Deep Neural Network Acoustic models for Multi-dialect Arabic Speech Recognition

Nadia Fathe Hmad

A thesis submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of Doctor of Philosophy

July 2015

Abstract

Deep Neural Network Acoustic models for Multi-dialect Arabic Speech Recognition

Nadia Fathe Hmad
Doctor of Philosophy
Graduate Department of School of Science and Technology
Nottingham Trent University
2015

Speech is a desirable communication method between humans and computers. The major concerns of the automatic speech recognition (ASR) are determining a set of classification features and finding a suitable recognition model for these features. Hidden Markov Models (HMMs) have been demonstrated to be powerful models for representing time varying signals. Artificial Neural Networks (ANNs) have also been widely used for representing time varying quasi-stationary signals. Arabic is one of the oldest living languages and one of the oldest Semitic languages in the world, it is also the fifth most generally used language and is the mother tongue for roughly 200 million people. Arabic speech recognition has been a fertile area of reasearch over the previous two decades, as attested by the various papers that have been published on this subject.

This thesis investigates phoneme and acoustic models based on Deep Neural Networks (DNN) and Deep Echo State Networks for multi-dialect Arabic Speech Recognition. Moreover, the TIMIT corpus with a wide variety of American dialects is also aimed to evaluate the proposed models.

The availability of speech data that is time-aligned and labelled at phonemic level is a fundamental requirement for building speech recognition systems. A developed Arabic phoneme database (APD) was manually timed and phonetically labelled. This dataset was constructed from the King Abdul-Aziz Arabic Phonetics Database (KAPD) database for Saudi Arabia dialect and the Centre for Spoken Language Understanding (CSLU2002) database for different Arabic dialects. This dataset covers 8148 Arabic phonemes. In addition, a corpus of 120 speakers (13 hours of Arabic speech) randomly selected from the Levantine Arabic dialect database that is used for training and 24 speakers (2.4 hours) for testing are revised and transcription errors were manually corrected. The selected dataset is labelled automatically using the HTK Hidden Markov Model toolkit. TIMIT corpus is also used for phone recognition and acoustic modelling task. We used 462 speakers (3.14 hours) for training and 24 speakers (0.81 hours) for testing.

For Automatic Speech Recognition (ASR), a Deep Neural Network (DNN) is used to evaluate its adoption in developing a framewise phoneme recognition and an acoustic modelling system for Arabic speech recognition. Restricted Boltzmann Machines (RBMs) DNN models have not been explored for any Arabic corpora previously. This allows us to claim priority for adopting this RBM DNN model for the Levantine Arabic acoustic models. A post-processing enhancement was also applied to the DNN acoustic model outputs in order to improve the recognition accuracy and to obtain the accuracy at a phoneme level instead of the frame level. This post process has significantly improved the recognition performance.

An Echo State Network (ESN) is developed and evaluated for Arabic phoneme recognition with different learning algorithms. This investigated the use of the conventional ESN trained with supervised and forced learning algorithms. A novel combined supervised/forced supervised learning algorithm (unsupervised adaptation) was developed and tested on the proposed optimised Arabic phoneme recognition datasets. This new model is evaluated on the Levantine dataset and empirically compared with the results obtained from the baseline Deep Neural Networks (DNNs).

A significant improvement on the recognition performance was achieved when the ESN model was implemented compared to the baseline RBM DNN model's result. The results show that the ESN model has a better ability for recognizing phonemes sequences than the DNN model for a small vocabulary size dataset. The adoption of the ESNs model for acoustic modeling is seen to be more valid than the adoption of the DNNs model for acoustic modeling speech recognition, as ESNs are recurrent models and expected to support sequence models better than the RBM DNN models even with the contextual input window.

The TIMIT corpus is also used to investigate deep learning for framewise phoneme classification and acoustic modelling using Deep Neural Networks (DNNs) and Echo State Networks (ESNs) to allow us to make a direct and valid comparison between the proposed systems investigated in this thesis and the published works in equivalent projects based on framewise phoneme recognition used the TIMIT corpus. Our main finding on this corpus is that ESN network outperform time-windowed RBM DNN ones. However, our developed system ESN-based shows 10% lower performance when it was compared to the other systems recently reported in the literature that used the same corpus. This due to the hardware availability and not applying speaker and noise adaption that can improve the results in this thesis as our aim is to investigate the proposed models for speech recognition and to make a direct comparison between these models.

Copyright Statement

Acknowledgement

I would like to express my deep gratefulness, admiration, and honour to my supervisor Dr. Tony Allen. With such richness in experience, knowledge and dedication, he supported and provided me with the direction to complete this work; always having time to listen, always prepared to help and always open to discussion. I must present my thanks to Dr. Joanna Hartley, my second supervisor for her advice and support during the research journey. I would also like to thank Professor Thomas Hain and Dr. Jonathan Tepper for their valuable feedback and suggestions.

All of my work would not have been possible without others' help. Especially, I would like to thank my colleague Tariq Tashan for his support, cooperation and assistance; he was always prepared to help and always open to discussion. Also I would like to thank Abdel-rahman Mohamed from University of Toronto and Sarah Al-Shareef from University of Sheffield for helping me in using the HTK toolkit and labelling the Levantine corpus. Abdulrahman Alalshekmubarak from University of Stirling helped me build the ESN online training using the stochastic gradient descent algorithm and provided the ranked list of information retrieval for the ESN experiments. My colleague Rowida Elforgani supported me with the required resources during my working outside the UK. Finally, I would like to present my thanks to all my colleagues in Nottingham Trent University (NTU) for their encouragement and support for the PhD years.

The support of my soul mate Abdulati and the encouragement that he provided is with no equivalence. I am always grateful to him for being very patient, supportive, dedicated and taking care of me during completing this work.

All the languages of the world cannot produce words that thank my parents for their great support to my education, eventually resulting in successful completion of this work. They were the cause, and the inspiration without which this work would have never been completed. I wish a happy long life for my Mum. I pray for my Dad, he dreamed to see me in this position, I always pray his soul be forgiven and I wish him a peace in paradise.

Nadia Hmad, July 2015

Dedication

To my parents, my husband and my sons

Abdulwahab, Yakoub and Yahya…

Table of Contents

Chapter Seven: Echo State Networks (ESNs) for phoneme classification and acoustic models 86

List of Tables

List of Figures

Abbreviations

AM	acoustic models
ANN	Artificial Neural Network
APD	Arabic Phoneme Database
ASR	Automatic Speech Recognition
BPNN	Back-Propagation Neural Networks
CA	Colloquial Arabic
CH	CallHome
CHMM	Continuous density Hidden Markov Model
CNNs	Convolutional Neural networks
CSLU2002	Centre for Spoken Language Understanding 2002
CTS	conversational telephone speech
CV	Consonant-Vowel
DBNs	Deep Belief Networks
DCT	Discrete Cosine Transform
DESN	Deep Echo State Networks
DNN	Deep Neural Networks
DR	Dynamic Reservoir
DTW	Dynamic Time Warping
ECA	Egyptian Colloquial Arabic
EM	expectation maximization
En	Energy
ESN	Echo State Network
FAR	False Accept Rate
FBIS	Foreign Broadcast Information Service
FFT	Fast Fourier Transform
FNNs	feed-forward Neural Networks
FPER	framewise phoneme error
FRR	False Reject Rate
FSN	finite state network
GMMs	Gaussian Mixture Models
HFCC	Human Factor Cepstral Coefficients
HMM	Hidden Markov models
ICSI	International Computer Science Institute
IPA	International Phonetic Alphabet
JOR	Jordan
KACST	Abdulaziz City for Science and Technology
KAPD	King Abdulalaziz Phonetic Database
KL	Karhunen-Loeve
LCA	Levantine colloquial Arabic
LDC	Linguistic Data Consortium
LEB	Lebanon

LM	language models
LPC	Linear predictive Coefficient
LPCC	Linear Predictive Coding Cepstral Coefficients
MADA	Morphological Analysis and Disambiguation for Arabic
LVCSR	large vocabulary continuous speech recognition
LVSR	large vocabulary speech recognition
MAER	Minimum Average Error Rate
MFCC	Mel-Frequency Cepstral Coefficient
MLP	Multi-Layer Perceptron
MMI	maximum mutual information
MPE	minimum phone error
MSA	Modern Standard Arabic
NFPN	Neural Fuzzy Petri Net
NN	Neural Network
OOV	Out-Of-Vocabulary
PAL	Palestine
PCA	Principle Component Analysis
PCs	Personal Computers
PER	Phone Error Rate
PFN	Phoneme Filter Neural Network
PLP	perceptual linear prediction coefficients
PNN	Probabilistic Neural Network
PSD	Power Spectral Density
RBF	Radial Basis Function
RNN	Recurrent Neural Network
ROS	Rates Of Speech
Rprop	Resilient Back-Propagation
SGD	stochastic gradient descent
SMLP	Sequence of MLP
SNN	Sub-Neural-Network
SNR	Signal to Noise Ratio
SoC	System-on-Chip
SVM	Support Vector Machine
SYR	Syria
TDNN	Time Delay Neural Network
TIMIT	Acoustic-Phonetic Continuous Speech Corpus
VC	Vowel-Consonant
WER	word error rates
WTA	winner-take-all
ZCR	Zero Crossing Rate

Chapter One

Introduction

Speech is a natural communication method between humans and is the fastest form of data input between humans and technology. Automatic speech recognition (ASR) is a process that converts an acoustic signal, captured by the device microphone or over a telephone line, to a set of textual words. Over the years, ASR systems have been developed for many via-voice applications. Examples include: speech to speech translation (Lavie et al., 1997, Wahlster, 2000), dictation (Murveit et al., 1993, Lee, 1997, Lee et al., 2009), Computer aided language learning (Witt and Young, 1997, Xu et al., 2009, Peabody, 2011), and voiced based information retrieval (Franz and Milch, 2002, Zue et al., 2000) etc. ASR also plays a significant role in helping handicapped people to interact with society. An additional advantage of speech input systems is that the speech capture devices are already built into a large variety of modern devices (mobile phones, PCs etc.). As a result, capture of the speech input requires no extra peripherals. Consequently, applications with speech interfaces are desirable.

Accurate acoustic models (AM) are a significant requirement of automatic speech recognizers. Acoustic modelling of speech describes the relation between the observed feature vector sequence, derived from the sound wave, and the non-observable sequence of phonetic units uttered by speakers. Hidden Markov Models (HMMs) (Rabiner, 1989) using Gaussian Mixture Models (GMMs) are one of the most common types of acoustic models. Each HMM state typically models a 10 msec frame of the spectral representation of the speech signal. Other natural-based models have also been effectively used to model acoustic sequences. This thesis investigates acoustic models based on Deep Neural Networks (DNN) (Hinton and Salakhutdinov, 2006) and a Deep Echo State Networks (Jaeger, 2005) trained on labelled data. When used together with an HMM system for phonetic boundary detection, the Hybrid HMM/NN system is investigated as a state-of-the-art speech recognition system for multi-dialect Arabic speech recognition.

1.1 Overview of speech recognition

Speech recognition can be basically classified into two modes. The first is isolated word recognition; where the words in this mode are surrounded by clear silence i.e. well known boundaries. The second is continuous speech recognition. This second mode is more difficult than isolated word recognition because the word boundaries are difficult to detect. A word may be uttered differently from one speaker to another due to the differences in dialects, gender and age; words may also be uttered in different ways by the same speaker as a result of emotion and illness. Temporal variability, due to differences in speaking rates, is easier to handle than acoustic variability introduced as a result of different pronunciations, accents, volumes, etc. (Tebelskis, 1995). In addition to speaker variability, all speech recognition systems are also affected by variability between environments. The environment may introduce corruption into the speech signal because of background noise, microphone characteristics, and transmission channels (Yuk, 1999, Jou, 2008).

Speech recognition systems can be further classified as being either a speaker dependent or a speaker independent system. A speaker dependent recognition system can extract uttered information from a specific speaker, or range of speakers, whose acoustic features have been previously installed from a training speech database. This type of system is called a ‘Closed-set’ speech recognition system as the training dataset contains uttered information for all speakers. An ‘Open-set’ speaker independent speech recognition system is one where there is no uttered information for the recognised speaker contained within the training dataset.

Speech recognition can be achieved at a variety of levels of speech (Phone/ Phoneme/ grapheme, syllable, word, phrase, etc.). A phoneme is generally considered as “the smallest meaningful contrastive units in the phonology of a language” (O'Shaughnessy, 1987). They are defined by “minimal pairs” which produce a change of meaning if any one phoneme is changed. Thus, phonemes are specific to a particular language. A phoneme can also be single or a set of phones. While, a phone is a single unit of speech sound, allophones are all possible spoken sounds that are used to pronounce a single phoneme. The human brain is presumed to perceive a given set of allophones as a particular phoneme (Hawkins, 1988). If a phoneme is defined as the smallest unit of sound which can differentiate meaning, then a grapheme can be defined as the smallest unit in the writing system of any language that can differentiate meaning. A grapheme can be a symbol or a letter. Sound–letter correspondence refers to the relationship between sound (or phoneme) and letter (or grapheme). A phonetic transcription system, like the International Phonetic Alphabet (IPA), aims to transcribe the pronunciation of a language in a standard form. While a phonemic transcription system usually disregards all allophonic differences and represents them using the same grapheme; it is also known as a representation of phonemic structure. However, graphemic system uses one-to-one letter to sound (phoneme to grapheme) rules for each word to generate a pronunciation dictionary. In this work, phonemes are the symbols used to produce the Arabic pronunciation dictionary and phones will be used only to describe a particular HMM designed to represent a specific sound.

Phonemes play a major role in most current continuous speech recognition systems, and they can be categorised into two main groups consonants and vowels. A definition of vowels and consonants as stated in (Mosa and Ali, 2009) is: "Vowels are produced without obstructing air flow through the vocal tract, while consonants involve significant obstruction, creating a noisier sound with weaker amplitude." In the Arabic language, consonants are further categorised into four classes. These are: voiced and unvoiced stops, voiced and unvoiced fricatives, nasal and the trill & lateral classes. Long and short vowels then make up a fifth class (Mosa and Ali, 2009).

Speech phoneme segmentation is a real challenge for continuous speech recognition systems. In limited vocabulary isolated word recognition, the problem can be easily solved by determining the correct boundary of the isolated words and rejecting the artefacts of speech such as noise and intra-word stops. With regard to large vocabulary continuous speech boundary detection, the problem becomes much more difficult because of the intra-word silences and other artefacts. These problems are commonly reduced by applying speech boundary detection algorithms.

Speech activity detection algorithms can be applied on pre-emphasised speech signals to detect silence/speech boundaries. The most common methods used for end point detection are Energy profile and Zero Crossing Rate (ZCR) (Chen, 1988). Typically, in the Energy profile and Zero Crossing Rate algorithms an adaptive threshold is applied, based on the characteristics of the energy profile, in order to differentiate between the background noise and the speech segments. However, this algorithm is very sensitive to the amplitude of the speech signal, such that the energy of the signal affects the classification results. This is especially a problem in noisy environments. Recently, a new end point detection algorithm has been proposed that uses Entropic contrast (Waheed et al., 2002). This algorithm uses features of the entropy profile of the speech signal, rather than the energy profile of the signal, for boundary detection. The calculation of the entropy is applied in the time domain. Crucially, this profile is less sensitivity to changes in amplitude of the speech signal.

Despite the use of the above techniques, current speech segmentation techniques do still introduce errors into the segmentation process. An alternative is to perform the recognition process without prior segmentation. This method includes the 'silence' as a phoneme, and the network is trained to recognize it in the same manner as the other patterns.

The phonemic representation of a given word is used in most speech recognizers to identify it. Thus, the availability of speech data that has time-aligned and labelled at phonemic level is a fundamental requirement for building speech recognition systems. Time-aligned phonemic labels can be either manually produced by expert human labellers or automatically produced using automatic methods. Though manually-aligned data is considered more accurate than automatically-aligned data (Cox et al., 1998, Hosom, 2000b), it is very time consuming to use manual aligning for large speech corpora. The most common method for automatic speech alignment is called "*forced-alignment.*" The most common method for forced alignment is to build a phonetic recognizer based on a Hidden Markov Model (HMM) (Brugnara et al., 1993, Ljolje et al., 1997, Wightman and Talkin, 1997, Hosom, 2000a, Toledano et al., 2003, Yuan and Liberman, 2008, Yuan et al., 2013).

Irrespective of whether the signal is segmented or not, all speech recognition systems use a feature extraction algorithm as the initial speech processing stage. The most common feature extraction techniques are Mel Frequency Cepstral Coefficients (MFCC) (Davis and Mermelstein, 1980), Fast Fourier Transform (FFT) (Loan, 1992) and perceptual linear prediction coefficients (PLP) (Hermansky, 1990). Typically, ASR system represents the speech signal with the state of the art Mel Frequency Cepstral Coefficients (MFCCs).

The major concerns of the automatic speech recognition are determining a set of classification features and finding a suitable recognition model for these features. HMMs (Rabiner, 1989), which are a special case of regular Markov models, have been demonstrated to be a powerful model for representing time varying signals as a parametric random process (X.D. Huang et al., 2001, Rabiner, 1989). Commonly, Hidden Markov Models (HMMs) are used for modelling the MFCCs observation sequence. These features are computed every 10 ms with an overlapping analysis window of 25 ms. Artificial Neural Networks (ANNs) such as Multi-Layer Perceptron (MLP), Deep Neural Networks (DNN) Recurrent Neural Networks (RNN), and Echo State Network (ESN), have also been widely used for representing time varying quasi-stationary signals.

1.2 Research Motivation

Speech is a desirable communication method between humans and computers and automatic speech recognition (ASR) has been achieved to a large extent for the English language. Arabic is one of the oldest living languages and one of the oldest Semitic languages in the world, it is also the fifth most generally used language and is the mother tongue for roughly 200 million people. Arabic speech recognition has been a fertile area of reasearch over the previous two decades, as attested by the various papers that have been published on this subject. Hence, the Arabic language is selected to be the target language for this research.

A phonemic represented Arabic speech database that is time-aligned and labelled at the phonemic level is a fundamental requirement for building an Arabic speech recognition system. The availability of such data is currently very limited for almost all Arabic corpuses. Thus, forced-alignment, based on a Hidden Markov Model (HMM), is used in this work as an automatic speech alignment method for large vocabulary speech recognition (LVSR).

Motived by the simple and efficient deep learning training algorithms available in different types of neural networks, and their success for modelling acoustic signals for many languages that achieved a higher classification accuracy than HMMs in many comparative speech recognition researches (Skowronski and Harris, 2006, Skowronski and Harris, 2007, Triefenbach et al., 2011, Dahl et al., 2012, Graves et al., 2013, Triefenbach et al., 2013, Mohamed, 2014), Deep neural networks (DNN, and Deep ESN), as well as conventional NN (MLP, ESN), are proposed in this work for creating Arabic acoustic models that can be used for continuous speech unit recognition to classify an input sequence phoneme into specific output classes.

1.3 Research Aim and Objectives

The central aim of this research is to investigate phoneme and acoustic models based on Deep Neural Networks (DNN) and a Deep Echo State Networks for multi-dialect Arabic Speech Recognition. The TIMIT corpus with a wide variety of American dialects is also used to evaluate the proposed models. This aim was achieved through the following objectives:

- To segment and label Arabic corpora that is suitable for implementing our aim. In order to investigate this objective, a developed Arabic phoneme database (APD) is manually timed and phonetically labelled. This dataset is constructed from the KAPD database for Saudi Arabia dialect and the Centre for Spoken Language Understanding (CSLU2002) database for different Arabic dialects. This dataset covers 8148 phonemes. In addition, a corpus of 13 hours of Arabic speech randomly selected from the Levantine Arabic dialect database that is used for training and 2.4 hours for testing are revised and transcription errors are manually corrected. The selected dataset is labelled using the HTK Toolkit.

◦ To analyse the extracted features using the state-of-the-art extraction technique, the Mel-frequency Cepstral Coefficients (MFCC), the Linear Predictive Code (LPC) algorithms are also used for comparison purpose.

◦ To develop and evaluate MLP-based NN classifiers (Shallow and Deep models) for Arabic phoneme recognition and acoustic models. Shallow and Deep MLP NN are investigated for Arabic phoneme speech recognition. The systems are trained on the developed King Abdul-Aziz Arabic Phonetics Database (KAPD) Saudi Arabia database and the Centre for Spoken Language Understanding (CSLU2002) database using MFCCs features only to train and test the systems. To evaluate the adoption of the DNN model in developing an acoustic modelling Arabic speech recognition system instead of a phoneme-based system. For large vocabulary Arabic dataset, the Levantine corpus is used to train DNN acoustic models using MFCCs features and online training algorithm (Stochastic gradient decent algorithm).

- To apply a post-processing enhancement to the framewise DNN outputs in order to improve the framewise phoneme recognition acoustic model accuracy and to find the accuracy in phoneme level instead of the frame level.

- To develop and evaluate an ESN for Arabic phoneme recognition with different learning algorithms. This investigated the use of the conventional ESN trained with supervised and forced learning algorithms. A novel combined supervised/forced supervised learning algorithm was also developed to enhance the performance of the ESN for speech recognition. In addition, deep ESN is implemented for Arabic phoneme recognition. These systems were trained on the developed KAPD Saudi Arabia dialect and the Centre for Spoken Language Understanding (CSLU2002) database using offline training algorithm. Both MFCCs and LPC techniques were used and compared as the input feature extraction technique. For large vocabulary Arabic dataset, the Levantine corpus is used to train ESN acoustic models using MFCCs features and online training algorithm (Stochastic gradient decent algorithm).

- To investigate deep learning for framewise phoneme classification and acoustic modelling using Deep Neural Networks (DNNs) and Echo State Networks (ESNs) on TIMIT corpus.

- To compare the respective performances of the RBM DNN and the ESN speech recognition systems as well as undertake a fine comparison between the performances of the proposed systems investigated in this thesis and the published works in equivalent projects based on framewise phoneme recognition using the various corpora.

1.4 Thesis Organization

This thesis is organized as follows:

- Chapter 2 provides a general overview of current methods for automatic speech alignment and recognition systems. A state-of-the-art HMM/GMM system is first presented. Previous ASR modelling efforts using the state of the art HMM and neural network techniques are then presented. Finally, existing Arabic speech recognition works are summarised in this chapter.

- Chapter 3 provides an explanation for the linguistic aspects of Modern Standard Arabic (MSA), primarily the orthography and pronunciation rules of Arabic language and Arabic transcription systems. The proposed Arabic databases that used in this dissertation are described and their limitations mentioned.

- Chapter 4 briefly describes the Front-end speech processes such as speech segmentation and feature extraction techniques that are used in this thesis. An Entropy-based algorithm is firstly presented and investigated for speech segmentation experiments, the Mel-frequency Cepstral Coefficients (MFCC) feature extraction techniques are also briefly described.

- Chapter 5 provides a description of the frame-based HMM that is considered the foundation for the forced-alignment algorithm which is used in this thesis for timing and labelling the Levantine Arabic dataset. The automatic speech alignment experimental setup including the proposed transcription system, text normalization, building the Arabic pronunciation dictionary and the frame work are presented.

- Chapter 6 briefly describes the conventional and deep NNs based models for Arabic phoneme recognition. Experiments and the results of using conventional Shallow and Deep MLP NNs for isolated speech unit recognition are first presented. Deep neural networks that are proposed in this work for creating Arabic acoustic models are then described.

- Chapter 7 details the conventional and deep Echo State Networks architectures with their experimentation applied on the proposed databases. Finally, fine comparisons between the performances of the conventional and deep MLP NN and the conventional and deep ESN, as well as between the ESN performance for Arabic speech and the HMM equivalent performance for different languages are presented.

- Chapter 9 explored deep learning for framewise phoneme classification and acoustic modelling using Deep Neural Networks (DNNs) and Echo State Networks (ESNs) on TIMIT corpus.
- Chapter 8 provides a conclusion of the work undertaken as well as a discussion on the future work required to produce a continuous Arabic speech recognition system.

1.5 Thesis Contributions

The major contributions of this thesis are:

- A developed Arabic phoneme database (APD) that is manually timed and phonetically labelled. This dataset is constructed from the KAPD database for Saudi Arabia dialect and the Centre for Spoken Language Understanding (CSLU2002) database for different Arabic dialects. This dataset covers 8148 phonemes.
- A corpus of 13 hours of Arabic speech randomly selected from the Levantine Arabic dialect database that is used for training and 2.4 hours for testing. The main contribution for the selected dataset is that we revised and corrected the transcription errors manually. In addition we labelled this selected dataset using the HTK Toolkit to be ready for further research.
- Deep Echo State Networks (DESN) acoustic models for Automatic Speech Recognition (ASR) are introduced. This new model is evaluated on the Levantine dataset and compared with the results obtained from the baseline Deep Neural Networks (DNNs). Both models we were first to introduce the for Arabic speech recognition system.
- A novel adaptation for the ESN is presented that can be used to enhance the performance accuracy for the ESN.
- Empirical comparison for the proposed machine learning techniques conducted on the three proposed corpora namely KAPD, CSLU2002 and Levantine Arabic speech are presented as well as to the TIMIT corpus with a wide variety of American dialects

is also aimed to evaluate the proposed models and to make a direct comparison them.

- A Neural Network toolkit, updated to C++ visual studio 2010. This toolkit includes the MLP, ESN, and DESN algorithms and is made freely available.
- The HTK toolkit is updated to C++ visual studio 2010.

Chapter Two

Background and Literature Review

State-of-the-art Automatic Speech Recognition (ASR) systems have, for many decades, used Hidden Markov Models (HMMs) for modelling the sequential structure of speech signals using a mixture of Gaussians. Since the early 90's an alternative approach has been used, with limited success compared to Gaussian Mixture Models (GMMs), that estimates the fit using types of Artificial Neural Networks (ANNs) that takes contextual window of frames as input and produce posterior probabilities for the HMM states as outputs. Recently, there has been a renewed interest in exploiting Artificial Neural Networks (ANNs) since the Deep machine learning field was introduced in 2006 by (Hinton et al., 2006). The ESN can also be utilized as probabilistic models for extracting features that can be used in modelling time series prolems (e.g. speech and image data) whithin hybrid systems (ESN-ANN, ESN-HMMs). The literature review in this Chapter presents a brief survey of relevant literature covering speech recognition technologies for a wide variety of languages and for Arabic language. A general overview of using Hidden Markov Models (HHMs) especially for speech alignment, as well as Artificial Neural Networks (ANNs) and Hybrid approaches for speech recognition tasks are presented. Specific attention is then applied to research that has explored Arabic speech recognition.

The rest of this Section is described as follows: Section 2.1 focuses on providing a background description of the HMMs for speech recognition, specifically speech alignment and labelling. Section 2.2 then describes the shallow and deep neural network architectures that have been effectively exploited for speech recognition, whilst Section 2.3 presents hybrid HMM/NN systems. In Section 2.4 an end-to-end neural network speech recognition system is described. Finally, a general literature review for Arabic speech recognition is presented.

2.1 A basic HMM model for ASR

The goal of automatic speech recognition is to produce an accurate transcription of spoken words. The major concerns of modern ASR systems are determining a set of classification features and finding a suitable recognition model for these features. HMMs (Rabiner, 1989) are a special case of regular Markov models that have been demonstrated to be a powerful model to represent time varying signals as a parametric random process (X.D. Huang et al., 2001, Rabiner, 1989). Typically, modern ASR system represent the speech signal using state-of-the-art Mel Frequency Cepstral Coefficients (MFCCs). The Hidden Markov Models (HMMs) are then used to model the MFCCs observation sequence.

These features are computed every 10 ms with an overlapping analysis window of 25 ms. The sequence of the acoustic feature vectors $Y_{1:T} = y_1, \dots, y_T$ that are extracted from the speech signal can be supposed as the observable data. The hidden state sequence $w_{1:L} = w_1, \dots, w_L$ supposed as the underlying symbol sequence, can then be inferred from the observation sequence. Each HMM state is modelled using a Gaussian mixture model (GMM) with several diagonal covariance Gaussians. HMMs are trained by maximizing the data likelihood:

$$\mathcal{F}_Y = \sum_{n=1}^{N} \log(p(Y_{1:T_n}^n | w_{1:L_n}^n)) \tag{2.1}$$

The decoder attempts to find $\widehat{w}$ that maximizes the posterior probability $P(w|Y)$ using the Viterbi algorithm:

$$\widehat{w} = \arg\ \max_w\{P(w|Y)\} = \arg\ \max_w\{P(Y|w)P(w)\}. \tag{2.2}$$

P(w|Y) is challenging to model directly. However, there are many proposals that have been made which use discriminative models (Gales, 2007), such as maximum mutual information (MMI) (P.S. Gopalakrishnan et al., 1991, Woodland and Povey, 2002) and minimum phone error (MPE) (Povey and Woodland, 2002, Povey, 2004) to maximize the posterior probability P(w|Y).

