow, the area written in red indicates where changes are introduced in this kind of experiments.

recogniser. 56 native Arabic speakers were asked to record a number of sentences. The total duration of the recordings is about 25.6 hours.

The pilot dataset uses 20 manually created sentences varying in length from 3 to 7 words. Although there are just 20 sentences, we made sure when creating those sentences to cover all the possible phonological variations in Arabic. The recordings were collected with the aid of 23 Arabic native speakers from different Arab regions and the total recording size is around 2.3 hours. More details about the datasets can be found in Chapter 4.3.
To compare the performance of each system, three recognition systems were developed for each dataset: a fixed dictionary-based system, a multi-pronunciation dictionary-based system, and a generated phonetic transcription-based system. The fixed dictionary-based system provides one possible transcription for each word whilst the multi-pronunciation dictionary provides a number of possible pronunciations. On the other hand, the generated phonetic transcription-based system gives a context-sensitive transcription for each word taking into account inter-word and cross-word pronunciation variations. This phonetic transcription is generated using the grapheme-to-allophone system that was introduced in Chapter 5. Technical details about embedding the grapheme-to-allophone system into the speech recogniser are given in Chapter 4.

The recordings were split into a training set and a testing set. The main dataset contained about 20.4 hours of audio used for training the system and about 5.1 hours used for testing. The recordings in the pilot dataset were divided into 1.7 hours audio files for training and about 28 minutes for testing.

For the best use of data and to avoid unfair testing, the research uses a 5-fold cross-validation approach as a way of assessing the proposed systems. This involves randomly partitioning the data into 5 equal size subsets to perform the training on 4 subsets and validate the performance on the other subset. This process is repeated 5 times, with each of the 5 subsets used only once as the validation data. The results from the 5 folds can then be averaged to compute a single estimation. The advantage of this method over the standard evaluation approach is that all observations are used for both training and testing, providing a more robust testing for experiments with limited data sources.

### 6.1.2 Results

Using a fixed pronunciation dictionary, the system used the main dataset reported 60.38% absolute accuracy\(^1\), whilst using a multi-pronunciation dictionary in building the system improved the recognition accuracy to 65.8%. On the other hand, by using the generated phonetic transcription for training, we achieved a further improvement to 69.36%. Using the generated phonetic transcription in the experiment that uses the pilot dataset has also improved the accuracy from 82.1% to 93.4%. These results are shown in Table 6.1.

\(^1\)The absolute accuracy is calculated by excluding the result of silences recognition from the word recognition result and this is the standard way we follow in calculating the accuracy.
Table 6.1: Results of using different pronunciation resources in training the speech recogniser with two different data sets. 5-fold cross-validation approach is used to test the system. The word recognition accuracy is provided as a common speech recognition accuracy metric.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Training size</th>
<th>Testing size</th>
<th>Absolute accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Fixed dict.</td>
<td>Multi dict.</td>
</tr>
<tr>
<td>Main</td>
<td>18,361 files</td>
<td>4590 files</td>
<td>60.4</td>
</tr>
<tr>
<td></td>
<td>20.4 hrs</td>
<td>5.1 hrs</td>
<td></td>
</tr>
<tr>
<td>Pilot</td>
<td>1600 files</td>
<td>400 files</td>
<td>82.1</td>
</tr>
<tr>
<td></td>
<td>1.7 hrs</td>
<td>28 mins</td>
<td></td>
</tr>
</tbody>
</table>

The reason for the impressive improvement obtained via the generated phonetic transcription is its accurate presentation of the sounds of the language. In Figure 6.2, we provide snapshots from the contents of the fixed pronunciation dictionary (typically used in building the ASR system), the contents of the multi-pronunciation dictionary (the predominant alternative to the fixed dictionary in the literature), and the contents of the generated phonetic transcription file. Obviously, the generated phonetic transcription is more accurate, for it provides the pronunciation of each grapheme according to the surrounding graphemes and the morphological structure of the word. At a certain point, the HTK alignment tool is forced to choose the pronunciation to be provided for the word following the generated phonetic transcription instead of allowing it to pick the best pronunciation found in the multi-pronunciation dictionary according to the acoustic evidence.

6.1.3 Evaluating the effectiveness of individual rules

Given that the overall performance of the speech recogniser has significantly improved by using the generated phonetic transcription, it is worth investigating which rules make the most contribution and, indeed, verifying that they are all useful. It would be also interesting to investigate the ratio between the number of times each rule was applied and how much improvement was introduced. The effectiveness of applying the individual pronunciation rules was evaluated by using backward elimination. This kind of testing is conducted by excluding one phonological phenomenon in each experiment while keeping all the other rules. Only the phonological phenomena which appear frequently were tested. The aim was
Figure 6.2: Snapshots of the contents of the generated phonetic transcription file, fixed pronunciation dictionary file, and the multi-pronunciation dictionary file.
to find out what impact each phenomenon has on the recognition performance. Below is a description for the process of rules elimination:

1. Eliminating “hamzatu Alwasl” deletion rules

The deletion of “hamzatu Alwasl” is one of the most applied rules with 22,614 appearances. This rule is applied to verbs as well as definite and non-definite nouns in certain contexts. Excluding “hamzatu Alwasl” deletion rules brings the word recognition accuracy to 66.3%. This means losing 3.1% of the accuracy compared to the system that uses a full set of pronunciation rules.

2. Eliminating the /r/ pharyngealisation rules

The /r/ consonant is pharyngealised in certain contexts. This set of rules was applied 17,754 times in the main training corpus. Despite the high frequency of this phenomenon, excluding the application of this set of rules deteriorates the accuracy by only 0.5% as the reported accuracy is 68.9%.

3. Eliminating spread of pharyngealisation rules

Having an emphatic or pharyngeal consonant within the word cause a spread of pharyngealisation to the neighbouring sounds. Spread of pharyngealisation rules were applied 13,793 times in our main training corpus. Excluding the application of these rules brought the accuracy to 67.2% with total of 2.2% drop in accuracy.

4. Eliminating definite article assimilation rules

Assimilation of the definite article has been found 6,828 times in our training corpus. This process may be accompanied with other processes such as long vowel neutralisation, hamza deletion, and vowel insertion. Eliminating the application of this set of rules has an obvious negative impact on the system’s performance as the system is deteriorated by 7%. The reported word recognition rate when excluding this set of rules is 62.4%.

5. Eliminating nasalisation rules

Nasalisation is one of the main assimilation aspects in Arabic which could be total or partial. This phenomenon has appeared 5098 in the main training corpus. Excluding this set of rules leads to 1.2% drop in the accuracy with a 68.2% word accuracy rate.
Table 6.2: Results of testing individual phonological rules. The table shows the type of the eliminated phonological rule, the number of occurrences, and the effect of eliminating this rule on the system’s performance.

<table>
<thead>
<tr>
<th>Type of eliminated rules</th>
<th>Appearances</th>
<th>Effect on accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>“hamzatu Awasl” deletion rules</td>
<td>22,614</td>
<td>-3.1%</td>
</tr>
<tr>
<td>The /r/ pharyngealisation rules</td>
<td>17,754</td>
<td>-0.5%</td>
</tr>
<tr>
<td>Spread of pharyngealisation rules</td>
<td>13,793</td>
<td>-2.2%</td>
</tr>
<tr>
<td>Definite article assimilation rules</td>
<td>6,828</td>
<td>-7%</td>
</tr>
<tr>
<td>Nasalisation rules</td>
<td>5,098</td>
<td>-1.2%</td>
</tr>
</tbody>
</table>

Table 6.2 summarises the results reported above by showing the type of rule which was eliminated, appearances, and the word recognition accuracy obtained.

It can be seen from Table 6.2 that although the definite article assimilation rules do not occur as often as any of the other three, they have a substantial effect on the recognition accuracy. Our speculation is that the other phenomena do not have as much impact on speech as the assimilation of the definite article, which renders the process in the HTK that maps characteristics of sounds to phonemes more robust in terms of mapping these items. This leads us to the notion that there is no direct correlation between the number of occurrences and recognition performance. The one phenomenon that has the most obvious effect on speech is that which has a considerable impact on the system’s performance.

6.1.4 Conclusion

In this section, we applied the developed grapheme-to-allophone algorithm in building a speech recogniser. We tested the effectiveness of this approach by building three different systems for each dataset (fixed dictionary-based system, multi-pronunciation dictionary-based system, and a generated phonetic transcription-based system). We reported a significant decrease in WER between 9% and 11.3%. The research confirms that using the generated phonetic transcription outperforms the use of multi-pronunciation dictionaries which are widely believed to be the best method for capturing pronunciation variations. The absolute accuracy of a speech recognition system depends on a large number of factors – size of the training data, perplexity of the language (which in turn depends on the size of the vocabulary and complexity of the grammar), quality of the recordings, homogeneity of the subject population, to mention but a few. It is therefore
difficult to provide a direct comparison between the performance of different systems, especially in the absence of a widely and easily available shared dataset. What matters here is the improvement produced by using the generated phonetic transcription. The two datasets have different vocabularies, different grammars, and different speaker-populations. In both cases, using context-sensitive phonetic transcriptions leads to substantial improvements in performance. We believe that similar results are likely to be obtained with other datasets, grammars, and vocabularies. These positive results have encouraged us to use the generated phonetic transcription in the rest of the experiments reported in this chapter.

6.2 Testing the effectiveness of incorporating stress value during the acoustic modelling

Developing precise word-stress rules for MSA could be of great value to enhancing the performance of speech recognition systems. In this section, the stress phenomenon is addressed as one of the supra-segmental characteristics of speech. The stress phenomenon is typically addressed in phonological literature as a non-phonemic feature of MSA, since it does not lead to a distinction in meaning [AA70] [Odi76]. According to [Bap86], stress can be identified from the standpoints of production and perception. Stressed syllables are produced with more muscular energy than unstressed syllables. From a perceptual standpoint, all stressed syllables have one feature in common which is prominence. Four different factors make the syllable prominent: pitch, length, loudness and vowel quality.

Stress is an important feature of any language as suggested by many researchers such as [LMS72] who stated that “stressed syllables can form the anchors around which a search for occurrences of words can be attempted”. In the domain of NLP, researchers have emphasised the importance of locating stressed syllables and have used this information as a strong constraint to enhance the acoustic modelling of speech. For instance, [AZ85] has found that augmenting word stress information can reduce the number of competing word candidates. In more recent studies, [ADA92] [SH97] have used the stress feature of separate phones during the recognition system’s training process to obtain a more accurate acoustic modelling. [HMM92] have used a knowledge-based approach to add vowel stress information to the recognition system’s lexicon and reported a significant WER reduction. Two more studies have introduced a classification
approach to determine the stressed vowels [JS96] [VKB99]. However, we have not found any attempt in the literature that uses stress information in conducting speech recognition experiments dedicated to Arabic. In this section, we aim to test the effectiveness of exploiting stress information in building Arabic speech recognition systems. The investigation being made here is based on the stress rules introduced in Chapter 5 which are formulated according to the weight and number of syllables that make up a word.

6.2.1 Experimental design

A number of experiments were conducted using the main dataset in order to test the effectiveness of adding stress value to syllables in the generated phonetic transcription. Unlike the experiments carried out in the previous section which were performed using a 5-fold validation approach, the experiments carried out here only used one subset of the data for training and a smaller subset for testing. The training corpus includes 18,361 utterances, whilst the testing corpus includes 4,590 utterances.

We considered three ways to carry out this testing: introducing a stressed version of every phoneme, a stressed version of just the vowels, or a separate stress marker following the stressed vowels.

In building a speech recogniser that uses a stressed version of every phoneme, the transcription system produces a context-sensitive phonetic transcription that uses all grapheme-to-allophone conversion rules in addition to the ones that mark the consonants and vowels of the stressed syllables. The stressed syllables are marked in the transcription by placing a distinctive symbol just after the stressed consonants and vowels. The following is an example of a generated transcription that includes stress markers:

**Sentence:** Alwaladu yatanAwalu AlTaEAm

**Transcription:** 
/Q a l w‘ a’ l a d u - y a t a n‘ a a’ w a l u - t˚ t˚ E˚ a a’ m˚/

The symbol “‘” indicates that the sound is stressed, so /w˚/ is a stressed phoneme while /w/ is unstressed. As far as the HTK is concerned, these are two unrelated phonemes.

Similarly, when introducing a stressed version of the vowels, the transcription system produces a context-sensitive phonetic transcription, marking only the vowels of the stressed syllables with “‘”, as in the following:

/Q a l w a’ l a d u - y a t a n a a˚ w a l u - t˚ t˚ E a a˚ m˚/
Introducing the stress marker as a separate phoneme makes the transcription of the mentioned sentence appear in this way:

/Q a l w a ’ l a d u - y a t a n aa ’ w a l u - t t a E a a ’ m/

In this transcription, the vowels are the same as the stress marker introduced as an additional phoneme.

6.2.2 Results

By activating the stress rules to mark the stressed phonemes, we did not expect that the overall performance of the system would deteriorate by 6.1%. The system reported 61.60% word recognition accuracy when attaching the stress marker to the contents of the stressed syllables, whereas in the baseline system the reported word recognition accuracy was 67.7%. We have always maintained that providing the most sensitive phonetic transcription would have a positive impact on the recognition performance. However, the reported result of activating stress rules does not support that assertion.

In order to investigate this negative finding more thoroughly, we ran another experiment using a phonetic transcription that marks only stressed vowels by attaching the stress value to each vowel as previously explained. The fact that the stress is more distinguishable with vowels than consonants further motivated us to carry out this type of testing. The testing results show that the word recognition accuracy increased to 65.0%, which, although not as good as excluding them from the provided transcription, shows an improvement over the full syllable marking.

The third way of investigating the stress effect was to introduce the stress marker to the vowels as a separate phoneme. This means that we are reducing the number of phonemes that the HTK has to deal with. In other words, instead of having two different representations of each vowel (a stressed version and a non-stressed version), only one representation of each vowel is used with an additional stress phoneme that follows the stressed vowels. Evaluating this system shows that the word recognition accuracy jumped to 69.4%, which outperforms all the systems previously mentioned.

This consistent improvement reported in the two systems previously mentioned might be explained by a reduction in the number of phonemes and HMMs accordingly. For instance, when blocking the position of the stress in the given transcription, the total number of phonemes in the recognition system was 44. On the other hand, by attaching the stress marker symbol to the stressed consonants
Table 6.3: Results of adding the stress value in the phonetic transcription. The table provides a comparison between the baseline system (no stress) and different levels of adding the stress marker.

<table>
<thead>
<tr>
<th>Phonetic transcription level</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline system (no stress)</td>
<td>67.7%</td>
</tr>
<tr>
<td>Stress marker attached to consonants and vowels</td>
<td>61.6%</td>
</tr>
<tr>
<td>Stress marker attached to vowels</td>
<td>65.0%</td>
</tr>
<tr>
<td><strong>Stress marker as a separate phoneme following vowels</strong></td>
<td><strong>69.4%</strong></td>
</tr>
</tbody>
</table>

and vowels, this number was increased to 81. This large amount of phonemes could lead to great confusion during the acoustic analysis which in turn weakens the overall performance of the system, as already observed. The word recognition accuracy was improved when the stress marker was attached only to vowels, thereby making the total number of phonemes reach 47. Adding the stress marker as a separate phoneme following the stressed vowels leads to the best performance with only 45 phonemes. The main advantage of this method lies in its ability to determine the stress positions in the word while keeping the number of phonemes to a minimum. Table 6.3 summarises the results reported in this section.