The likelihood P(Y|w) of Y given w, and the prior probabilityP(w) of the word sequence w are determined by an *acoustic model* and a *language model* respectively. The language model is an *N-gram* model trained separately on an enormous text in which the probability of each word $P(w)$ is conditionally obtained on its *N-1* predecessors.

The acoustic model for any given word w is produced by concatenating phoneme models of the word as defined by a pronunciation dictionary. Multiple pronunciations for each word can be found in the pronunciation dictionary. The sequence of the base phonemes K_w is defined as $q_{1:K_w}^{(w)} = q_1, \dots, q_{K_w}$ for each pronunciation. Therefore, the likelihood $P(Y|w)$ can be calculated over these pronunciations using this formula:

$$P(Y|w) = \sum_Q P(Y|Q)P(Q|w) \tag{2.3}$$

where the $\sum_Q$ denoted by the overall pronunciation sequences for given w and Q is a specific sequence of pronunciations

$$P(Q|w) = \prod_{l=1}^{L} P(q^{(w_l)}|w_l)$$

where $q^{(w_l)}$ is a one possible pronunciation for word w_l.

Each HMM associated with a phoneme is called a phone model. Phones are used as the main speech unit to be modelled by HMMs, where each phone is modelled using left-to-right HMMs with three states plus the entry and exit states as shown in Figure 2.1.

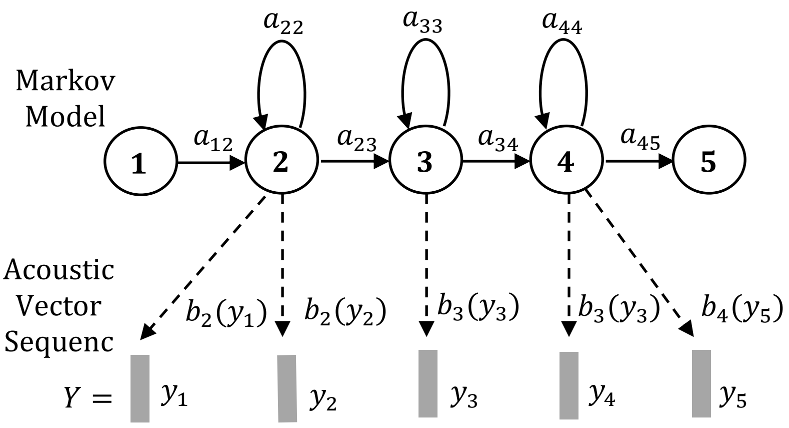


Figure 2.1- HMM-based phone model (Gales and Young, 2007).

HMMs make a transition from the current state to one of the neighbour states every time step t. A transition probability $\{\mathrm{a_{ij}}\}$ is associated for each arc between two connected states. On entering a state s_j, a feature vector is generated with the output distribution $\{b_{\mathrm{j}}(\,)\}$ associated with the state being entered. This form of the HMM process makes two unrealistic conditional independence assumptions:

- The current state is conditionally independent from all other states given the prior state.
- Observation frames are conditionally independent of all other observation frames, in the utterance, given the current state.

For classification of frames, in state-of-the-art ASR, a context window of frames is taken. This context window includes the feature vector of the current frame and may include feature vectors of the surrounding frames. Gaussian Mixture Models (GMMs) are easily trained using the expectation-maximization (EM) algorithm (Dempster et al., 1977) to estimate the phonetic likelihoods. Only one frame is covered in the contextual window, and the delta features, which implicitly contain information about neighbouring frames, are included. With a NN classifier, the input feature vector is often a context window that includes multiple of frames in addition to their delta features. These classifiers estimate the posterior probability distribution over the categories. These posterior probabilities are passed to the Viterbi decoder to search for all the possible word sequences for the utterance using pruning to eliminate the unlikely hypotheses. A description of Viterbi decoding algorithm can be found in (Huang et al., 2001). Finally, the most likely word sequence is considered as output. The Viterbi search is also used to generate the timing of each state for each input utterance using a dictionary and word level transcriptions.

The major reasons for the widespread use of HMM based systems are that the mathematics are well formulated and their performances are often the most superior to other approaches. However, the disadvantages of HMM systems are the amount of data required to train these systems and the fragility of these recognizers when applied under various conditions. In addition, independence between values of the frame cannot be fully justified. Also, since no speech specific knowledge (e.g. data-driven clustering of phonetic categories and perceptually related warping of the spectrogram) is used in the HMMs likelihoods computation at each frame of each phone, these values are estimates of phonetic qualities of the speech as well as the channel and noise conditions of the data.

2.2 Methods of automatic alignment systems

Speech recognizers are used to identify the phonemic representation of a given word. Thus, the availability of speech data with time-aligned phonemic labels is a fundamental requirement for building many speech recognition systems.

Time-aligned phonemic labels can be manually transcribed by expert human labellers or automatically produced using an automatic method. Though manually-aligned data is considered more accurate than automatically-aligned data (Cox et al., 1998, Hosom, 2000b, Cosi. et al., 1991, Ljolje et al., 1994), it is very time consumption to use manual methods for aligning large corpora. Hence, a fast and accurate automatic alignment system is needed for time-aligned phonemic labels in large vocabulary speech recognition systems (LVSRS).

A wide variety of approaches has been used for automatic alignment systems. The most common method is called "*forced-alignment*" several phonemic recognizers based on Hidden Markov Model (HMM) have been created (Brugnara et al., 1993, Ljolje et al., 1997, Wightman and Talkin, 1997, Hosom, 2000a, Toledano et al., 2003, Yuan and Liberman, 2008, Yuan et al., 2013). Dynamic Time Warping (DTW) is another approach that has been used to align sequences. Dynamic Time Warping (DTW) is a dynamic programming approach that is used to align the features of the input utterance against the features of a reference utterance. Many publications have used DTW for phonetic alignment of speech (e.g. (Wagner, 1981, Malfrère and Deroo, 1998, Campbell, 1996)). In addition to HMM and DTW, a wide variety of other approaches have attempted to align and label speech based on different techniques. For instance, van Santen and Sproat in (Santen and Sproat, 1999) tried to align the speech based on detecting phonetic boundaries using edge detectors on the energy information and spectral-domain representations in different frequency bands.

Typically, the same process that is used for speech recognition is used for automatic forced alignment. Recognition of the utterance is, usually, performed using a search procedure, such as Viterbi search, that is constrained to prior known sequences of phones. This search produces the locations and the identities of the phones. The phonetic sequences are often prior determined by a pronunciation dictionary. Using this way, the forced alignment method can produce the phonetic alignment by forcing the recognition result to obtain the utterance phonetics locations and identities.

Many researchers have used this approach to perform the alignment of speech. For example, Brugnara et al. (Brugnara et al., 1992, Brugnara et al., 1993) developed an automatic alignment and labelling speech system using a forced alignment HMM based system using a combination of spectral variation features and the standard cepstral-domain features to compute the state occupation likelihoods. These additional features enhanced the system with a 2% reduction in error. Wightman and Talkin in (Wightman and Talkin, 1997) also developed an automatic alignment system based on HMMs, termed "the Aligner." The Viterbi search was implemented using the HTK toolkit (Young et al., 2003) and trained on the TIMIT phonetic sequences. Ljolie and Riley (Ljolje et al., 1997) built their HMM alignment system based on the availability of the date; i.e HMMs with different types of phonetic models. A complete triphone model is used if the training data is enough for the left and right contexts of the given phone. Alternatively, a quasi-triphone model (one side-state is dependent on the other side) is tried if sufficient data is not available. If the training data is still not enough, context-independent phone models are applied. Svendsen and Kvale (Svendsen and Kvale, 1990) segmented the speech signal into equal length segments of speech, and then the HMM was used within the segment during the Viterbi search. State transitions are then only allowed at the segments boundaries. A phonetic alignment system based on a hybrid HMM/ANN model was developed by Hosom (Hosom, 2000a) using the CSLU Toolkit.

A wide verity of languages have been segmented and labelled using the HTK Toolkit. Rapp in (Rapp, 1995) trained the forced alignment system for German read speech. Sjölander (Sjölander, 2003) also developed an automatic time-aligned phone transcription system of spoken Swedish. Wightman and Talkin in (Wightman and Talkin, 1997) used the same HTK Toolkit for English language trained on the TIMIT database. The EasyAlign (Goldman, 2011) system, based on the HTK Toolkit, was implemented for aligning English and French speech.

The reported performances for these systems, range from 80%-89% agreement within 20 ms of the standard HMM based forced alignment systems when compared to manual segmentation on the TIMIT corpus (Hosom, 2000a, Hosom, 2009). Many researches have attempted to develop alignment systems with higher performance. For instance, Yuan et al. in (Yuan et al., 2013) investigated the use of phone boundary models within the forced alignment HMMs system. They achieved a 93.92% agreement within 20 ms using the HTK Toolkit.

It is clear from the literature that aligning the speech is a fundamental task for large vocabulary speech recognition systems. Since there is no available aligned and labelled Arabic dataset that can be used for speech recognition area, the Levantine Arabic corpus that is described in Chapter Three has been aligned and labelled in this work using the HTK Toolkit. Details and the experimental results of forced alignment are presented in Chapter Five.

2.3 Machine learning algorithms

In general, machine learning algorithms have been classified into three main fields:

- **Supervised learning**

Supervised learning systems use a training dataset constructed from features of instances with its labels as pairs, represented by $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. The aim of this learning is to predict the output y for any given input feature x. There is an interesting type of supervised learning technique that replaces the actual output $y(t)$ by the teacher output (desired output) $d(t)$ during the training stage (hmidt et al., 2010, Williams and Zipser, 1989). This technique is called Teacher-Forced supervised learning.

- **Unsupervised learning**

Unsupervised learning systems use an unlabelled training dataset, represented only by the features of instances $\{x_1, x_2, \ldots, x_n\}$. The main use of this type of learning is for clustering purposes, i.e k-Means (Lloyd, 1982).

- **Reinforcement learning**

Reinforcement learning systems observe an environment x, then implement an action a, and obtain a reward r (Duda et al., 2001). Maximizing the future rewards is the main goal of this learning and this is achieved by choosing actions that lead to this maximum rewards.

2.3.1 Static Neural Networks

2.3.1.1 Feed-forward Neural Networks

The feed-forward Neural Networks (FNNs) are a widely used artificial neural network architecture (ANN). There are several architectures of neural networks (NNs); each with its own unique set of connections, training algorithms, and applications. They are all, however, based on common principles. Neural networks such as Multi-Layer Perceptron (MLP) are constructed from a large number of processing elements (called neurons, nodes or units). Each neuron influences the other's behaviour via weighted connections; where each neuron output is computed as the sum of the product of nonlinear weights with its inputs. This output is broadcasted to all other units over its outgoing connections.

Initially inspired by neurobiology, MLP connectionist systems have since developed into an interdisciplinary field that classify static features, covering physics, mathematics, linguistics, computer science and electrical engineering. Nowadays, the properties of neural computation are used in a wide range of research (Tebelskis, 1995). These properties are:

- **Trainability**. Neural network can be trained to learn the relationships between any set of input and output patterns. This can be used to teach a neural network to categorize speech patterns into phoneme classes.

- **Generalization**. Rather than the network memorizing the training data; the underlying patterns are learnt by the neural network, so the new examples can be generated from the training data. This property is vital in a speech recognition task since the auditory input patterns are never precisely the same.

- **Nonlinearity**. Nonlinear functions can be computed by neural networks on their input. This enables arbitrarily complex transformations to be performed on the data. This is useful because of the highly nonlinear processing nature of speech.

- **Robustness**. Noisy training data can help a neural network to perform better generalization (Tebelskis, 1995). Speech patterns are notoriously noisy, so this property is valuable.

- **Parallelism.** Neural networks implement massive parallel computers since they are inherently parallel in nature. This potentially leads to very fast processing of data.

Until recently, most machine learning techniques have exploited shallow-structured architectures which contain a single layer of nonlinear feature transformation. Examples of these architectures are support vector machines (SVMs), kernel regression, logistic regression, and multi-layer perceptron (MLP) neural networks that consist of a single hidden layer. The main properties of shallow learning models are that they have a simple architecture and are effective in solving simple problems. However, these shallow architectures have limited representational and modeling power that can lead to difficulties when dealing with perceptual problems such as human speech.

2.3.1.2 Deep Belief Networks

Deep Neural Networks (DNNs) are believed to achieve high performance on complicated real applications such as vision and speech by better representing their complex functions. Deep learning is a category of machine learning techniques, where hierarchical architectures are used to process natural signals (e.g. speech signals) using several non-linear information stages such signals often contain features that are inherently in nature. Feed-forward neural networks (FNNs) (e.g., Multi-Layer Perceptron (MLPs)) with many hidden layers are considered a good example of the deep models. Usually MLPs use the Backpropagation (BP) algorithm for learning the network weights. However, this learning algorithm does not work well for learning networks with several hidden layers (Bengio, 2009, Glorot and Bengio, 2010). The BP algorithm often gets trapped in local optima and the severity of this non-optimal learning increase as the depth of the neural networks increases.

DNNs are a feed-forward Neural Networks that has many layers of non-linear hidden units between their inputs and their outputs. Such a network can be trained one layer at a time as a generative model of the input data, (i.e. a window of speech coefficients with one frame delay), using Restricted Boltzmann Machines (Smolensky, 1986). A subsequent discriminative fine-tuning phase using the backpropagation algorithm, then fine-tunes the features in every layer to make them more valuable for discrimination. This type of network is also called Deep Belief Networks (DBNs) in the literature (e.g., (Dahl et al., 2011, Mohamed et al., 2010, Mohamed et al., 2009, Mohamed et al., 2012)). A

DBN can be effectively used to initialize the DNN training; such a network is called DBN–DNN (Hinton et al., 2012). The sparse encoding symmetric machine (SESM), developed by Ranzato et al. (Ranzato et al., 2007), has a similar approach to RBMs by way of building blocks of a DBN. SESM can also be used to initialize the DNN training.

There are two key properties for deep learning techniques. A DNN is trained as a generative model[1], and then an additional top layer is used to perform the discriminative tasks. An unsupervised pre-training phase, which train multilayer generative neural network one layer at a time, that makes them effective in extracting structures that represent the input features in large unlabelled training data is also employed. Most of the work, that uses deep learning, can be categoraized depending on how the architectures are intended for use into three main classes:

1) Generative deep architecture: this architecture aims to distinguish the high-order correlation properties of the visible or observed data for pattern synthesis or analysis purposes. Deep generative models are not always required to be probabilistic in nature (i.e., the parametric feature encoding techniques used in DNN auto-encoders, whose targets are the observed data itself, or “bottleneck-layer" that produce low-dimensional codes for the original inputs). In this architecture, DNNs are similar to other dimensionality reduction methods such as Principle Component Analysis (PCA). Generally, the use of generative models played a significant role in the interest in deep learning for feature coding and speech recognition (Hinton et al., 2012, Dahl et al., 2011, Deng et al., 2010).

2) Discriminative deep architecture: this type of architecture is, often used to characterize the posterior distributions of classes, conditioned on the observed data, to provide discriminative power for pattern classification. Discriminative training using the back-propagation algorithms has been used in many Neural Networks (NNs) approaches for acoustic modeling. The discriminative approach was first proposed in (Brown, 1987) where generative training is replaced by discriminative modeling for acoustic modeling.

3) Hybrid deep architecture: the goal of this architecture is discrimination. The outcomes of the generative model that are used in (1) are fed into the discriminative

[11] A **generative model** is a model for randomly generating observable-data values.

architicture in (2). such an approach is termed DBN-DNN (Mitra et al., 2014). This hybrid system is different from one that uses NN-HMMs such as (Bengio et al., 1991, Bourlard and Morgan, 1993, Morgan, 2012, Mohamed, 2014), for more discussion about NN-HMMs hybrid systems see Section 2.4.

For large vocabulary speech recognition, deep neural networks must be trained with a large number of parameters. Therefore, overfitting is a potentially serious problem. The term overfitting refers to the gap between training error and test error; i.e. the neural net has learned the training examples very well, but has 'lost' its ability to generalize to a new situation. There are several techniques used to prevent overfitting that can be (e.g. cross-validation, early stopping, regularization, pruning). These techniques can be used to indicate when more training can lead to worse generalization. More recently, Dropout (Srivastava et al., 2014, Srivastava et al., 2012) is a regularization technique that has been used for addressing overfitting. This technique randomly drops units from the neural net during training to prevent units from co-adapting too much. They showed that dropout can improve the performance of the deep neural networks on supervised learning tasks in speech recognition and benchmark data sets. Grubb and Bagnell in (Grubb and Bagnell, 2013) also used a stacking training method that trains multiple simultaneous predictors to simulate the overfitting problem in early layers of a network.

Deep learning has been successfully exploited for feature learning and pattern classification in many research (Hinton et al., 2006, Bengio, 2009, Bengio et al., 2013b). A recent progress in learning algorithms for deep neural network learning has had a strong impact with breakthrough results in classification tasks (Bengio et al., 2007, Ahmed et al., 2008, Lee et al., 2009), in regression (Salakhutdinov and Hinton, 2008), dimensionality reduction (Hinton and Salakhutdinov, 2006, Salakhutdinov and Hinton, 2007a), modeling motion (Taylor and Hinton, 2009), object segmentation (Levner, 2008), information retrieval (Ranzato and Szummer, 2008, Salakhutdinov and Hinton, 2007b), robotics (Hadsell et al., 2008), natural language processing (Collobert and Weston, 2008, Mnih and Hinton, 2009, Weston et al., 2008), and most notably in area of large vocabulary automatic speech recognition (LVASR) (Dahl et al., 2010, Seide et al., 2011, Dahl et al., 2012, Mohamed et al., 2012, Hinton et al., 2012).

Motivated by the significant gains over the state-of-the-art GMM/HMM speech recognition systems on a variety of small and large vocabulary speech tasks (Hinton et al., 2012, Vesel´y et al., 2013, Kingsbury et al., 2012, G.-Dominguez et al., 2014, Sainath et al., 2014, Sainath et al., 2013, Mohamed, 2014), Deep Neural Networks (DNNs) are used in this work for modelling the acoustic signals. Weights in the DNN are generatively pre-trained to maximize the data likelihood before a fine tuning phase using the back-propagation algorithm. Then in the second phase, a deep network is built by adding many hidden layers with many hidden units in each hidden layer. State-of-the-art DNNs (Hinton et al., 2006) is used in Chapter Six and Eight to model the acoustic signals. This system is evaluated on the Levantine and the TIMIT corpora for a framewise phoneme recognition task.

2.3.2 Dynamic Neural Networks

2.3.2.1 Recurrent Neural Networks

RNNs are nonlinear dynamical systems which map input sequences to output sequences. The depth of the RNN depends on the length of the observed data sequence, and it can be as large as that sequence data length. RNNs are dynamical in that output activations of a unit in one layer can feed back as input to units in preceding layers. Since the hidden state of the RNNs is a function of all preceding hidden states; they are inherently deep in time. RNNs are a very powerful model that can be used for modelling sequence data such as speech and text input. These types of task require a system that can store and update context-information, computed from the past inputs sequences, to produce desired output sequences. RNNs are well suited for those tasks as their internal states can represent context-information. There are many types of training algorithms that can be used to train recurrent networks: real-time recurrent learning (RTRL), back-propagation through time (BPTT), and Extended Kalman Filtering techniques (EKF). Commonly these algorithms result in suboptimal solutions with slow convergence. Although RNNs gradient based learning algorithms (e.g. (Werbos, 1990, Williams and Zipser, 1989)) have simple computation for their gradients, they are relatively difficult to train on problems that have long range temporal dependencies. This problem increases as the duration of the dependencies increases (Bengio et al., 1994, Hochreiter and Schmidhuber, 1997, Martens and Sutskever, 2011). For instance, Cyclic dependencies in RNNs can result in bifurcations during training, where infinitesimally slight changes to the RNN parameters compound into large discontinuous changes in its behaviour, this phenomenon can lead to non-converging (Doya, 1992). Another well-known RNNs training problem is the "vanishing gradient" problem, in which RNNs based on the gradient learning algorithm take too much time as errors vanish with the backpropagation learning.

There have been many attempts to resolve the RNN problems. Hochreiter and Schmidhuber (Hochreiter and Schmidhuber, 1997) developed a modified architecture named the Long Short term Memory (LSTM) and implemented it successfully for speech and handwritten text recognition (Graves and Schmidhuber, 2005, Graves and Schmidhuber, 2009). In this architecture, the standard RNN is augmented with "memory-units" to transmit long-term information along with set of "gating-units" that allow the memory-units to interact selectively with the standard RNN hidden state. Another recent effort, that has received much attention for solving the RNN training problems easily, is the Echo-State-Network (ESN) (Jaeger, 2001, Jaeger, 2007) . ESNs typically have thousands of hidden units. One disadvantage of ESN is that they must be initialized with a set of manually fixed connections correctly-scaled to drive the hidden states. Recent developments in Hessian-free optimization in (Martens, 2010) have to some extent overcome these problems by using stochastic curvature estimates or second-order information.

RNN can be categorized depending on how the architectures are intended for use into three main classes:

1) Generative deep architecture: RNNs can be considered as deep generative architectures which can be used to generate and model sequential data (e.g., (Sutskever et al., 2011)). In the recent work of (Martens and Sutskever, 2011), RNNs are used as a generative deep model for speech recognition tasks. Many other researcher have demonstrated the ability of RNN models for generating sequential text characters (Bengio et al., 2013a), and language models (LM) (Mikolov et al., 2010).

2) Discriminative deep architecture: for example, Hermans et al. In (Hermans and Schrauwen, 2013) they reached state-of-the-art performance for Deep RNNs in

character-level language modeling that was trained with the simple stochastic gradient descent.

3) Hybrid deep architecture: for instance to the ESN-ANN described in (Schrauwen and Busing, 2010).

2.3.2.2 Echo State Networks

Echo state networks (ESNs), invented by Jaeger, are a novel structure of recurrent neural networks (RNNs) that contain a large, random, and untrained recurrent “dynamic reservoir” network (several thousand hidden units) which represent a method for utilizing recurrent ANNs for online computations (classification, regression, prediction/generation) on time series. The learning algorithm of the ESNs is very simple and linear, in that only the weights from the reservoir to the outputs neurons are adapted (see Figure 2.2). As a result, the learning process is fast. The optimal output weights for the ESN are obtained when the *MSE* is minimized.

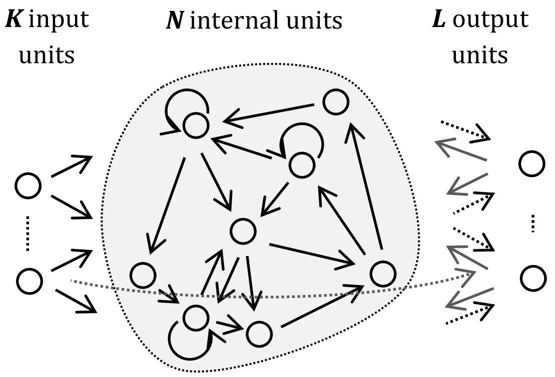


Figure 2.2 - The basic ESN architecture. Dotted arrows indicate trained weights in the ESN approach. Shaded arrows point to optional connections (Jaeger, 2005).

A comparison between conventional RNNs techniques and the ESNs technique in Figure 2.3 shows the main difference between them. In part A-RNN, all the RNN weights are changed during training whilst in part B-ESN only the ESN output weights are changed during training. The Dynamic reservoir in the ESNs, also, contains more neurons than used in the RNNs (Jaeger, 2005). In addition, whilst the RNN has full connectivity, the ESN has designed connectivity, leading to less computation overhead during training. As mentioned in Section 2.1.1.2, there is also a regulation technique that can be used (Srivastava et al., 2014) for addressing overfitting, where units are randomly dropped from the neural net during training.

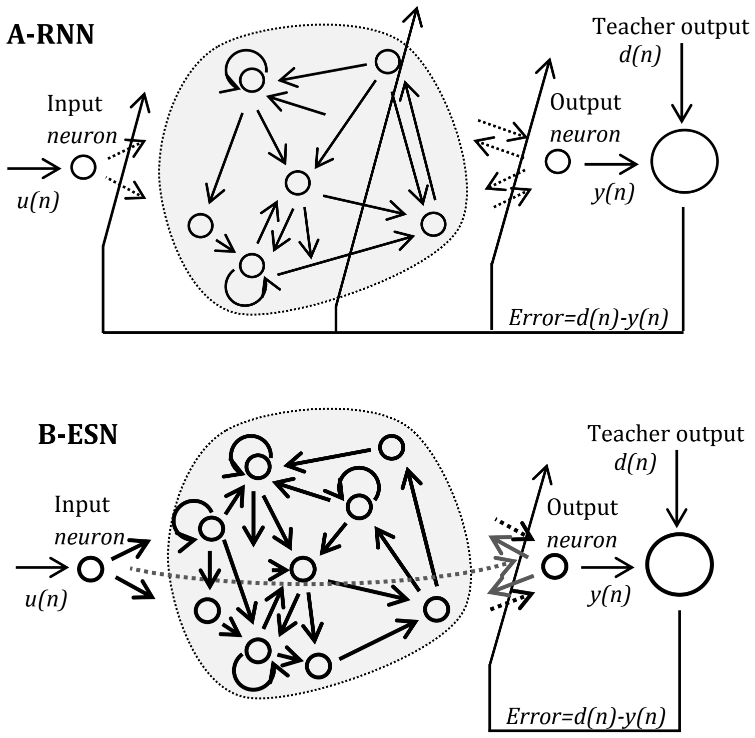


Figure 2.3 - Shows a comparison between RNNs and ESN architecture and training.

ESNs are capable of modelling nonlinear systems with multiple inputs and multiple outputs. Also, being a recurrent network, its outputs can be used as additional inputs by feeding them back to the input stage. This provides a possibility to train ESNs as generators or oscillators (Holzmann, 2008). Based on its ability to model nonlinear complex dynamic systems, Echo state networks have been used on a variety of benchmark tasks. ESN are particularly well suited for classifying temporal signals as part of a dynamic pattern recognition task (Ozturk and Principe, 2007). An ESN with delay and sum readout has been used to perform as a nonlinear identification system “An Audio signal identification” (Holzmann, 2008), and for time series modelling “Mackey-Glass example” (Jaeger and Haas, 2004) and as a 10th order NARMA system (Jeager, 2003, Verstraeten et al., 2007). ESNs have also demonstrated their capability for modelling nonlinear systems and control (Dai et al., 2010). They have been used for predicting the speed deviations in multi-machine power system (Venayagamoorthy, 2007), and for Motor Control (Salmen and Ploger, 2005). Moreover, ESNs have shown good ability for learning grammar structure as part of a natural language task(Tong et al., 2007, Sakenas, 2010) . Some of most impressive applications of the ESNs are on pathological long range dependencies problems (Jaeger, 2012). ESNs have also been used for pattern recognition tasks.

In approaches focused on isolated speech recognition, Japanese Vowels classification was effectively performed using an ESN (Jaeger et al., 2007). At a higher level, an ESN was also successfully used to classify ten isolated English digits (Verstraeten et al., 2007, Skowronski and Harris, 2007, Schrauwen and Busing, 2010, Ozturk and Principe, 2007). In (Skowronski and Harris, 2007) and (Ozturk and Principe, 2007) an echo state network (ESN) was combined with a competitive state machine to construct a predictive ESN classifier. The ESN classifier out-performed the HMM baseline on the same dataset. An ESN was also successfully used to classify ten isolated Arabic digits in (Alalshekmubarak and Smith, 2014). The ESN was combined with the Extreme Kernel machines which named ESNEKM. This system performance is also out-performed the baseline HMM. The ESNs in (Jalalvand et al., 2011) also demonstrate their ability to perform better than the standard HMM approaches on noise-robust recognition research. ESN has been implemented successfully for framewise phoneme and vowel classification task. An ESN with force supervised learning demonstrated the best performance for English vowel classification when compared with other networks (hmidt et al., 2010). Also, in (Hmad and Allen, 2013) an ESN-based Arabic phoneme recognition system trained with supervised, forced and combined supervised/forced supervised learning algorithms using a deep architecture was evaluated on the KAPD and CSLU2002 datasets.