6.2.3 Conclusion

This section aimed at testing the effectiveness of incorporating an advanced level of phonological constraints for speech recognition tasks. Given this experimental evidence, we may conclude that although excluding the stress rules seems to have a positive impact from the first instance, what is better is to find other ways of introducing the stress phenomenon. We found that by introducing stress as a separate phoneme following stressed vowels, the achieved word recognition accuracy is 69.4%. The obtained accuracy outperforms the baseline system by 1.7%.

6.3 Phoneme-based models vs. word-based models experiments

As in any speech recognition engine, the HTK requires a dictionary which provides an initial phone level transcription and a grammar that defines all of the legal word sequences. Words are typically the basic speech unit used in the provided
scripts to build HMM models. This section is concerned with examining the influence of using phonemes as basic units in building HMM models. Using phonemes as terminal symbols in designing ASR systems has previously been suggested by [Alo10] and [AAA10a]. The advantages found in phoneme-based systems are related to their effectiveness in capturing phonetic details, thereby outperforming the word-based systems.

[Alo10] used the Arabic alphabets and / or digits to build an isolated word small vocabulary speech recogniser with both word and phoneme level representations. The researcher reported an improvement in the recognition accuracy by 0.77%, 5.67%, and 5.62% for digit recognition, alphabet recognition, and for both digits and alphabets (alphadigits), respectively.

In this section, several experiments have been conducted to compare the performance of a word level based system with a phoneme level based system. We are looking to see what improvement (if any) can be obtained by using phonemes rather than words in training the recognition system. Figure 6.3 gives a summary of the HTK workflow in such experiments.

### 6.3.1 Experimental design

The aim of this work is to investigate the effectiveness of using phonemes in building HMM models by comparing this approach with the performance of the word level based systems that were provided in Section 6.1. We want to see whether using phoneme level HMMs can change the performance of the dictionary-based system as well as the generated phonetic transcription-based system. The obtained results will be compared with the performance of the word level based systems reported in Section 6.1.

In the phoneme level experiments, utterances are presented as strings of successive phonemes, the dictionary contains a sorted list of phonemes rather than words, and the grammar considers every phoneme as a single word. Figure 6.4 illustrates the contents of the three source files (prompts-dictionary-grammar).

As with the experiments presented in Section 6.1, we will use both datasets, while the 5-fold cross validation approach is also used to achieve the best usage of the data. In five phases, every time the first system was trained each time with different 18,448 utterances (20.4 hours) and tested with different 4612 utterances (5.1 hours). Meanwhile, the second system is trained with 1600 different utterances (1.7 hours) and tested with 400 different utterances (28 mins.).
6.3.2 Results

As described in Section 4.2.2.3, the HTK results’ analysis gives the sentence level and word level statistics as well as indicating the number of times in which the test sentences and words have been recognised correctly. In the case of phonemes being used as the basic units in designing the system, the word recognition accuracy given by the HTK refers to phonemes not words. In order to avoid confusion, we will name the word recognition accuracy given by the HTK in the phoneme based experiments as phoneme recognition accuracy.
Figure 6.4: A snapshot of the source files contents: prompts, dictionary, and grammar for a phoneme level system.

Table 6.4: Experiments’ results using the main and pilot datasets. The table provides a comparison between the use of phonemes and words as the HMM units. Two systems are reported: one using a fixed pronunciation dictionary and another used the generated phonetic transcription. 5-fold cross-validation approach is used to test the reported systems.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HMM unit</th>
<th>Absolute accuracy (%)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fixed dict.</td>
<td>Phonetic transcription</td>
<td></td>
</tr>
<tr>
<td>Main</td>
<td>Word level</td>
<td>60.4</td>
<td>69.4</td>
</tr>
<tr>
<td></td>
<td>Phoneme level</td>
<td>74.5</td>
<td>82.1</td>
</tr>
<tr>
<td>Pilot</td>
<td>Word level</td>
<td>82.1</td>
<td>93.4</td>
</tr>
<tr>
<td></td>
<td>Phoneme level</td>
<td>87.6</td>
<td>98.2</td>
</tr>
</tbody>
</table>

If we glance at the average unit recognition accuracy of each system as given in Table 6.4, we can see an impressive improvement in the system’s performance when using phoneme units. However, looking at the sentence recognition accuracy for both types of systems, we found that phoneme level based systems are not as good as they seem to be. Obviously, the sentence recognition accuracy is typically less than word or phoneme level accuracy, as it looks for the correctness of the whole sentence rather than individual words or phonemes. Nevertheless, sentence recognition accuracy is expected to be strongly correlated with any increase or decrease in the word recognition accuracy. In the case of the results given in Table 6.4, the sentence recognition accuracy is higher with the word level ASR system than with the phoneme level system, which indicates that the overall performance
Table 6.5: Phoneme-level and word-level experiments’ results with sentence accuracy.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HMM unit</th>
<th>Absolute accuracy (%)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fixed dict.</td>
<td>Phonetic transcription</td>
</tr>
<tr>
<td>Main</td>
<td>Word level</td>
<td>59.9</td>
<td>66.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phoneme level</td>
<td>52.3</td>
<td>63.7</td>
<td></td>
</tr>
<tr>
<td>Pilot</td>
<td>Word level</td>
<td>79.6</td>
<td>91.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phoneme level</td>
<td>71.2</td>
<td>85.4</td>
<td></td>
</tr>
</tbody>
</table>

of the recogniser is not actually improving when using phoneme level HMMs. Table 6.5 gives the sentence accuracy results for the designed systems in order to clarify this point.

The sentence accuracy results are reported here because we do not have a meaningful word recognition accuracy for phoneme-based systems in order to make a fair comparison. If the sentence recognition accuracy had been higher with phoneme level experiments, we would have looked for a meaningful word recognition accuracy. However, since it has not shown any improvement, there is no point in doing a complicated alignment task so as to obtain word recognition accuracy for the phoneme-based system.

It is apparent from reading the results given in tables 6.5 and 6.4 that phoneme representation is not superior -despite appearances- in building Arabic ASR systems. By looking at the confusion matrix for the phoneme level experiments, we can explain this high accuracy in the phoneme recognition rate in terms of the existence of lots of coincidences. For instance, all Arabic definite nouns start with the definite article “Al” and a good number of verbs have identical diacritic sequences. The following is the confusion matrix we get from recognising the sentence “AintabihuwA liQi§Arapi Almurawr” which has been mistakenly recognised as “AirtifAEu QasEArî Al*ahab”:

```plaintext
/*test3100437
same A, A
same i, i
different n, r, 1.500000
same t, t
deleted a
deleted b
same i, i
```
deleted h
deleted u
different w, f, 1.500000
same A, A
different l, E, 1.500000
different i, u, 1.500000
same Q, Q
inserted a
different i, s, 1.500000
different $, E, 1.500000
same A, A
same r, r
deleted a
deleted p
same i, i
same A, A
same l, l
deleted m
different u, *, 1.500000
different r, a, 1.500000
different u, h, 1.500000
different w, a, 1.500000
different r, b, 1.500000

This indicates that some phonemes were coincidentally counted as well-recognised. With word level experiments, we would get all words wrong in the sentence and consequently the recogniser will report 0% word accuracy. However, as we can see from the confusion matrix given above, the recogniser will consider 11 out of 29 phonemes in the sentence to be recognised accurately, giving us 37.9% phoneme recognition accuracy.

6.3.3 Conclusion

From the results given above, we can conclude that although we have seen a much higher unit recognition accuracy when using phonemes rather than words as the base units, when looking at it in more detail we found that there is no real gain
from using phoneme-level HMMs. The reason behind this significant improvement is the fairly high possibility that phonemes would occur in a certain order, which gives the recogniser a better chance to come up with the right phoneme. The observation that phoneme level experiments do not pay off was established by looking at the sentence recognition accuracy, which was found to be higher with words, and by looking at the outcome of the confusion matrix for the recogniser output, which confirms the accidental recognition of certain phonemes.

6.4 Using phonetically underspecified text materials

Most written Arabic scripts lack important phonetic information, a result of not including short vowels and other diacritics (more information about Arabic diacritics can be found in Section 3.1.2). The absence of diacritics in Arabic text materials has concerned many researchers working in the field of Arabic language processing. Most researchers find non-diacriticised scripts to have considerable ambiguity for the development of NLP tools. [VK04] explained this ambiguity in terms of the difficulty of building accurate acoustic models for short vowels if they are not identified or their position in the signal is unknown. Furthermore, researchers found that the lack of diacritical marks leads to a less predictive language model compared to those trained on fully diacriticised texts.

The problem of non-diacriticised text materials has been discussed by many researchers, many of whom considered this as one of the main obstacles in processing the Arabic language, and have therefore developed automatic and semi-automatic methods for restoring the missing diacritics, as in [VK04], [ANB05] [KV05], [MGL06], [ZSS06], and [HR07]. The process of diacriticising Arabic text is part of the efforts made towards developing high performance speech recognition systems.

In contrast, more recent studies have reported a better recognition accuracy using non-diacriticised text materials than when using diacriticised text materials. For instance, results reported from developing a transcription system for Arabic broadcast news in [AEAM09] have shown that the recognition accuracy is higher when using non-diacriticised texts. The system, which is trained using about 7 hours of speech, obtained a recognition accuracy of 90.78% when using a fully
diacriticised corpus, whereas it obtained 93.04% recognition accuracy when using non-diacriticised texts. Similarly, results introduced in [AAZ+12] have shown a slight increase in word recognition accuracy when using a corpus without diacritical marks. Two systems were developed, one with different speakers but similar sentences which achieved 95.92% and 96.29% recognition accuracy with and without diacritical marks, respectively. The second system uses different speakers and different sentences and achieved a recognition accuracy of 89.08% with fully diacriticised texts and 90.23% with non-diacriticised texts.

This section investigates the influence of using phonetically underspecified text materials in building the acoustic and language model of the ASR system. The outcomes are then compared with the results of using fully diacriticised text materials given in Section 6.1. Figure 6.5 summarises the workflow of the HTK when using phonetically underspecified text materials.

6.4.1 Experimental design

The aim of the designed experiments is to determine the kind of impact from using non-diacriticised text materials in training and testing the speech recognition system with the available datasets. Since our datasets are originally fully diacriticised, we will carry out a set of phonetic transcription-based experiments besides the fixed dictionary-based experiments. Acquiring a context-sensitive phonetic transcription for a given text depends on the properties of the surrounding vowels, so the grapheme-to-allophone system will not process a non-diacriticised input text. In the phonetic transcription-based experiments, we blurred the diacritics after extracting the phonetic transcription which reflects all context-dependent pronunciation variations. The diacritics were removed gradually; firstly we only removed the short vowels and sukoons. Then, we removed all tanweens in addition to the short vowels and sukoons. In the final stage, we removed all diacritics including shadda as well as short vowels, sukoons, and tanweens.

As with the experiments carried out in the previous sections, a 5-fold cross validation approach is applied to obtain best usage of the data.
6.4.2 Results

Table 6.6 shows the results obtained from using the underspecified transcription in its different phases for both datasets with a fixed dictionary and context-sensitive phonetic transcription. In this table, the non-blurred version is included as reference.

The average scores of the experiments given in Table 6.6 provide clear evidence of the positive effect of using underspecified text materials for designing an Arabic ASR system. The rate of improvement varies according to the level of
Table 6.6: Underspecification experiments’ results using two different datasets. The table reports the results of different systems using different underspecification levels with both fixed dictionary and generated phonetic transcription. 5-fold cross-validation approach is used to test the system.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Underspecification level</th>
<th>Absolute accuracy (%)</th>
<th>Fixed dict.</th>
<th>Phonetic transcription</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>Non-blurred</td>
<td>60.4</td>
<td>69.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blurring short vowels+sukoons</td>
<td>62.3</td>
<td>72.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blurring short vowels+sukoons+tanweens</td>
<td>67.5</td>
<td>74.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blurring short vowels+sukoons+tanweens+shadda</td>
<td>67.3</td>
<td>72.7</td>
<td></td>
</tr>
<tr>
<td>Pilot</td>
<td>Non-blurred</td>
<td>82.1</td>
<td>93.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blurring short vowels+sukoons</td>
<td>89.4</td>
<td>94.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blurring short vowels+sukoons+tanweens</td>
<td>92.1</td>
<td>94.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blurring short vowels+sukoons+tanweens+shadda</td>
<td>90.1</td>
<td>93.0</td>
<td></td>
</tr>
</tbody>
</table>

underspecification and the source of pronunciation definition.

Blurring only short vowels and sukoons improves the performance by 1.9 and 2.9 percent with the main dataset using a fixed dictionary and a context-sensitive phonetic transcription. Furthermore, using this level of blurring improves the accuracy in the pilot dataset with a fixed dictionary and derived phonetic transcription by 7.3 and 0.9 percent, respectively.

Another interesting observation is that using the second level of blurring, which also includes blurring tanweens, gives the best overall results. Applying this level of blurring improved the accuracy by 7.1% when using the fixed dictionary and 4.7% when using the generated phonetic transcription with the main dataset. Moreover, the recognition accuracy was increased by 10% using the dictionary and 1.3% when using the phonetic transcription with the pilot dataset.

Using the third level of underspecification which includes blurring shadda on top of blurring short vowels, sukoons, and tanweens, did not yield any further improvement in the performance. The total improvement reported using the main dataset is 6.9 and 2.3 percent with a dictionary and phonetic transcription, respectively. With the pilot dataset, the recognition accuracy increased by 8% with the dictionary and decreased by 0.4% when using the generated phonetic transcription.

The reason behind the observed improvements in the recognition accuracy is mainly related to the minimisation that occurs in the system’s vocabulary. In
the fully diacriticised experiments, for instance, the lexicon includes 310 different vocabulary inputs. Meanwhile, in the underspecified experiments the lexicon only has 253 vocabulary inputs. This makes the recogniser’s task considerably easier with fewer distinctions to make, and thus less chance of making a mistake. For example, in the underspecified experiments, we have an identical representation for three distinct words, the words are: “yaSiIwuna” (meaning “to arrive”), “yuSal~uwn” and “yuSal~uwna” (both meaning “to pray”). These words were all presented as “ySlwn” when underspecified. Looking at the error analysis of the three words in the non-blurred experiment, we found that the word “yaSiIwuna” occurred 9 times and was recognised correctly 4 times. The word “yuSal~uwn” occurred 22 times and was recognised correctly 18 times. The word “yuSal~uwna” occurred 19 times and was recognised correctly 11 times. The total number of occurrences for these words is 50, in 33 contexts they were recognised correctly (66% accuracy). In the meantime, with a total of 50 occurrences for the underspecified form of the words “ySlwn”, we found that it has been correctly recognised 46 times (92% accuracy).

6.4.3 Conclusion

This section set out to determine the impact of using non-diacriticised text sources in building an Arabic speech recogniser. This issue has concerned many researchers in the field, many of whom have yet to agree on whether it has a positive or a negative impact on the system’s performance. Roughly speaking, we found that using non-diacriticised text materials can positively affect the recognition system’s performance. The rate of improvement varies according to the level of underspecification and the dataset used in training and testing the system. Besides the advantage of improving the accuracy of the system, using non-diacriticised texts without any deterioration in accuracy makes it easier to obtain a training corpus for the purpose of designing an Arabic ASR system. This means that we will not have to manually diacriticise the text or even use any of the diacriticising tools. However, in order to derive a context-sensitive phonetic transcription for the text, which is shown to have a significant impact towards improving the results, we will need to have a fully diacriticised text and do the underspecification after generating the phonetic transcription. When using the underspecified text, the output of the recogniser will be very similar to a non-diacriticised written text which can be useful for many ASR applications.
like dictation, subtitling, etc. The non-diacriticised output will be of less help in designing an ASU system since a disambiguation tool has to be provided to restore the missing diacritics and ultimately to find the meaning.

6.5 Speakers’ community experiments

It is widely known that speaker-dependent systems outperform speaker-independent systems, as already discussed in Section 2.1. A typical speaker-dependent ASR system can achieve a significant WER reduction by more than 30% with a comparable speaker-independent ASR system [HL93] [HAH+01].

This is due to the fact that training a speech recognition system with random and different speakers carrying all kinds of variations will make the task of the recogniser much more complicated. For instance, every single speaker has a special vocal anatomy which results in a unique set of speech sounds. In addition to the unique physiological features that everybody has, the speaker’s accent, gender, age, educational level, and speaking style are all major sources of speech variation.

Although speaker-dependent systems report higher recognition accuracy, they are not feasible in every application as they require a training process with the user’s data prior to usage, so all the speaker’s specific data should be available. Furthermore, building speaker-dependent systems requires a large amount of training data. Roughly speaking, speaker-dependent systems perform better but they are not as practical as speaker-independent systems.

Hence, we would like trying to develop a method in order to reduce the gap in performance between the two systems. We argue that the ideal system is one that can identify similarities between speakers to form speaker community groups as a step towards controlling the speaker’s variability effect. The basic idea is to take subsets of the training data as seeds and try to grow communities out of them by clustering speakers with speech similarities. The ideal approach of doing it is to use a general purpose clustering algorithm like $K$-means to group speakers in sets by calculating the distance between them. Despite the simplicity of this algorithm in solving clustering problems, it is not the best way in our case for several reasons. Firstly, it is a time-consuming process because we will need to do too many comparisons with every single pair so as to converge the differences between the speakers, thereby rendering it too distant to be feasible.
Furthermore, this method requires training with a single speaker, which does not normally give a robust acoustic modelling that we can trust.

So, what we did instead was to adopt some predefined classes that were suggested by the literature and see how we can fit speakers into these classes. The work starts by sorting out speakers in the database according to the main classes of gender, region, and age group.

It is widely reported that the mismatch in gender between training speakers and testing speakers significantly affects the ASR system performance [AKZ01]. After classifying our speakers according to their gender, we built a gender-specific model before testing each speaker on both models. Because we do not have sufficient age varieties, as most of our speakers are between the age of 20 and 30 years, the age factor is neglected in our study. Dialect is another important classification and a prime source of variation as discussed in Section 2.2.2.2. For instance, [DDNK97] reported a dramatic decrease in recognition accuracy when the speaker-independent system is trained on the Stockholm dialect and tested on speakers from the Scania region of Sweden. We do not have dialectal variation in our dataset as the speakers were asked to read sentences that were written according to MSA rules. However, having speakers from different Arab regions will result in multiple accent variations for the utterances (for more on the difference between dialect and accent, see 2.2.2.2 and more on the accent's effect on MSA speech can be found in Section 3.1.1). The main accent groups that can be found in our main dataset are Gulf accent and Levantine accent (a detailed description of the speakers can be found in Chapter 4.3).

In this section we will start by considering the gender effect on speaker-independent speech recognition systems. The lack of Levantine male speakers makes it difficult to do rigorous accent-specific experiments. However, the experiments carried out in this section include some investigations into accents.

As reported in Section 6.1, we can achieve a better performance by using the generated phonetic transcription rather than the fixed dictionary. As a result, the generated phonetic transcription will be used in all the experiments reported in this section. A detailed overview of the experiments carried out to identify the best speaker communities is given below.
6.5.1 Gender-dependent experiments

As a first step to find the best community for each speaker, we used gender as a basic predefined category to sort out our speakers.

With a total data comprising 23,060 utterances, 7900 utterances were recorded by 20 male speakers and 15,160 were recorded by 36 female speakers. To ensure a fair distribution of the training size, the female data were pruned in order to have an equal number of utterances and speakers in training each model.

A total of 7900 utterances were used in building the first training model recorded by 20 female speakers. The same number of utterances was used in building the second training model with 20 male speakers. Both systems contain speakers with different accents and from different age groups. All speakers were tested against both models, with the speaker removed from the training set before the model on which they were tested was constructed. Results of the gender-dependent experiments are given in Table 6.7.

In order to confirm the usefulness of training on a specific gender, we carried out some experiments to test each speaker against a mixed-gender model. The model is trained with 7900 utterances by 20 speakers from both genders. Detailed results of the experiments are given in Appendix H. Generally speaking, speakers do not attain their best performance when tested on the mixed-gender training model.

We did, however, notice a striking phenomenon, namely, there were males who did better when tested using a model trained on female speakers, and likewise there were females who did better when tested using a model trained on male speakers (bolded). The testing shows that 19 of the 56 speakers did not do well with their associated gender. This finding has not been previously reported in the literature as far as we are aware.

We then supposed that perhaps the speakers who have been treated wrongly are acoustically close to the other gender. So we worked on extracting the pitch value for each speaker to investigate whether this happened because of the pitch. Pitch frequency is defined as that frequency where the vocal folds vibrate when producing voiced sounds [Cry03]. As already mentioned in Section 2.2.2.1, male speakers tend to have a lower pitch than female speakers. The adopted method for pitch detection was created with the aid of PRAAT.

We established that this is not a matter of pitch frequency, as all our male speakers had a pitch from 106 to 166 Hz, and all our female speakers had a pitch
Table 6.7: Gender-dependent experiments’ results. The first system uses 20 female speakers for the training while the second training system uses 20 male speakers. A total of 56 speakers were used in testing.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Absolute accuracy (%)</th>
<th>ID</th>
<th>Gender</th>
<th>Absolute accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Female model</td>
<td></td>
<td></td>
<td>Male model</td>
</tr>
<tr>
<td>1</td>
<td>male</td>
<td>43.0</td>
<td>29</td>
<td>male</td>
<td>39.7</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>77.9</td>
<td>30</td>
<td>female</td>
<td>31.6</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>79.9</td>
<td>31</td>
<td>male</td>
<td>17.2</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>73.1</td>
<td>32</td>
<td>female</td>
<td>6.7</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>66.5</td>
<td>33</td>
<td>male</td>
<td>86.4</td>
</tr>
<tr>
<td>6</td>
<td>male</td>
<td>63.7</td>
<td>34</td>
<td>female</td>
<td>81.5</td>
</tr>
<tr>
<td>7</td>
<td>male</td>
<td>89.1</td>
<td>35</td>
<td>female</td>
<td>82.6</td>
</tr>
<tr>
<td>8</td>
<td>female</td>
<td>80.8</td>
<td>36</td>
<td>male</td>
<td>76.6</td>
</tr>
<tr>
<td>9</td>
<td>male</td>
<td>80.6</td>
<td>37</td>
<td>female</td>
<td>51.4</td>
</tr>
<tr>
<td>10</td>
<td>female</td>
<td>89.1</td>
<td>38</td>
<td>male</td>
<td>57.5</td>
</tr>
<tr>
<td>11</td>
<td>male</td>
<td>75.0</td>
<td>39</td>
<td>female</td>
<td>50.7</td>
</tr>
<tr>
<td>12</td>
<td>male</td>
<td>4.4</td>
<td>40</td>
<td>female</td>
<td>80.9</td>
</tr>
<tr>
<td>13</td>
<td>male</td>
<td>18.9</td>
<td>41</td>
<td>female</td>
<td>91.9</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>36.9</td>
<td>42</td>
<td>female</td>
<td>77.9</td>
</tr>
<tr>
<td>15</td>
<td>female</td>
<td>83.0</td>
<td>43</td>
<td>female</td>
<td>68.9</td>
</tr>
<tr>
<td>16</td>
<td>female</td>
<td>75.7</td>
<td>44</td>
<td>male</td>
<td>54.4</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>75.7</td>
<td>45</td>
<td>female</td>
<td>91.2</td>
</tr>
<tr>
<td>18</td>
<td>female</td>
<td>83.0</td>
<td>46</td>
<td>female</td>
<td>43.1</td>
</tr>
<tr>
<td>19</td>
<td>female</td>
<td>78.5</td>
<td>47</td>
<td>female</td>
<td>29.6</td>
</tr>
<tr>
<td>20</td>
<td>female</td>
<td>22.6</td>
<td>48</td>
<td>female</td>
<td>88.5</td>
</tr>
<tr>
<td>21</td>
<td>female</td>
<td>80.5</td>
<td>49</td>
<td>female</td>
<td>90.3</td>
</tr>
<tr>
<td>22</td>
<td>male</td>
<td>32.7</td>
<td>50</td>
<td>female</td>
<td>80.9</td>
</tr>
<tr>
<td>23</td>
<td>female</td>
<td>96.8</td>
<td>51</td>
<td>male</td>
<td>64.4</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>80.0</td>
<td>52</td>
<td>male</td>
<td>50.9</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>13.5</td>
<td>53</td>
<td>female</td>
<td>63.8</td>
</tr>
<tr>
<td>26</td>
<td>female</td>
<td>18.3</td>
<td>54</td>
<td>male</td>
<td>52.7</td>
</tr>
<tr>
<td>27</td>
<td>female</td>
<td>54.0</td>
<td>55</td>
<td>female</td>
<td>73.0</td>
</tr>
<tr>
<td>28</td>
<td>female</td>
<td>66.0</td>
<td>56</td>
<td>female</td>
<td>87.9</td>
</tr>
</tbody>
</table>
from 200 to 264 Hz as shown in Figure 6.6.

![Figure 6.6: The average pitch rate for our male and female speakers.](image)

This confirms that although the gender separation criterion works in most of the cases, gender alone does not give the optimal results for all speakers.

We may thus speculate that speakers who have been misclassified are speaking in some way more like the other class. As the study carried out by [Byr94] has found, male and female speakers also differ in their speech rate and the way they produce sounds (for more details, go to 2.2.2.1). Speakers differ according to these stylistic features even though we cannot easily identify such features in speech so as to use them to split people up.

As a way to investigate this problem and propose suitable solutions, we regrouped our speakers according to their best performance and retrained them until we reach the optimal results with the best-speaker combination.

It is worth mentioning here that some of the speakers achieved comparatively poor results. By listening to the recordings, we can explain this poor performance through the existence of background noise or the use of low-quality microphones.
6.5.2 Speakers’ groups

The fact that speakers do not necessarily achieve their best performance when tested against their associated gender model has encouraged us to find other ways than classifying according to the typical predefined classes. Therefore, we regrouped our dataset and added pseudo-females (males who performed better when tested with the female model than with the male one) to female speakers and pseudo-males to male speakers. The first group is made up of 33 speakers (25 females and 8 males), while the second group consists of 23 speakers (12 males and 11 females). The number of sentences is 13,600 and 9460 in the first and second group, respectively. In order to achieve a fair comparison between the two groups and between these experiments with the experiments discussed in the previous section, the dataset was pruned to have as much as possible the same number of speakers and sentences with the same variety in accents. Hence, the first group of speakers consists of 20 speakers (12 females and 8 males) with a total of 7900 utterances. The second group of speakers consists of 20 speakers (10 males and 10 females) also with 7900 utterances. Each individual speaker has been excluded from the training dataset whenever used in testing. The results from individually testing each speaker against the two models are given in Table 6.8.

It is obvious from the results listed in Table 6.8 that 17 out of the 19 pseudo-gender speakers show an improvement when tested against the group to which (we believe) they belong. The improvements in the recognition accuracy range from 0.6% to 26.1%. This confirms that speech similarity exists between those speakers. Apart from that, Speaker 11, a pseudo-female, performed adequately in both models with a 5% increase in WER. On the other hand, speaker 16, a female, has switched to group2 comprising males and pseudo-males with 16.9% improvement in recognition accuracy.

Interestingly, for most of the speakers the accuracy goes up when tested against both models (except the ones marked with a down arrow). The recognition accuracy improvement is between 0.2% and 41.2% when testing the speakers in a group that they do not belong to. This is a surprising result as speakers were expected to perform better only with their group where they have something in common. Nonetheless, they did better with the other group than with the original wrong model, as shown in Table 6.7. This unanticipated finding seems to be difficult to explain, but it might be related to the fact that when a model
Table 6.8: Results of speakers’ group experiments. The first system is trained on female and pseudo-female speakers, while the second one is trained on male and pseudo-male speakers. All of the 56 speakers are used in testing both systems.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Group1 model</th>
<th>Group2 model</th>
<th>ID</th>
<th>Gender</th>
<th>Group1 model</th>
<th>Group2 model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>male</td>
<td>39.7</td>
<td>69.4</td>
<td>29</td>
<td>male</td>
<td>38.9</td>
<td>82.8</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>82.6</td>
<td>89.0</td>
<td>30</td>
<td>female</td>
<td>38.2</td>
<td>70.6</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>82.9</td>
<td>66.0</td>
<td>31</td>
<td>male</td>
<td>23.5</td>
<td>46.2</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>74.4</td>
<td>53.1</td>
<td>32</td>
<td>female</td>
<td>12.4</td>
<td>46.6</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>75.6</td>
<td>69.9</td>
<td>33</td>
<td>male</td>
<td>89.1</td>
<td>46.0</td>
</tr>
<tr>
<td>6</td>
<td>male</td>
<td>76.4</td>
<td>89.3</td>
<td>34</td>
<td>female</td>
<td>83.8</td>
<td>66.9</td>
</tr>
<tr>
<td>7</td>
<td>male</td>
<td>92.3</td>
<td>77.3</td>
<td>35</td>
<td>female</td>
<td>84.4</td>
<td>72.1</td>
</tr>
<tr>
<td>8</td>
<td>female</td>
<td>81.4</td>
<td>76.5</td>
<td>36</td>
<td>male</td>
<td>79.8</td>
<td>74.7</td>
</tr>
<tr>
<td>9</td>
<td>male</td>
<td>86.6</td>
<td>89.3</td>
<td>37</td>
<td>female</td>
<td>55.4</td>
<td>77.5</td>
</tr>
<tr>
<td>10</td>
<td>female</td>
<td>94.4</td>
<td>84.3</td>
<td>38</td>
<td>male</td>
<td>64.2</td>
<td>56.3</td>
</tr>
<tr>
<td>11</td>
<td>male</td>
<td>70.0</td>
<td>70.0</td>
<td>39</td>
<td>female</td>
<td>64.5</td>
<td>56.2</td>
</tr>
<tr>
<td>12</td>
<td>male</td>
<td>7.3</td>
<td>22.8</td>
<td>40</td>
<td>female</td>
<td>81.7</td>
<td>79.1</td>
</tr>
<tr>
<td>13</td>
<td>male</td>
<td>23.6</td>
<td>53.1</td>
<td>41</td>
<td>female</td>
<td>93.4</td>
<td>84.0</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>37.1</td>
<td>82.6</td>
<td>42</td>
<td>female</td>
<td>81.3</td>
<td>86.2</td>
</tr>
<tr>
<td>15</td>
<td>female</td>
<td>85.6</td>
<td>91.9</td>
<td>43</td>
<td>female</td>
<td>73.9</td>
<td>52.8</td>
</tr>
<tr>
<td>16</td>
<td>female</td>
<td>81.5</td>
<td>92.6</td>
<td>44</td>
<td>male</td>
<td>62.4</td>
<td>70.3</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>87.3</td>
<td>65.0</td>
<td>45</td>
<td>female</td>
<td>92.3</td>
<td>88.1</td>
</tr>
<tr>
<td>18</td>
<td>female</td>
<td>85.7</td>
<td>57.5</td>
<td>46</td>
<td>female</td>
<td>46.3</td>
<td>89.1</td>
</tr>
<tr>
<td>19</td>
<td>female</td>
<td>83.6</td>
<td>45.8</td>
<td>47</td>
<td>female</td>
<td>41.5</td>
<td>65.0</td>
</tr>
<tr>
<td>20</td>
<td>female</td>
<td>14.2</td>
<td>73.2</td>
<td>48</td>
<td>female</td>
<td>91.0</td>
<td>82.4</td>
</tr>
<tr>
<td>21</td>
<td>female</td>
<td>86.1</td>
<td>51.2</td>
<td>49</td>
<td>female</td>
<td>91.9</td>
<td>87.8</td>
</tr>
<tr>
<td>22</td>
<td>male</td>
<td>48.0</td>
<td>35.3</td>
<td>50</td>
<td>female</td>
<td>82.5</td>
<td>69.7</td>
</tr>
<tr>
<td>23</td>
<td>female</td>
<td>94.2</td>
<td>79.2</td>
<td>51</td>
<td>male</td>
<td>68.1</td>
<td>58.3</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>78.5</td>
<td>86.3</td>
<td>52</td>
<td>male</td>
<td>59.1</td>
<td>74.2</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>12.5</td>
<td>23.5</td>
<td>53</td>
<td>female</td>
<td>65.2</td>
<td>59.9</td>
</tr>
<tr>
<td>26</td>
<td>female</td>
<td>21.9</td>
<td>21.1</td>
<td>54</td>
<td>male</td>
<td>54.8</td>
<td>76.6</td>
</tr>
<tr>
<td>27</td>
<td>female</td>
<td>55.6</td>
<td>65.2</td>
<td>55</td>
<td>female</td>
<td>79.4</td>
<td>62.9</td>
</tr>
<tr>
<td>28</td>
<td>female</td>
<td>67.2</td>
<td>61.1</td>
<td>56</td>
<td>female</td>
<td>89.0</td>
<td>84.2</td>
</tr>
</tbody>
</table>
is built using speakers with many features in common, the model itself will be
tidier. Therefore, although some speakers do not belong to it, the model behaves
in a cleaner manner and somehow this compensates for the difference between
the speaker and the other group.