The ESN can also be utilized as probabilistic models for extracting features that can be used in modelling time series prolems (e.g. speech and image data) whithin hybrid systems (ESN-ANN, ESN-HMMs). For instance, it has been shown that ESNs have the capability to extract low dimensional features from a dynamic reservoir for a handwriting recognition task. The main use of the ESN in this process was to generate the features of the data in a high dimensional representation (Sakenas, 2010). Schrauwen et al. in (Schrauwen and Busing, 2010) proposed the Temporal Reservoir Machine (TRM) probabilistic model which operated on sequential speech data and aimed to gain the advantages of ESNs and RBM-based models. In this model, they used a Principle Component Analysis (PCA) dimensionality reduction technique, with a random recurrent network utilized as a feature extractor and the directed graph of RBMs for modelling the time series. This architecture was successfully tested on the classification of spoken English digits. In addition, competitive phone recognition rates was achieved in (Triefenbach et al., 2011) based on a front-end, ESN-HMM hybrid setup on the TIMIT speech recognition benchmark task using 20,000 hidden units. In the recent work of (Triefenbach et al., 2013), the Deep hierarchical architecture of an ESN with bi-directional reservoir architectures was used for acoustic modeling. This produced a very competitive Phone Error Rate (PER) on the TIMIT speech dataset.

ESN has impressive performance in various applications. However, high-performing models potentially need millions of parameters that increase the calculation and memory capacity.

In this thesis, a Deep ESN architecture is proposed for Arabic framewise phoneme classification and Arabic acoustic modeling. This architecture is applied on the proposed datasets namely the KAPD, CSLU2002, and the Levantine datasets.

2.4 Towards End-to-End ASR with Neural Networks

The aim of the neural network in this case is to perform a mapping from the input sequence of frames of speech into a sequence of phoneme labels associated with these frames. When training neural networks for discrimination, typically, a pre-segmented and labelled training data are needed (forced-alignment using HMMs is using for this purpos). Also, a post-processing stage is required to transform the outputs of the discriminative network into label sequences.

However, An interesting approach was proposed in (Graves et al., 2006, Graves and Schmidhuber, 2009, Graves, 2012, Graves, 2008) which removed the necessity of pre-segmenting and labelling the training data as well as the need for post-processing the outputs. This approach enables the RNNs to purely perform sequence classification by themselves. The main idea of this method is interpreting the RNN outputs as the conditional distributions over all probable label sequences given the input frame sequence. Then, a differentiable objective function can be used to optimize those conditional distributions through the correct label sequences. Consequently, no pre-segmentation of data is needed. This approach has been successfully exploited for speech recognition (Graves et al., 2013, Graves and Jaitly, 2014).

In this work, we adopted a first approach with a post-processing stage to transform the outputs of the discriminative network into phoneme sequences.

2.5 Hybrid HMMs/ANN for speech recognition systems

Many research laboratories have developed hyprid HMMs/NN systems; such as Cambridge University (Robinson, 1994), Toronto University (Mohamed, 2014), and International Computer Science Institute (ICSI) (Bourlard and Morgan, 1993), …etc. The main use for the ANNs in these hybrid systems is to estimate phonetic probability (observation probabilities) that are then used as parameters for HMMs instead of estimating them using GMMs as in the standard HMM ASR. Feed-forward MLP NNs have been applied for this function in many ASR systems (Bengio, 1991, Bourlard and Morgan, 1993, Morgan et al., 2005, Yuk, 1999). Recently, The use of neural networks features for ASR have become a component of state-of-the-art recognition systems when they are also refered to as *probabilistic features*.

Artificial Neural Networks which have been discriminatively trained have attractive advantages over GMMs: ANNs provide a better discrimination, they offer better probability density functions estimation, and the input features of ANNs do not necessitate being uncorrelated. Therefore, some of these HMM/NN hybrid-based solutions improved the ASR systems. Significant improvements are reported in the recent work in (Triefenbach et al., 2012, Triefenbach et al., 2013) with 0.9% better recogntion. This improvement was attained over the standard HMM baseline by using Reservoir Computing with a HMM (RC-HMM) system for large vocabulary continuous speech recognition (LVCSR). Also, Mohamed in (Mohamed, 2014) has significantly improved acoustic models on small and large vocabulary tasks by replacing the GMM components with Deep Neural Networks (DNNs) by 2% better performance and Convolutional Neural networks (CNNs) by 2.4% better discrimination as Tandem features.

A Tandem features model is another sucsseful way of using ANNs with HMMs in hybrid systems. In the Tandem model (Hermansky et al., 2000), the log of posterior probabilities, are the output of the NN (sometimes augmented with the PLP or MFCC features). These are decollelated via the Karhunen-Loeve (KL) transform and fed as feature vectors to the standerd HMM/GMM system. Hence, this model is a compromise between the modeling power of ANNs and all of the tools and algorithms previously developed for GMMs. The Tandem model has been applied in much recent research and has improved the recognition performance ASR for many corpora (Hermansky et al., 2000, Triefenbach et al., 2012, Morgan et al., 2004, Ellis et al., 2001).

Usually, a context window of n frames, shifted by 10 ms, is used as an input acoustic feature vector. The context window 10*n ms covers all the frequency bands of the vertical slices from the MFCC spectrogram. In the TRAPS system (Hermansky and Sharma, 1998), horizontal slices represent the features, where each slice focuses on frequency band over 1sec time and a NN band filter is used to produce the posterior probability distribution through class labels. The outputs of these filters for all bands are then combined using another NN to generate the posterior probability distribution for every single frame that can be used in tandem or hybrid systems. The TRAP system was shown to achieve competitive recognition performance with the spectral-based ASR techniques (Hermansky and Sharma, 1998).

In addition, MLP neural networks with a bottle-neck layer have been used to generate features for a GMM-HMM system; these features being termed "bottle-neck features" (Gr´ezl et al., 2007, Gre´zl and Fousek, 2008). This type of features are directly obtained from the neural network without the need of converting output probabilities and are used as features for the subsequent GMM-HMM systems. The MLP network used had more than 3 layers, usually with an odd number of layers and a bottle-neck layer in the middle layer. After training, the outputs of the bottle-neck layer are used as features for the GMM-HMM system. These type of features have improved the recognition system performance (Gre´zl and Fousek, 2008).

The main weakness of the hybrid HMM/ANN systems is the amount of time needed to estimate the observation probabilities during training of the Artificial Neural Network classifier. However, using a special hardware chip with multiple DSPs (Morgan, 1990) produce a reduction in the time required for training neural networks and made them competitive with the GMMs for speech classification and recognition.

A hybrid HMM-ANN systems is used in this work, where HMMs are used for alignment and labelling the uttrences, and the ANN types are used for recognition task. The observation probability distribution (the output vectors are normalised using the softmax function) through the class labels are estimated during the recognition by the proposed neural networks.

2.6 The related work on automatic Arabic speech recognition

Arabic speech recognition has been a fertile research area over the previous two decades as attested by the various papers has been published on this subject. Most previous effort on Arabic ASR has focused on developing recognizers for the formal Arabic language (Modern Standard Arabic MSA). MSA is a linguistic standard used in the Arabic world and is usually employed in the formal media (e.g. broadcast news), courtrooms, lectures, etc. Recent recognizers for MSA in Arabic broadcast news could achieve word error rates (WER) of 15-20% (Billa et al., 2002b, Billa et al., 2002a). However, MSA is not the natural language for native speakers throughout Arabic world. In fact, everyday communication for Arabic speakers is carried out in one of four regional dialects. These Arabic dialects are: Levantine, Gulf Arabic, Egyptian, and North African.

When developing ASR systems for dialectal Arabic, the early work was performed on the CallHome task for Egyptian Colloquial Arabic (ECA) within the framework of the 1996/97 NIST benchmark evaluations with a performance of 61% word error rate (WER) (Zavagliakos et al., 1998). More recent systems obtain around 56% WER on the same task, which is still significantly higher than the word error rates on CallHome data in other languages. A significant word error rate improvements on the same dataset (53.7%) was achieved in more recent system by (Kirchhoff et al., 2002). This improvement was achieved by developing approaches to automatic vowel diacritization to address the lack of short vowels (a, i, u) and other pronunciation information in Arabic texts. However, many of the research implemented script-based recognizers used a grapheme rather than phoneme systems to train the acoustic models in modern and colloquial Arabic (Billa et al., 2002a, Billa et al., 2002b, Al-Shareef and Hain, 2011).

One of the most challenging problems in developing an accurate ASR for Arabic is the lack of diacritized text material. Arabic texts are usually non- diacritized as diacritics are used to prevent missunderstandings for formal texts and beginner' books for students of Arabic. A number of researchers deal with the lack of diacritics by developing systems for automatic diacritization of various Arabic corpuses. Vergyri and Kirchho in (Vergyri and Kirchhoff, 2004) use various procedures to insert the missing diacritics into the transcription. They evaluated the performance of these techniques against manually and automatically diacritized training data using acoustic models trained on these generated data. They, also, explored the effect of increasing the amount dialect training data by adding MSA data and the results showed that there was no much improvement over the baseline (CallHome data only). Soltau et al. (Soltau et al., 2007) also addressed the lack of diacritized text material by generating vowelized pronunciations using the Buckwalter Morphological Analyzer and the Arabic Treebank. Their results showed error reduction of 1.5% with vowelization. In a similar study offered by Al-Shareef and Hain (Al-Shareef and Hain, 2012), their results showed an improvement in the recognition performance by 1.8% in Levantine colloquial Arabic by training the acoustic module on automatically diacritized data.

In this thesis, two different diacritized datasets were used for our experiments. The first is manually labelled and diacritized by the Author. The KAPD and the CSLU2002 Arabic datasets were used for phoneme classification experiments. The second was fully diacritized Levantine data offered by the LDC and was used for the acoustic modelling experiments.

Mapping between the orthography and pronunciation (phonetization) is challenging work in many languages (El-Imam, 2004). Typically, a set of well-defined rules for phonetizing a written text is used to solve the pronunciation ambiguities in the text and to produce accurate phonetic transcriptions that can significantly improve phone recognition results. The correspondence (mapping issues) between graphemes and phonemes in Arabic is a relatively simple one to one mapping as compared to other languages such as French and English. Many researchers have addressed the impact of using the most common Arabic rules on the system performance. Fadi Biadsy (Biadsy et al., 2009) demonstrate that using Arabic pronunciation rules can achieve improvement on phone and word recognition for Modern Standard Arabic (MSA). They obtained a significant improvement in accuracy in ASR of 4.1% and they obtained a significant enhancement in phone recognition accuracy of 3.77%-7.29%.

Due to the lack of short vowels and constant doubling (shadda) in the transcription of the Arabic news, Billa et al. in (Billa et al., 2002b) used grapheme-based modelling. They only addressed the effect of applying the hamza/alef (glottal stop) rule on the recognition performance of Arabic broadcast news. They mapped the glottal stop into a single character and they gained an improvement of 2.6% in WER. They showed that a further improvement of 3.8% in WER (19.1% to 15.3%) can be achieved by removing the glottal stop from the transcription as it is not pronounced by native speakers. Also Vergyri et al. addressed the impact of using a pronunciation dictionary on the system's performance (Vergyri et al., 2008). They verified that pronunciation rules lead to small improvement in performance; only 1.1% improvement in WER. This result was reported for both BN and BC speech in various GALE datasets.

Most of the common rules used by (Biadsy et al., 2009, Vergyri et al., 2008) for MSA, are also used by Al-Shareef and Hain in (Al-Shareef and Hain, 2011) and Vergyri et al. (Vergyri et al., 2005) for colloquial Arabic (CA) on Levantine telephone conversations. Al-Shareef and Hain evaluated the effect of each rule independently on ASR performance by generating a multiple dictionaries rule-based. Also, a final dictionary was generated to assess the effect of all rules on the recognition performance when they are applied jointly. They found that none of these rule-based dictionaries outperform the baseline grapheme dictionary. Indeed the impact of applying all rules hindered rather than helping increasing the WER by 5.1% (59.3% to 64.4%).

The approaches for the transcription of written text into sounds are various in the literature. A part from the differences mentioned above, most of the rules used in this thesis are derived from (Biadsy et al., 2009). For more details see Section 5.3.3.

The Levantine Arabic dataset is fully diacritized and the statistics of 13 hours of speech dataset randomly selected from this Levantine Arabic dataset showed that vowels are the most frequent phonemes in Arabic language where it represents about 52.35% of the total number of phonemes in the Levantine dataset (more details about these statistics see Chapter Three Section 3.3). Removing the diacritics may lead to difficulties for both acoustic and language modeling, also, can affect the accuracy of the system performance. Therefore, diacritics should not be ignored from the Levantine Arabic dataset. This encouraged using the phonemic-based rather than the graphemic-based system for Arabic speech recognition.

Chapter Three

Arabic linguistic background and proposed Arabic corpora

The Arabic language has two variant forms; Modern Standard Arabic (MSA) is considered the formal written language of the culture, media and education throughout the Arab world. The morphological, syntactical, and phonologically of MSA are based on Classical Arabic. MSA, typically, is not a native spoken language of any Arab. The Arabic dialects are the true native spoken language forms and are not written at all. The Arabic dialects basically originate from historical dealings between languages of the contemporaneous cultures and Classical Arabic. For example, Algerian and Marco Arabic dialects have been influenced by French as well as Berber[2]. Significantly, Morphology, phonology, syntax and lexical choice of the Arabic dialects differ from MSA.

What follows is a limited overview of the linguistic concepts of Arabic, for more details see (Habash, 2010, Biadsy, 2011, Elmahdy et al., 2012a).

3.1 Arabic language: Orthography and Pronunciation

MSA is written in Arabic script using an alphabet accented by optional diacritics. The Arabic language has 34 phonemes including 28 consonants, 3 long and 3 short vowels, and can transcribed using 36 basic letters (graphemes) accented with 9. In the most cases the Arabic letters are mapped to Arabic phonemes using a one-to-one conversion; though, there are some exceptions.

[2] The **Amazigh** languages or dialects.

3.1.1 Hamza and morpho-phonemic spelling

- **The Hamza** /ء /: The glottal stop Hamza /</ has various forms in Arabic writing: *alif* with *madda /|,آ/, hamza* above *alif* />,أ /, and *hamza* bellow *alif* /<,إ/ *wa* /&,ؤ /, and *ya* /},ئ /, *hamza* /',ء/. Primarily, Hamza spelling depends upon its vocalic context. Also, Arabic writers very often substitute "Hamzated" consonant with "unhamzated" form i.e. (آ,أ , إ → ا), (ؤ →و) and (ئ →ى). In addition, Arabic speakers in most of the cases do not pronounce the Hamza forms. However, all the forms of Hamza can be pronounced as a glottal stop.

- ***Hamzat-Wasl*** /A, ا/: it appears at the beginning of the words. If the word initial Hamzat-Wasl and appears at the beginning of the sentence/utterance, the Hamzat-Wasl is pronounced as a glottal stop; however, if the sentence medial Hamzat-Wasl, it is silent. For instance, /Ainkataba kitAbN/ is pronounced like /<inkataba kitAbun/, on the other hand, /kitAbN Ainkataba/ is pronounced as /kitAbun inkataba/.

- ***Definite Article*** /Al, ال/: the Arabic definite article comes only at the beginning of nouns and there are two pronunciation cases for the definite article in Arabic language. It is pronounced as /Al/ if the followed letter is one of the Moon letters, and it is silent if it is followed by one of the Sun letters3: For instance, the word /Alqamar/ means 'the moon' is pronounced as /Alqamar/, whereas word /Al$ams/ means `the sun' is pronounced as /a$$ams/.

- ***Ta-marboutah*** /p, ة/: it is a word finally and it is also called a feminine ending. It is pronounced as sound /t/ when followed by an optional diacritic or sound /h/ when it appears at the end of the sentence; otherwise it is not pronounced. For instance, /madrasapN/ `a school' is pronounced / madrasatun/, also is pronounced as /madrasah/ or /madrasa/.

- ***Silent Letters*** /A, ا/, /w, و/, /Y, ى/: silent Alif in plural conjugation in verbs appears in the morpheme /uwA/, for example, ward /yajnuwA/ is prounounced as /yajnuw/ and the alif is silent here. Also, /w/ comes silent in some irregular spelling such that in the name /Eamrw/ is pronounced as /Eamr/ where the final w here is silent.

[3] The Sun Letters:/t,ت/, /v,ث/, /d,د/, /*,ذ/, /r,ر/, /z,ز/, /s,س/, /$,ش/, /S,ص/, /D,ض/, /T,ط/, /Z,ظ/, /l,ل/, and /n,ن/.

Finaly, *Alif-Maqsura* (Y, ى) is always appears at the end of a word and preceded by the short vowel /a/. It is always silent. For example, /mashaY/ `has gone' is pronounced /masha/.

- ***Foreign phonemes*** /ق,g/ Egyptian geem, the English letters /ڤ,v/, /P/ : in the MSA, these phonemes may possibly not be treated as additional sounds. They can be mapped to the closest pronunciation sound in the Arabic language. For instance, phonemes /ف, f/ and /ڤ,v/ may be treated as the one phoneme. The same manner might be applied on /ب,b/ and the English sound /p/. The main reason for mapping these phonemes using that manner is that foreign phonemes are not often used in MSA.

3.1.2 Optional Diacritics

- ***Short vowels*** (*fatha* /َa/, *kasra* /ِi/, *damma* /ُu/ and *sukoun* /ْ/): The *sukoun* mark indicates that the consonant is not followed by a short vowel, so there is no mapping for this mark.
- ***Shadda diacritic*** */~/*: doubles the previous consonant, for example word */kat~aAb/* (means the person who is writing) is pronounced like /*kattaAb*/.
- ***Three nunation diacritics*** (F /ًan/, N /ٌun/, K /ٍin/) are a combination of a short vowel and the sound /n/.

Arabic diacritics can only follow a consonant. These diacritics narrow the gap between the spoken and written words in Arabic. However, Arabic text is not commonly discretised with these marks.

3.2 Arabic transcription systems

Phonemic and graphemic systems are the most common systems used to transcribe of Arabic. In a graphemic system, one-to-one letter to sound rules are used for each word to generate a phonemic Arabic dictionary. For example, the Arabic word “ktab-كتاب” means “book” in English, is written in the dictionary as:

ktAb[4]	/k/ /t/ /A/ /b/

This scheme generates 36 graphemes (28 consonants, 4 different *alif: alif, alif* with *madda, hamza* above *alif* and *hamza* bellow *alif,* variants of *wa* and *ya* (*hamza* above), *hamza* and *ta-marboutah*). The simplicity of this scheme allows the generation of an Arabic dictionary by automatically splitting a word into letters. In the graphemic scheme, the short vowels (/a/ *fatha*, /i/ *kasra* and /u/ *damma*), diacritics (*sukoun*, *shadda*), and word final *nun* /n/ are not used as Arabic text is not commonly marked with these symbols.

An alternative system is to use a phonemic scheme. In this approach, the pronunciation of each word includes the short vowels and *nun* moves. Thus, the same word "book" becomes in the dictionary as:

KtAb	/k/ /i/ /t/ /a/ /A/ /b/.

In the phonemic system the variants of *alef*, *wa* and *ya* were symbolized to their simple forms. Also, the Egypt G (the /J/ is pronounced /G/) phoneme is added to the Arabic phonemes list in this work. Thus, a total of 33 phonemes are used to build the Arabic phonemic lexicon.

Arabic phonemes can be categorised into consonants and vowels, and the consonants can be further categorised into four classes as described in Figure 3.1 (Mosa and Ali, 2009).

[4] The Arabic text transcribed using the Buckwalter Arabic transliteration Romanization scheme in this thesis. Note, it is written from left to right (see the appendix A).

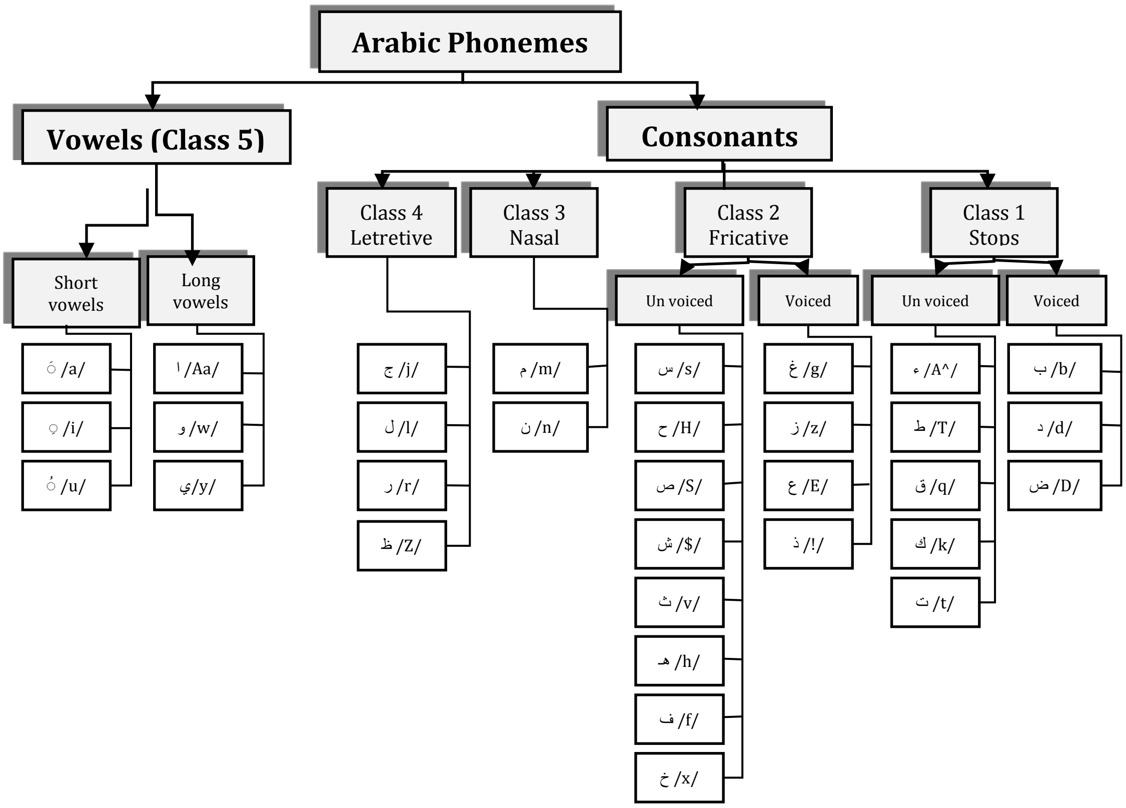


Figure 3.1 - Arabic Phoneme categorisation.

3.3 The frequency of Arabic phonemes

Many studies have been conducted for addressing and analysing the Arabic language for speech recognition and speech synthesis (Nahar et al., 2012, Al-Radaideh and Masri, 2011). Providing valuable information about the phoneme length and frequency can be helpful to enhance the design of appropriate speech recognition systems. Phoneme's length is a variant within the given language and can vary from one language to another. For example, glottal stops in the Arabic language are very short compared with the Fricatives such as /s,س /, /$,ش /, /f,ف / phonemes. Phoneme length is important for deciding the length of frames (frame size). In other words, the maximum frame size is limited by the size of the smallest phoneme in the dataset, i.e. the frame size should be less than the smallest phoneme in the dataset. For example , in Arabic language the size of most glottal stop phonemes /'ء, ^', 'ب, b', 'د, d', 'ت, t', 'ض, D', 'ط, T', 'ق, q', 'ك, k'/ was found to be very short (between 7 ms and 10 ms), thus dictating the selection of frame size to be 10 ms, in order to cover all phonemes . This is in agreement the frame size used in state-of-the-art ASR systems.

Studying the phonemes frequency can also give an indication as to whether a graphemic or phonemic system is useful for speech recognition.

The frequency order of the Arabic phonemes from the speech of the 34 CSLU2002 speakers (see Figure 3.2) is as follows from left to right:

(/a, vowel a/,/i, vowel i /, /u,vowel u/, /A,ا /, /b,ب /, /y,ي /, /t,ت /, /k,ك /, /w,و /, /m,م /, /l,ل/, /d,د /, /H,ح /, /f,ف /, /E,ع /, /s,س /, /n,ن /, /r,ر /, /h,ه /, /T,ط /, />,ء /, /$,ش /, /j,ج /, /x,خ /, /z,ز /, /q,ق /, /S,ص /, /g,غ /, /&,ذ /,/v,ث /, /D,ض /, /Z,ظ /)

This order is very close to the order of Al-Radaideh and Masri's study (Al-Radaideh and Masri, 2011).

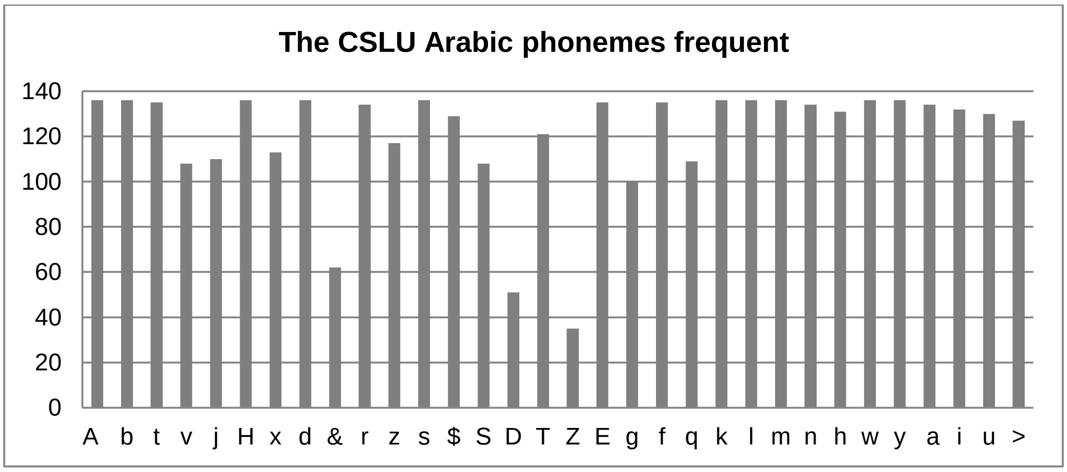


Figure 3.2 - The frequency of Arabic phonemes based for the 17 female and 17 male speakers from the CSLU2002 database.

Phonemes (/a, vowel a/,/i, vowel i /, /u,vowel u/, /A,ا /, /b,ب /, /y,ي /, /t,ت /, /k,ك /, /w,و /, /m,م /, /l,ل/, /d,د /, /H,ح /, /f,ف /, /E,ع /, /s,س /, /n,ن /, /r,ر /, /h,ه /) are the most frequent phonemes in the CSLU2002 dataset, therefore, they are easiest to find and to segment. Conversely, phonemes (/T,ط /, />,ء /, /$,ش /, /j,ج /, /x,خ /, /z,ز /, /q,ق /, /S,ص /, /g,غ /, /&,ذ /,/v,ث /, /D,ض /, /Z,ظ /) are not easy to find because they are less frequent than the other phonemes in this work's study. Also, some phonemes, such as (/T,ط /, /D,ض /, /q,ق /, /Z,ظ /) are completely missed in some of the Arabic dialects i.e. Egyptian and Lebanese in the CSLU2002 dataset.

The statistics of 105295 phonemes that were derived from the Levantine dataset using the HTK Toolkit, tend to agree with the CSLU2002 study. These statistics shows the frequency order of the Arabic phonemes is as follows:

(/a, vowel a/,/i, vowel i /, /A,ا /, /y,ي /, /l,ل/, /w,و /, /t,ت /, /n,ن /, /u,vowel u/, /b, ب /, /m,م /, /r,ر /, /E,ع /, />,ء /, /k,ك /, /d,د /, /h,ه /, /s,س /, /$,ش /, /f,ف /, /H,ح /, /j,ج /, /T,ط /, /x,خ /, /S,ص /, /z,ز /, /D,د /, /g,غ /, /&,ذ /,/q,ق /,/v,ث /,/Z,ظ /)

As shown from the Figure 3.3 the phonemes (/a, vowel a/,/i, vowel i /, /A,ا /, /y,ي /, /l,ل/, /w,و /, /t,ت /, /n,ن /, /u,vowel u/, /b,ب /, /m,م /, /r,ر /, /E,ع /, />,ء /, /k,ك /, /d, د/, /h,ه /, /s,س /, /$,ش /, /f,ف /, /H,ح /, /j,ج /, /T,ط /, /x,خ /, /S,ص /) are the most frequent. However, phonemes (/z,ز /, /D,د /, /g,غ /, /&,ذ /,/q,ق /,/v,ث /,/Z,ظ /) are rarely found in this dataset and represent no more than 3.87% of the total number of phones in the dataset. As is clear from the Figure 3.3, vowels are the most frequent phonemes in Arabic language, representing about 52.35% of the total number of phonemes in the Levantine dataset. This indicates that using the phonemic system for Arabic speech recognition is more reasonable than using a graphemic system.

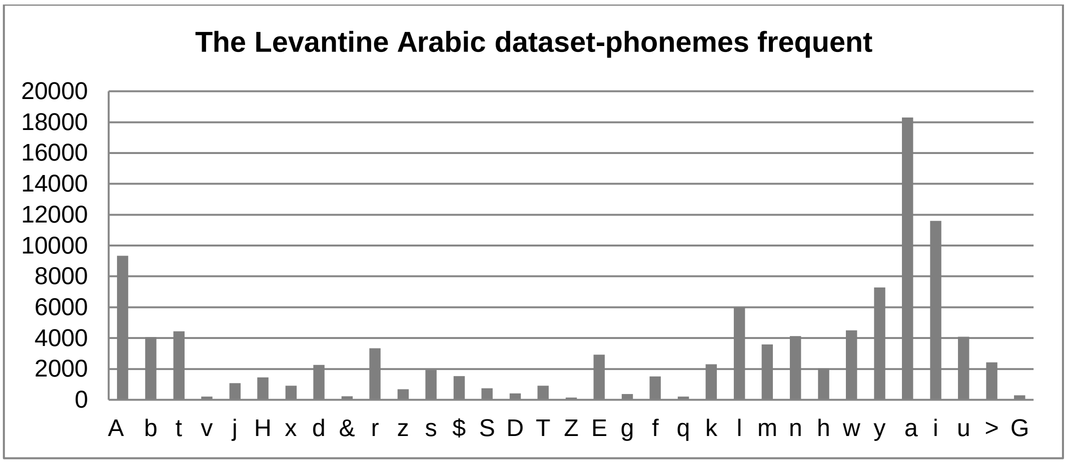


Figure 3.3 - The frequency of Arabic phonemes extracted from the Levantine database.

3.4 The Proposed Arabic databases

Speech performance is sensitive to the transmission channel used to transmit the speech, due to the changes produced in the speech spectral characteristics by the transmission type and background noise (Rabiner and Juang, 1993). Speech can be recorded over the different channels with a variety of qualities. Telephone line and microphone channels have been widely used for capturing the speech signal in the literature with a range of sampling frequencies from the 8 kHz used for telephone line channels to the 32 kHz used, for instance, in High-quality digital wireless microphones. In this research, two different sampling frequencies are used: 10 kHz for the KAPD database, recorded over microphone line, and 8 kHz for the CSLU2002 and Levantine databases, recorded over telephone line.

The performance of a speech recognition system is also very sensitive to the amount of available training data for each reference pattern (Rabiner and Juang, 1993). In the literature review, there have been several speech databases used to train and evaluate the speech recognition systems across the various languages. The key factor for evaluating the efficiency of any speech recognition system is to use standard speech databases (Tashan, 2012). In this Section, three Arabic databases are presented and used to train and evaluate the proposed Arabic speech recognition systems.

3.4.1 The King Abdul-Aziz Arabic Phonemics Database (KAPD)

The KAPD was recorded at the King Abdul-Aziz City for Science and Technology (KACST) in Saudi Arabia. This database consists of only 7 male speakers; where each speaker has recorded the 340 semi-Arabic words that were artificially created to construct the KAPD Saudi Arabia dialect database. All 340 words were recorded 7 times in the same environment. The speech was recorded using a microphone at a sampling frequency of 10 kHz and 16 bits.

A total of 4346 pure Arabic phoneme samples were manually extracted in the frequency domain from the seven male speakers in the KAPD database to represent the 33 different Arabic phonemes (26 consonant plus 3 short vowels /a/, /i/, /u/ and 3 long vowels /A/, /w/, /y/). Also, in this project, the silence between phones and words is represented as a phoneme. A total of 4346 phonemes were therefore segmented and labelled to build the KAPD Arabic phoneme database.

Figure 3.4 shows an example of the boundaries of the phonemes of the word /إرْتِفَاعَاً/ in both the time and frequency domains.

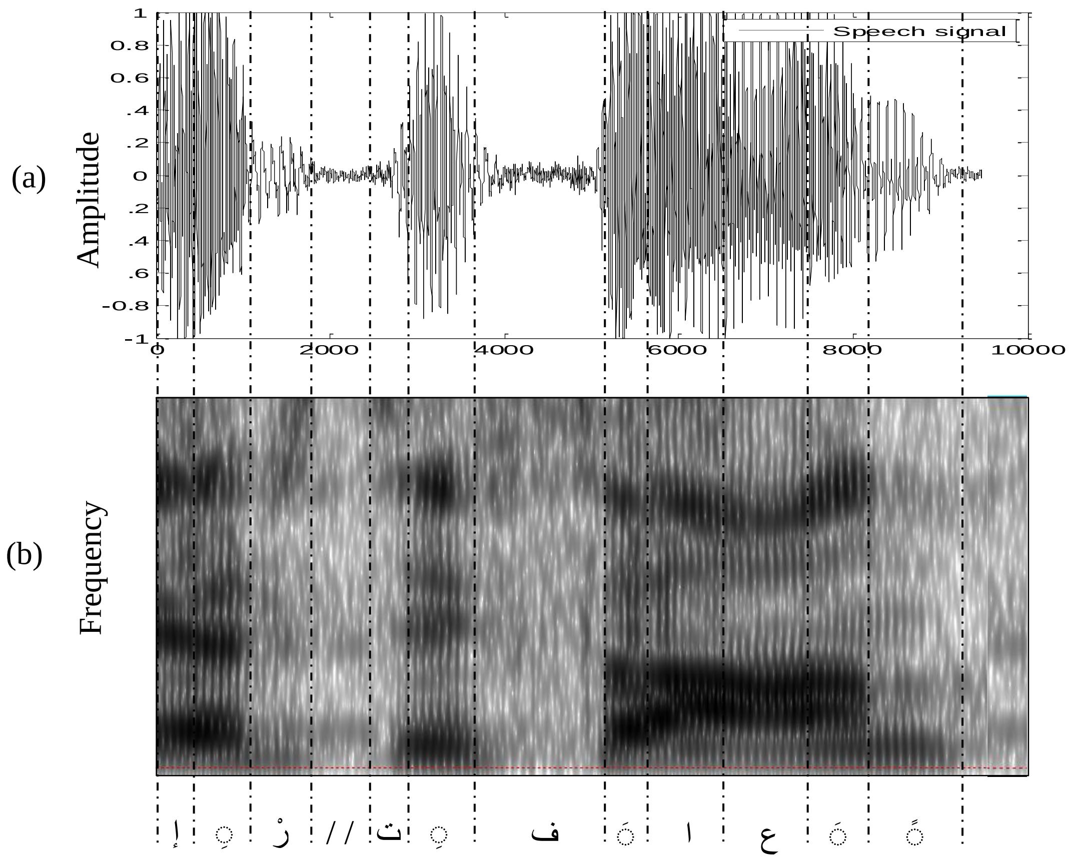


Figure 3.4- Phoneme classification for the utterance of the word /إرْتِفَاعاً/. (a) The word in time domain (spectrograph). (b) The word in frequency domain (spectrogram).

3.4.2 The Centre for Spoken Language Understanding (CSLU2002) database

The CSLU2002 database consists of 22 datasets of telephone speech for 22 languages. Eastern Arabic is one of these languages. The Arabic corpus contains fluent continuous speech of 98 native callers (5 Children, 23 females and 70 males) from more than 12 Arabic countries. Each speaker in this corpus talks about fixed topics (e.g. address, time home town, weather etc.). A total of 1997 sentence files were collected over telephone lines in different environments with an 8kHz 8-bit sampling rate. 34 speakers (17 females and 17 males) were selected arbitrarily from this Arabic dataset to evaluate the Arabic phoneme recognition systems developed in this work.

A total of 3802 phonemes were manually segmented and labelled from the CSLU2002 database in the frequency domain to represent the 33 Arabic phonemes and to build the CSLU2002 Arabic phonemes database.

3.4.3 The Levantine colloquial Arabic (LCA) database

The Levantine colloquial Arabic (LCA) corpora, called Appen corpora (Appen, 2007), is distributed by the Linguistic Data Consortium (LDC). This corpus makes available about 41 hours of Arabic speech with fully-discretised orthographic transcriptions using MSA scripts. The dataset consist of a collection of Levantine dialects including Syria (SYR), Jordan (JOR), Lebanon (LEB) and Palestine (PAL) dialects. This database contains 982 Levantine Arabic speakers taking part in spontaneous conversational telephone speech (CTS) in Colloquial Levantine Arabic. The average duration per side is between 5 and 6 minutes. This corpus contains 478 and 12 two-sided conversation for the train2c and devtest datasets respectively. The devtest dataset represents a reasonably balanced sample selected from the total collected calls. This dataset makes available 2.24 hours of Arabic speech for the same dialects. All audio files are 8 kHz, 8-bit.

A dataset of 120 Levantine Arabic female and male speakers (60 conversational telephone speech (CTS) two-channel recordings) was randomly selected per dialect to use as training dataset in our experiments. This training dataset consist of 13 hours of speech covering the four dialects. The whole devtest dataset is used as the testing dataset in our experiments. Table 3.1 illustrates these major dialects and the distribution of gender.

Table 3.1 - Amount of training and testing data (hours) for the major Levantine dialects and speaker gender.

Dialect	Training dataset			Dialect	Testing dataset		
	Size (hours)	Gender			Size (hours)	Gender	
		M	F			M	F
SYR	6	4.48	1.12	**SYR**	0.36	0	0.36
PAL	2.24	1.24	1	**PAL**	0.36	0.18	0.18
JOR	2.24	0.48	1.36	**JOR**	0.36	0.24	0.12
LEB	2.12	1.24	0.48	**LEB**	0.36	0.30	0.06
Total	13	8.24	4.36	**Total**	2.24	1.12	1.12

A word-list of 10k words was used for phonemic system, and about 284k words of training data were used. The acoustic model was trained on 6298 utterances (13 hours) of acoustic data.

3.4.4 Challenges of The Proposed Arabic Databases

The KAPD database is very limited as it consists of only 7 male speakers. In addition, one of the speakers' dataset is very noisy, meaning that it was excluded from the experiments. This dataset is, however, used here as it represents one of the few standard microphone based Arabic speech databases available.

Both of the CSLU2002 and Levantine colloquial Arabic (LCA) databases are recorded using a telephone channel. As a result, the quality in general is low and is dependent on the speakers' environments, with several of the utterances containing street and home noise. Moreover, both of the databases are unbalanced speech. A number of speakers were talkative, producing many phones. Some of recorded speech, however, contains very limited speech, producing recorded utterances that contained few or even zero examples of some phones.

An additional problem is that speakers were from 22 different Arabic countries; consequently, they have different dialectics. Several of these dialectics omit some phones. For example, the Egyptian dialectic misses phones /q,ق/, /j,ج/, /!,ذ/, /Z,ظ/ whilst the Lebanese dialectic misses /!,ذ/ and /Z,ظ/. Furthermore, phoneme pronunciation is completely different from one dialectic to another. For example, phoneme /!,ذ/ in the Egyptian dialectic is pronounced as /z,ز/ - a standard Arabic sound. Likewise, the phoneme /q,ق/ in the Yamane dialectic has a sound that has no similar in standard Arabic sounds. Finally, different speakers had different speed of speech during recording.

Dialectal Arabic Speech Recognition is very challenging. Preferably, a large speech dataset should be collected to train an acoustic model for each Arabic dialect. Unfortunately, collecting dialect speech data is very difficult in comparison to MSA. The difficulties are primarily due to the inability to record high quality speech using microphone for a huge number of speakers. In addition, a pronunciation dictionary for each Arabic dialect is not available and creating the set of dictionaries would be a challenge. Dialects in Arabic are only spoken and not written. There is no accepted standard for Arabic dialect orthography. Moreover, Arabic dialects are written without diacritics, so it hard to estimate the short vowels. Finally, automatic discretization systems are mainly not designed for Arabic dialects.

Chapter Four

Front-end speech process (Speech Segmentation and Feature Extraction)

It is not possible to perform speech recognition on the speech signal directly. Hence, the front-end speech process is concerned with the modelling of perception. The aim of this process is to produce features that are optimally used for the speech recognition task and robust to variant factors such as speaker variability, microphones and noise environment. The front-end processing phase can be divided into the segmentation and feature extraction stages (Duda et al., 2001).

Speech segmentation is a real challenge especially in noisy environments, and there are a number of different algorithms that can be used to automatically segment a speech signal. Selecting a feature vector type is also significant for the speech recognition task and a variety of features have been widely used in the literature. In this chapter, the results obtained using Entropy-based algorithm for automatic speech segmentation technique is presented in Section 4.1. The proposed Mel-Frequency Cepstrum Coefficients (MFCCs) feature extraction technique is then described in Section 4.2.

4.1 Automatic Speech Segmentation

Having selected a speech database, the first major challenge in designing a speech recognition system is which speech activity detection algorithm to apply on the recorded speech databases. Speech segmentation techniques are used to divide the speech signal into recognizable chunks. This involves segmentations at several levels (sentence, word, phoneme, and pause or silence level) (Anwar et al., 2006). Speech segmentation systems can be used in a number of applications such as speaker identification, speech recognition, speech documentation and speech translators.

There are several techniques for automatically segmenting the speech signal into speech regions (syllables and phonemes), and non-speech regions (silence, noise etc.). Power Spectral Density (PSD), Zero Crossing Rate (ZCR), and Entropy-based speech segmentation algorithms are widely applied for automatic syllable segmentation. Phoneme boundaries detection is a greater challenge due to the difficulty of detecting the beginning and ending of phonemes, where there is no specific point to separate phonemes. Success in detecting phoneme boundaries leads directly to success in classification of phonemes into categories such as consonants and vowels. A successful phoneme boundary detection technique is also significant for phoneme recognition.

In this work, the Entropic contrast algorithm (Waheed et al., 2002) was initially proposed for this function, and details of this algorithm are described in Chapter Four. However, since the performance of this algorithm is shown to have a percentage of error, and this error affects the subsequent performance of the recognition stage, this step was subsequently excluded from the system and a silence phoneme was proposed to allow for segmentation-free phoneme detection, as described in Chapter Six.

4.1.1 Entropy-based algorithm speech segmentation

An entropy-based contrast function is proposed to distinguish between the speech signal and the background noise. This algorithm has been shown to have less sensitivity to the amplitude changes of the speech signal compared to the energy-based algorithms (Waheed et al., 2002).

The first step of this algorithm is a pre-processing stage. In this stage the incoming speech data is modified by applying a pre-emphasis filter to reduce the glottal pulses effects (Waheed et al., 2002). Then a low-pass filter is used to remove the low frequency noise of the background. In this stage, the speech is framed using 50% overlapped frames where the size of each frame is 25 ms of speech.

A histogram of N bins is constructed to determine the probability distribution of each individual frame. Normalization of the histogram is then done. The selection of N (number of bins) for the histogram is based on computational load and sensitivity. Generally, the range of N bins is between 50 and 100.

The computation of the entropy for each frame is then computed using Equation 4.1 (Waheed et al., 2002).

$$H = -\sum_{k=1}^{N} P_k \log P_k \tag{4.1}$$

It is assumed that there is entropy profile ξ for the whole speech data presented, where:

$$\xi = [H_1 \; H_2 \; \ldots \; H_m] \tag{4.2}$$

and m is the total frames in the speech data.

Waheed and et al. (2002) assumed that this entropy profile can be used to guess an appropriate threshold γ as follows:

$$\gamma = \frac{\max(\xi) - \min(\xi)}{2} + \mu \min(\xi); \qquad \mu > 0 \tag{4.3}$$

μ is chosen to make threshold value is a little higher than the mean entropy profile. Actually, this is not always true. The threshold which is taken from this formula will be greater than the mean of the entropy profile. Hence, there are some parts of data speech that will be classified as silence or noise when this threshold is applied on the speech data. Practically, to avoid eliminating these parts of speech data, which have low energy, it is useful to use the threshold formula:

$$\gamma = meanentropy - (meanentropy * \alpha) \tag{4.4}$$

where $0 < \alpha < 1$.

After a threshold has been selected, any data greater than the threshold is considered as speech and any data less than the threshold is considered as noise or silence. i.e.:

$$\xi = \begin{cases} \zeta_i & if \;\; \xi_i > \gamma \\ 0 & otherwise \end{cases} \quad i = 1, 2, \ldots m \tag{4.5}$$

Even with the improved threshold formula (Equation 4.4), the result of applying the threshold on the speech data will, in many cases, cause some valid speech data frame to be rejected as silence and some invalid speech data frame to be falsely reported as speech; it is therefore important to include post-processing criteria to eliminate these false results.

The first post processing criterion is to apply an intra-segment distance algorithm. This algorithm is used to remove very small silence (unvoiced) speech segments between syllables. These segments are then collected together to form words using a fragment size based on a speed of speech normalization algorithm. Because humans do not generate very small duration sounds, this process be applied if the fragments of speech (sub-segments) are smaller than a given minimum length of speech sample. The collection of small fragments were also prevented from becoming longer than the maximum word length of speech. i.e. if the length of i^{th} speech segment is λ_i:

$$\lambda_i = e_i - s_i \tag{4.6}$$

Where: s_i is the starting sample of the i^{th} frame.

e_i is the ending sample of the i^{th} frame.

Then the intra-segment distance algorithm can connect fragment λ_i with λ_{i-1} if the collection of $\lambda_i + \lambda_{i-1} + d_{i-1}$ is smaller than the maximum word length of speech, and λ_i is smaller than the minimum length of speech as determined by the length of shortest phoneme in any language (Al-manie et al., 2006) – see Figure 4.1.

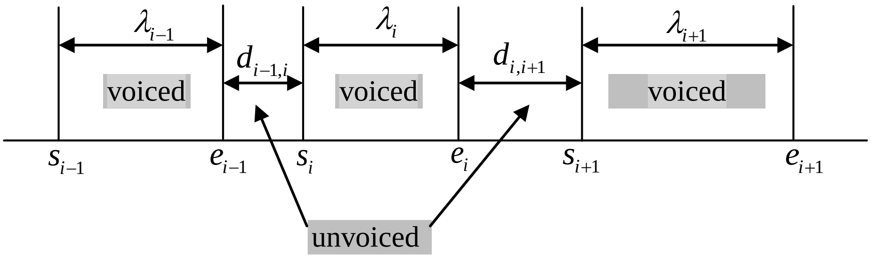


Figure 4.1 - Speech segments (Waheed et al., 2002).

In the Arabic language, the longest word is 'فأسقيناكموها', which is pronounced in English as 'faaskinakomoha'. This word contains twelve letters and seven syllables. However, this word is not that frequent, therefore, it is not sensible to use the length of this word as a maximum word length of Arabic speech, or to use it as a reference for the speed of speech normalization algorithm. The length of the most frequent Arabic words (Gammas et al., 2013), is between 2 to 8 phonemes. Thus, the maximum word length used in this work is 8 phonemes, and the minimum Arabic word used in this work 2 phonemes. Experimentally, the smallest phonemes were found to be stops phonemes ('ء, ^', 'ب, b', 'د,

d’, ‘ت, t’, ‘ض, D’, ‘ط, T’, ‘ق, q’, ‘ك, k’) with phoneme length between 7 ms and 10 ms. The longest Arabic phoneme were found to be fricatives phonemes (‘ث,v’, ‘ج,j’, ‘ح,H’, ‘خ,x’, ‘ذ,!’, ‘ز,z’, ‘س,s’, ‘ش,$’, ‘ص,S’, ‘غ,g’, ‘ف,f’, ‘ه,h’, ‘ع,E’) with phoneme length up to 24 ms.

The second post processing criterion was to remove very small silence periods based on the average silence length. This criterion is based on the length of any given silence period and the length of the neighbouring sub-segments of speech of this silence period. For example, in Figure 4.1, if $d_{i-1,i}$ is very small then the collection of the two neighbour sub-segments λ_{i-1}, and λ_i, plus $d_{i-1,i}$ will be compared to the maximum word length.

The final output of the Entropy-based algorithm is the detection of the syllables in the utterance. To achieve this successfully, it is important to normalize the speech based on the normalization of rate of speech (ROS).

4.1.2 Normalization of speech

One of the key factors in normalizing the speech is to normalize the Rate of Speech (ROS). This factor participates in the complexity of the mapping process between the phonetic categories and the acoustic signal (Benzeghiba et al., 2007).

According to Benzeghiba and et al. (2007) there is a strong relationship between the performance of speech recognition and speaking rate variation; fast speaking has higher error rate than low-rate speaking. However, speaking very slow introduce pauses (short silence) within syllables. As a result the performance of speech recognition systems degrades.

To normalize the speech, it is crucial to measure the rate of speech. Most ROS measure methods are based on the number of linguistic units (phonemes, syllables and words) in an utterance. Currently, there are two main types of ROS measurements: lexically-based and acoustically-based measures. The lexically-based technique counts the number of linguistic units (words and phonemes) per second (Benzeghiba et al., 2007). Siegler and Stern (1995) suggests that the most meaningful measure is phoneme rate; as compared to word rate which is the more common (Siegler and Stern, 1995). However, use of this technique is not practical in the speech recognition field because phonemes and words are the final outputs of the system and ROS is required in the early stages of processing. Therefore, an acoustically-based measure is proposed as an alternative technique that can directly estimate the ROS from the speech signal. Morgan and Fosler-Lussier have found that (1998) counting the syllables per second and dividing it by the segment length can be a sensible and reasonable measure of ROS.

The Entropy-based algorithm can be used to count the number of syllables forming an utterance, so, it is relatively easily to estimate the rate of speech directly from the utterance by counting the number of syllables per second; Equation 4.4 is used for normalizing the ROS for speakers depending on the threshold value α. ROS can have other impacts on the pronunciation that effect the speech recognition accuracy (Benzeghiba et al., 2007) (Morgan and Fosler-Lussier, 1998).

4.1.3 Experiment results

Results were obtained when the entropy-based algorithm was applied on 31 sentences for 9 speakers, one female and 8 male. The speakers used different rates of speech (ROS) and were from two databases: KAPD and CSLU corpora. The training sentences were chosen randomly from these Arabic databases. Several experiments were performed on the selected sentences using the Entropy based algorithm to examine the effectiveness of the selected number of bins (N) on the word segmentation performance. Moreover, to illustrate the effect of pre-emphasis and low pass filter factors on the results, four situations where tested:

1. Speech signal without any changes.
2. Speech signal with applying pre-emphasis only.
3. Speech signal with applying low pass filter only.
4. Speech signal with applying pre-emphasis and low pass filter.

A profile constructed manually for each sentence describes the real word boundaries of the proposed sentence. False accept rates (FAR) for any sentence was calculated automatically by counting false boundaries generated by the Entropy algorithm when compared with its manual profile. Similarly, false reject rates (FRR) were calculated by counting true boundaries rejected by the Entropy algorithm when compared with its manual profile.

Figure 4.2 shows the average false accept and false reject error rates for the different numbers of bins when the entropy-based algorithm was applied on the four situations.

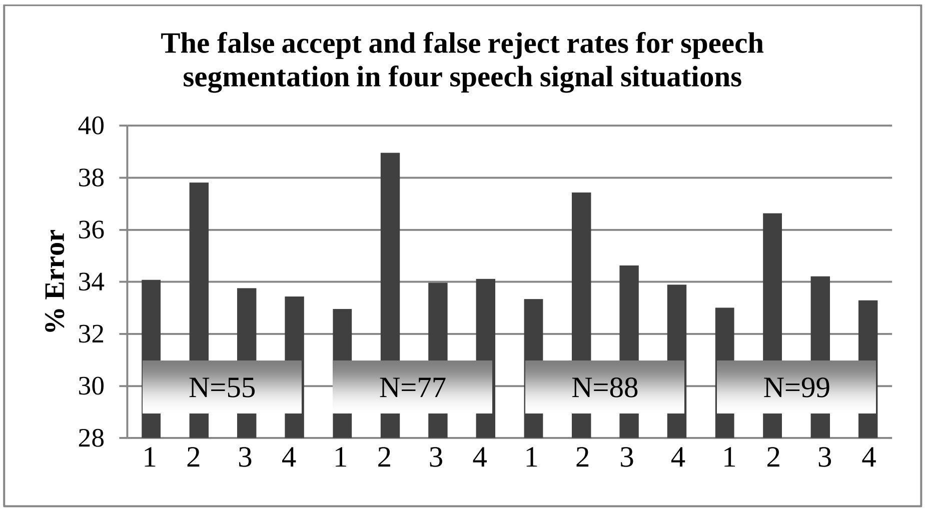


Figure 4.2 - The averages of false accept and false reject error in different bins (N), and different situations: 1. Original speech signal. 2. Pre-emphasis applied. 3. Low pass filter applied 4. Per-emphasis and low pass filter applied.

It is clear from Figure 4.2 that there is very limited effect on word segmentation performance with respect to the number of bins (N). In contrast, applying pre-emphasis to the signal has a detrimental effect on the results across all number of bins. The worst case of error (38.96%) is when N=77 with pre-emphasis only applied to the speech signal. The best word segmentation accuracy occurs for N=77 with the raw signal. The error obtained when N=77 is 32.95%.

The results presented in Fig 4.2, clearly show the difficulty in performing word segmentation on 'real-world' continuous speech using conventional word segmentation algorithms. One possible reason for this is the absence of any phonetic information within the segmentation algorithm. As a consequence, it was agreed that the emphasis of the following work should concentrate on performing segmentation-free recognition by including a silence phoneme with the recognition process.

4.2 Feature extraction process

The choice of feature extraction technique plays an important role in speech recognition systems. Features can be extracted from speech signal in the frequency or time domains. Analytically, however, the speech signal is more understandable in the frequency domain than in the time domain (Rabiner and Schafer, 1978). Thus, all of the techniques found in the literature (MFCCs, LPC, and PLP … etc.) extract features from the speech signal in the frequency domain.

The speech signal, usually captured by a microphone or telephone line, is digitised and sampled over time. The feature extraction process is designed to deliver a feature vector to any efficient classifier for categorization of the patterns. Mel frequency cepstrum coefficients (MFCC) (Davis and Mermelstein, 1980) is a popular speech feature representation. The most important property of the MFCC technique is that it is more robust in the presence of background noise than other feature parameters. Generally speaking, other feature vector formats contain more information than an MFCC vector, but this greatly increases the computational load in terms of feature vector extraction and the subsequent recognition stages. The MFCC technique offers a trade-off between the computational and performance requirements. The MFCC is a Discrete Cosine Transform (DCT) (Huang et al., 2001) of the logarithmic Mel-Spectrum; where, the Mel-Spectrum is the output of Mel Frequency filters.

Feature extraction is typically carried out from speech frames every 10ms with an overlapping analysis window[5] of 25 ms of speech., where, the waveform in this length is assumed to be stationary (Schafer and Rabiner, 1975) . The discrete Fourier transform (DFT) (Huang et al., 2001) is the state-of-the-art technique applied to compute the speech spectrum, where the peaks in the spectrum are denoted as “formants”. The Principal Component Analysis (PCA) has been investigated as a suitable transformation to compute the MFCCs instead of the DCT in many research (Vetter et al., 1999, Lee et al., 2001, Lima et al., 2004, Lima et al., 2005). However, in all our experiments the MFCCs were extracted using the DCT for computing the speech spectrum.

A sequence of triangular band-pass filters are applied for each frame to procedure a vector of filter bank coefficients. The number of filters is usually between 20-40 filters. The band-pass filters are spaced according to the linear frequency scale (*Mel* scale) as follows:

$$mel(f) = 2595 * \log_{10}(1 + {}^{f_{Hz}}/_{700})$$

[5] A *Hamming* window is commenly applied to each frame.

In fact, the human ear does not do a linearly react to the amplitude of the speech signal. So, the logarithm of filter output values was taken rather than using the output values in the previous equation. The features derived from this step are called the “MFSC” or the“log mel-frequency” spectral Coefficients. These coefficients were found to be highly correlated. Therefore, to reduce this correlation between the filter bank coefficients, a further transformation using the Discrete Cosine Transform (DCT) is applied on these coefficients. The DCT transform is given by

$$o_{ti}^{mfcc} = \sqrt{\frac{2}{B}\sum_{b=1}^{B} \log\left(o_{tb}^{fb}\right) \cos\left(\frac{i(b-0.5)\pi}{B}\right)}$$

Where $\mathrm{o}_{\mathrm{tb}}^{\mathrm{fb}}$denoted by the bth filter coefficient, and $\mathrm{o}_{\mathrm{ti}}^{\mathrm{mfcc}}$ is the *i*th Mel frequency cepstral coefficient.