6.5.3 Is it an accent-specific effect?

We wondered whether there is any accent-specific effect hidden among the speak-
ers which drives us to these results. In other words, did the speakers who did
better with the model that was trained with the other gender only do so be-
cause that model contained their accent? For this reason, we carried out the
same testing but with a female model and male model that were trained using
strictly speakers from the Gulf region. Our dataset contains 18 male and 21 fe-
male speakers with the Gulf accent. The total number of utterances is 6077 and
6720 for male and female speakers respectively. Three speakers were eliminated
from the female training set to minimise the variation sources. The final number
of utterances in the female model after deleting three speakers was 6060.

Looking at the results obtained from testing on a gender-specific and an
accent-specific model, we can confirm that it is not the accent effect as the same
speakers have been misclassified (except speaker 36). This eliminates the possi-
bility that the effects discussed in Section 6.8 were due to biased distributions of
accents within the split based on gender.

More interestingly, most of the Gulf speakers achieved a significant improve-
ment when compared to their performance with the models described in the
previous two sections. 36 speakers performed better compared to the gender-
dependent model with an average improvement of 8.7%. On the other hand, 31
speakers reported a better performance compared to the groups model with an
average improvement of 5.2%. A table of the results is included in Appendix G.

This significant improvement encouraged us to carry out further adjustmen-
to the model to see if we can achieve better results using a combination of the best
speakers. We re-ran the group experiments but this time the model is trained only
on the Gulf-accented speakers who belong to the same group. In the collected
dataset, we have 39 speakers from the Gulf region. 21 speakers achieved better
with the first group (14 females and 7 males) and 18 speakers achieved better
with the second group (11 males and 7 females). The 21 speakers who belong to
the first group have recorded a total of 6116 utterances, while the 18 speakers in

179
The second group recorded 6507 utterances. Because we want to build two models with the same number of speakers and utterances, we removed 4 speakers from the first group’s training set who have a comparatively small number of utterances. After their removal, the total number of utterances becomes 6060, obtained from 17 speakers. We have also deleted one speaker from the second training set, so that it contains 6060 utterances recorded by 17 speakers. After preparing two models that share the same number of utterances and speakers, we also tested each Gulf speaker against both models to investigate the effect of constraining the accent in building the recognition system. Results of the experiments are shown in Table 6.9.

The results presented in Table 6.9 reveal two main outcomes. Firstly, all speakers with no exceptions achieved better with the group to which (we believed) they belong (pseudo-females with females and pseudo-males with males). Furthermore, compared to the results shown in Table 6.8, all the Gulf accented

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Absolute accuracy</th>
<th>Group1</th>
<th>Group2</th>
<th>ID</th>
<th>Gender</th>
<th>Absolute accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>male</td>
<td>59.2</td>
<td>78.0</td>
<td></td>
<td>32</td>
<td>female</td>
<td>38.5</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>85.3</td>
<td>96.1</td>
<td></td>
<td>33</td>
<td>male</td>
<td>94.3</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>88.7</td>
<td>71.1</td>
<td></td>
<td>36</td>
<td>male</td>
<td>86.4</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>79.0</td>
<td>64.2</td>
<td></td>
<td>38</td>
<td>male</td>
<td>73.4</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>85.4</td>
<td>73.8</td>
<td></td>
<td>40</td>
<td>female</td>
<td>92.4</td>
</tr>
<tr>
<td>6</td>
<td>male</td>
<td>84.2</td>
<td>92.1</td>
<td></td>
<td>41</td>
<td>female</td>
<td>96.7</td>
</tr>
<tr>
<td>7</td>
<td>male</td>
<td>88.9</td>
<td>84</td>
<td></td>
<td>42</td>
<td>female</td>
<td>88.4</td>
</tr>
<tr>
<td>11</td>
<td>male</td>
<td>93.3</td>
<td>76.0</td>
<td></td>
<td>43</td>
<td>female</td>
<td>80.4</td>
</tr>
<tr>
<td>12</td>
<td>male</td>
<td>33.1</td>
<td>54.0</td>
<td></td>
<td>44</td>
<td>male</td>
<td>60.9</td>
</tr>
<tr>
<td>13</td>
<td>male</td>
<td>49.9</td>
<td>78.1</td>
<td></td>
<td>45</td>
<td>female</td>
<td>93.4</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>65.2</td>
<td>93.1</td>
<td></td>
<td>47</td>
<td>female</td>
<td>52.7</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>92.8</td>
<td>79.0</td>
<td></td>
<td>48</td>
<td>female</td>
<td>94.1</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>87.1</td>
<td>93.8</td>
<td></td>
<td>50</td>
<td>female</td>
<td>93.5</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>31.7</td>
<td>51.3</td>
<td></td>
<td>51</td>
<td>male</td>
<td>69.7</td>
</tr>
<tr>
<td>26</td>
<td>female</td>
<td>34.1</td>
<td>33.8</td>
<td></td>
<td>52</td>
<td>male</td>
<td>70.6</td>
</tr>
<tr>
<td>27</td>
<td>female</td>
<td>81.1</td>
<td>87.4</td>
<td></td>
<td>53</td>
<td>female</td>
<td>79.1</td>
</tr>
<tr>
<td>28</td>
<td>female</td>
<td>83.9</td>
<td>81.8</td>
<td></td>
<td>54</td>
<td>male</td>
<td>69.4</td>
</tr>
<tr>
<td>29</td>
<td>male</td>
<td>61.3</td>
<td>89.7</td>
<td></td>
<td>55</td>
<td>female</td>
<td>88.4</td>
</tr>
<tr>
<td>30</td>
<td>female</td>
<td>74.9</td>
<td>90.2</td>
<td></td>
<td>56</td>
<td>female</td>
<td>93.2</td>
</tr>
<tr>
<td>31</td>
<td>male</td>
<td>39.3</td>
<td>62.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
speakers in these experiments performed better when tested against both group models (except speakers 45 and 48 who did not perform better with the other group). This significant overall improvement might be explained in terms of the positive effect of training and testing with the same accent when building an ASR system.

By looking at the best score for each speaker, we can confirm that all speakers (except speaker number 7) achieved better with an average improvement of 10.5% in recognition performance. This is strong experimental evidence of the effect of restricting speakers’ accents while maintaining the best matching speakers’ groups. We do not have enough data to determine whether a parallel set of experiments aimed at regrouping people according to their accent would be worth the effort. However, the fact that every speaker improved when the training set was accent-specific suggests that there will not be many people who are wrongly classified.

6.5.4 Conclusion

The experimental work in this section was carried out to determine the effect of defining speakers’ communities on improving the recognition performance. The work starts by sorting people according to their gender and then building gender-dependent training models. Compared to a model trained with mixed-genders, most speakers achieved better when tested on a model trained only with their gender. Interestingly, we found that about a third of speakers achieved better with the other gender. This led the research to carry out further testing on models that are trained on speakers with similarities. The majority of speakers achieved better in their associated group with an interesting improvement in their overall performance. In order to exclude the possibility of having an accent’s effect on the results, another testing phase was carried out on models that trained on Gulf-accented female and male speakers, respectively. In general, most of the speakers still act in the same way as was found in the first experimental set. The improvements observed in accent-specific experiments encouraged us to build new training models based on the idea of grouping speakers with similarities only in terms of the Gulf accent. A noticeable improvement in recognition performance was reported. We may thus conclude by saying that looking for similarities between speakers and trying to grow speakers’ communities out of these similarities will undoubtedly improve the performance of the ASR system.
6.6 Influence of size of training data

Accurate acoustic modelling contributes greatly to improving the ASR system’s performance. It is widely assumed that the more training data is used in building the acoustic models, the better the recognition performance. However, whilst this seems plausible *prima facie*, we have not found many studies in the literature that explicitly verify this assumption. One of the few studies concerning the effect of the size of training materials has been made by [EM99]. These researchers have observed a consistent improvement in recognition accuracy when increasing the amount of the training data. The average decrease in WER was approximately 5.3% in the model that was trained with 74 hours of data compared to the model that used only 9.25 hours of data. Figure 6.7 shows the improvement in recognition performance as reported in [EM99].

![Figure 6.7: Average WER percentage for speech recognition system with differently-sized training materials as reported by [EM99].](image)

The relationship between the WER and the amount of training data is also demonstrated by Lamel et al. who used training corpora that ranged in duration from 10 minutes to 140 hours [LGA00] [LGA02]. A summary of the improvements reported in the two published papers is given in Table 6.10 and Table 6.11.

[Pov03], in his PhD thesis, studied the training data size as a contributing factor in reducing the WER. With a widely varying amount of data (from 1.125 hours to 265 hours), the researcher has confirmed the positive effect of increasing the training data amount in reducing the WER. Figure 6.8 provides a logarithmic scale for the amount of training data as given by the researcher.

182
Table 6.10: WER for increasing quantity of training data as reported by [LGA00].

<table>
<thead>
<tr>
<th>Hrs</th>
<th>Ave. WER (%)</th>
<th>Hrs</th>
<th>Ave. WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>26.4</td>
<td>6</td>
<td>25.7</td>
</tr>
<tr>
<td>17</td>
<td>25.2</td>
<td>13</td>
<td>23.7</td>
</tr>
<tr>
<td>28</td>
<td>24.3</td>
<td>21</td>
<td>22.5</td>
</tr>
<tr>
<td>76</td>
<td>22.4</td>
<td>57</td>
<td>21.1</td>
</tr>
<tr>
<td>140</td>
<td>21.0</td>
<td>108</td>
<td>19.9</td>
</tr>
</tbody>
</table>

Table 6.11: WER for varying amounts of training data in supervised, unsupervised, and unsupervised with reduced language model training acoustic models as shown in [LGA02].

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Unsupervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs WER</td>
<td>Hrs WER</td>
<td>Hrs WER</td>
</tr>
<tr>
<td>10min 53.1%</td>
<td>4 37.3%</td>
<td>10min 65.3%</td>
</tr>
<tr>
<td>1 33.3%</td>
<td>12 31.7%</td>
<td>4 54.1%</td>
</tr>
<tr>
<td>33 20.7%</td>
<td>27 27.9%</td>
<td>12 47.7%</td>
</tr>
<tr>
<td>67 19.1%</td>
<td>53 26%</td>
<td>27 43.7%</td>
</tr>
<tr>
<td>123 18%</td>
<td>135 23.4%</td>
<td>53 41.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>103 39.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>135 37.4%</td>
</tr>
</tbody>
</table>
As with the experiments carried out in the previous sections, we wanted to find out whether these common observations would be replicated in our data. We therefore split our dataset, which contains 23,060 utterances (about 25 hours of recordings), to see what impact it would have on the system’s performance. In the experiments described here, we will mainly be using the generated phonetic transcription in training the system. However, the fixed dictionary-based experiments are also provided in the first experiment set to make sure that the obtained results are not an effect of using the generated phonetic transcription.

### 6.6.1 Experimental design

We worked on splitting our main corpus, so we started by training the system with 2.7 hours of recording. The amount of training size is increased gradually to determine what impact it would have on the system’s performance. In splitting the corpus, we were careful to maintain the balance between speakers and vocabulary in every experiment to avoid biased results. One speaker was excluded from the corpus to be used in testing each system with a total of 954 recordings. Two versions of the system are compared: one with a fixed dictionary and another using the phonetic transcription generated from the developed grapheme-to-allophone system.
Table 6.12: Accuracy rate for varying amounts of training data with a fixed dictionary-based system and a generated phonetic transcription-based system.

<table>
<thead>
<tr>
<th>Training size</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs</td>
<td>Utterances</td>
</tr>
<tr>
<td>2.7</td>
<td>2401</td>
</tr>
<tr>
<td>4.9</td>
<td>4401</td>
</tr>
<tr>
<td>7.0</td>
<td>6326</td>
</tr>
<tr>
<td>9.2</td>
<td>8315</td>
</tr>
<tr>
<td>11.5</td>
<td>10356</td>
</tr>
<tr>
<td>13.7</td>
<td>12342</td>
</tr>
<tr>
<td>16.0</td>
<td>14467</td>
</tr>
<tr>
<td>18.7</td>
<td>16861</td>
</tr>
<tr>
<td>20.2</td>
<td>18236</td>
</tr>
<tr>
<td>22.0</td>
<td>19822</td>
</tr>
<tr>
<td>24.3</td>
<td>22106</td>
</tr>
</tbody>
</table>

6.6.2 Results

Details of the results are given in Table 6.12 and Figure 6.9.

Figure 6.9: Accuracy rate for speech recognition system with different-sized training materials.

Table 6.12 compares the results obtained from the fixed dictionary-based system and the system based on the generated phonetic transcription. The results vary greatly depending on the size of the training data. It is apparent from the provided table and figure that the recognition accuracy increased consistently from the smallest training size 2.7 hours to a 16-hours training size, with
27.4% and 38.5% improvements in the fixed dictionary-based system and phonetic transcription-based system, respectively. However, there is a clear trend of decline after this point until we reach the largest training size. The system’s performance deteriorates by 16% in the last 4 systems. This is a surprising result as the expectation was to have seen consistent improvement in the system’s performance as the training data increased: “there is no data like more data” [Moo03].

This unexpected deterioration in the system’s performance can have two explanations. The first is that we are overtraining\(^2\) the system, which consequently affects performance in a negative manner. The overtraining may result from training on the same examples too many times with the same speakers. In order to investigate the overtraining possibility, we removed all the repetitions in the corpus so that each speaker has a maximum of 300 sentences. This process results in a total of 7613 utterances (after excluding the testing speaker). These utterances were split into five phases to see how increasing the data without repeated utterances will affect the recognition performance. However, no clear evidence of overtraining was found as the results were erratic.

The other possible reason for performance deterioration is that training on noisy recordings may affect the stability of the system. The speaker used in the testing can also have an impact on the reported results. For this reason, we filtered our database so that it includes only speakers with recordings above 700 utterances. This is to insure that we do not have bad quality recordings in our database. The filtered database contains a total of 18,580 utterances from 20 speakers. 800 sentences were reserved for testing, with 4 speakers (2 males and 2 females) having different accents. The filtered database was split by size so that we start the first experiment with 1500 utterances (1.6 hours) recorded by twenty speakers. The training data size increases gradually until we reach the maximum point where we can go with the available data (17,780 utterances which are equivalent to 19.7 hours of recordings) recorded by the same speakers. Results of the experiments are given in Table 6.13.

Results reported in Table 6.13 are quite revealing. For instance, unlike the results reported in Table 6.12, there is a consistent improvement in the recognition

\(^2\)More details about over training can be found in [Alp04].
Table 6.13: Accuracy rate for varying amount of training data in the filtered database.

<table>
<thead>
<tr>
<th>Training size Hrs</th>
<th>Utterances</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6</td>
<td>1500</td>
<td>66.3</td>
</tr>
<tr>
<td>2.8</td>
<td>2540</td>
<td>69.2</td>
</tr>
<tr>
<td>5.6</td>
<td>5080</td>
<td>71.3</td>
</tr>
<tr>
<td>8.4</td>
<td>7620</td>
<td>73.6</td>
</tr>
<tr>
<td>11.2</td>
<td>10160</td>
<td>74.8</td>
</tr>
<tr>
<td>14.1</td>
<td>12700</td>
<td>76.3</td>
</tr>
<tr>
<td>16.9</td>
<td>15240</td>
<td>80.9</td>
</tr>
<tr>
<td>19.7</td>
<td>17780</td>
<td>80.7</td>
</tr>
</tbody>
</table>

accuracy when increasing the training data. However, this noticeable improvement in performance has levelled out in the last experiment where the performance deteriorated slightly from 80.9% to 80.7% (as can be seen in Figure 6.10). This deterioration (or might be called stability as it is not significant) in the system’s performance might be explained in terms of the sufficiency in training the system with these limited sentences (in vocabulary and grammar). Although training with more data would be useful, it seems that there may be a point where the data suffices.

Figure 6.10: Accuracy rate for speech recognition system with different-sized filtered training materials.
6.6.3 Conclusion

This section has investigated the effect of training size on the performance of the ASR system. The most obvious finding to emerge from the experiments discussed in this section is that increasing the amount of training data is not always useful. We saw in the original set of experiments which used the whole data corpus in training the system that we may end up with overtraining which results in deterioration of the system’s performance. On the other hand, with the filtered and well-balanced data, we achieved a noticeable overall increase in accuracy and linear improvement in performance by up to 16.9 hours of training data. Taken together, these results suggest that there is no data like more good quality data. Our concern should be shifted from working on collecting more and more data to finding the best ways of exploiting the available data. This section together with Section 6.5 suggest two ideas for making the best use of data. The first is finding similarities between speakers and sorting them accordingly to build a robust ASR system. The second suggests cleaning up the data by excluding poor quality recordings before training the system to obtain better accuracy.
Chapter 7

Conclusion and future work

In this thesis we have sought to investigate different factors that can affect the performance of the ASR system. The proposed work is part of a larger project aims at developing an ASU system dedicated to MSA. Developing an ASU system requires, first of all, having a high performance speech recogniser. A large amount of work and effort is needed to sort out the many problems raised from different sources that hinder developing an accurate recognition system. In this thesis, various Arabic-related problems have been considered and investigated in order to improve the ASR system’s performance. Furthermore, the research addressed some technical problems that might have an effect on the system’s performance.

The thesis starts by introducing the research problems and objectives and outlining the thesis structure, as presented in Chapter 1. In Chapter 2, a general overview of the ASR systems is given, highlighting its main applications, types of ASR systems, challenges confronted when developing an ASR system, the general architecture of the ASR system, and an overview of the available recognition toolkits. Arabic-related problems are discussed in Chapter 3 from the perspective of ASR. A survey of works dedicated to developing an Arabic ASR is also given in this chapter. Chapter 4 lays out the experimental dimensions of the developed systems by introducing the main component for the acoustic analysis which is HMMs, describing how the recognition toolkit (the HTK) works, introducing the amendments made into the HTK processing stages to accommodate the developed grapheme-to-allophone rules, and finally describing the collected datasets that were used in running the research experiments. After the introduction of ASR, Arabic language, and the experimental set up details, Chapter 5
introduces the work done in developing a grapheme-to-allophone conversion system to generate a context-sensitive phonetic transcription that can replace the fixed dictionary in training an ASR system. The structure of the developed system is reviewed and the related examples are given before testing and validating the proposed system have taken place. Finally, Chapter 6 describes the experiments carried out to investigate some factors believed to have an effect on the recognition system performance.

This chapter revisits the the research objectives that were set in Chapter 1, pointing out the main findings and contributions of the thesis. The chapter also presents possible directions for future work.

7.1 Objectives revisited

- The first objective of the thesis was to develop a set of comprehensive conversion rules to automatically produce a context-sensitive phonetic transcription for a given Arabic text. These rules were expected to capture inter-word and cross-word pronunciation variations to control the boundary effect on pronouncing letters in MSA. These pronunciation rules were collected from the literature, along with some novel pronunciation rules obtained by analysing Arabic speech from different dialect speakers. The rules were implemented as a set of two-level finite state automata. Testing the grapheme-to-allophone system shows that it produces the phonetic transcriptions that correspond to our expectations in all the cases except with items that do not follow the phonetic patterns of Arabic such as misspelled items, proper names, abbreviations, and symbols. In order to verify that these phonological aspects have audible consequences in natural speech, an independent human annotator with experience in Arabic phonetics was asked to transcribe a total of 100 sentences spoken by 5 native Arabic speakers from different dialectal origins. Comparing the annotator's transcription with the system's output shows that they do match in 98.3% of the cases. After testing and validating the developed rules, the grapheme-to-allophone system was accommodated within the speech recognition system in order to achieve the goal of creating these rules, which is to generate the phonetic transcription needed for the speech recogniser. Different processing steps were canceled or modified when running the speech recogniser,
namely those that use the pronunciation dictionary to obtain the phone level transcription. A set of experiments were carried out using two different datasets to investigate the effectiveness of using the generated phonetic transcription. Three different systems were developed to evaluate the effectiveness of the proposed approach. The first system used a typical fixed dictionary with a single pronunciation for each word. The second system used a multi-pronunciation dictionary, which gives many possible pronunciations of the words. The third system used the generated context-sensitive phonetic transcription. The experiments results show that by applying the developed grapheme-to-allophone algorithm in building a speech recogniser a significant improvement of 10.1% was achieved compared to the standard use of the fixed dictionary. Furthermore, using the context-sensitive phonetic transcription achieved 5.2% improvement in recognition accuracy compared to the use of a multi-pronunciation dictionary. The research confirms that using the generated phonetic transcription outperforms the use of the standard fixed dictionary and the multi-pronunciation dictionary which is widely believed to be the best method for capturing pronunciation variations. Besides the great advantage of improving the recognition system’s performance, another advantage of the grapheme-to-allophone system is to create the phonetic transcription that the speech recogniser needs without the effort of manually creating a pronunciation dictionary for the system’s vocabulary, especially with the lack of available Arabic pronunciation dictionaries.

In order to tackle the problem of having too many similar forms of a single word, the research investigates the use of an underspecified representation of the text in training and testing the recognition system. The underspecification is done by removing diacritics from the words. Removing diacritics from the text was done gradually in order to determine the impact of removing each diacritic group solely. The research found that using non-diacriticised text materials can positively affect the recognition system’s performance. The rate of improvement varies according to the level of underspecification and the dataset used in training and testing the system. For instance, using the non-blurred version of the main dataset and the generated phonetic transcription, the system reported 69.4% accuracy. On the other hand, removing short vowels, sukoon, and tanween when using
same settings brings the accuracy to 74.1%. Besides the great advantage of improving the system’s performance, using non-diacriticised texts without any deterioration in accuracy makes it easier to obtain a training corpus to design an Arabic ASR system. This should compensate for the lack of available fully-diacriticised corpora and it means that we will not have to manually diacriticise the text or even use any of the diacriticising tools.

- The third objective of the research was to test another way of tackling the problem of complex morphology of Arabic. The proposed method tests using two different units in building HMMs, phonemes and words. The unit recognition accuracy was found to be much higher when using phoneme level HMMs. However, by looking at the sentence recognition accuracy we established that using phonemes as the base units is not as good as it looks to be. This was further investigated by analysing the confusion matrix of the system’s output which confirms the accidental recognition of certain phonemes. The reason behind this apparent significant improvement is the fairly high possibility that phonemes would occur in a certain order, which gives the recogniser a better chance to come up with the right phoneme. This means that using phoneme-level based HMMs does not outperform the standard use of word-level HMMs based system.

- The fourth objective of the research was to investigate the effectiveness of training the system on sub-populations of speakers rather than the general population. The motivation behind this approach is to reduce the amount of variation among speakers, which negatively influences the system’s performance. The speaker sorting process starts by using predefined categories such as gender to group the speakers and train the system on subpopulation corpora. In the first such experiment, each speaker was tested against the mixed-gender corpus and the gender-specific corpus. Testing results show that, as widely reported, most speakers achieved better when tested on a model trained only with their gender compared to a model trained with mixed-genders. Results analysis also shows, however, that about third of the speakers perform better with the other gender. This interesting finding encouraged us to carry out further testing. This time the system is trained only on speakers with similarities (females and pseudo-females model, males and pseudo-males model). This method shows an interesting improvement
in the overall performance and the majority of speakers achieved better when tested against their associated group. In order to exclude the possibility of accent effects on the results, another testing phase was carried out on models trained on Gulf-accented female and male speakers. In general, most of the speakers still act in the same way as was found in the first experimental set with a noticeable increase in the accuracy. The improvements observed in accent-specific experiments motivated us to build new training models based on the idea of grouping speakers with similarities only in terms of the Gulf accent. A noticeable improvement in recognition performance was reported. We may thus conclude by saying that looking for similarities between speakers and trying to grow speakers’ communities out of these similarities will undoubtedly improve the performance of the ASR system.

• Besides the previously mentioned problems, the research also investigated finding the correlation between the size of the training corpus and the performance of the system. Firstly, the whole main study corpus was arbitrarily split into different sizes maintaining the same number of speakers in every subset. The experiments’ results show that the accuracy of the system increases as we increase the training size, but that after a while the accuracy starts to deteriorate. This deterioration might be explained in terms of overtraining the system, which results from training on the same examples too many times with the same speakers. The other possible explanation is the existence of some noisy files in the training set which affect the stability of the performance. For this reason, we cleaned up the database so that it includes only speakers who recorded above 700 utterances to insure the quality of the recordings. Results of the experiments carried out on different sizes subsets confirm that there is a consistent improvement in the recognition accuracy when increasing the training data. These results suggest that there is no data like more good quality data. Our concern should be shifted from working on collecting more and more data to finding the best ways of exploiting the available data.
7.2 Future work

There are a number of possible directions for future research and also some remaining open questions related to the field that are worth further investigation. For instance, the speech recognition systems introduced in this research are created with the aid of the HTK as a portable toolkit for building and manipulating HMMs. It will be worth trying other available recognition toolkits such as CMU Sphinx to ensure that our approach is robust even when applied in different recognition engines. CMU Sphinx toolkit is of particular interest to us due to the fact that it is open source and well-documented and maintained.

Furthermore, the main focus of this work was on developing a substantial set of grapheme-to-allophone conversion rules to produce a context-sensitive phonetic transcription for a given MSA text. This could be followed up by looking at the pronunciation variation in MSA speech caused by the dialectal background of the speaker. Especially that when testing the rules, the dialect was found to be a major source of mismatch between the generated phonetic transcription and the transcription obtained from a phonetician. Since using the generated phonetic transcription in training the ASR system was found to have a significant impact in improving the recognition accuracy, we assume that extending the scope of the rules will lead to further enhancements.

In addition, it would be advantageous to extend the work by developing dialect-specific grapheme-to-allophone rules in a similar framework using the common dialect’s pronunciation rules. Besides the possible advantage of improving the ASR system’s performance, this system could be used to cover the lack of available pronunciation dictionaries dedicated to dialects which are essential in building ASR systems. Creating multiple pronunciation modules will presumably require finding a way to automatically identify the speaker’s dialect in order to switch to the desired model. Investigating ways of automatically identifying speaker’s dialect is a substantial work.

Moreover, the research suggests the use of underspecified transcription to tackle the problem of having many similar words in training and testing the speech recogniser. The underspecified form is similar to the use of non-diacriticised text materials. This method shows a considerable improvement in the system’s performance. This improvement is encouraging to integrate the recogniser into the ASU system and test the effectiveness of using underspecification in understanding the speech.
Bibliography


[AAKEAM12] D. AbuZeina, W. Al-Khatib, M. Elshafei, and H. Al-Muhtaseb. Within-word pronunciation variation modeling for Arabic ASRs:


Appendix A

Text dataset for the pilot corpus

1. yabsu'Tu All'hu Alrizq
2. tasAqaTat Alvuluwju fiy AlEaSimap
3. >a$jAru AlriyADi Tawiylap
4. muwaZ~afuw Albanki mujtahiduwn
5. AirtifAEu >asE Ari Al*ahab
6. ziAAdapu Eadadi Al*ukuwri biAlnisbapi lil<inAv
7. AintabihuwA li<i$Arapu Almuruwr
8. AinqA*u AlmutaDaririyna mina AlfayaDanAt
9. lam >atamak~an min ru&yapi Aljaziyrap
10. tujraY AIAintixAbAtu fiy mawEdihA
11. AinaSarat <irAdapu Al$aEb
12. AiDrAbu AIEm³~Ali lita>ax~uri Sarfi rawAtibihim
13. Eawdapu AI AintixAbAti liAit~iHAdi AljamEiy~ati AltaEAwuniy~api
14. AiDTar~a Al~AEibu lixuruwjii mina AlnubArap
15. AIdawri AlvAniy limajlisí AljamìEap
16. >ajAba AlTAlibu Ean kAf~api Al>as{ilap
17. badŠu AI AintiHAnAti gadAF Al>arbiEAŠ
18. takal~ama AlxaTiybu min EalaY Alminbar
19. AISilqu min faDA}ili Al>axIAq
20. xu* vulva Almablæ
Appendix B

Details of the speakers participated in the pilot corpus
Table B.1: Details of the speakers participated in recording the pilot corpus. Showing their gender, regional accent, age, total number of recordings, number of repetitions, and the average pitch.