Commonly, 12 MFCC coefficients are computed, along with the 0th order cepstral coefficient to produce a 13 MFCCs dimensional static feature. Dynamic features (Furui, 1986) are typically appended to enhance the recognition performance. The delta coefficients $\Delta \mathrm{o}_{\mathrm{t}}$ (also known as first order dynamic features) can be computed as follows:

$$\Delta o_t = \frac{\sum_{i=1}^{d} w_i * (o_{t+i} - o_{t-i})}{2\sum_{i=1}^{d} {w_i}^2}$$

Where w_{i} is the regression coefficient, and d is the regression window size. The above equation can be used to produce the delta-delta coefficients and so on. However, It is presented in Huang et al. (Huang et al., 2001) that a greater number of cepstral coefficients (more than the second-order) do not improve speech system accuracy.

These dynamic coefficients are added to the static coefficients to form the speech feature vectors. Typically, the state-of-the-art dimensional feature vector for speech recognition is 39. This dimensional feature vector is useful for different Neural Networks and for HMM systems. So, this dimensional feature vector is used for all the experiments conducted in this thesis.

Chapter Five

Automatic speech alignment system

The phoneme is considered the basic unit of speech in much of speech processing. The phonemic representation of a given word is used in most speech recognizers to identify it. Thus, the availability of speech data that is time-aligned and labelled at phonemic level is a fundamental requirement for building speech recognition systems.

Time-aligned phonemic labels may be manually transcribed by expert human labellers or automatically produced using automatic methods. There is common agreement that manual labelling is more precise than automatic labelling (Cosi. et al., 1991, Cox et al., 1998, Ljolje et al., 1994). However, although manually-aligned data is considered more accurate than automatically-aligned data (Cox et al., 1998, Hosom, 2000b), it is very time consuming to use manual methods for aligning large corpora. Automatic alignment methods are reported to consume 0.1 to 0.4 seconds to align one phoneme, whereas manual alignment method can require 11 to 30 seconds to align one phoneme (Kvale, 1994, Leung and Zue, 1984). Hence, there is a need for fast and accurate automatic alignment systems. However, manually-aligned data is needed to evaluate the accuracy of the automatic alignment systems (Hosom, 2000a).

5.1 Methods of automatic alignment systems

The most common method for automatic speech alignment is called "*forced-alignment.*" In the main, there is a link between automatic forced alignment and speech recognition systems, in that they can use the same processes for both tasks. Recognition of utterance is performed in some systems using a search procedure constrained to prior known sequences of phonemes. This search produces the locations and the identities of the phonemes. Used in this way, the forced alignment method can yield the phonetic alignment by forcing the recognition result to obtain the utterance phonetics locations and identities. The phonetic sequence is usually prior determined by a pronunciation dictionary. The most common method for forced alignment is to build phonetic recognizer based on a Hidden Markov Model (HMM) (Brugnara et al., 1993, Ljolje et al., 1997, Wightman and Talkin, 1997, Hosom, 2000a, Toledano et al., 2003, Yuan and Liberman, 2008, Yuan et al., 2013).

The next section will explain the mathematical formulation of the HMM model for forced alignment.

5.2 HMMs for forced alignment

As mentioned earlier, HMMs speech recognizer can be used to yield the phonemic alignments for a given utterance using the *forced alignment* process. In this process, the HMMs are projected to recognize the input speech using the Viterbi search constrained to the correct phonemics sequence. Consequently, the phonemic alignment is obtained with this Viterbi search. For given words, the phoneme sequence of each word is generated by phonemic dictionary. Then the sequences of these words are concatenated together to produce a phoneme sequence for the utterance. Forced alignment is useful in training to derive phoneme level transcriptions automatically.

We implemented the acoustic and language modeling of a HMM speech recognition system (HTK v3.4.1 (Woodland et al., 1995)) which is used in forced alignment mode. However, to setup the HTK toolkit, several issues must be prepared. For example, Arabic transcriptions for the utterances and the Arabic phonemic dictionary are needed. Also, in Arabic language, text normalization (mapping rules) should be addressed.

5.3 Automatic speech alignment experiment setup

5.3.1 Data

The experiment investigated the use of MFCC features and delta features in the Viterbi search. The Training was done on the Levantine colloquial Arabic (LCA) corpora, described in Chapter Three. This corpus makes available about 41 hours of Arabic speech with fully-discretised orthographic transcriptions using MSA scripts. The dataset, as described, consists of a collection of Levantine dialects including Syria (SYR), Jordan (JOR), Lebanon (LEB) and Palestine (PAL) dialects. A dataset of 120 Levantine Arabic female and male speakers (60 conversational telephone speech (CTS) two-channel recordings) randomly selected per dialect to use as training dataset on our experiments. This training dataset consist of 13 hours of speech covering the four dialects. The devtest dataset is selected to use as testing dataset on our experiments.

A word-list of 10k words was used for phonemic system, and about 284k words of training data were used. The Levantine Arabic acoustic model was trained on 6298 utterances (13 hours) of acoustic data.

5.3.2 Arabic Transcription

As mentioned early in Chapter Three, the most common Arabic transcription systems are the graphemic and phonemic systems. This thesis used the phonemic scheme for the reasons explained in Section 3.3. In this approach, the pronunciation of each word includes the short vowels and *nun* moves.

Most of the Morphological Analysers do not attempt to generate the complete phonetic dictionary covering all the words in the lexicon used in the experiments. For example, the Buckwalter Morphological Analyser (Buckwalter, 2002) generates about 75% of the lexicon words. Therefore, missed pronunciations must be added manually. In this work, more than 6k words were added using a script. A total of 59k words were subsequently created to produce the Arabic phonemic dictionary that used in the experiments of this thesis. In the phonemic system the variants of *alef*, *wa* and *ya* were symbolized to their simple forms. Thus, a total of 33 phonemes are used to build the Arabic phonemic lexicon.

5.3.3 Text Normalization

Mapping between the orthography and pronunciation (phonetization) is challenging work in many languages (El-Imam, 2004). Typically, a set of well-defined rules for phonetizing a written text is used to solve the pronunciation ambiguities in the text and to produce accurate phonetic transcriptions that can significantly improve phone recognition results as is demonstrated in (Billa et al., 2002b, Vergyri et al., 2008, Biadsy et al., 2009). However, Al-Shareef and Hain in (Al-Shareef and Hain, 2011) showed none of these rules help to improve the performance of Levantine colloquial Arabic and none outperform the raw grapheme dictionary performance.

The approaches for the transcription of written text into sounds are various in the literature. For instance, many forms of the glottal stop in the Arabic language /<,>,',&/, these forms of glottal stops mapped into a single character as performed by (Billa et al., 2002b, Biadsy et al., 2009, Vergyri et al., 2008). In this thesis, the glottal stop is also normalized to a single letter /A/ (see Table 5.1). Nunnations /AF,F,K,N/ are mapped as "n" sound preceded by a short vowel similar to (Vergyri et al., 2008, Al-Shareef and Hain, 2011, Biadsy et al., 2009). The Geminates (*shadda*) letter /~/ is mapped differently in the literature, for example, the word ba$~ara is mapped into ba$ara by (Biadsy et al., 2009). They ignored the *shadda* and dealt with the constant letter as normal. On the other hand, Vergyi et al. (Vergyri et al., 2008) doubled the consonant; the same word is mapped into ba$$ara. In this thesis, the *shadda* is doubled similar to (Vergyri et al., 2008) where it is close to its pronunciation. *Tah-marbutah* is either pronounced as "t" or "h" or is silent. This rule is applied by (Al-Shareef and Hain, 2011), this rule is also used in this thesis, whereas (Biadsy et al., 2009) and Vergyri (Vergyri et al., 2008) map *tah-marbota* into t and ignored other cases of t. Solar lam /l/ is removed if the definite article /Al/ is followed by a *sun letter* /t, v, *, d, r, z, s, S, $, T, D, Z, l, and n/. There is an agreement for mapping the Solar lam in the literature. Also, most of the references replaced *Alif maqswra* /Y/ as it is pronounced which is normal *alif* /A/. Apart from the differences mentioned above, most of the rules used in this thesis are derived from (Biadsy et al., 2009). A summary of these rules is shown in Table 5.1.

Table 5.1 Summary of rules used in this thesis compared with (Biadsy et al., 2009).

Rule	**Source**	**Target by Biadsy et al.(Biadsy et al., 2009)**	**Target used in this thesis**
Long Vowel (Dagger alif)	/‘/	/A/	/A/
Long Vowel *(Madda)*	/|/	/G A/	/< A/
Nunnation	/F/	/a n/	/a n/
	/K/	/i n/	/i n/
	/N/	/u n/	/u n/
Glottal Stop (Hamza)	/’}&< >/	/G/	/</
Tah-marbutah (word ending)	/p/	/t/	/t/ /A/ or sil.
Geminates (shadda)	/~/	/ /	/previous consonant is doubled/
Suffixe (Waw Al-Jama’a)	/uwoA/	/U/	/uw/
Hamzat-Wasl	/{/	/ /	/ /
Sun letters (t, v, *, d, r, z, s, S, $, T, D, Z, l, and n.)	/Al+ Sun letter/	/A+ Sun letters/	/A+ Sun letters/

These rules were basically designed for MSA, even so they can be used for mapping Arabic dialects. However, these rules are not always right for the all Arabic dialects. For example, in the Levantine corpus used in this work, the rule of glottal stop as it explained earlier is converted into single letter /A/ is not always true, for example, word “is burning” is written as “biyHri}” in the transcription of Levantine dataset. The transcription of this word is the same with the pronunciation of the Syrian and Lebanon dialects but it is pronounced as “biHriG” in the Jordan and Palestine dialects, where is ends with /G/ “*Egyptian Geem*”. Table 5.2 shows many of similar examples. Many of these words were handled manually.

Table 5.2 Shows examples of many words pronounced differently in the Levantine dialects.

Word in the Levantine corpus	**Meaning in English**	**Pronounced in**			
		Syrian	**Lebanese**	**Jordanian**	**Palestinian**
/suwriyap/	Syria	/suwriyay/	/suwriyay/	/suwriyap/	/suwriyap/
/Al>abiy~ip/	Proud	/Al>abiy~iy/	/Al>abiy~iy/	/Al>abiy~ip/	/Al>abiy~ip/
/ha*aA/	This	/hazaA/	/haydaA/	/haDaA/	/haA*/
/mavalan/	For example	/masalan/	/masalan/	/mavalan/	/mavalan/
/bi*aAt/	Exact	/bizaAt/	/bizaAt/	/bi*aAt/	/bi*aAt/

5.3.4 Building the Arabic Pronunciation Dictionary

A pronunciation dictionary maps words into one or more phonetic pronunciation variants. A single pronunciation dictionary that has one pronunciation per word or multi pronunciations per word is typically used for training and testing (decoding) in ASR systems. Results of experiments conducted on the effect of using the single or multi pronunciation dictionary in (Hain, 2001) showed that the word error rate (WER) is slightly lower when only a single pronunciation per word was used. Usually, most data available to train LVASR systems is not phonetically (orthographically) transcribed. Vowelized data, for instance, in Arabic transcription gives multiple pronunciations for each word. Hence, in our experiments, an Arabic dictionary with multiple pronunciations was built for training the acoustic model. There are two training types in ASR: firstly, *bootstrap* training; this training is used when certain data available is phonetically annotated. Secondly, *flat start* training is used when bootstrap data is not available (Young et al., 2006). The pronunciation dictionary is typically used, in the flat start training, for mapping the orthographic transcription of the utterance to sequences of phonetic labels. This flat start training is used to train the first monophoneme models. Then the dictionary is applied again to generate networks of probable pronunciations that may be used to find the most likely phoneme sequence using force alignment in order to match the acoustic data. The followed step is to re-estimate the monophoneme acoustic models.

In the pronunciation dictionary, every entry is a fully diacritized word which is mapped into a set of possible pronunciations. Usually, automatic morphological tagger such as MADA and Buckwalter (Buckwalter, 2002) are used for generating the most likely diacritic marks with some success.

In our work, the Levantine colloquial Arabic pronunciation dictionary is generated from the Arabic lexicon offered by the LDC. This contains 51730 words and is generated using the Buckwalter Morphological Analyser. However, for training, the Levantine acoustic data, 6708 words were manually added to generate a Levantine colloquial pronunciation dictionary that had all the Arabic words used in the input utterances.

5.3.5 Frame work

The system used in this thesis followed a standard recipe. Timings suggested by the corpus are used to segment the audio data, and 13 Mel Frequency Cepstral Coefficients (MFCCs) features plus their first and second derivatives were extracted using 10 ms using a windowed speech signal of 25 ms., each conversation side was normalized using cepstral mean and variance normalisation as well as vocal tract length normalisation (VTLN). It was shown in (Hain and Woodland, 1998), that VTLN is more effective if it is used per conversation side variance normalisation. The Maximum Likelihood function was used to train all the models with context-dependence triphones. Each phone was modelled using left-to-right HMM with three states. A binary decision tree (Young and Woodland, 1994) with phonologically phonetic questions were used to cluster the HMM states. A single Gaussian mixture component for each state with a total of 2083 clustered states was used. The HTK Hidden Markov Model toolkit was utilized to perform the automatic labelling using forced alignment mode. The labelled Arabic dataset thus produced to be used for further experiments in Chapters six and seven.

Chapter Six

Deep Neural Networks (DNNs) for phoneme classification and acoustic modelling

Feed forward deep learning models have proven to be very successful in training artificial neural networks (ANNs) and have resulted in a wide adoption of the ANNs technique in many fields, including speech recognition applications. The acoustic modelling of speech is considered as a key component of most state-of-the-art speech recognition systems. Research in this area has recently used Deep Neural Networks (DNNs) with high success for acoustic modelling of the English language (Mohamed et al., 2009, Dahl et al., 2012, Hinton et al., 2012, Mohamed et al., 2012). This success encouraged us to be the first to attempt to investigate this type of network for developing acoustic models for an Arabic speech recognition system. The aim of this Chapter is therefore to explore deep learning for phoneme classification and acoustic modelling using Deep Neural Networks.

In order to achieve this aim, we have explored different network architectures based on the Deep Neural network introduced in (Hinton et al., 2006). This chapter is organised as follows: the main concepts of the Shallow NNs supervised learning model and its use for Arabic framewise phoneme classification is first described in Section 6.1. Deep NNs models used in this work as Arabic acoustic modelling system are then described in Section 6.2. The systems' concepts with their related technical details such as the hyper-parameters optimisation process and the software methodologies used are then discussed. This is followed by experiments results using small vocabulary datasets (the KAPD and the CSLU2002 phoneme datasets) to train Deep Neural Networks (DNNs) with the Back-propagation supervised learning algorithm for framewise phoneme recognition. Then the classified frames are processed in a post processing stage to detect the phonemes. These datasets are used in order to address the behaviour and optimize the DNNs. Then, the Deep Neural Network (DNN) is used for large vocabulary speech recognition (the Levantine dataset) which we automatically labelled it using the HMMs using the HTK Toolkit as it was described in Chapter Five. This classification stage is also followed by a post processing stage to detect the acoustic sequences. A conclusion is finally drawn in the third Section.

6.1 Shallow Multilayer Perceptron (MLP)

Feed-forward Neural Networks (FNNs) are a widely used artificial neural network architecture. Multilayer Perceptron (MLP) NNs, trained using the Back-propagation (BP) algorithm, have been widely used for solving many patterns recognition problems. The network architecture of a MLP is illustrated in Figure 6.1. As shown, a MLP NN consists of several layers: an input layer, one or more hidden layers, and an output layer. Neurons in the input layer operate as buffers for distributing the input patterns to the neurons in the hidden layer. Also, biases ***b*** are connected with bias units. Each neuron influences the other's behaviour via weighted connections; where each neuron output is computed as the sum of the product of nonlinear weights with its inputs in the previous layer.

The output of each neuron is broadcasted to all other units over its outgoing connections. These outputs are calculated using equations of the following form:

$$h^{(i+1)} = f^{(i+1)}(W^{(i)}l^{(i)} + b^{(i)}) \tag{6.1}$$

where l is the hidden units, W is the weights, b is a bias with value 1 and f is the activation function where it is usually a non-linear function.

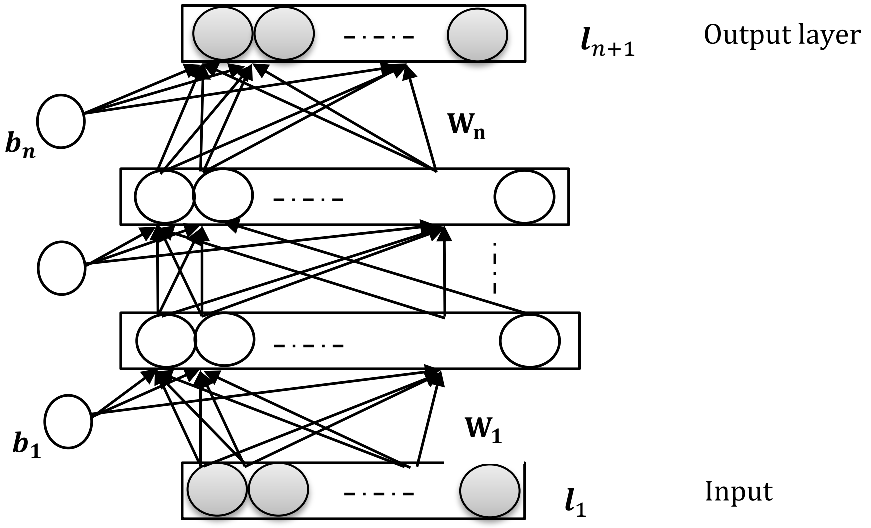


Figure 6.1 - Multilayer Perceptron (MLP).

The activation function is often the asymmetric sigmoidal, hyperbolic tangent or softmax function:

$$sigmoid(x) = \frac{1}{1 + e^{-\alpha x}} \quad , \quad 0 \leq f(x) \leq 1 \tag{6.2}$$

$$\mathrm{tanh(x)} = \frac{\mathrm{e}^{\alpha \mathrm{x}} - \mathrm{e}^{-\alpha \mathrm{x}}}{\mathrm{e}^{\alpha \mathrm{x}} + \mathrm{e}^{-\alpha \mathrm{x}}} \quad , \quad -1 \leq \mathrm{f(x)} \leq 1 \tag{6.3}$$

$$\mathrm{softmax(x)_i} = \frac{\mathrm{e}^{\mathrm{x_i}}}{\sum_{\mathrm{j=0}}^{\mathrm{O}} \mathrm{e}^{\mathrm{x_j}}} \quad , \quad 0 \leq \mathrm{f(x)} \leq 1 \tag{6.4}$$

where α is a slope control parameter on the sigmoid function and x is the network input. The most important advantage of these functions is their ability to be differentiable; as required when deriving the weight updating function in the back-propagation learning algorithm. Selecting the output layer activation function is task dependant. For example in binary classification the sigmoid is used as it is guaranteed to have [0, 1] values. The "*softmax*"[6] activation function is used for multi-way classification

[6] Is a generalization of the logistic function that "squashes" a O-dimensional vector x of arbitrary real values to a O-dimensional vector f(x) of real values in the range (0, 1) that add up to 1.

which learn to classify correctly O classes, where it is not applied on the activations of the whole output vector. Instead it is applied independently to each neuron in the layer. The softmax is differ to the other functions in that it has a statistical view, where its outputs are posterior probabilities for O classes

All Neural Network training algorithms are used to adjust the network weights in a network training phase. The Conjugate Gradient algorithm was used to adjust the weights in order to speed-up the learning phase. The adjustment is repeated over several training epochs until an adequately small value of error $E(t)$ is achieved or a given number of epochs are reached.

A C++ visual studio code has been written to implement the Conjunction Gradient algorithm to adjust the weights for the MLP NN (for more details see (Salimi et al., 2012)).

6.1.1 Shallow MLP Hyper-parameters Optimisation

Optimising the model's hyper-parameters is one of the major problems when developing a shallow MLP classifier. There is no agreed methodology to select the number of hidden layers or the number of nodes in each hidden layer. This is commonly performed in a problem-based fashion. Typically, a logistic sigmoid function is selected to be used for activating each node. The network's weights between [-1, 1] are randomly initialised. The Conjunction Gradient algorithm was used to adjust the weights using a suggested learning rate of 0.3, and momentum 0.9.

6.1.2 Shallow MLP Experiments and results

For the experiments conducted in this Section, a standard feature extraction technique was used. The 13 MFCCs and their first and second derivatives were extracted to create standard MFCCs feature vectors covering 10ms of speech, where each frame vector is normalized for DNNs experiments using Equation (6.5).

$$\text{Normalized Feature} = \frac{\text{feature} - \text{MinCoef}}{\text{MaxCoef} - \text{Mincoef}} \tag{6.5}$$

Where MaxCoef is the Maximum cepstral coefficient in the MFCCs feature vector and MinCoef is the Minimum cepstral coefficient in the MFCCs feature vector. In this thesis, all experiments conducted on the three different corpora were repeated at least five times to ensure a valid conclusion.

The main purpose of the experiment conducted in this Section was to compare the results of the shallow MLP networks that were trained with different dimension of MFCCs vector size. The networks were trained for a frame-based phoneme classification using the KAPD dataset. The first experiment used input vectors with the 13 MFCCs only. The second experiment used the 13 MFCCs plus their first and second derivatives. The results are presented in Table 6.1.

Table 6.1 - The Framewise Phoneme Recognition for shallow MLP models trained with different dimension of MFCCs feature. The 13 MFCCs is published in ((Hmad and Allen, 2012)).

MFCCs	MFCCs + ΔMFCCs + Δ^2MFCCs
47.36%	**47.71%**

The performance of the system with 13 MFCCs plus their first and second derivatives feature vectors was about 0.35% better than the base 13 MFCCs. A two-tailed Matched Pairs T-test significance test using Matlab was conducted with the null-hypothesis that there is no performance difference between the shallow MLP models trained with two different MFCCs feature vector sizes. Since the P-value (0.5175) is greater than the significance level (0.05), the test cannot reject the null hypothesis. However, the 13 MFCCs plus their first and second derivatives feature vectors are adopted over all experiments conducted in the remainder of this thesis unless otherwise indicated.

6.2 Deep learning for acoustic modelling

Until recently, Feed-forward neural networks have had limited success when compared to GMMs, in applied ASR systems (Bengio, 1991, Bourlard and Morgan, 1993, Morgan et al., 2005). However, there has recently been an upsurge in interest using Deep Neural Networks for speech recognition systems. The significant gains over the state-of-the-art GMM/HMM speech recognition systems seen on a variety of small and large vocabulary speech, as presented in the literature reviews, motivated our use of Deep Neural Networks (DNNs) for two small vocabulary phoneme classification tasks and for modelling the acoustic signals using a large vocabulary Arabic dataset. Weights in the DNN are generatively pre-trained to maximize the data likelihood before a fine tuning phase using the back-propagation algorithm. Then in the second phase, a deep network is built by adding hidden layers with many hidden units in each hidden layer. This structure of the state-of-the-art DNN (Hinton et al., 2006) is used in this chapter to model the acoustic signals. In the following sections, the structure of the DNN is first described and then evaluated on the proposed datasets for phoneme recognition and acoustic modelling tasks.

6.2.1 Deep Neural Networks Basics

DNNs are feed-forward Neural Networks that have many layers of non-linear hidden units between their inputs and their outputs. This type of network is first trained one layer at a time as a generative model of the input data using Restricted Boltzmann Machines (RBMs) (Smolensky, 1986) to create the deep neural network. A subsequent discriminative fine-tuning phase using the backpropagation algorithm, then fine-tunes the features in every single layer to make them more valuable for discrimination.

6.2.1.1 Restricted Boltzmann Machines (RBMs) and their Generalizations

In RBMs, all visible units that represent observations are connected to all binary, stochastic hidden units. RBMs are restricted where there are no connections between visible nodes or between hidden nodes (see Figure 6.2-(a)). The hidden and visible units in the simple RBM are binary and stochastic. However, in a Gaussian-Bernoulli RBM (GRBM) (Freund and Haussler, 1994., Hinton and Salakhutdinov, 2006), the hidden units are binary and the visible units are linear. Gaussian RBMs have been used in many applications for modeling real-valued inputs such as speech recognition, image classification, and video action recognition (Lee et al., 2009, Taylor et al., 2010, Mohamed et al., 2012). We will first explain the RBM.

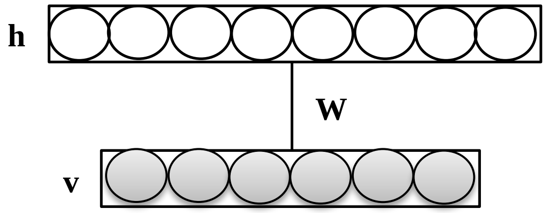


Figure 6.2 – (a) The RBM structure from (Mohamed et al., 2009)).

6.2.1.1.1 Binary Restricted Boltzmann Machines

In the binary RBMs model, the stochastic visible variables are $\mathrm{v} \in \{0,1\}^{\mathcal{V}}$ and the stochastic hidden variables $\mathrm{h} \in \{0,1\}^{\mathcal{H}}$. The model defines the energy function $\mathrm{E}\colon \{0,1\}^{\mathcal{V}} \mathrm{x} \{0,1\}^{\mathcal{H}} \rightarrow \mathrm{R}$

$$\mathrm{E(v,h|\theta)} = -\sum_{i=1}^{\mathcal{V}} \sum_{j=1}^{\mathcal{H}} \mathrm{W_{ij} v_i h_j} - \sum_{i=1}^{\mathcal{V}} \mathrm{b_i v_i} - \sum_{j=1}^{\mathcal{H}} \mathrm{a_j h_j} \qquad (6.6)$$

where $\theta = \{\mathrm{a, b, W}\}$ are the RBM parameters: $\mathrm{W_{ij}}$ represents the weight between visible unit i and hidden unit j, and $\mathrm{b_i}$ and $\mathrm{a_j}$ are bias terms. $\mathcal{V}$ and $\mathcal{H}$ represent the numbers of visible and hidden units.

The probability that RBMs assign to visible and hidden vectors (v,h) is:

$$\mathrm{P(v|\theta)} = \frac{\exp(-\mathrm{E(v,h)})}{\sum_{\mathcal{V}} \sum_{\mathcal{H}} \exp(-\mathrm{E(v,h)})} \qquad (6.7)$$

where $1/\sum_{\mathcal{V}} \sum_{\mathcal{H}} \exp(-\mathrm{E(v,h)})$ is the normalizing constant.

Since there are no visible-visible and no hidden to hidden connections, the conditional distributions over the visible and hidden are factorial and given by:

$$\mathrm{P(h|v,\theta)} = \prod_{j=1}^{\mathcal{H}} \mathrm{p(h_j|v)}, \text{with } \mathrm{p(h_j = 1|v)} = \mathrm{g}\left(\mathrm{a_j} + \sum_{i=1}^{\mathcal{V}} \mathrm{W_{ij} v_i}\right) \qquad (6.8)$$

$$\mathrm{P(v|h,\theta)} = \prod_{i=1}^{\mathcal{V}} \mathrm{p(v_j|h)}, \text{with } \mathrm{p(v_i = 1|h)} = \mathrm{g}\left(\mathrm{b_i} + \sum_{j=1}^{\mathcal{H}} \mathrm{W_{ij} h_j}\right) \qquad (6.9)$$

where $\mathrm{g(x)} = 1/(1 + \exp(-\mathrm{x})))$ is the logistic function. The gradient of the log-likelihood of the training data with respect to the model parameters is given by:

$$\frac{\partial}{\partial \mathrm{w_{ij}}} \log \mathrm{p(v;\theta)} = \langle \mathrm{v_i h_j} \rangle_{\mathrm{data}} - \langle \mathrm{v_i h_j} \rangle_{\mathrm{model}} \qquad (6.10)$$

The first term in Equation 6.5 $\langle \mathrm{v_i h_j} \rangle_{\mathrm{data}}$ denotes the conditional probability of h given v. This term is reasonably straightforward to compute. However, the term $\langle \mathrm{v_i h_j} \rangle_{\mathrm{model}}$ denotes the expectation under the distribution defined by the model over all (v, h) joint configurations, and to compute it exactly, exponential time is needed. Hence, the Contrastive Divergence (CD) (Hinton, 2002) training procedure is often used to estimate this term.