<table>
<thead>
<tr>
<th>ID</th>
<th>gender</th>
<th>accent</th>
<th>age</th>
<th>number of utterances</th>
<th>number of repetitions</th>
<th>ave. pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>female</td>
<td>Levantine</td>
<td>25-30</td>
<td>40</td>
<td>20</td>
<td>218</td>
</tr>
<tr>
<td>2</td>
<td>female</td>
<td>Egypt</td>
<td>20-25</td>
<td>45</td>
<td>25</td>
<td>202</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>Gulf</td>
<td>35-40</td>
<td>40</td>
<td>20</td>
<td>257</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>Gulf</td>
<td>45-50</td>
<td>40</td>
<td>20</td>
<td>213</td>
</tr>
<tr>
<td>5</td>
<td>female</td>
<td>Gulf</td>
<td>20-25</td>
<td>200</td>
<td>180</td>
<td>249</td>
</tr>
<tr>
<td>6</td>
<td>female</td>
<td>North Africa</td>
<td>40-45</td>
<td>40</td>
<td>20</td>
<td>198</td>
</tr>
<tr>
<td>7</td>
<td>female</td>
<td>North Africa</td>
<td>40-45</td>
<td>20</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>8</td>
<td>female</td>
<td>Gulf</td>
<td>30-35</td>
<td>80</td>
<td>60</td>
<td>256</td>
</tr>
<tr>
<td>9</td>
<td>female</td>
<td>Levantine</td>
<td>20-25</td>
<td>40</td>
<td>20</td>
<td>191</td>
</tr>
<tr>
<td>10</td>
<td>female</td>
<td>Iraq</td>
<td>25-30</td>
<td>64</td>
<td>44</td>
<td>239</td>
</tr>
<tr>
<td>11</td>
<td>female</td>
<td>Gulf</td>
<td>45-50</td>
<td>82</td>
<td>62</td>
<td>209</td>
</tr>
<tr>
<td>12</td>
<td>female</td>
<td>North Africa</td>
<td>45-50</td>
<td>40</td>
<td>20</td>
<td>229</td>
</tr>
<tr>
<td>13</td>
<td>female</td>
<td>Gulf</td>
<td>50-55</td>
<td>40</td>
<td>20</td>
<td>232</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>Gulf</td>
<td>30-35</td>
<td>181</td>
<td>161</td>
<td>162</td>
</tr>
<tr>
<td>15</td>
<td>female</td>
<td>Gulf</td>
<td>55-60</td>
<td>41</td>
<td>21</td>
<td>241</td>
</tr>
<tr>
<td>16</td>
<td>male</td>
<td>Iraq</td>
<td>30-35</td>
<td>91</td>
<td>71</td>
<td>132</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>Iraq</td>
<td>25-30</td>
<td>60</td>
<td>40</td>
<td>240</td>
</tr>
<tr>
<td>18</td>
<td>female</td>
<td>Iraq</td>
<td>40-45</td>
<td>40</td>
<td>20</td>
<td>228</td>
</tr>
<tr>
<td>19</td>
<td>male</td>
<td>Gulf</td>
<td>30-35</td>
<td>120</td>
<td>100</td>
<td>139</td>
</tr>
<tr>
<td>20</td>
<td>female</td>
<td>North Africa</td>
<td>35-40</td>
<td>40</td>
<td>20</td>
<td>263</td>
</tr>
<tr>
<td>21</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>100</td>
<td>80</td>
<td>264</td>
</tr>
<tr>
<td>22</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>46</td>
<td>26</td>
<td>238</td>
</tr>
<tr>
<td>23</td>
<td>female</td>
<td>Egypt</td>
<td>40-45</td>
<td>20</td>
<td>0</td>
<td>233</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>Levantine</td>
<td>35-40</td>
<td>41</td>
<td>21</td>
<td>217</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>80</td>
<td>60</td>
<td>214</td>
</tr>
<tr>
<td>26</td>
<td>male</td>
<td>Gulf</td>
<td>20-25</td>
<td>120</td>
<td>100</td>
<td>119</td>
</tr>
<tr>
<td>27</td>
<td>male</td>
<td>Levantine</td>
<td>30-35</td>
<td>20</td>
<td>0</td>
<td>135</td>
</tr>
<tr>
<td>28</td>
<td>male</td>
<td>Gulf</td>
<td>15-20</td>
<td>120</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>29</td>
<td>male</td>
<td>Gulf</td>
<td>15-20</td>
<td>80</td>
<td>60</td>
<td>166</td>
</tr>
</tbody>
</table>
Appendix C

Text dataset for the main corpus

1. AlmusAfrA n i ya$HanAni AlHaqAQib
2. Alwaladu yatanA w aHU AI TaE A m
3. AlbintAni tastariyHAni fiy AISAlap
4. Hamala AIQawlAdu AI HaqAQib
5. taxorju bintAn
6. AlbanAtu Ai$tارyna AlgadAQ
7. AlmusAfrapu ta$ohabu QilaY AKAEap
8. Sal~aY Alraju1An
9. AlmusAfru yaQku1u AI TaE A m
10. Qaka1at AlmusAfrapu AI TaE A m
11. Qayna yuSa1~iy AlgrijAl
12. AlmusAfru ya$tariy AI E a$AQ
13. AistarAHa Alraju1Ani fiy AlmA T E a m
14. hal xaraja AlmusAfruwn
15. Alwaladu yaxruj
16. gAdara AlmusAfruwna filisTiyna SabAHAF
17. AistamaEat AlbintAni QilaY AltaEliymAt
18. yaxoruju Alwaladu mina AlmaHaT~ap

220
19. Alrajuulu yuSal~iy
20. tasaw~aqã AlmusAfiruwna fiy AlmaTAr
21. AlmusAfiratAni tastariyHAñi fiy AlqAEap
22. Qayna yantaZihu AlrijAl
23. AlmusAfiruwna QalgawoA Alta*karap
24. AlbanAtu taSilha AklawHapa masAQF
25. tur$ïdu AlmuwaZ~afapu AlqAdimiyn
26. AistarAHa AlqAdimuwna fiy AlqAEap
27. mataY Sal~aY AlrajuLAN
28. Qar$adat AlmuwaZ~afapu AlmusAfiriyn
29. AlmusAfirAni Ai$tarayA AlEa$AQ
30. mataY yaQkulu AlmusAfirAni AlgaDAQ
31. AlmusAfirapu ta*hahu QilaY AlmaTAr
32. AlwaladAni yaQkulAni AlgaDAQ
33. AlqAdimuwna yadofaEuwna AunuquwdA ìiAlbint
34. AlbintAni AintaZaratA fiy qAEapi Alta$riyfAt
35. AlmusAfirAni QakaA AITAeAm
36. AlmuwaZ~afu yur$ïdu AlrajuLAN
37. AlbintAni AistarAHatA fiy AlqAEap
38. AlmusAfiru yaSilu maTAra QabuwZabiyy EaSrAF
39. Alrajuulu ya$tariy AITAeAm
40. ta$tariy Albintu QifTAeAF
41. Albintu Aistalamat Alta*karap
42. yuSal~iy rijAIN fiy gurfapi AlSalAp
43. AlrijAlu AistarAHawoA fiy AlmaHAT~ap
44. yuSal~iy AlrijAl

221
45. gAdara AlrijAlu QabuwZabiya masAQF
46. AlmusAfirAni xarajA mina AlmaTAr
47. Ai$taraY AlmusAfiruwna AlQifTAr
48. taQkulu AlmusAfirAtu fiy AlmaTEam
49. yugAdiru AlQawlAdu bagdAda SabAHAF
50. yaxruju AlqAdimuwna mina AlmaHaT~ap
51. AlrijAlu yaxrujuwna min baw~Abapi Alxuruwj
52. AlrijAlu yuSal~wna fiy gurfapi AlSalAp
53. Qar$adat muwaZ~afapu AlQistiqbAli Alrajulayn
54. AlmusAfirAtu ta$tariyana AlEa$AQ
55. Alwaladu dafaEa AlnuquwdA liAlmuwaZ~af
56. ya$tariy AlqAdimuwna AlEa$AQ
57. yaQkulu AlmusAfiruwna fiy AlmaHaT~ap
58. Albintu Qalgat Ata*karap
59. yaQkulu AlrijAlu AlTaEAm
60. AlmusAfiru waSala AlkawHapa EaSrAF
61. yasQalu AlqAdimuwna AlmuwaZ~afa Ean mawEidi AlQiqlAE
62. AlmuwaZ~afa yuElinu Ean AltaQxiyr
63. Qakala AlmusAfiruwna AlQifTAr
64. AlmugAdiruwna AintaZaruwA fiy Albank
65. yulgiiy AlrijAlu AlriHlap
66. tugAdiru AlbanAtu AlkawHapa masAQF
67. AlmusAfirapu tasaw~aqat fiy AlmaTAr
68. AlmusAfirAni yaxoruJAn
69. $aHanat AlmusAfirAtu AlQamtiEap
70. Albintu AintaZarat fiy Almarkaz
71. Alwaladu AintaZara fiy AlmaTEam
72. tanAwalat AlmusAfirAtu TaeAmu AlgadAQ
73. Aistalama AlQawwAdu AlHaqAQib
74. Alwaladu yaxoruj
75. ya*ohabu AlmusAfiru QilaY AlmaHaT~ap
76. AlmusAfirapu Qakalat AlQifTAr
77. Albintu tugAdiru QabuwZabiy EaSrAF
78. yuSal~iy rajuLAn
79. AlmusAfirtAn tawaj~ahatA QilaY AlmaTAr
80. AlmusAfirAni QakAlA AlgadAQ
81. AlmusAfirapu Qakalat fiy ALSAlap
82. AlmusAfirAni Ai$tarA Alta*karap
83. Qar$ada rajuLu AlQamni Albint
84. AlqAdimuwna Ai$tarawO ta*karapa safar
85. Alrajulu Aistalama AljawAz
86. Hamala AlmusAfiruwna AlHaqAQib
87. AlmusAfirtAni tastariyHAni fiy SAapi AlmutAdiriyn
88. AlmuwaZ~afu Qar$ada Alwalad
89. yuSal~iy AlrijAI
90. Albintu tastaLimu AlHaqAQib ZuhrAF
91. xaraja AlrijAI
92. Qalgat AlbintAni Alta*karap
93. AlmusAfiru Qakala AlEa$AQ
94. yaQkulu AlqAdimuwna AlEa$AQ
95. yur$du AlmuwaZ~afu AlrijAI
96. tatanAwalu AlmusAfirAtu AlEa$AQ
97. AlwaladAni gadarA TahrAna ZuhrAF
98. Qakala AlrajulAni AIQifTAr
99. gAdara AlmusAfiru bagdAda EaSrAF
100. AlbanAtu Aistalamna AIQamtiEap
101. waSala AIQawlAdu AldawHapa EaSrAF
102. AintaZarat AlbanAtu fiy AlmaTAr
103. AlbintAni xarajatA mina AlmaTAr
104. AlmuwaZ~afu Qar$ada AlmugAdiriyn
105. Qar$ada muwaZ~afu AIQanmi Albint
106. tugAdiru AlbintAni masqaTa EaSrAF
107. AlmusAfirAni ya$tariyAn AIrTaEAm
108. yaQkuulu AIQawlAdu fiy AlmAIEam
109. hal xarajat Albint
110. yuSal~iy AlrajulAn
111. tur$du AlmuwaZ~afapu AlmugAdiriyn
112. AlwaladAni gAdarA masqaTa ZuhrAF
113. AlmugAdiruwna AistalamuwA AlTa*karap
114. AlwaladAni yugAdirAni TahrAna SabAHAF
115. tanAwala AIQawlAdu AlgadAQ
116. Qakalt AlbanAtu AlgadAQ
117. AIqAdimuwna *ahabuwA QilaY AlmaTAr
118. AlmuwaZ~afu yuElliu Ean rukuwbi AIrTAQirap
119. Alrajul Sal~aY
120. ya$tariy AIQawlAdu AIQifTAr
121. mataY laEiba AIQawlAd
122. mataY gAdarat AlbintAni AldawHap
123. AlmusAfirapu Ai$tarat gadaAQF
124. AlbintAni *ahabatA QilaY Albaw~Abap
125. AlbanAtu xarajna
126. Alrajulu yastalimu AlbiTAqapa ZuhrAF
127. hal Qar$ada AlmuwaZ~afu AlmugAdiriyn
128. Albintu tanAwalat AlgadAQ
129. Albintu ta$tariy AlQiTTara mina AlmaTEam
130. AlmusAfiru yaxoruj
131. Albanku yaqaEni fiy Aldawri AskAmis
132. Albintu tulgiy AlriHlapa Almutawaj~ihapa QilaY AlriyAD
133. mataY Qakalat AlbintAni AIQiTTAr
134. Qayna *ahabat Albint
135. AistarAHat AlmusAfirtAn fiy AlSAlap
136. taxoruju AlbintAn
137. AlmusAfirwna waSalwA bagdaAda ZuhrAF
138. AlbanAtu AistaraHna fiy AlSAlap
139. AlmusAfirAni Ai$tarayA AlTaEAm
140. xaraja AlrijAl
141. Alrajulu yantaZiru fiy Albank
142. tatasaw~aqu AlbanAtu fiy AlmaTAr
143. AlmusAfirAni $aHanA AlHaqAQib
144. yatanAwalu AlwaladAni AIEa$AQ fiy SAlapi AlmaTAEim
145. AlmusAfirtAn xarajTA min qAEapi AlmugAdiriyn
146. AlrajulAni yuSal~iyAn
147. yuriydu Alrajulu Qan yuSal~iy
148. AlmusAfirAtu $aHan~a AlHaqAQib
149. يوريودى أرارجالني قان يلعغيْأ ألتاكرَاب
150. تُغّأديْأرُ ألمُعُاسْأْفْرَاتُ أليليْأْدا قِلْأْيْ أَلْخُرُتْوَم
151. أرارجالني ينسلْيْأْيْان
152. يسايل ألمُعُادْجُرْيوْنْأا فِلْيْسْيْتُئْنْا سَابْأّهْأْف
153. قارّسَادا ألموُوْوادّاف أُلِرْيأْل
154. ألمُعُاسْأْفْرَابْعَةَ أُتْهَوْبْأْيْ قِلْأْيْيْ بَأْوْ أَلْخُرْوعْوْج
155. ألمُعُاسْأْفْرَاتُ أَنْتْأْزارْأا فِيْ أَلْمَاّتّأْر
156. يخآروْوج ألمُعُادْجُرْيوْن
157. أرارجالني خارْجأ
158. ألمُعُاسْأْفْرَانْيْيْ يَانْتْأْزَيْرْأْنْيْيْ أَينْدِا أَلْبَأْوْ أَبَأْو
159. ألمُعُاسْأْفْرَانْيْيْ يَاهْمِيْلْأْن أَلْهَأْقّأْأْيْب
160. ألموُوْوادّاف قارّسَادا أَلْقْوْلّأّد
161. *أّهَابْأ ألمُعُادْجُرْيوْنْأا قِلْأْيْيْ أَلْقّأْإَإْب
162. قارّسَادا ألموُوْوادّاف أَلْقّأْإَإْب أَلْوَالادْأْيءَن
163. سالْأّيْ أَلْرْيْلأْن
164. تانْأْوَأْل أَلْبَنْتُ أَلْإَّفْأّيْ
165. أَلْبَنْتُ أَيْ إِيْثرْأْتْأ أَلتاكرَاب
166. يورّسْيدو ألموُوْوادّاف أَلْوَالادْأْيءَن
167. ألمُعُاسْأْفزْرَوْنْأا يَؤْسَاليمْوْنْأا أَلْهَأْقّأْيىْبا إِأْسْرّأْف
168. ألمُعُاسْأْفْرَتْأْنْيْيْ تاْهْمِيْلْأْن أَلْيْإَيْأْيْإْأْب
169. أَلْبَنْتُ أَيْ إِيْثرْأْتْأ أَلْإَّفْأّيْأْبات
170. ألمُعُاسْأْفزْرَانْيْبُ هَأْمْلْنَأْ أَلْيْإَيْأْيْإْأْب
171. ألمُعُاسْأْفْرَتْأْنْيْيْ إِيْثرْأْتْأ <يْإْيْأْيْإْأْف
172. أَلْوَالادْأْيءَنْيْيْ يَانْتْأْزَيْرْأْيْ فِيْ أَلْمَاّكْتَب
173. أَنْتْأْزَارْأا أَلْبَنْتُ فِيْ أَلْمَاّتّأْر
174. تاسْأْلْو أَلْبَنْتُ أَمْوُوْوادّافْيْ أَنْجّأَنْجَدْأْلْيْ أَرْيّحْلأْت
201. tugAdiru AlbanAtu tanzAnyA masA’F
202. AlmusAfirapu tugAdiru TahrAna EaSrAF
203. yuSal~iy AlrajuAn
204. yastamiEu Alwalad <ilaY AltaEliymAt
205. yantaZir AlqAdimuwna fiy AlmabaY AlxAmis
206. AlmusAfirAtu >akalna fiy AlmaTEam
207. tastariyHu AlmusAfirAtu fiy AlqAEap
208. ta$tariy Albintu TaEAmAF
209. >akalat AlmusAfirapu AlgadA’
210. AlrajuAni yaxorujAn mina Almarkaz
211. Aistalama AlmusAfiruwna Alta*karap
212. AlbintAni AistalamatA AlTa*karap
213. AlbintAni tanZirAni fiy Albank
214. AlwaladAni yugAdirAni TahrAna SabAHAF
215. tanAwalat AlbintAni AlIEa$A’
216. AlbanAtu AintaZarna Einda Albaw~Abap
217. AlmusAfirAni ya~kuAni AlgadA’
218. AlmusAfirAni AistalamA Al>amtiEap
219. AlrajuAni $aHanA Al>amtiEap
220. ta*habu AlbanAtu <ilaY AlmaTaR
221. yantaZiru AlmugAdiruwna fiy AISAlap
222. yaxoruju AlwaladAn
223. Alwaladu >algaY AlTa*karap
224. turiydu AlmusAfirAtu >an taHmilna AIHaqA}ib
225. AlmusAfiruwna AistalamuwA AlTa*karap
226. AlmusAfirAni yatanAwalAn Al<ifTaR fiy AlmaHaTa~ap
227. AlmusAfiruwna ya>okulwna AlgadA’
228. AlmusAfirAtu Hamalna AlHaqA}ib
229. dafaEat AlbanAtu AlnuqyuwdA lilmugAdiriyn
230. Sal~aY Alrajul
231. AlmusAfirAni wSalA masqA}Ta EaSrAF
232. AlqAdimuwna $aHanwA AlHaqA}ib
233. >ar$adat AlmuwaZ~afapu AlmugAdiriyn
234. AlqAdimuwna ya>okulwna AI<l5TAr
235. Aistalama AlmusAfiruwna HaqA}ibA Alsafar
236. Alrajulu xaraja mina AlmaHaT~ap
237. Alrajulu >akala AlgadA’
238. yaxoruju AlwaladAni mina AlqAEap
239. yatasaw~aqI AlmugAdiruwna fiy AlmaTAr
240. AlmusAfirAni yulgIyAn AlriHIap
241. AlmusAfirapu ta$tariy AIEa$A’
242. Alwaladu tasaw~aqa fiy AlmaTAr
243. AlbintAni tasta$rIyHAni fiy AlmA$eam
244. AlmusAfirAtu tatasaw~aqna fiy AlmaTAr
245. taxoruju AlmusAfiratAni mina Almaktab
246. AlrajulAni yatanAwalAn AlgadA’
247. >ayna ta*habu AlmusAfirAt
248. AlrijAlu yuSal~uwn
249. AlrajulAni ya>okulAni fiy AlSAIap
250. xaraja waladAn
251. >ar$sada AlmuwaZ~afu AlwaladAy
252. AlmuwaZ~afapu >ar$sadat Alwalad
253. Albintu ta$tariy AlTaEAm
254. AlmusAfirAni gAdarA bagdAda ZuhrAF
255. tasaw~aqa AlrajulAni fiy maTAr AlxurTuwm
256. AlrajulAni yugAdirAni TahrAna EaSrAF
257. yuriydu AlmusAfiru >an yaxoruja mina AlmaTAr
258. tuElinu AlmuwaZ~afapu Ean <ixIA’i AlmaTAr
259. AlmusAfirAtu ta>okulna fiy AqAEap
260. AlmusAfirAni Ai$tarA AlTa*karap
261. AlmusAfirapu gAdarat tanz.AnyA SabAHAF
262. Alrajul Ai$taraY AlTa*karap
263. ya$tariy Alwaladu A1<ifTAr
264. AlmuwaZ~afu >ar$ada AlmugAdiriyn
265. Albintu tantazeru fiy AlmaHaT~ap
266. tuElinu AlmuwaZ~afapu Ean ta>xiyri AlriHlap
267. yastalimu Alrajulu AlHaqiiba ZuhrAF
268. ta$tariy Albintu AIe$A’
269. A1>awIAdu yaSiHuwna >abuwZabiys masA’F
270. hal Sal~aY AlrijAl
271. AlmusAfiruwna yugAdirwna AlriyADa EaSrAF
272. Alrajulu gAdara TahrAna EaSrAF
273. yatanAwalu AlwaladAni A1<ifTAr
274. AlmusAfirapu >algat AlTa*karap
275. AlmuwaZ~afapu >ar$adat AlmusAfiriyn
276. AlqAdimuwna ya*ohabwna <ilaY AlmaTAr
277. xaraja AlwaladAni mina AISAlap
278. yuElinu AlmuwaZ~afu Ean <iqIAEi AITA}irap
279. التأعالات مسافر العربية <التي> رفع إلى الياة لثمن
280. مسافر أistar أنف مسافر الفوائد
281. التأعالات التأعالات <التي> مسافر تانا
282. >النار *احابة مسافر
283. مسافر أistar كرك�نود <التي> قد <التي>
284. *احابة مسافر <التي> كركشانود <التي> مسافر
285. أistar مسافر أistar <التي> العش يكار
286. أبنت أنكر مسافر أكار ورابع <التي>
287. *احابة مسافر أistar كركشانود <التي> مسافر
288. مسافر كركشانود التأعالات <التي> مسافر العش
289. أكار <التي> كركشانود <التي>
290. التأعالات <التي> مسافر أكار ورابع <التي>
291. كركشانود أبنت أنكر كركشانود <التي> مسافر
292. كركشانود مسافر أكار ورابع <التي>
293. أكار <التي> كركشانود <التي>
294. كركشانود مسافر أكار ورابع <التي>
295. مسافر أestar أصار <التي> مسافر
296. مسافر أestar أصار <التي> مسافر
297. كركشانود مسافر أestar أصار <التي> مسافر
298. >النار أكار كركشانود
299. كركشانود مسافر أكار ورابع <التي>
300. أبنت كركشانود كركشانود

231
Appendix D

Details of the speakers participated in the main corpus
Table D.1: Details of the speakers participated in recording the main corpus. Showing their gender, regional accent, age, total number of recordings, number of repetitions, and the average pitch.

<table>
<thead>
<tr>
<th>ID</th>
<th>gender</th>
<th>accent</th>
<th>age</th>
<th>utterances</th>
<th>repetitions</th>
<th>pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>male</td>
<td>Gulf</td>
<td>15-20</td>
<td>613</td>
<td>313</td>
<td>106</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>Gulf</td>
<td>25-30</td>
<td>901</td>
<td>301</td>
<td>121</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>Gulf</td>
<td>30-35</td>
<td>60</td>
<td>0</td>
<td>257</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>50</td>
<td>0</td>
<td>234</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>Gulf</td>
<td>55-60</td>
<td>101</td>
<td>0</td>
<td>149</td>
</tr>
<tr>
<td>6</td>
<td>male</td>
<td>Gulf</td>
<td>15-20</td>
<td>14</td>
<td>0</td>
<td>152</td>
</tr>
<tr>
<td>7</td>
<td>male</td>
<td>Gulf</td>
<td>25-30</td>
<td>21</td>
<td>0</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>920</td>
<td>620</td>
<td>256</td>
</tr>
<tr>
<td>9</td>
<td>male</td>
<td>Levantine</td>
<td>15-20</td>
<td>893</td>
<td>593</td>
<td>161</td>
</tr>
<tr>
<td>10</td>
<td>female</td>
<td>Levantine</td>
<td>30-35</td>
<td>898</td>
<td>598</td>
<td>249</td>
</tr>
<tr>
<td>11</td>
<td>male</td>
<td>Gulf</td>
<td>20-25</td>
<td>6</td>
<td>0</td>
<td>108</td>
</tr>
<tr>
<td>12</td>
<td>male</td>
<td>Gulf</td>
<td>30-35</td>
<td>100</td>
<td>0</td>
<td>129</td>
</tr>
<tr>
<td>13</td>
<td>male</td>
<td>Gulf</td>
<td>20-25</td>
<td>34</td>
<td>0</td>
<td>112</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>Gulf</td>
<td>30-35</td>
<td>1051</td>
<td>751</td>
<td>162</td>
</tr>
<tr>
<td>15</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>210</td>
<td>0</td>
<td>251</td>
</tr>
<tr>
<td>16</td>
<td>female</td>
<td>Egypt</td>
<td>35-40</td>
<td>896</td>
<td>596</td>
<td>201</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>887</td>
<td>587</td>
<td>246</td>
</tr>
<tr>
<td>18</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>905</td>
<td>605</td>
<td>248</td>
</tr>
<tr>
<td>19</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>902</td>
<td>602</td>
<td>239</td>
</tr>
<tr>
<td>20</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>34</td>
<td>0</td>
<td>263</td>
</tr>
<tr>
<td>21</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>466</td>
<td>166</td>
<td>264</td>
</tr>
<tr>
<td>22</td>
<td>male</td>
<td>Levantine</td>
<td>20-25</td>
<td>1013</td>
<td>713</td>
<td>137</td>
</tr>
<tr>
<td>23</td>
<td>female</td>
<td>Levantine</td>
<td>20-25</td>
<td>44</td>
<td>0</td>
<td>253</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>26</td>
<td>0</td>
<td>217</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>Gulf</td>
<td>20-25</td>
<td>30</td>
<td>0</td>
<td>209</td>
</tr>
<tr>
<td>26</td>
<td>female</td>
<td>Gulf</td>
<td>30-35</td>
<td>201</td>
<td>0</td>
<td>217</td>
</tr>
<tr>
<td>27</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>415</td>
<td>115</td>
<td>219</td>
</tr>
<tr>
<td>28</td>
<td>female</td>
<td>Gulf</td>
<td>30-35</td>
<td>103</td>
<td>0</td>
<td>250</td>
</tr>
</tbody>
</table>
Table D1: Details of the speakers participated in recording the main corpus. Showing their gender, regional accent, age, total number of recordings, number of repetitions, and the average pitch.

<table>
<thead>
<tr>
<th>ID</th>
<th>gender</th>
<th>accent</th>
<th>age</th>
<th>utterances</th>
<th>repetitions</th>
<th>pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>male</td>
<td>Gulf</td>
<td>20-25</td>
<td>708</td>
<td>408</td>
<td>114</td>
</tr>
<tr>
<td>30</td>
<td>female</td>
<td>Gulf</td>
<td>50-55</td>
<td>1002</td>
<td>702</td>
<td>262</td>
</tr>
<tr>
<td>31</td>
<td>male</td>
<td>Gulf</td>
<td>30-35</td>
<td>83</td>
<td>0</td>
<td>166</td>
</tr>
<tr>
<td>32</td>
<td>female</td>
<td>Gulf</td>
<td>20-25</td>
<td>46</td>
<td>0</td>
<td>232</td>
</tr>
<tr>
<td>33</td>
<td>male</td>
<td>Gulf</td>
<td>60-65</td>
<td>1040</td>
<td>740</td>
<td>157</td>
</tr>
<tr>
<td>34</td>
<td>female</td>
<td>Levantine</td>
<td>30-35</td>
<td>955</td>
<td>655</td>
<td>257</td>
</tr>
<tr>
<td>35</td>
<td>female</td>
<td>Levantine</td>
<td>35-40</td>
<td>856</td>
<td>556</td>
<td>248</td>
</tr>
<tr>
<td>36</td>
<td>male</td>
<td>Gulf</td>
<td>30-35</td>
<td>22</td>
<td>0</td>
<td>138</td>
</tr>
<tr>
<td>37</td>
<td>female</td>
<td>Levantine</td>
<td>15-20</td>
<td>36</td>
<td>0</td>
<td>249</td>
</tr>
<tr>
<td>38</td>
<td>male</td>
<td>Gulf</td>
<td>25-30</td>
<td>50</td>
<td>0</td>
<td>156</td>
</tr>
<tr>
<td>39</td>
<td>female</td>
<td>Levantine</td>
<td>30-35</td>
<td>1163</td>
<td>863</td>
<td>240</td>
</tr>
<tr>
<td>40</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>60</td>
<td>0</td>
<td>242</td>
</tr>
<tr>
<td>41</td>
<td>female</td>
<td>Gulf</td>
<td>35-40</td>
<td>79</td>
<td>0</td>
<td>223</td>
</tr>
<tr>
<td>42</td>
<td>female</td>
<td>Gulf</td>
<td>20-25</td>
<td>300</td>
<td>0</td>
<td>206</td>
</tr>
<tr>
<td>43</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>920</td>
<td>620</td>
<td>238</td>
</tr>
<tr>
<td>44</td>
<td>male</td>
<td>Gulf</td>
<td>20-25</td>
<td>251</td>
<td>0</td>
<td>135</td>
</tr>
<tr>
<td>45</td>
<td>female</td>
<td>Gulf</td>
<td>30-35</td>
<td>576</td>
<td>276</td>
<td>227</td>
</tr>
<tr>
<td>46</td>
<td>female</td>
<td>Levantine</td>
<td>45-50</td>
<td>15</td>
<td>0</td>
<td>231</td>
</tr>
<tr>
<td>47</td>
<td>female</td>
<td>Gulf</td>
<td>20-25</td>
<td>32</td>
<td>0</td>
<td>219</td>
</tr>
<tr>
<td>48</td>
<td>female</td>
<td>Gulf</td>
<td>40-45</td>
<td>8</td>
<td>0</td>
<td>222</td>
</tr>
<tr>
<td>49</td>
<td>female</td>
<td>Levantine</td>
<td>25-30</td>
<td>102</td>
<td>0</td>
<td>234</td>
</tr>
<tr>
<td>50</td>
<td>female</td>
<td>Gulf</td>
<td>30-35</td>
<td>95</td>
<td>0</td>
<td>238</td>
</tr>
<tr>
<td>51</td>
<td>male</td>
<td>Gulf</td>
<td>60-65</td>
<td>92</td>
<td>0</td>
<td>111</td>
</tr>
<tr>
<td>52</td>
<td>male</td>
<td>Gulf</td>
<td>25-20</td>
<td>101</td>
<td>0</td>
<td>124</td>
</tr>
<tr>
<td>53</td>
<td>female</td>
<td>Gulf</td>
<td>25-30</td>
<td>832</td>
<td>532</td>
<td>220</td>
</tr>
<tr>
<td>54</td>
<td>male</td>
<td>Gulf</td>
<td>20-30</td>
<td>789</td>
<td>489</td>
<td>135</td>
</tr>
<tr>
<td>55</td>
<td>female</td>
<td>Gulf</td>
<td>20-30</td>
<td>1000</td>
<td>700</td>
<td>261</td>
</tr>
<tr>
<td>56</td>
<td>female</td>
<td>Gulf</td>
<td>20-30</td>
<td>53</td>
<td>0</td>
<td>238</td>
</tr>
</tbody>
</table>
Appendix E

The PROLOG extensions

The following description of the Prolog extensions used in this report is extracted from Allan Ramsay’s *Parasite manual*

I use an extended version of Prolog. The extensions serve two purposes. Firstly, I think they are more concise and legible than standard Prolog. You have to get used to them, but I am now very happy about the improved legibility that results from using them. Secondly, they result in considerable performance gains, at the cost of making programs much harder to debug. So much harder, in fact, that if you are having trouble with a program and you want to spy it you may want to turn the optimisations off. No sooner said than done.