6.2.1.1.2 Gaussian-Bernoulli RBMs

In Gaussian RBMs, the visible units are real-valued $v \in R^{\mathcal{V}}$, and the hidden units are binary stochastic $h \in \{0,1\}^{\mathcal{H}}$, The energy of the Gaussian RBM is defined as:

$$E(v,h|\theta) = -\sum_{i=1}^{\mathcal{V}}\sum_{j=1}^{\mathcal{H}} \tfrac{v_i}{\sigma_i} W_{ij} h_j - \sum_{i=1}^{\mathcal{V}} \frac{(v_i - b_i)^2}{2{\sigma_i}^2} - \sum_{j=1}^{\mathcal{H}} a_j h_j \tag{6.11}$$

To keep Equation 6.6 simple, the Gaussian noise level σ is assumed to be fixed at 1 for all the visible units. Equation 6.11 can therefore be simplified to:

$$E(v,h|\theta) = -\sum_{i=1}^{\mathcal{V}}\sum_{j=1}^{\mathcal{H}} W_{ij} v_i h_j - \sum_{i=1}^{\mathcal{V}} \frac{(v_i - b_i)^2}{2} - \sum_{j=1}^{\mathcal{H}} a_j h_j \tag{6.12}$$

Since there are no connections between visible-visible units, the conditional distribution over the visible units is given by:

$$P(v,h|\theta) = \frac{\sum_{\mathcal{H}} \exp(-E(v,h|\theta))}{\int_{\mathcal{V}} \sum_{\mathcal{H}} \exp(-E(v,h|\theta))} \tag{6.13}$$

Similar to the RBMs, the conditional distribution is factorial and is given by:

$$P(h|v,\theta) = \prod_{j=1}^{\mathcal{H}} p(h_j|v), \text{ with } p(h_j = 1|v) = g\left(a_j + \sum_{i=1}^{\mathcal{V}} W_{ij} \frac{v_i}{\sigma_i}\right) \tag{6.14}$$

$$P(v|h,\theta) = \prod_{i=1}^{\mathcal{V}} p(v_j|h), \text{ with } p(v_i|h,\theta) = \mathcal{N}\left(b_i + \sigma_i \sum_{j=1}^{\mathcal{H}} W_{ij} h_j\, , \sigma_i^2\right) \tag{6.15}$$

where $\mathcal{N}(\mu,\sigma^2)$ indicates to a Gaussian distribution with mean and variance (μ,σ^2). The derivative of the log-likelihood of the training data with respect to the model parameters is given by:

$$\frac{\partial}{\partial w_{ij}} \log p(v;\theta) = \langle \frac{v_i}{\sigma_i} h_j \rangle_{\text{data}} - \langle \frac{v_i}{\sigma_i} h_j \rangle_{\text{model}} \tag{6.16}$$

6.2.1.2 RBM for phoneme recognition

The states of the visible units of the RBM are fed with a frame of the feature vector. When training the RBM to model the joint distribution of a frame with $\mathcal{L}$ possible phone (phoneme) labels, as in this thesis, the HMM states are produced by a forced alignment algorithm using a pre-existing ASR model using the HTK toolkit (Young et al., 2003). Hence, the energy function becomes:

$$E(v,l,h|\theta) = -\sum_{i=1}^{\mathcal{V}}\sum_{j=1}^{\mathcal{H}} W_{ij}h_j v_i - \sum_{k=1}^{\mathcal{L}}\sum_{j=1}^{\mathcal{H}} W_{kj}h_j l_k - \sum_{j=1}^{\mathcal{H}} a_j h_j - \sum_{k=1}^{\mathcal{L}} c_k l_k - \sum_{i=1}^{\mathcal{V}} \frac{(v_i - b_i)^2}{2\sigma_i{}^2} \tag{6.17}$$

$$p(l_k = 1|h;\theta) = \text{softmax}\left(c_k + \sum_{j=1}^{\mathcal{H}} W_{kj}h_j\right) \tag{6.18}$$

Also, p(l|v) can be computed using:

$$P(l|v) = \frac{\sum_{\mathcal{H}} \exp(-E(v,l,h))}{\int_{\mathcal{L}} \sum_{\mathcal{H}} \exp(-E(v,l,h))} \tag{6.19}$$

The visible-hidden weights are updated using:

$$\Delta w_{ij} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \tag{6.20}$$

6.2.1.3 Stacking RBMs to construct a Deep Neural Network

After training an RBM on speech data, the states of the hidden units can be treated as data to train another RBM. This generative “pre-training” process can be repeated many times to create many nonlinear feature detectors (see Figure 6.2 (b)). “Pre-training” a multilayer generative neural network, one layer at a time, has been shown to consistently give good phone recognition (Mohamed, 2014), and good MNIST handwritten digit recognition (Hinton and Salakhutdinov, 2006). The RBMs in stacks are then combined, similar to (Hinton et al., 2006), to produce only one, multilayer generative model named a Deep Neural Network (DNN). A subsequent discriminative fine-tuning phase is followed using all the generative weights in the opposite way to initialize the feature detecting layers using the feed-forward standard backpropagation algorithm. The features are then adjusted in every layer for a more useful discrimination.

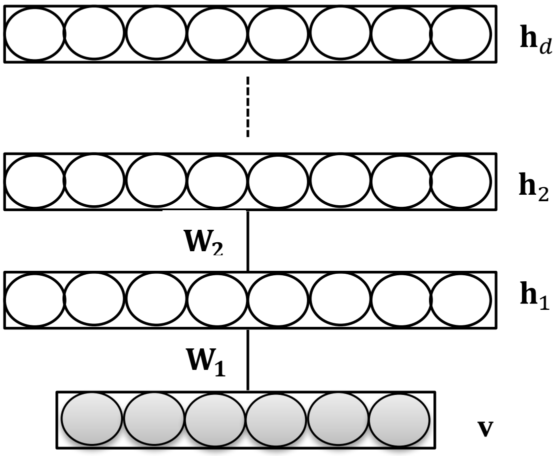


Figure 6.2 – (b) The DNN is composed of RBMs (from (Mohamed et al., 2009))

6.3 RBM Deep Neural Networks Experiments with the KAPD corpus

The aim of this set of experiments is to firstly compare Sallow MLP and Deep Neural Networks for frame-based classification system using the KAPD and CSLU2002 corpora. Deep Neural Networks are then used to develop Arabic acoustic models using the Levantine corpus. Several DNNs are developed and evaluated by comparing the obtained results to other results reported in the literature, where possible, with regard to the three proposed corpora.

6.3.1 RBM DNN Hyper-parameters Optimisation

For the developed systems in this section, the all conducted experiments followed a standard recipe. The stochastic gradient decent algorithm was used for pre-training phase with a mini-batch size of 100 training cases using learning rate 0.1, weight cost 0.0002 and momentum 0.9. For the structure of the RBMs in the pre training stage, we used three Gaussian RBMs, and trained them using 100 epochs for the Gaussian RBMs and 0.1 for a learning rate with different number of nodes in each RBM layer. In the fine-tune stage we used 10000 epochs for the KAPD and CSLU2002 datasets, and 5000 epochs for the Levantine dataset due to the hardware availability. We ran the experiments on the Matlab code offered by Geoffrey Hinton[7] to implement the developed systems.

6.3.2 Optimizing the size of the hidden layers in the RBM DNN

As common in neural network, there are no clear rules to select the number of hidden layer and the number of nodes in each layer. Thus, As suggested by Mohamed (Mohamed, 2014) to optimize the number of nodes in the hidden layers, number of nodes in the hidden layers should be changed once a time. Figure 3.3 shows the effect of the size the final hidden layer on the framewise phoneme recognition accuracy using 3 input frames KAPD dataset. For simplicity, the network used was optimised by fixing the size for the first and second hidden layers.

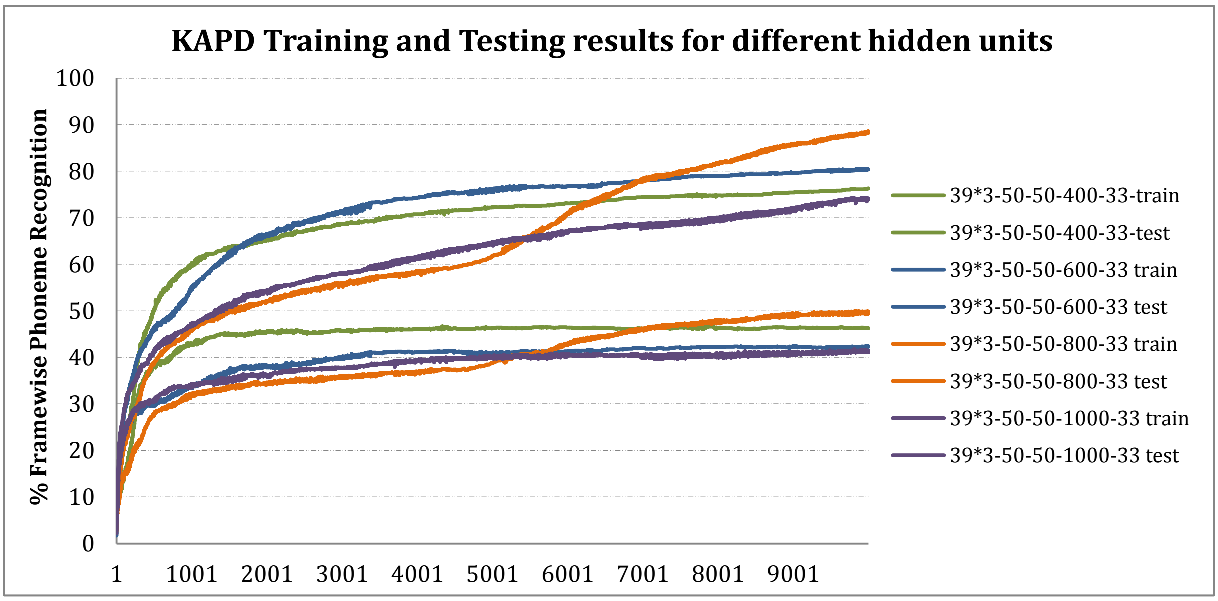


Figure 6.3 - Framewise phoneme recognition Accuracy on the training and the core test set as a function in the size of the final hidden layer, using 3 input frames.

These results were also proved using the HResult HTK performance analysis tool by comparing the label files obtained from the DNNs with the corresponding reference transcription files, where the accuracy is computed by:

[7] http://www.cs.toronto.edu/~hinton/

$$\text{Accuracy} = \frac{H - I}{N} \times 100\% \qquad (6.20)$$

where H is the number of correct labels (frames), I number of insertions and N is the total number of labels defined in the transcription files.

The main trend visible in Figure 3.3 is that adding more hidden units up to 800 nodes in the final hidden layer gives better performance in both training and testing datasets. However, 1000 nodes in the final hidden layer gives the worst perfomance compared with 400, 600, and 800 nodes in that layer.

The effect of number of nodes in the first and second hidden layers, on the framewise phoneme recognition accuracy using 3 input frames and a fixed number of 800 nodes in the first layer, and the results evaluated on the training and testing datasets are shown in Figure 3.4.

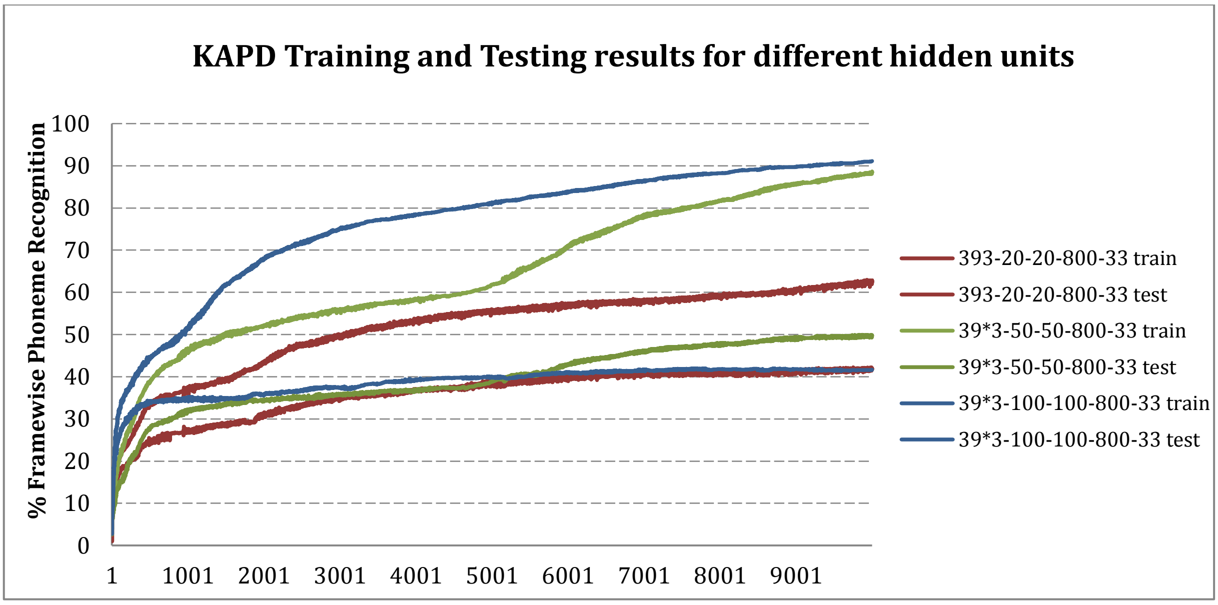


Figure 6.4 - Framewise phoneme recognition Accuracy on the training and the core test set as a function in the size of the first and second hidden layers, using 3 input frames.

Fixing the number of hidden units in the final hidden layer to 800 and varying the number of hidden nodes in the first and second hidden layers (Figures 6.4) shows that the best performance on the testing set is obtained using 50-50 hidden nodes in the first and second hidden layers respectively. Much smaller (20-20 hidden nodes) and much bigger (100-100 hidden nodes) in the first and second hidden layers give significantly worse performance. Therefore, the structure of the hidden nodes for the experiments with the CSLU2002 and the Levantine datasets was chosen based on these optimized experiments to be 50-50-800 hidden nodes for the first, second and final hidden layers respectively.

6.3.3 Optimizing the Range of Context (size of input windows) for the RBM DNN

As discussed earlier, non-recurrent NN systems cannot handle dynamic systems directly. Thus, a window of feature vectors is used to represent a sequence of frames. This window was used as input to the RBM DNN. As a consequence, one of the most important parameters to be optimized is the number of frames in the input window that is fed to the network. For non-recurrent NNs, the amount of context is dependent on the number of frames in the input window (size of the time-window). To optimise this context, the network was trained with different number of frames of window as inputs to the network. A study was conducted on the segmented phonemes of the KAPD and CSLU2002 dataset to infer the suitable number of frames in window size as input to the network.

The average length per phoneme was measured based on the number of frames per each phoneme and converted to the frame unit to represent the phoneme length in frames. Figure 6.5 shows the length of Arabic phonemes in “frame unit”. The window size sellected was 25ms with overlapped frames every 10 ms. The main finding shown in Figure 6.5 is that the average number of frames that could represent a suitable input window is found to be 5 frames.

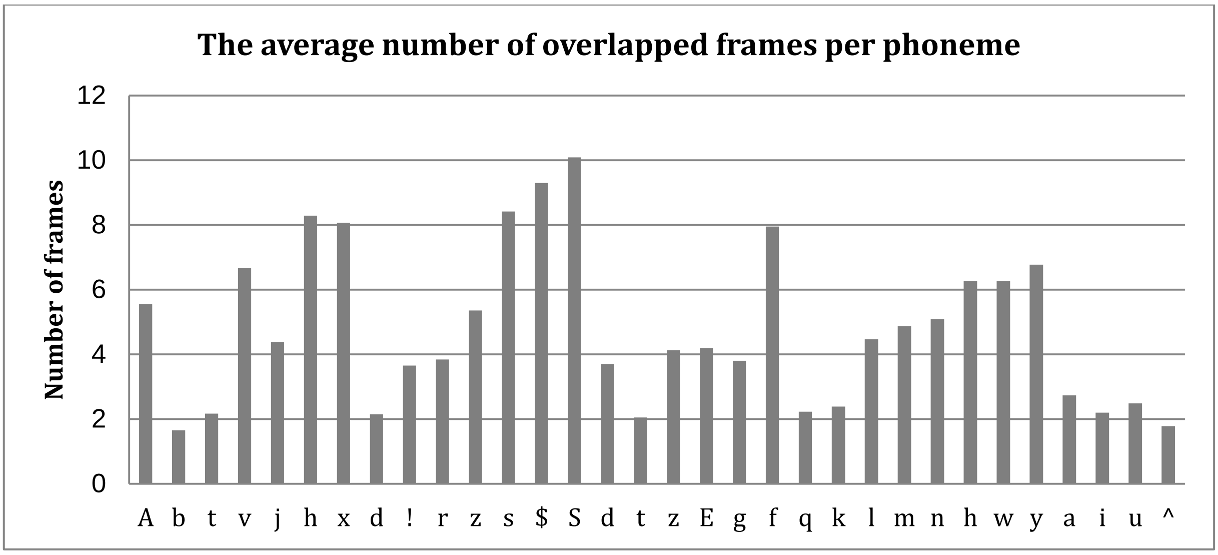


Figure 6.5 - The length of Arabic phonemes in “frame unit”.

The range from 70ms to 110ms roughly covers the average size of phonemes in the KAPD and CSLU2002 datasets that are used in this work. However, 46% of the Arabic phonemes in these datasets are smaller than 45ms. Most of these phonemes are the stop phonemes (/ء, ^/, /ب, b/, /د, d/, /ت, t/, /ض, D/, /ط, T/, /ق, q/, /ك, k/). Any smaller input windows can miss vital discriminative information in the context, whereas networks with a larger window size are more likely to be confused by irrelevant information that is far from the centre of the context frames of window. For segmented frames, similar to that found in the KAPD and the CSLU2002 dataset, context frames of windows between 3 and 5 frames as input windows work best. However, the Levantine dataset is labelled using triphones which suggestes that the network may work better with larger input windows, in order to provide the network with additional information that can help for discriminative.

Figure 6.6 and Figure 6.7 show the effect of varying the number of input frames in the context windows and the size of the final hidden layer with the RBM DNN network with 3 hidden layers. The results for the RBM DNN with no time-window (i.e. represents only the current frame, i.e. 1frame) provide a baseline for performance without context information. We used the same number of hidden nodes 50-50 in the first and second hidden layers respectively.

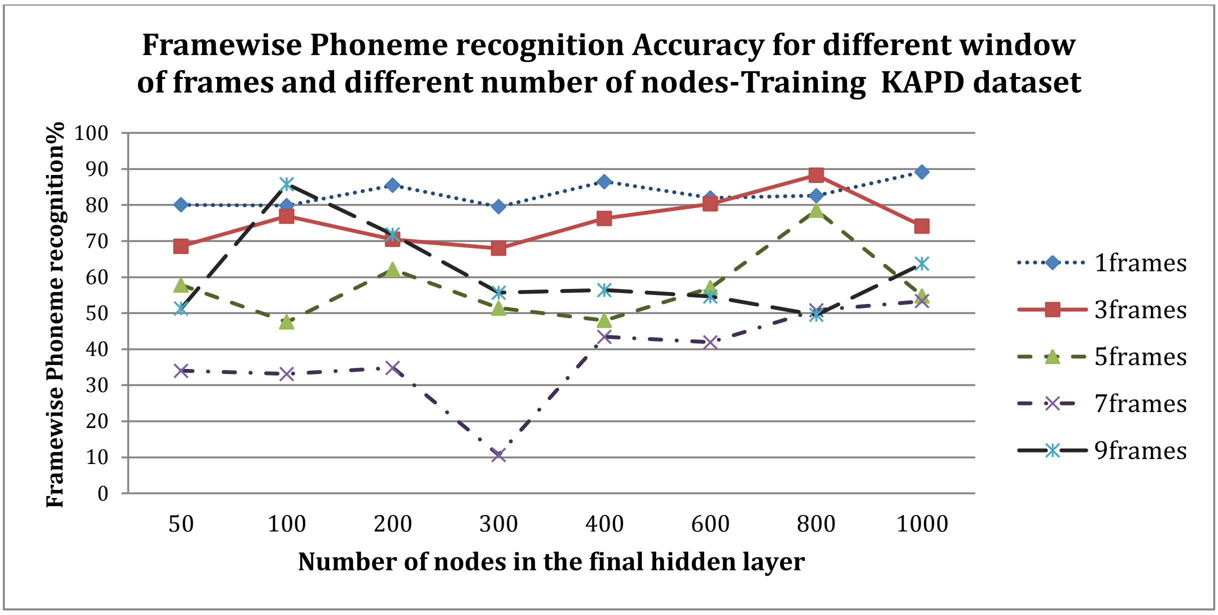


Figure 6.6 - The effect of varying the number of frames in the context input window and the number of nodes in the final hidden layer on the KAPD training dataset.

It is obvious from Figure 6.6 that smaller input windows provide better discrimination than the larger window size. 1frame and 3frame window sizes gives the best classification performance. It is likely that, networks with window size with more than 3 frames are distracted by irrelevant information that is far from the centre of the context frames of window.

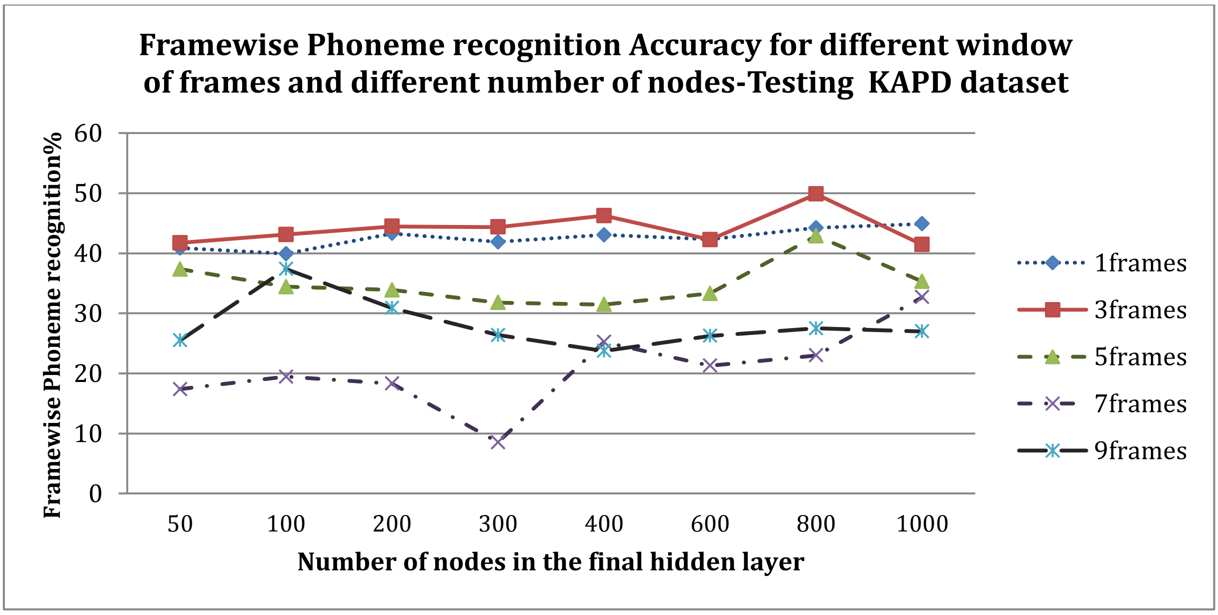


Figure 6.7 - The effect of varying the number of frames in the context input window and the number of nodes in the final hidden layer on the KAPD testing dataset.

Figure 6.7 shows the effect of using different size input features on the framewise phoneme recognition. The main trend visible in this figure is that adding more frames in the input window more than 5 frames gives worse performance. There is no significant difference in performance between the 1frame and 3frames window features found when using a two-tailed Matched Pairs T-test significance test using Matlab. Table 6.2 shows the best achieved framewise phoneme recognition for each type of input window features on the test set. By analyzing the performance that different systems produce using a two-tailed Matched Pairs T-test significance test, it is found that a significant difference between 3frames input system and 5frames, 7frames, and 9frames input systems at the level of p=0.01.

Table 6.2 - The average Framewise phoneme recognition Accuracy on the KAPD core test using various input window sizes.

Number of frames in input window	**Framewise phoneme recognition%**
1frame-1000nodes	44.89
3frame-800nodes	**49.85**
5frame-800nodes	42.84
7frames-1000nodes	32.69
9frames-100nodes	37.38

The results, summarised in Table 6.2, show the performance of the five models that were built to evaluate the optimum number of nodes in the final hidden layer. It is clear from the chart in Figure 6.7 and Table 6.2 that windows of 3frame size provide the best classification performance across all structures of the network. Increasing the number of hidden nodes in the final hidden layer upto 800 hidden nodes improves the classification performance of the network. The significance tests also prove that the 3frames-50-50-800 structure gives the best RBM DNN.

6.3.4 A comparison between Shallow and Deep NNs

As the KAPD corpus had been manually produced for this study, it was impossible to evaluate the developed systems in this chapter with the other systems reported in the literature. However, the Shallow MLP model results provided in Section 6.1.2, can be compared directly with the RBM DNN model results. Table 6.3 shows the performance of the experiments conducted on the KAPD dataset using the Shallow and Deep NNs models.

Table 6.3 - A comparison between Shallow and deep MLP Neural Networks.

Shallow MLP NN	RBM DNN
47.71%	**49.85%**

As can be seen in Table 6.3, the performance of the RBM DNN is about 2.13% better than the Shallow MLP NN. A two-tailed Matched Pairs T-test significance test was conducted with the null-hypothesis that there is no performance difference between the shallow MLP model and the RBM DNN model trained on the KAPD dataset. The test finds a significant difference at the level of p=0.05, with P-value (0.0147). Suggesting that RBM DNN to be adopted for the remainder work in this chapter.

6.3.5 Post-Processing Experiment

A post processing enhancement was also applied to the DNN outputs in order to improve the framewise phoneme recognition accuracy and to find the accuracy at the phoneme level instead of the frame level. This was achieved by selecting the phoneme classification of all frames within a phoneme as those given by the majority of frames within that phoneme i.e. if two narrowly separated blocks of frames are classified as the same phoneme then the intervening frames are also classified as that phoneme. A small example of this is shown in Table 6.4.

Table 6.4 A small example of chart of phonemes (framewise phoneme recognition).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ
	15						15						12		20	
							20				14				13	
	19	13					32		27			3	12	5	14	
							30		9							
				15			23					5	12	13	22	
				15					4				12		10	2
		15										12			21	
		14					20					12				
				22		22			9			12	12	17		
								17			8	14			14	
			13	16			13		4			12				
						20	5		19		19				4	
								19	24			12	15		14	
		13	20		18	4	16	19	25		8	3	12		14	
			5		18		20	19	25				12	20		
		4						19	25			12			20	
						20		19				12	5	9	4	
		12						10	26					12		
		20		15			3	33	24					12		
				16			4		24						15	

The main reason for including this step was to overcome the effect of distorted/noisy frames within a phoneme and to construct the phoneme level. As a result, the accuracy of frames and phonemes recognition was significantly improved over the raw results as shown in Table 6.5.

Table 6.5 - Shows the effect of the post-processing on the Framewise Phoneme Recognition using DNN model and the KAPD dataset.

Framewise phoneme recognition		**Phoneme recognition**	
Before the post-processing	**After the post-processing**	**Before the post-processing**	**After the post-processing**
49.85%	**72.23%**	50.55%	**66.09%**

Table 6.5 shows the effect of using the post-process algorithm on the framewise phoneme recognition output to produce phoneme recognition results. It is obvious that applying the post-process significantly improved the framewise phoneme recognition results and constructed the phoneme level. The two-tailed Matched Pairs T-test significance test finds a significant difference at the level of p=0.05 that was conducted with the null hypothesis that there is no performance difference between the framewise phoneme recognition before and after applying the post-process. Likewise, this significance test finds similar finds when conducted on the phoneme level performance.

However, this post-processing process can only be done on isolated phoneme speech samples. For acoustic modelling a different post-processing process is applied as presented in Section 6.4.3.

6.3.6 RBM DNN experiments with the CSLU2002 corpus

A different set of experiments were conducted for the RBM DNN with the CSLU2002 dataset for a range of input frames fed to the RBM DNN network. Figure 6.8 shows the effect of varying the number of input frames in the context windows on the framewise phoneme recognition using 50-50-800 hidden nodes for the first, second and final hidden layers respectively. For these experiments, we used the same parameters of the RBM DNN as used for the KAPD dataset experiments.

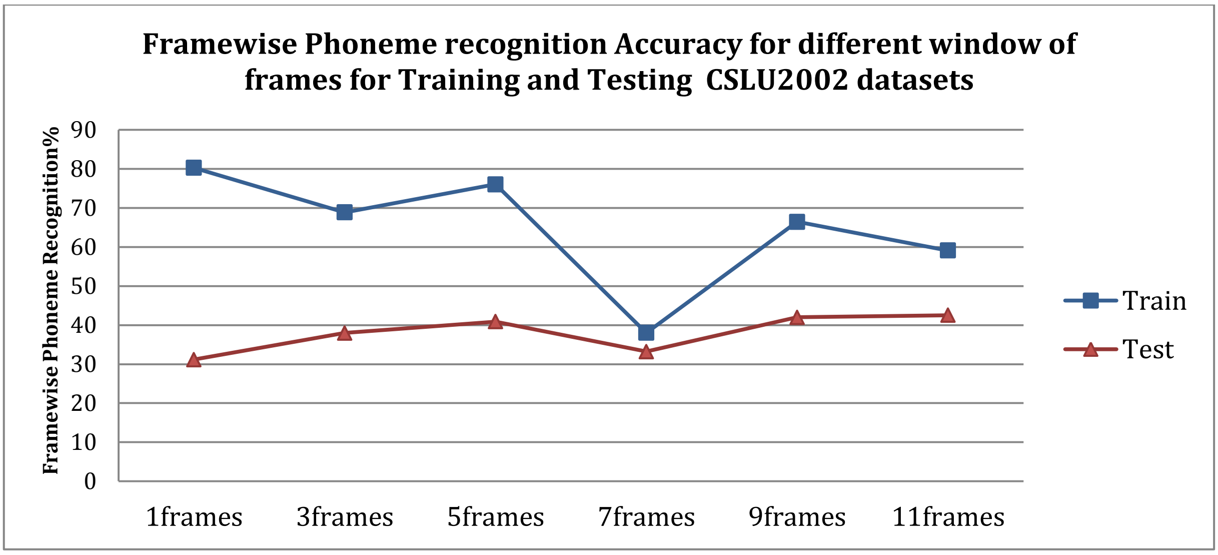


Figure 6.8 - The effect of varying the number of frames in the context input window on the CSLU2002 training and testing datasets.

The main trend visible in the Figure 6.8 is that adding more frames in the input window generally gives better test set performance. It is obvious that the larger 5frames, 9frames and 11frames input windows seem to be discriminated more than the smaller window size. 9frame and 11frame window sizes give the best test set classification performance with accuracy of 42.02% and 42.53% respectively. By analysing the performance of the different systems using a two-tailed Matched Pairs T-test significance test, the test finds a significant difference between the baseline system 1frame input window and the 5frames, 9frames, and 11frames input systems at the level of p=0.05. However, the performances of 9frames and 11frames systems are about 1.1% and 1.62% better than the 5frames system. A two-tailed Matched Pairs T-test significance test was conducted with the null-hypothesis that there is no performance difference between the 9frames, and 11frames systems and the 5frames system. The test could not reject the null hypothesis. Consequently, the 5frames input system is found to be the optimal input time-window for the Arabic language. This finding coincides with finding of the study conducted in Section 6.3.3 with the KAPD and CSLU2002 dataset.

The relatively low Arabic phoneme recognition performance on the small sizes speech datasets is probably due to the dialect differences in the speakers who are from different countries. Moreover, the limited number of examples of some phonemes leads to less training for the models for these phonemes. Finally, noise from the environment and from the speakers has a corrupting effect on the RBM DNN performance.

In the next section we explore acoustic modelling for Arabic speech recognition with the large vocabulary Levantine corpus.

6.4 The Levantine dataset Experiments

The Levantine Arabic Corpus is unlike the corpora used in the previous sections, in terms of the size and type of the data-labelled speech. This allows us to explore the potential gain in building a RBM DNN based acoustic models for Arabic speech recognition. It thus allows us to directly compare our results with those of others presented in the literature.

6.4.1 RBM DNN Hyper-parameters optimisation with the Levantine corpus

As it described in Chapter five, 13 hours of conversational speech were randomly selected to produce the training dataset. However, the testing dataset that suggested from the LDC is completely maintained as a testing dataset for our experiments on this corpus to ensure a valid comparison of this acoustic modelling Arabic speech recognition system with any other developed systems published work in future. The experiments on this corpus is followed the same standard recipes that described in Section 6.3.1. The inputs were decided to be 5frame input window based on the optimisation experiments that conducted in Sections 6.3.3 and 6.3.6 on the KAPD and the CSLU2002 corpora. However the structure of the network is slightly different due to the size of this dataset and the hardware availability. For this experiment we used 30, 20, and 30 nodes in the hidden layers. These numbers of nodes were experimentally chosen within limit number of nodes in each layer.

6.4.2 Experiment's Results

A different set of experiments were conducted for the Levantine corpus for a range of input frames fed to the RBM DNN network. Table 6.6 presents results of three systems with varying number of input frames in the context windows for framewise phoneme recognition. For these experiments, we used the same hyper-parameters of the RBM DNN as used for the KAPD and the CSLU2002 datasets experiments.

Table 6.6 - The average Framewise phoneme recognition Accuracy on the Levantine core test using various input frames.

Number of frames in input window	**Framewise phoneme recognition%**
1frame	26.19
5frame	**28.80**
7frames	28.22

The relatively low Arabic phoneme recognition performance on the Levantine corpus is probably due to the nature of the corpus that is conversational telephone speech for different dialects. See Table 6.7 for the performance of most published work for conversational telephone speech on the Levantine dialect for various corpora and sizes. It is difficult to make a direct comparison on these corpora as they all have different training and testing sizes and use different data catalogues. In addition, the metrics is different of the metric of our woe as all the published used the HMMs that offers the language model that uses the Word Error Rate (WER) metric instead of the Phone Error Rate (PER) or Frame-wise Phone Error Rate (FPER) metrics.

Table 6.7 - The performance of most published work for conversational telephone speech on the Levantine dialect for various corpora and sizes.

Model	Size of data in hours		WER	Reference
	Train	test		
HMMs-MFCCs-grapheme-based	70 h-Conversational Telephone Speech	3 h	46.7%	(Vergyri et al., 2005)
HMMs-MFCCs-grapheme-based	50 h- microphone speech	10 h	28.8%	(Elmahdy et al., 2012b)
HMMs-PLP-grapheme-based	41 h-Conversational Telephone Speech	1.12 h	70.6%	(Al-Shareef and Hain, 2012)
HMMs-PLP-grapheme-based	143.3 h-Conversational Telephone Speech	5.1 h	60.3%	(Al-Shareef and Hain, 2012)
HMMs-PLP-grapheme-based	175 h-Conversational Telephone Speech	5.1 h	59.3%	(Al-Shareef and Hain, 2011)
HMMs-MFCCs-grapheme-based	32 h-Conversational Telephone Speech	1 h	80.1%	(Heintz, 2010)
HMMs-MFCCs-phoneme-based	1 h Conversational Telephone Speech	2.5h	75.2%	(Novotney et al., 2011)
	10 h		60.9 %	

6.4.3 A post processing for acoustic modeling investigated on the Levantine corpus

A post-processing enhancement was also applied to the DNN acoustic model outputs in order to improve the recognition accuracy and to obtain the accuracy at the phoneme level rather than the frame level. This was achieved by a two-step process. The first step was to '*window'* the sequence by choosing a window size with multiple frames then classify this window of frames according to the most frequent frame within the window. For example if we have chosen the window size is 3 frames then we will classified these selected frame to the most frequent one. i.e the window classification of all frames within a window as those given by the majority of frames within that window. This step will reduce the sequence by 1/w, where w is the window of frame size.

For example if we take the following sequence of frames representing the acoustic utterances and we chose a '*winding'* of 3 then

/sil//A//A//A//l//l//l//l//l//l//u//u//u//u//u//w//w//w//w//w//w//w//w//sil/

/A/ /l/ /l/ /u/ /u/ /w/ /w/ /w/

As we can see a new acoustic sequence was produced from the windowing step. This new sequence has a number of repeating frames or labels. So, the second step was to remove the repeating frames in the sequence. The second step was applied by (Graves, 2012) also as a post process stage. Thus the result of the sequence after the second step is:

/A/ /l/ /u/ /w/

These steps are applied on the outputs of the DNN framewise acoustic model with 5frames that achieved from Section 6.4.2. Figure 6.9 shows the accuracy as defined by Equation 6.20.

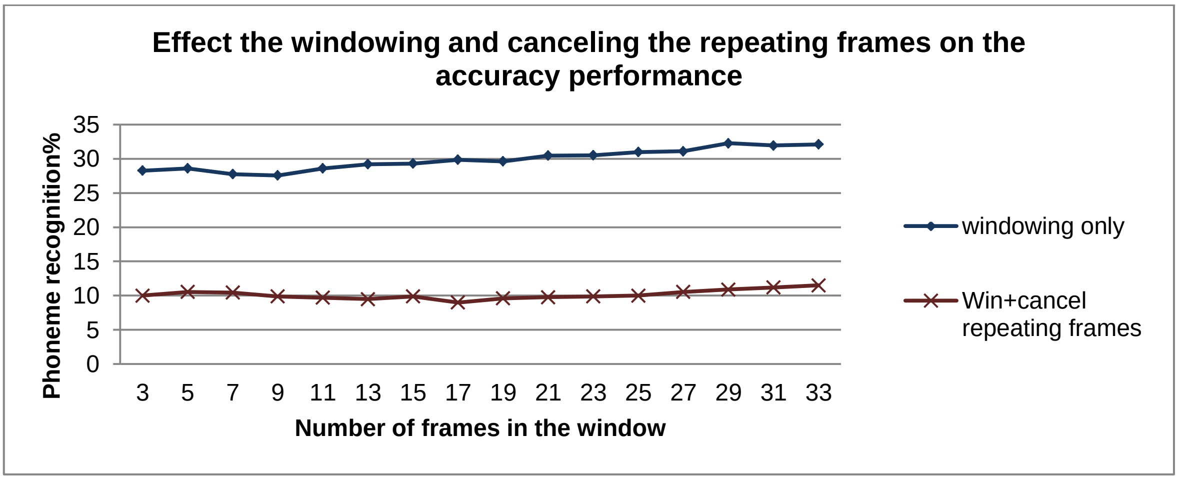


Figure 6.9 - Effect the windowing and cancelling the repeating frames on the accuracy performance.

It is obvious from the Figure 6.9 that applying various size of window has significantly improved the accuracy of phoneme recognition performance. About 3.84% phoneme recognition was increased in the accuracy performance by applying window size of 29 frames to the baseline which is the framewise phoneme recognition. By analyzing the accuracy performance for the base line and the best accyracy performance was gained by applying the 'window' procedure using a two-tailed Matched Pairs T-test significance test, the test finds a significant difference between framewise phoneme recognition, which represent window with one frame, and the windwing with 29 frames procedure at the level of p=0.05.

However, applying the cancellation procedure harmfully affected the accuracy performance due to the increase in the intersections in the sequence.

6.4.4 External works comparison

The results from our experiments should be compared to other published work on the same corpus. However, since we randomly select 13 hours of speech from the total size of the Levantine corpus, as described in Chapter five, there is no equivalent acoustic modelling published work on the same selected dataset. However, much work has been done on the Levantine dialect on many corpora and Table 6.7 shows comparison across previous research of WER using various systems for various Levantine corpora.

The results, summarised in Table 6.7, show the performance of most published work on the Levantine dialect for various corpora and sizes. It is difficult to make a direct comparison on these corpora as they all have different training and testing sizes and use different data catalogues. However, the main trend visible in the Table 6.7 is that HMMs-based models are the dominant technique used to investigate acoustic modelling for the Levantine Arabic dialect. So, to the best of our Knowledge, RBM DNN models have not been explored for any Levantine corpora. This allows us to claim priority for adopting this RBM DNN model for Levantine Arabic acoustic models. In addition, the Microphone-based Levantine corpus system outperformed the conversational Telephone-base speech corpus systems. This is similar to that demonstrated by our results of the small corpora (the KAPD is a microphone-based corpus and the CSLU2002 a Telephone line-based corpus). The nearest corpus in terms of training and testing sizes and conversational telephone line speech is the work of (Novotney et al., 2011) but with a different catalogue. The work of (Al-Shareef and Hain, 2012) was done on the same corpus from the same catalogue on the same testing dataset, the only one difference being the size of the training dataset at 41 hours. Our work is on selected 13 hours from these 41hours. Unfortunately all these reported work used a word error (WER) metric, which is different from our framewise phoneme error (FPER) and phoneme error (PER) metrics. Thus, we use the work presented in this section as a baseline for our work in Chapter seven in order to make legal and direct comparison.

6.5 Conclusion

The objective of this chapter was to explore the potential advantages of developing a RBM DNN-based system for Arabic speech recognition. In order to achieve this aim, several RBM DNN-based Arabic speech recognition systems were developed and evaluated across various corpora.

In this section, we discuss the findings of the conducted experiments through all of the proposed corpora. The performance of a state-of-the-art Shallow MLP based phoneme recognition system trained on the 13 MFCCs plus their first and second derivatives found to be about 0.35% better than the base system using only the 13 MFCCs. Statistically there was no significant difference in performance between the two different MFCCs feature sizes. However, the state-of-the-art 13 MFCCs plus their first and second derivatives features was adopted over all experiments that conducted in this thesis.

Deep Neural Networks (DNNs) were then developed and evaluated on small vocabularies (the KAPD and the CSLU2002 corpora) for phoneme classification task and for modelling the acoustic signals using a large vocabulary (the Levantine corpus) Arabic dataset. A direct comparison experiment was then conducted on the KAPD corpus between the results of the Shallow MLP model and the RBM Deep MLP model results. It was found that, the performance of the RBM DNN was about 2.13% better for Arabic phoneme recognition than the Shallow MLP NN.

A study was conducted on the segmented phonemes of the KAPD and CSLU2002 datasets to infer the suitable number of frames in window size as input to the network. The main finding from this study is that the average number of frames that represent a suitable input window is 5 frames. By anlysing the performance results of experiments conducted on the KAPD and the CSLU2002 corpora, were in agreement with 5 frames being more suitablt than 1, 3, 7, 9, and 11 frames to represent the time-window for Arabic speech recognition.

A post-processing enhancement was also applied to the DNN outputs in order to improve the framewise phoneme recognition accuracy and to find the accuracy at a phoneme level instead of the frame level. We found that applying the post process was significantly improved the frame wise phoneme recognition and successfully constructed the phoneme level. However, this post-processing process can be done only on isolated phoneme recognition. Hence, for acoustic modeling a different post-processing process is applied.

To evaluate the adoption of the DNN model in developing an acoustic modelling for Arabic speech recognition system instead of a phoneme-based system, the Levantine Arabic corpus was used to explore the potential gain in building an acoustic modelling Arabic speech recognition system using the DNN model. To the best of our knowledge, RBM DNN models have not been explored for any Levantine corpora. This encouraged us to investigate the DNN model for Arabic acoustic modelling on one of the proposed Levantine corpus. This allows us to claim priority for adopting this RBM DNN model for Levantine Arabic acoustic models.

A post-processing enhancement was also applied to the DNN acoustic model outputs in order to improve the recognition accuracy and to obtain the accuracy at a phoneme level instead of the frame level. This was based on '*windowing'* the acoustic sequence followed by cancelling the repeated frame in the sequence. The '*windowing'* post process was significantly improved the phoneme recognition performance. About 3.84% phoneme recognition was increased in the accuracy performance by applying window size of 29 frames to the baseline which is the framewise phoneme recognition. However, applying the cancelling procedure, harmfully affected the accuracy performance due to the increase in the number of intersections in the sequence as a result of this procedure.

Since the most published work on the Levantine dialect were for various corpora and sizes. It was difficult to make a direct comparison between other work using these corpora and our work as they have different training and testing sizes and different data catalogue. Thus, we use the work presented in this chapter as a baseline for our work in Chapter seven in order to make legal and direct comparison.

Despite the success of the RBM DNN, reported in this work, for phoneme classification and acoustic modeling, there is an inherent limitation in using feed-forward neural network, even when using a deep structure with multiple context frames. Fundamentally, feed forward NNs are not capable of handling dynamic data directly. In the next chapter, the recurrent Echo State Network dynamic model is adopted for phoneme classification and acoustic modelling using the same corpora for Arabic speech recognition.

Chapter Seven

Echo State Networks (ESNs) for phoneme classification and acoustic models

Echo State Networks (ESNs) have recently been applied to various applications with impressive success. ESNs models have proven to be very successful in training for acoustic modelling; which is considered as the main component of the-state-of-the-art speech recognition systems. This success encouraged us to adopt ESNs models for developing Arabic speech recognition systems. The aim of this Chapter is to investigate Echo State Networks (ESNs) for phoneme classification and acoustic modelling. In order to achieve this aim, we have explored different ESN architectures. Similarly to the DNNs systems, described in Chapter six, these models were initially trained on small vocabularies (the KAPD and the CSLU2002 phoneme) corpora and then the large vocabulary (the Levantine) corpus. To the best of our knowledge, this work is the first effort in the literature to adopt ESNs for acoustic modelling of Arabic speech recognition. This Chapter is organised as follows: a description of the main concepts of the proposed Echo State Networks (ESNs) as an Arabic framewise phoneme classifiers and acoustic modelling system is first introduced in Section 7.1. Section 7.2 then describes the Echo State Network structures and algorithm implementations with a series of results and analysis for Arabic phoneme recognition systems with the KAPD and CSLU2002 corpora. Section 7.3 then presents ESN acoustic models for the Levantine corpus. A comparison between the best results achieved in this thesis and the results of other related works are presented in Section 7.4. Finally, in Section 7.5, conclusions from the ESN experiments are provided.

7.1 Conventional Echo State Networks (ESN) algorithm

ESNs are capable of modelling nonlinear systems with multiple inputs and multiple outputs. Also, being a recurrent network, its outputs can be used as additional inputs by feeding them back to the input stage. This gives a possibility to train ESNs as generators or oscillators (Holzmann, 2008).

The ESN training stages can be divided into three main stages as follows:

7.1.1 ESN activations

Assuming an ESN with K input neurons, N internal neurons (forming a dynamic reservoir) and L output neurons. The input vector $u(t) = (u_1(t), \dots, u_k(t))$ is the activation of the input neurons at time step t, and the reservoir vector $x(t) = (x_1(t), \dots, x_N(t))$ is the activation of the internal neurons. The output vector $\hat{y}(t) = (y_1(t), \dots, y_L(t))$ is the activation of the output neurons, and $y(t) = (y_1(t), \dots, y_L(t))$ is the target output vector. The connection weights from the input neurons to the reservoir neurons are composed of a $N \times K$ weight matrix W^{in}. The weights of the reservoir connections are collected in an $N \times N$ weight matrix W and the weights of the connections from the reservoir to the output are in L× $(K + N + L)$ weight matrix W^{out}. The weights of the connections from the output to the internal neurons are in $N \times L$ weight matrix W^{back}.

Updating of the reservoir neurons is according to:

$$x(t+1) = g(W^{in}u(t+1) + Wx(t) + W^{back}y(t)) \tag{7.1}$$

Where g is the nonlinear activation functions of the reservoir neurons hyperbolic tangent *(tanh)*, asymmetric sigmoidal function etc. The output neurons are then activated according to:

$$y(t+1) = g(W^{out}[x(t+1); u(t+1)]) \tag{7.2}$$

where g is linear or nonlinear output activation function, and $[x(t+1); u(t+1)]$ is a concatenation of the reservoir states and the input vectors.

7.1.2 Echo state property

ESNs require specific properties to model a nonlinear system. The echo state property is determined by the untrained reservoir weights W, input weights W^{in} and feedback weights W^{back}. Although, there are no known specific algebraic conditions for the echo state property, there are several conditions specified in (Jaeger, 2005) to select the initial conditions and to generate W matrix weight that has an echo state property.

The most important parameters for the success of modelling any nonlinear problem using an ESN are the spectral radius α and the size of the network. The ultimate success of an ESN mainly depends on diligently selecting the spectral radius α. Herbert Jaeger in (Jaeger, 2005) gives a reason for that as follows:

> "*This is because α is intimately connected to the intrinsic timescale of the dynamics of the Dynamic Reservoir (DR) state. Small α means that one has a fast DR, large α (i.e., close to unity) means that one has a slow DR. The intrinsic timescale of the task should match the DR timescale*".

7.1.3 ESN training algorithms

Generally, the function of a training algorithm is to compute an optimal mapping from the collected state vector $x(t)$ to a teacher output y_{teach} over time. ESNs use a regression algorithm to find a global minimum of the mean square error in order to find the right mapping (Holzmann, 2008). There are offline and online training algorithms for an ESN used to train a read-out function that lead to learning the output layer's weights W^{out}.

7.1.3.1 Offline ESN training algorithm

Jaeger in (Jaeger, 2005) demonstrated an algorithm to calculate the output weights W^{out} which minimize the mean squared error MSE during training.

At time step t the error is calculated as the difference between a teacher output y_{teach} and the actual output obtained from the network:

$$e_{train}(t) = y_{teach} - W^{out}[x(t+1); u(t+1)] \tag{7.4}$$

Then the MSE is calculated as:

$$MSE_{train} = \frac{1}{t_{max}-t_{min}} \sum_{t_{min}}^{t_{max}} e_{train}(t)^2 \tag{7.5}$$

where t_{max} is the number of vectors and t_{min} is the number of washout vectors. The dynamic reservoir usually needs t time steps to eliminate the initial transients, and these t samples are usually not considered for calculating W^{out}.

The offline algorithm is mainly based on the pseudo-inverse method to calculate the output weights W^{out} and is obtained as follows:

$$W^{out} = ((A^T A + \alpha^2 I)^{-1} A^T B)^T \tag{7.6}$$

where I is the identity matrix, $(\mathrm{A})^{-1}$ is the pseudo inverse matrix of A which is the accumulated the autocorrelation matrix (state matrix) A of size $(t_{max} - t_{min}) \times (\mathrm{N}+\mathrm{K})$ from the state vector $x(t)$ and the input vector $u(t)$ for each time step t, and A^T donates the transpose of matrix A. Also, B is the accumulated the cross-correlation matrix (output matrix) B of size $(t_{max} - t_{min}) \times \mathrm{L}$ from the target output $\mathrm{y_{teach}(t)}$ for each time step t. α is the smoothing factor, where $\alpha \leq 1$, the strongest regularization is obtained when $\alpha = 1$. When $\alpha = 0$ the system becomes a linear regression:

$$W^{out} = ((A^T A)^{-1} A^T B)^T \tag{7.7}$$

7.1.3.2 Online an ESN training algorithm

The online training algorithm can be used to train an ESN in each time step t. The early stages of the online training algorithm are similar to the offline training algorithm. Generating and initializing the matrixes input weight matrix W^{in}, Internal weight matrix W, feedback weight W^{back}, and output weight W^{out}. The states in the dynamic reservoir are then calculated in a similar manner to the offline training algorithm using Equation 7.1 and 7.2. With online training, it is no longer necessary to accumulate the states as a new row of an autocorrelation matrix A at time t.

The network output $y(t+1)$ is obtained by:

$$y(t+1) = x(t+1) W^{out}(n) \tag{7.8}$$

Updating of the output weights W^{out} in the online training algorithm is completely dependent on the error vector of the same time step (only W^{out} are updated) this is similar to the Widrow-Hoff rule which is also known as the delta rule:

$$e_y(t+1) = y_{teach}(t+1) - y(t+1) \tag{7.9}$$

The output weights are then calculated using:

$$W^{out}(t+1) = W^{out}(t) + \eta x(t+1)^T e_y(t+1) + \gamma x(t)^T e_y(t) \tag{7.10}$$

where η is the learning rate, and γ is the momentum and their valves are within $[0,1]$. These parameters have similar properties to those used in other kinds of Neural Networks training algorithms. The optimal output weights W^{out} is such that it minimizes the Mean Square Error MSE, calculated using Equations 7.4 and 7.5.

In a regime in which the size of data is very large, the Widrow-Hoff rule can be implemented during an online or mini-batch training. This type of training was used for training the DNNs and the ESNs acoustic models for the Levantine corpus.

For implementing the Echo State Network algorithm, C++ code has been written that follows the steps of the algorithm described above.

7.2 ESN phoneme recognition experimental results

Based on the ability of ESNs to model nonlinear complex dynamic systems, and the simplicity of their learning algorithm, ESNs were proposed for Arabic speech recognition. The KAPD Saudi Arabia dialect corpus and the CSLU2002 multi-dialect corpus were used to train and test an ESN with supervised and forced supervised learning algorithms. Furthermore, a novel combined supervised/forced supervised learning algorithm was developed and tested on an optimised Arabic phoneme recognition echo state network using both datasets.

7.2.1 Experiments of ESN with supervised learning (conventional ESN)

The architectures of the ESN used in this work was based on the work of Ted et al. (hmidt et al., 2010) and Jaeger (Jaeger, 2005). The ESN connections used are shown in

Figure 7.1. During ESN training and testing, the activations of the reservoir and output neurons were calculated using either a supervised learning or forced supervised learning algorithms.

The activations of reservoir and output neurons are calculated using Equations (7.12) and (7.13) respectively:

$$x(t+1) = tanh\left(W^{in}u(t+1) + Wx(t) + W^{back}y(t)\right) + leftover \quad (7.12)$$

Where $x(t+1)$ is the reservoir state for time step $(t+1)$, $u(t+1)$ is the input vector, $y(t)$ is the calculated output for the supervised learning for time step (t), and the $leftover$, as used in (hmidt et al., 2010), is calculated as:

$$leftover = x(t) \times (1 - d\_leakRate) \quad (7.13)$$

The calculated output $y(t+1)$ is then given by:

$$y(t+1) = tanh(W^{out}[x(t+1); u(t+1)]) \quad (7.14)$$

Where $y(t+1)$ is the calculated output states for time step $(t+1)$.

Weight vectors W^{in}, W, W^{back} and, W^{out} are initially generated with random values between -1 and 1, and with a connectivity parameter between 0 and 1 in order to generate random connections between neurons (see Figure 7.1).

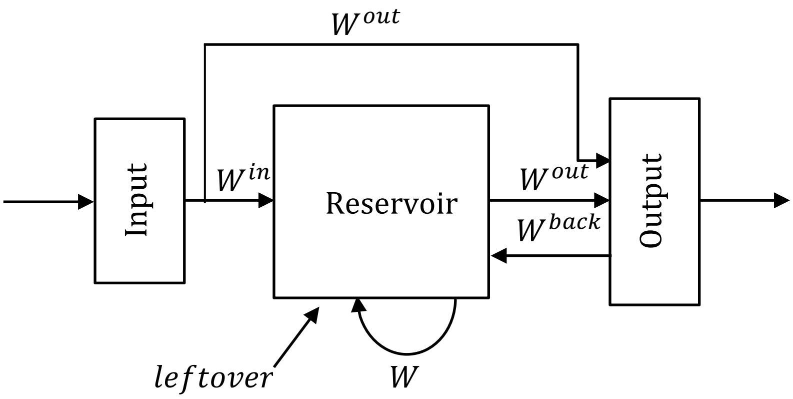


Figure 7.1 - ESN connections with supervised learning.

The ESN network was implemented and optimized for the proposed Arabic speech databases, and the hyper-parameters values used for the experiments in this Chapter are shown in Table 7.1.

Table 7.1 - The ESN hyper-parameters.

Parameter	Value
Reservoir size:	400
Connectivity:	0.2
Spectral Radius:	0.996
Activation function:	tanh
Input size:	39
Input connectivity:	0.5
Input Shift:	0
Input Scale:	1
Output size:	33
Output activation:	tanh
Feedback connectivity:	0.3
Feedback Shift:	0
Feedback Scale:	1
D_leakRate	0.6
Alfa	0.33
Wash-out time	0

ESNs are designed especially for learning pattern sequences. Hence, it is crucial to order the Arabic frames as sequences. Consequently, all frames of each phoneme were connected to build one sequence to represent this phoneme. From the initial results it was found that the ESN had not learnt short sequences as well as it had learnt the long sequences.

A validation dataset were used for early stopping and to save the output weights W^{out} that lead to the best performance (minimum ESN error). As is shown by Figure 7.2, training of the ESN with supervised learning continues until the ESN error increased at time step $t + 1$ Training was then stopped and the network weights were saved for time t.