The key point underlying these extensions is that you often write Prolog procedures which define classes, i.e. single argument procedures which contain no branches or recursive calls. For instance, I might decide that a noun is something which is marked `[cat=[+n, -v], bar=phrasal]`. There is no point in calling the definition of such a procedure every time you want to check whether something is a noun: simple unification with the pattern generated by calling the procedure on an uninstantiated variable will do the job, and do it a great deal faster.

So if you have such a procedure you have a choice of how to call it: you can either call it normally, so that if `verb` and `tensed` are both procedures like this you could define

```prolog
tensed_verb(X) :-
    verb(X),
    tensed(X).
```

Or you can have the procedures executed at compile time, by defining it as
tensed_verb(X) :-
    X <> [verb, tensed].

The second version will run very much faster, since all it will have to do is unify its argument with the pattern resulting from calling verb and tensed on an uninstantiated variable. This is much faster than calling verb and tensed, particularly if they in turn are defined by procedures with the same 'no branching, at most one solution' character.

The same idea lies behind the other extensions. Firstly, I have a number of procedures which access the constituents of complex structures: consider the feature active, for instance. Because I am relying largely on standard Prolog unification, I need signs all to have the same shape, so that active appears in the same place in every sign. There is therefore a very important file, types.pl, which contains definitions laying out the position of every feature that the system supports. These definitions enable you to call a procedure looking like active(X => A) to make A unify with the value of active for X. If you call this with X uninstantiated and a concrete value for A, for instance, you will be able to see where active fits:

\[ \text{?- active(X => +), pretty(X).} \]

sign(A,
    syntax(nonfoot(B,
        head(C,
            vform(vfeatures(D,
                E,
                F,
                +active,
                G,
                H,
                I),
            J),
        K,
        L),
    M,
    N),

236
Again, these are unbranching procedures and hence can be executed at compile time, so that you have a choice between defining e.g.

```prolog
active_verb(X) :-
    verb(X),
    active(X => +).
```

and

```prolog
active_verb(X) :-
    X <> verb,
    active@X <-> +.
```

with the latter being a great deal quicker and a bit more legible.

There are two further refinements:

(i) you very often want to use the values + and - for features which can be either present or absent. You can write +active@V instead of active@V <-> +, and likewise for -.

(ii) you very often want to constrain two items to share values for a number of features: you can write

```
[active, tensed, semantics]@X1 <-> [active, tensed, semantics]@X2
```

to say that X1 and X2 should share values for these features. This is more concise than the usual notation, and also saves you having to think up names for boring variables: you would otherwise have to write something like

```prolog
active(X1 => A),
ative(X2 => A),
tensed(X1 => T),
tensed(X2 => T),
semantics(X1 => S),
semantics(X2 => S),
```


It is also possible to say that two items should agree on every subfeature of something except for some nominated element. So

\[
\text{nonfoot}[\text{specf, target, modifiable}] @ \text{result}@X
\]
\[<->\]
\[
\text{nonfoot}[\text{specf, target, modifiable}] @ \text{target}@X,\]

says that \(X\) and its target should share all their nonfoot features apart from \text{specf}, \text{target} and \text{modifiable}. This notation\(^1\) does not say that these subfeatures must be different, just that they may be.

The notation is neat (at least I think so); the resulting programs are fast; but there is a downside, which is that they are also extremely opaque. If you are debugging you may well want something that looks more like the long-winded version, since otherwise you will often just get a unification failure without being able to see very easily where it came from. There are therefore various compilation options.

To use this extended notation, you obviously can’t just use ordinary Prolog. You first have to load the files \texttt{useful.pl} (which contains various useful list processing procedures and other basic stuff) and \texttt{translat.pl}. You can then load extended Prolog files in the usual ways, since Prolog allows you to perform arbitrary operations on your source code before its is actually passed to the interpreter or compiler. You can use the \texttt{feature@SIGN} notation when typing commands to the Prolog interpreter. The other extensions are only available in program source files.

\(^1\) introduced and implemented by Helen Gaylard
Appendix F

Stress assigning rules

findStressedSyll(L, X):-
    +heavy:syll@syllable:char@S,
    reverse(L, R),
    member(X, R),
    member(S, X),
    !,
    +stress:syll@syllable:char@S.

findStressedSyll(L, X):-
    +vowel:char@S,
    +long:char@S,
    reverse(L, R),
    member(X, R),
    member(S, X),
    !,
    +stress:syll@syllable:char@S.

findStressedSyll(L, X):-
    X <-> [S | _],
    reverse(L, [_ | _ | X | _]),
    !,
    +stress:syll@syllable:char@S.

findStressedSyll(L, X):-

X <-> [S | _],
reverse(L, [_, X | _]),
!,
+stress:syll@syllable:char@S.

findStressedSyll([X], X):-
X <-> [S | _],
+stress:syll@syllable:char@S.

addDefaultStress([], _X).

addDefaultStress([Y | SS], X):-
Y <-> [S | _],
stress:syll@syllable:char@S <-> STRESS,
(X == Y -> true; STRESS = -),
addDefaultStress(SS, X).

assignStress(S3):-
(markSyllables(S3) -> true; fail),
splitSyllables(S3, U),
findStressedSyll(U, X),
addDefaultStress(U, X).

assignStressToWords([]).

assignStressToWords([W | WORDS]):-
assignStress(W),
assignStressToWords(WORDS).
Appendix G

The gender-specific and accent-specific results
<table>
<thead>
<tr>
<th>ID</th>
<th>gender</th>
<th>accent</th>
<th>accuracy(%)</th>
<th>ID</th>
<th>gender</th>
<th>accent</th>
<th>accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>female model</td>
<td></td>
<td></td>
<td></td>
<td>female model</td>
</tr>
<tr>
<td>1</td>
<td>male</td>
<td>Gulf</td>
<td>41.3</td>
<td>29</td>
<td>male</td>
<td>Gulf</td>
<td>49.1</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>Gulf</td>
<td>84.1</td>
<td>30</td>
<td>female</td>
<td>Gulf</td>
<td>71.9</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>Gulf</td>
<td>85.6</td>
<td>31</td>
<td>male</td>
<td>Gulf</td>
<td>27.4</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>Gulf</td>
<td>83.1</td>
<td>32</td>
<td>female</td>
<td>Gulf</td>
<td>43.1</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>Gulf</td>
<td>76.8</td>
<td>33</td>
<td>male</td>
<td>Gulf</td>
<td>87.8</td>
</tr>
<tr>
<td>6</td>
<td>male</td>
<td>Gulf</td>
<td>81.9</td>
<td>34</td>
<td>female</td>
<td>Levantine</td>
<td>69.9</td>
</tr>
<tr>
<td>7</td>
<td>male</td>
<td>Gulf</td>
<td>91.7</td>
<td>35</td>
<td>female</td>
<td>Levantine</td>
<td>84.4</td>
</tr>
<tr>
<td>8</td>
<td>female</td>
<td>Levantine</td>
<td>75.1</td>
<td>36</td>
<td>male</td>
<td>Gulf</td>
<td>51.9</td>
</tr>
<tr>
<td>9</td>
<td>male</td>
<td>Levantine</td>
<td>76.0</td>
<td>37</td>
<td>female</td>
<td>Levantine</td>
<td>39.9</td>
</tr>
<tr>
<td>10</td>
<td>female</td>
<td>Levantine</td>
<td>91.7</td>
<td>38</td>
<td>male</td>
<td>Gulf</td>
<td>67.3</td>
</tr>
<tr>
<td>11</td>
<td>male</td>
<td>Gulf</td>
<td>95.0</td>
<td>39</td>
<td>female</td>
<td>Levantine</td>
<td>50.5</td>
</tr>
<tr>
<td>12</td>
<td>male</td>
<td>Gulf</td>
<td>8.8</td>
<td>40</td>
<td>female</td>
<td>Gulf</td>
<td>81.9</td>
</tr>
<tr>
<td>13</td>
<td>male</td>
<td>Gulf</td>
<td>37.4</td>
<td>41</td>
<td>female</td>
<td>Gulf</td>
<td>95.1</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>Gulf</td>
<td>49.9</td>
<td>42</td>
<td>female</td>
<td>Gulf</td>
<td>83.3</td>
</tr>
<tr>
<td>15</td>
<td>female</td>
<td>Levantine</td>
<td>79.6</td>
<td>43</td>
<td>female</td>
<td>Gulf</td>
<td>74.9</td>
</tr>
<tr>
<td>16</td>
<td>female</td>
<td>(Egypt)</td>
<td>81.0</td>
<td>44</td>
<td>male</td>
<td>Gulf</td>
<td>69.6</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>Gulf</td>
<td>90.1</td>
<td>45</td>
<td>female</td>
<td>Gulf</td>
<td>94.6</td>
</tr>
<tr>
<td>18</td>
<td>female</td>
<td>Levantine</td>
<td>81.2</td>
<td>46</td>
<td>female</td>
<td>Levantine</td>
<td>42.1</td>
</tr>
<tr>
<td>19</td>
<td>female</td>
<td>Levantine</td>
<td>76.3</td>
<td>47</td>
<td>female</td>
<td>Gulf</td>
<td>55.5</td>
</tr>
<tr>
<td>20</td>
<td>female</td>
<td>Levantine</td>
<td>32.1</td>
<td>48</td>
<td>female</td>
<td>Gulf</td>
<td>90.0</td>
</tr>
<tr>
<td>21</td>
<td>female</td>
<td>Levantine</td>
<td>74.9</td>
<td>49</td>
<td>female</td>
<td>Levantine</td>
<td>88.7</td>
</tr>
<tr>
<td>22</td>
<td>male</td>
<td>Levantine</td>
<td>32.1</td>
<td>50</td>
<td>female</td>
<td>Gulf</td>
<td>89.1</td>
</tr>
<tr>
<td>23</td>
<td>female</td>
<td>Levantine</td>
<td>93.6</td>
<td>51</td>
<td>male</td>
<td>Gulf</td>
<td>62.8</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>Gulf</td>
<td>82.8</td>
<td>52</td>
<td>male</td>
<td>Gulf</td>
<td>62.4</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>Gulf</td>
<td>21.0</td>
<td>53</td>
<td>female</td>
<td>Gulf</td>
<td>69.1</td>
</tr>
<tr>
<td>26</td>
<td>female</td>
<td>Gulf</td>
<td>29.0</td>
<td>54</td>
<td>male</td>
<td>Gulf</td>
<td>63.3</td>
</tr>
<tr>
<td>27</td>
<td>female</td>
<td>Gulf</td>
<td>67.1</td>
<td>55</td>
<td>female</td>
<td>Gulf</td>
<td>78.8</td>
</tr>
<tr>
<td>28</td>
<td>female</td>
<td>Gulf</td>
<td>79.5</td>
<td>56</td>
<td>female</td>
<td>Gulf</td>
<td>93.2</td>
</tr>
</tbody>
</table>
Appendix H

Mixed genders experiments results
Table H.1: Results of testing on a mixed-gender model.

<table>
<thead>
<tr>
<th>ID</th>
<th>gender</th>
<th>absolute accuracy(%)</th>
<th>ID</th>
<th>gender</th>
<th>absolute accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>male</td>
<td>46.6</td>
<td>29</td>
<td>male</td>
<td>63.9</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>73.2</td>
<td>30</td>
<td>female</td>
<td>58.7</td>
</tr>
<tr>
<td>3</td>
<td>female</td>
<td>71.6</td>
<td>31</td>
<td>male</td>
<td>30.7</td>
</tr>
<tr>
<td>4</td>
<td>female</td>
<td>62.9</td>
<td>32</td>
<td>female</td>
<td>25.8</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>49.8</td>
<td>33</td>
<td>male</td>
<td>67.1</td>
</tr>
<tr>
<td>6</td>
<td>male</td>
<td>63.7</td>
<td>34</td>
<td>female</td>
<td>45.1</td>
</tr>
<tr>
<td>7</td>
<td>male</td>
<td>62.1</td>
<td>35</td>
<td>female</td>
<td>52.8</td>
</tr>
<tr>
<td>8</td>
<td>female</td>
<td>61.6</td>
<td>36</td>
<td>male</td>
<td>68.4</td>
</tr>
<tr>
<td>9</td>
<td>male</td>
<td>64.5</td>
<td>37</td>
<td>female</td>
<td>63.0</td>
</tr>
<tr>
<td>10</td>
<td>female</td>
<td>71.2</td>
<td>38</td>
<td>male</td>
<td>48.3</td>
</tr>
<tr>
<td>11</td>
<td>male</td>
<td>65.7</td>
<td>39</td>
<td>female</td>
<td>43.6</td>
</tr>
<tr>
<td>12</td>
<td>male</td>
<td>13.4</td>
<td>40</td>
<td>female</td>
<td>74.7</td>
</tr>
<tr>
<td>13</td>
<td>male</td>
<td>38.7</td>
<td>41</td>
<td>female</td>
<td>77.9</td>
</tr>
<tr>
<td>14</td>
<td>male</td>
<td>71.1</td>
<td>42</td>
<td>female</td>
<td>70.9</td>
</tr>
<tr>
<td>15</td>
<td>female</td>
<td>77.8</td>
<td>43</td>
<td>female</td>
<td>42.3</td>
</tr>
<tr>
<td>16</td>
<td>female</td>
<td>59.5</td>
<td>44</td>
<td>male</td>
<td>42.5</td>
</tr>
<tr>
<td>17</td>
<td>female</td>
<td>64.3</td>
<td>45</td>
<td>female</td>
<td>76.4</td>
</tr>
<tr>
<td>18</td>
<td>female</td>
<td>73.5</td>
<td>46</td>
<td>female</td>
<td>51.8</td>
</tr>
<tr>
<td>19</td>
<td>female</td>
<td>59.0</td>
<td>47</td>
<td>female</td>
<td>44.5</td>
</tr>
<tr>
<td>20</td>
<td>female</td>
<td>61.6</td>
<td>48</td>
<td>female</td>
<td>72.1</td>
</tr>
<tr>
<td>21</td>
<td>female</td>
<td>64.6</td>
<td>49</td>
<td>female</td>
<td>84.5</td>
</tr>
<tr>
<td>22</td>
<td>male</td>
<td>31.5</td>
<td>50</td>
<td>female</td>
<td>63.2</td>
</tr>
<tr>
<td>23</td>
<td>female</td>
<td>79.8</td>
<td>51</td>
<td>male</td>
<td>48.6</td>
</tr>
<tr>
<td>24</td>
<td>female</td>
<td>77.6</td>
<td>52</td>
<td>male</td>
<td>60.6</td>
</tr>
<tr>
<td>25</td>
<td>female</td>
<td>7.9</td>
<td>53</td>
<td>female</td>
<td>51.7</td>
</tr>
<tr>
<td>26</td>
<td>female</td>
<td>14.8</td>
<td>54</td>
<td>male</td>
<td>50.8</td>
</tr>
<tr>
<td>27</td>
<td>female</td>
<td>60.3</td>
<td>55</td>
<td>female</td>
<td>61.2</td>
</tr>
<tr>
<td>28</td>
<td>female</td>
<td>61.8</td>
<td>56</td>
<td>female</td>
<td>80.9</td>
</tr>
</tbody>
</table>