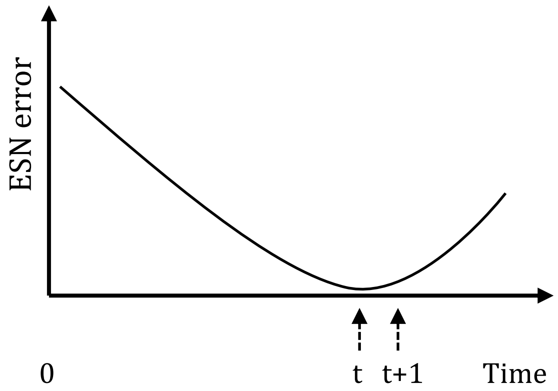


Figure 7.2 - Time of saving the network and stop training.

The system performance was recorded over many different reservoir sizes; from 50 to 700 neurons. The accuracy of the system, as shown in Figure 7.3, tends to improve significantly as the reservoir size increases up to 400 neurons. However, only 1% difference in the performances of the system were obtained between reservoir sizes of 400 and 700 neurons. Figure 7.3 shows the system performance for the CSLU2002 Arabic phonemes with the different reservoir sizes.

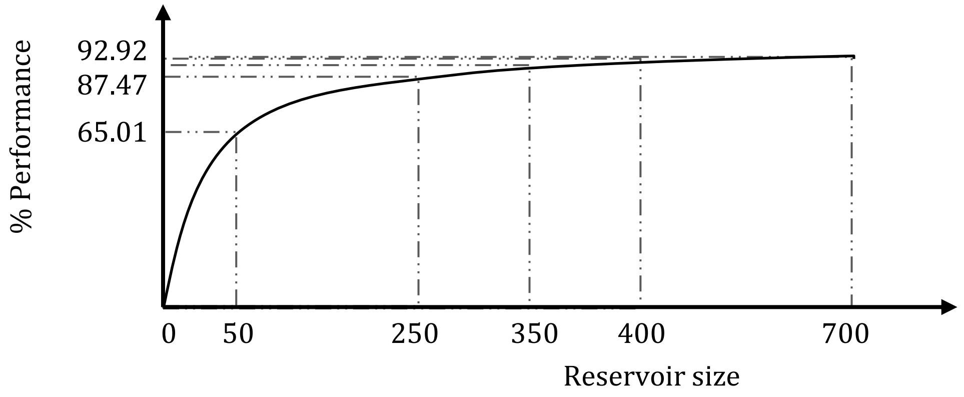


Figure 7.3 - The performance of the ESN with different reservoir sizes.

7.2.1.1 Experiments on the KAPD database

ESNs differ from feed forward NNs in the way they access the context information. As mentioned in Chapter six, for feed forward NNs, the amount of context is dependent on the number of frames in the input window (size of the time-window). However, through their recurrent connections ESNs had access to the whole input sequence (i.e to the complete context in one direction only).

The frames of the normalized KAPD dataset that were used to train the MLP NN in Section 6.2 are again used to train the ESN with the supervised learning algorithm. The ESN architecture used was as described in Figure 7.1 with the parameters presented in Table 7.1. The raw KAPD database was also used to train the ESN for comparison purposes.

7.2.1.1.1 Experiment results

An ESN was trained using all 33 Arabic phonemes extracted from the KAPD speakers. The average Arabic phonemes recognition results obtained are presented in Figure 7.4.

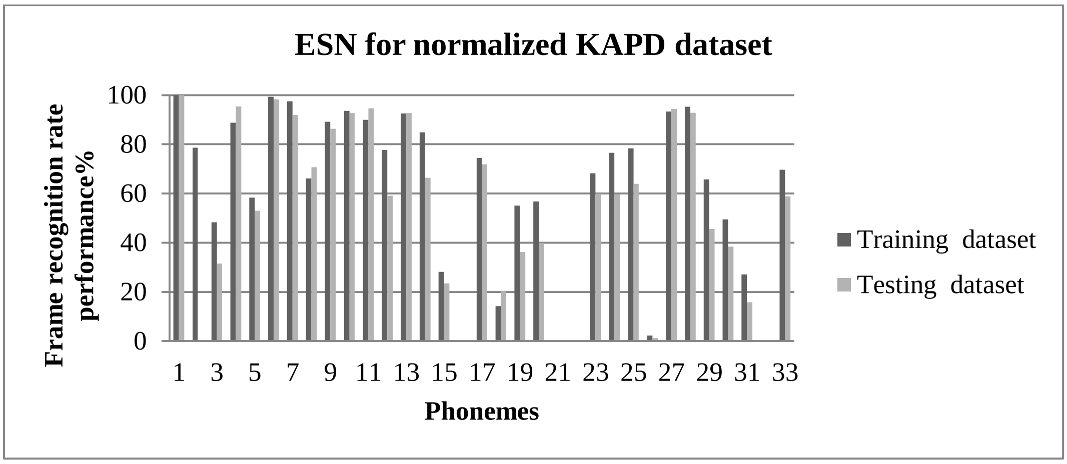
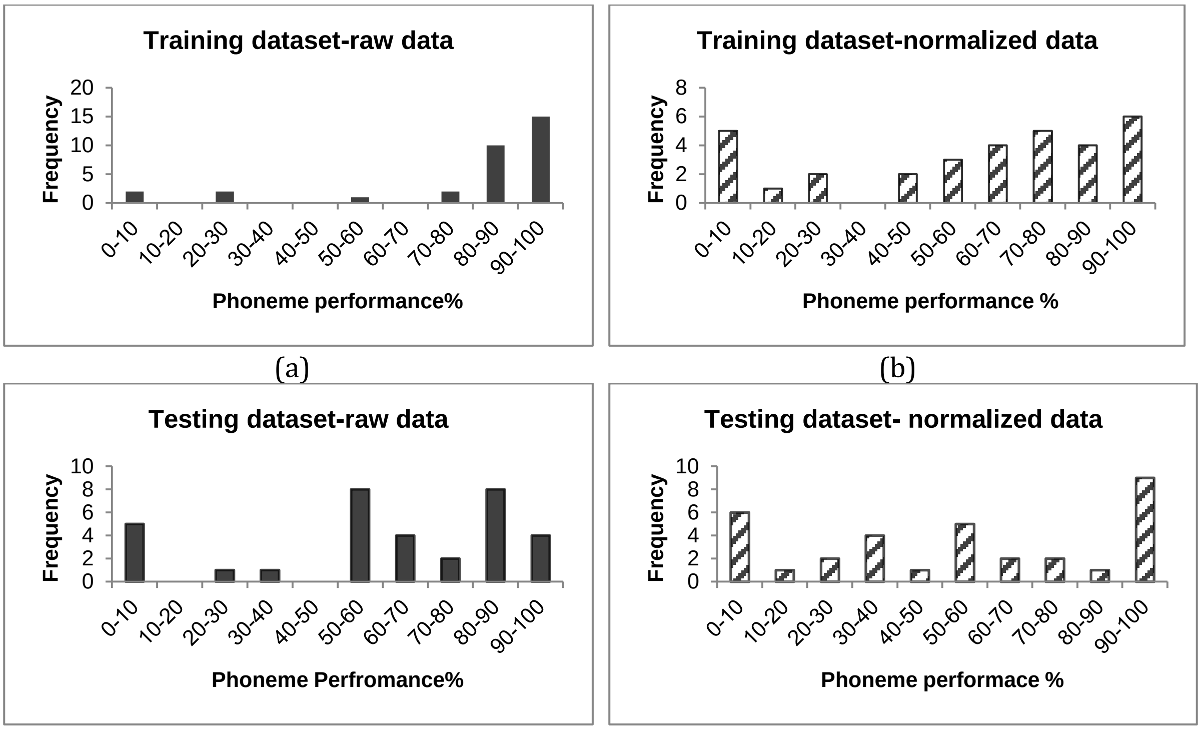


Figure 7.4 - The average train and test datasets results of the normalized KAPD for training and testing datasets.

It is clear from Figure 7.4 that, several phonemes of the normalized KAPD Arabic phonemes were recognized with accuracy below 60.0%. The average performances for the KAPD were 68.62% and 58.02% for training and testing datasets respectively. Figure 7.5 shows a comparison between the histograms of the average Arabic phonemes recognition results obtained for the raw and normalized KAPD training and testing datasets.



(c) (d)

Figure 7.5 - The histograms for Arabic phonemes recognition performances for training dataset for raw data (a) and normalized data (b), and for testing dataset for raw data (c) and normalized data (d).

The main trend visible in Figure 7.5 is that most of Arabic phonemes in the raw KAPD training dataset were recognized with performance between 80% and 100%; few phonemes are recognized below 70%. In comparison most of phonemes in the normalized KAPD training dataset were classified with only 40% to 100% accuracy. Moreover, for the testing dataset, most of the KAPD raw Arabic phonemes were recognized with superior performance 73.40% accuracy; with only 7 of the testing phoneme dataset being recognized with performance less than 50%. In contrast, approximately 14 phonemes of the KAPD normalized dataset were recognized with performance less than 50% with the average recognition of the KAPD normalized Arabic phonemes being 66.61%.

The two-tailed Matched Pairs T-test significance test finds a significant difference at the level of p=0.05 with the null hypothesis that there is no performance difference between the accuracy of the raw KAPD dataset and the accuracy of the normalized dataset using the same corpus.

For comparison between the ESN performance and the performance of the RBM Deep NN for the same database (the KAPD) see Table 7.2.

Table 7.2 - A comparison between the overall average test dataset performance of the ESN and of the RBM Deep NN for the KAPD testing dataset.

Neural Network type	**Database type**	**Performance**
RBM DNN	Normalized dataset	49.85%
Conventional ESN	Raw dataset	**73.40%**

It is clear from Table 7.2 that the frame recognition performance of Arabic phonemes recognition using the KAPD were much improved (more than 23% difference) using the ESN with supervised learning model compared with the performance of the Arabic phoneme recognition when using the DNN model. The results clearly show that the ESN model has a better ability for recognizing phonemes sequences than the DNN model for the small vocabulary size dataset. A two-tailed Matched Pairs T-test significance test was conducted with the null-hypothesis that there is no performance difference between the DNN model and the ESN model trained on the KAPD dataset. The test finds a significant difference at the level of p=0.05.

7.2.1.1.2 MFCC's features verses LPC features using supervised learning

In order to ensure that the performance of the ESN system was optimal, it was decided to perform an experiment to compare the performances of the ESN with supervised learning algorithm when trained on the KAPD Arabic phoneme datasets using both LPC and MFCC feature extraction techniques.

Table 7.3 - A comparison between the KAPD testing dataset performance of the ESN with supervised learning algorithm when using 39 MFCC and 12 LPC features.

Feature extraction technique	Performance
MFCC	**73.40 %**
LPC	66.75%

It is clear from Table 7.3 that the ESN can correctly recognize more of the KAPD Arabic frames with the MFCCs feature extraction technique with an average performance of 73.40% frame recognition on the testing dataset. In contrast, only 66.75% average frame recognition performance was obtained when using LPC's features with the same ESN learning algorithm. Therefore, the MFCC technique is confirmed as the best format for extracting features for Arabic speech recognition – all further experiments used the MFCC extraction algorithm as the input feature extraction stage.

7.2.1.2 Experiment on the CSLU2002 database

In order to investigate Multi-dialect Arabic speech recognition, the ESN network architecture with supervised learning algorithm, described in Figure 7.1 with the parameters presented in Table 7.1, was also investigated using the CSLU2002 database for 34 female and male speakers. This dataset includes speakers from 12 Arabic countries. The average phoneme recognition results for this dataset are presented in Figure 7.6.

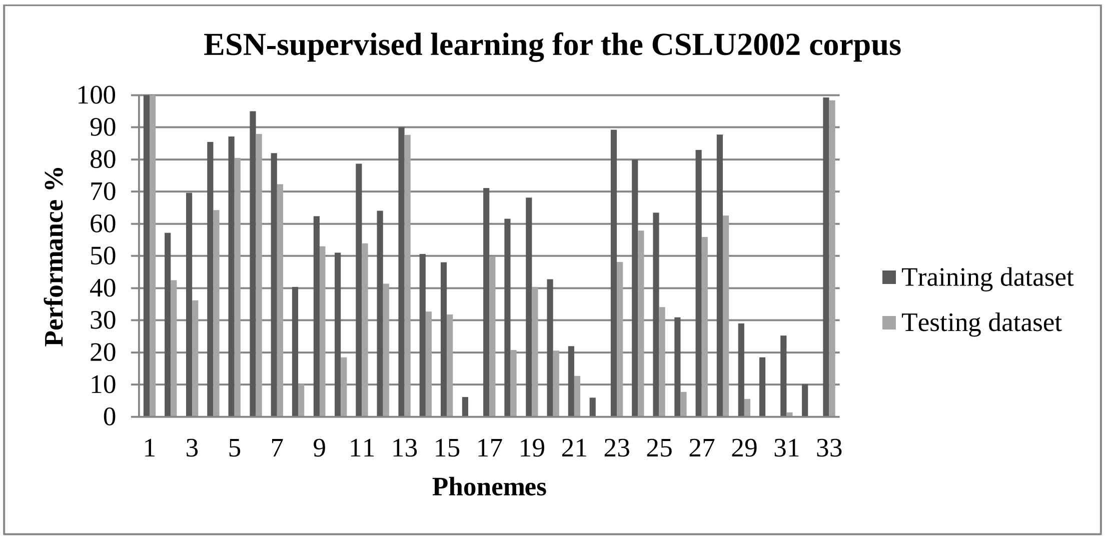


Figure 7.6 - The average results of phonemes for female speakers' dataset when it trained and tested on the ESN with supervised learning.

As can be seen from Figure 7.6, for the training dataset, the average phoneme recognition result was 69.52%. The ESN with supervised learning algorithm can recognize most of the training dataset of CSLU2002 corpus with performance above 60%. On the other hand, the testing dataset has an average phoneme recognition performance of **51.72%**.

7.2.2 An ESN with forced supervised learning algorithm

For standard forced supervised training, the activations of the reservoir and output neurons are calculated using Equations (7.12) and (7.13):

$$x(t+1) = tanh\left(W^{in}u(t+1) + Wx(t) + W^{back}\hat{y}(t)\right) + leftover \qquad (7.15)$$

where $x(t+1)$ is the reservoir state for time step $(t+1)$, $u(t+1)$ is the input vector, $\hat{y}(t)$ is the desired output vector (target output vector) for time step t. The $leftover$ was calculated using Equation (7.13), and $\hat{y}(t+1)$ the calculated output states for time step t using Equation 7.14

When initializing the ESN network hyper-parameters, the initial weight vectors W^{in}, W, W^{back} and, W^{out} are randomly generated as described previously using the connections shown in Figure 7.7. The main parameters used are as described in Table 7.1.

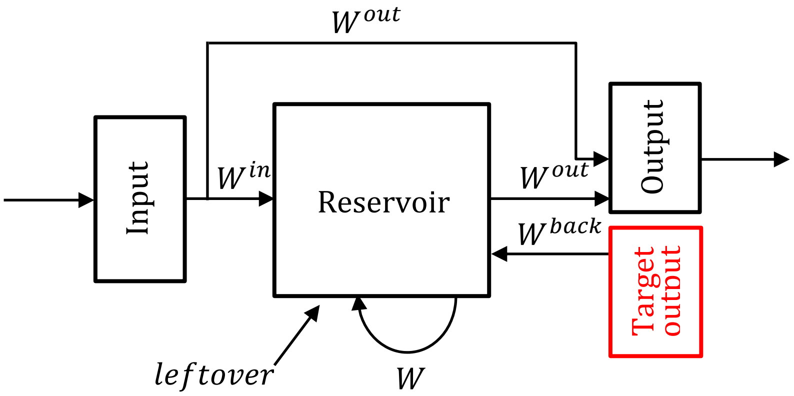


Figure 7.7 - ESN connections with forced supervised learning algorithm.

7.2.2.1 Experiment on the CSLU2002 dataset using forced supervised learning ESN

The 34 CSLU2002 speakers dataset was used to train and test an ESN using a forced supervised learning algorithm. Firstly, each speaker was tested on a separate (speaker dependant) ESN with forced supervised learning algorithm are described in Figure 7.7 and Table 7.1. Figure 7.8 shows the average results of the Arabic phonemes for each speaker. The complete CSLU2002 phonemes dataset was then trained and tested on one speaker independent ESN using the forced supervised learning algorithm. The average phonemes recognition results for the speaker independent system is presented in Figure 7.9.

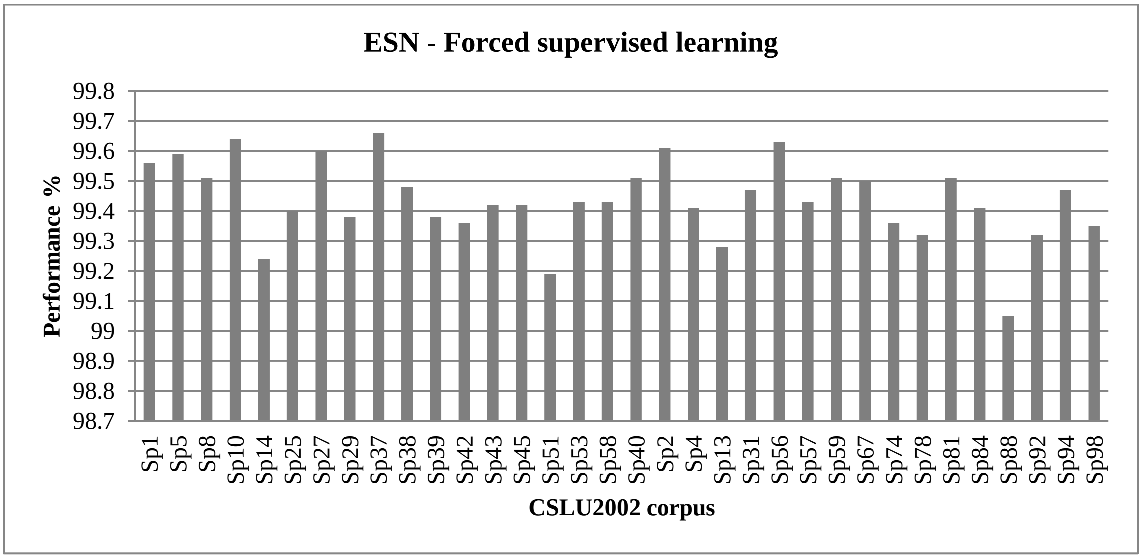


Figure 7.8 - The average train dataset results of 34 speakers of the CSLU2002 database trained and tested on individual ESN with forced supervised learning algorithm.

As is clear from Figure 7.8, that the average phoneme classification results for each speaker were more than 99% when using individual ESN classifier trained with a forced supervised learning algorithm on the CSLU2002 corpus. Essentially an ESN trained with a forced supervised algorithm can be trained perfectly for phonemes classification. However, forced supervised learning is not suitable for general purpose speech recognition problem as the reservoir calculation depends on the target or desired output. In reality, this target output is not available in the testing stage for a speech recognition problem.

7.2.3 Unsupervised Adaptation (Modified ESN with a novel combined supervised/forced supervised learning algorithm)

Mismatches between the training model conditions and the input conditions may occur due to differences between speakers, environmental noise, and differences in channels. Many adaptation approaches have been suggested to compensate for these variabilities. Maximum likelihood linear regression (MLLR) and Maximum a posteriori (MAP) are the most popular adaptation methods for HMMs. The idea behind MLLR adaptation approach is to apply a set of linear transformation matrixes to transform the learned model parameters and to maximize the likelihood on the test data under the new model (Gales, 1998, Leggetter and Woodland, 1995). This adaption approach is practically used to transform the model means and covariances in order to improve the system performance for each test speaker. The MAP adaptation approach is used in statistical modeling and requires prior knowledge about the distribution of the model parameters for robust estimating of model parameters more than the EM estimation for small amount of data.

There are two modes of adaptation, supervised and unsupervised. In supervised adaptation, an adaptation-data transcript is required during the estimation of the approach such as in the MLLR adaptation approach. However, in an unsupervised adaptation mode, the adaptation-data transcript is provided before the transform is estimated. Speaker adaptation can also be done using on-line adaptation or batch adaptation (Zavaliagkos et al., 1995). On-line adaptation is done after obtaining one utterance, whereas batch adaptation is done after collecting all the available utterances.

Feed forward Neural networks such as MLPs have also used different adaptations approaches for speaker adaptation in an attempt to minimize test errors on the adaptation data. For example, speaker identity vectors (i-vectors) (Dehak et al., 2011) have been used to extract low-dimensional representations of the speaker features and are used as an additional input feature to the network with the regular acoustic features for ASR as is adopted in (Saon et al., 2013, Miao et al., 2014).

All the adaptation approaches mentioned above, attempt to deal with the poor fit between the training model conditions and the input conditions that may occur due to differences between speakers and for environmental noises. However, in this work we investigate an adaptation that is intended to minimize classification errors. Similar to the i-vector approach, this adaptation approach can be completely extracted in an unsupervised way.

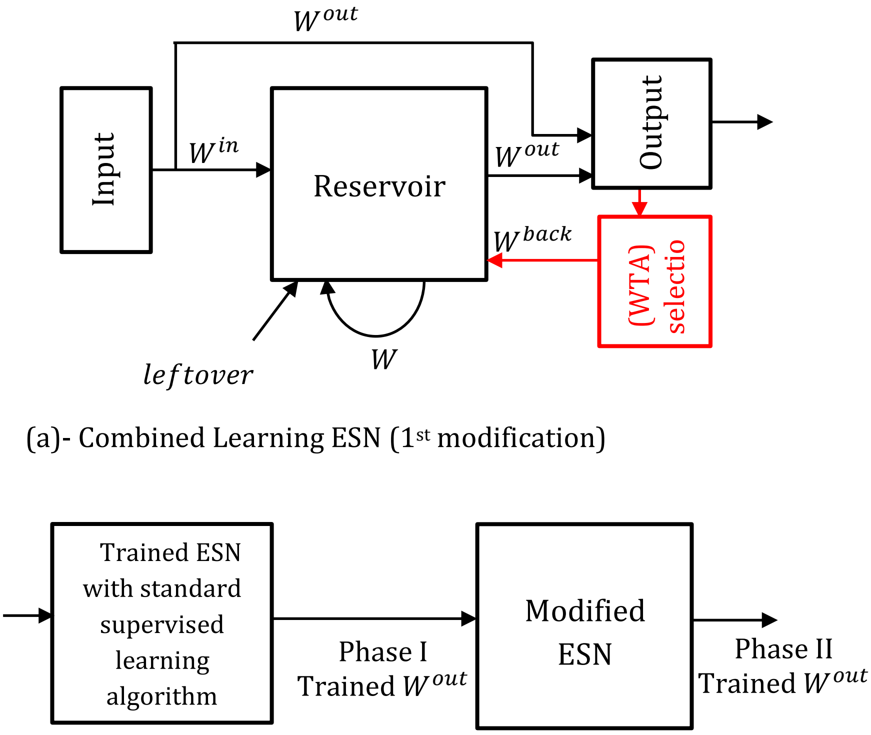
As mentioned previously, an ESN trained with the standard forced supervised learning algorithm is not suitable for general purpose speech recognition purposes because the target output would not be available for calculating the reservoir states during the testing stage. To address this shortcoming, a combined supervised/forced supervised learning was implemented (Unsupervised Adaption). In this novel algorithm, the calculated network outputs are passed through a winner-take-all (WTA) selection during training. This converts the output y(t) into a binary vector to effectively produce binary teacher outputs before being fed back into the reservoir (see Figure. 7.9). This adaptation is applied only on the hidden-to-output layer. The target outputs are thus used as feedback connections with binary values zero or one, whilst the calculated outputs are floating point numbers between -1 and 1 as a result of the activation function *tanh*. During testing, all the outputs are converted from floating point numbers into zeros or ones using the winner-take-all (WTA) selection algorithm.

For the modified ESN with supervised/forced supervised training, the activations of the reservoir and output neurons are calculated using

$$x(t+1) = tanh\left(W^{in}u(t+1) + Wx(t) + W^{back}\hat{\hat{y}}(t)\right) + leftover \qquad (7.17)$$

where $x(t+1)$ and $u(t+1)$ are the reservoir state and the input vector respectively, and the $leftover$ was calculated using Equation (7.13). The Initial weight vectors

W^{in},W, and W^{back} are randomly initialised and the output matrix W^{out} is updated using the ESN offline training algorithm, as in Equation 7.7. $\hat{\hat{y}}(t)$ is the modified calculated output vector for time step t and is calculated using Equation 7.14 , where $\hat{\hat{y}}(t)$ is the winner-take-all (WTA) selection of the calculated output vector $y(t)$ for time step t.


W^{out}
Input
W^{in}
Reservoir
W^{out}
Output
W^{back}
(WTA) selectio
$leftover$
W
Trained ESN with standard supervised learning algorithm
Phase I Trained W^{out}
Modified ESN
Phase II Trained W^{out}

(a)- Combined Learning ESN (1st modification)

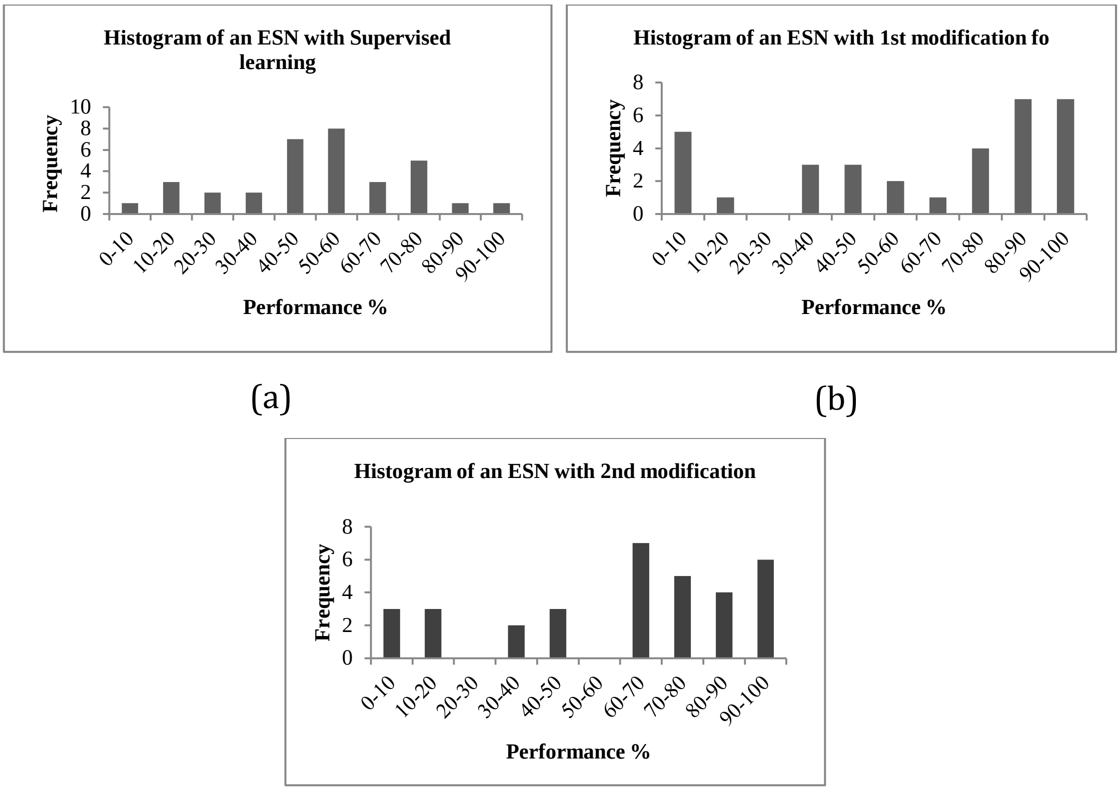
(b) – Deep combined supervised/forced supervised ESN (DESN) (2nd modification)

Figure 7.9 – (a)- Combined Learning ESN (1st modification). (b)- Deep combined supervised/forced supervised ESN (DESN) (2nd modification).

For deep ESN model, the standard supervised learning algorithm was first used to train the conventional ESN until optimum weights are obtained. Training was then resumed using the modified forced learning algorithm described by Equations 7.14 and 7.17. This 2nd phase modification significantly enhanced the speech recognition performance for the multi-dialect CSLU2002 training dataset (see Table 7.4).

7.2.3.1 Experiments on the CSLU2002 corpus using the Modified ESN

Two different feature vector dimensionalities were used to represent the CSLU2002 corpus. The basic 13 coefficients MFCCs and the extended 39 coefficients MFCCs feature vector datasets were extracted for training and testing these datasets were then applied to three different ESN learning algorithms namely, conventional ESN, ESN with the Adaption (1st modification) and deep learning ESN (2nd modification). Firstly, the results of the 13 MFCCs dataset are compared together in Figure 7.10.



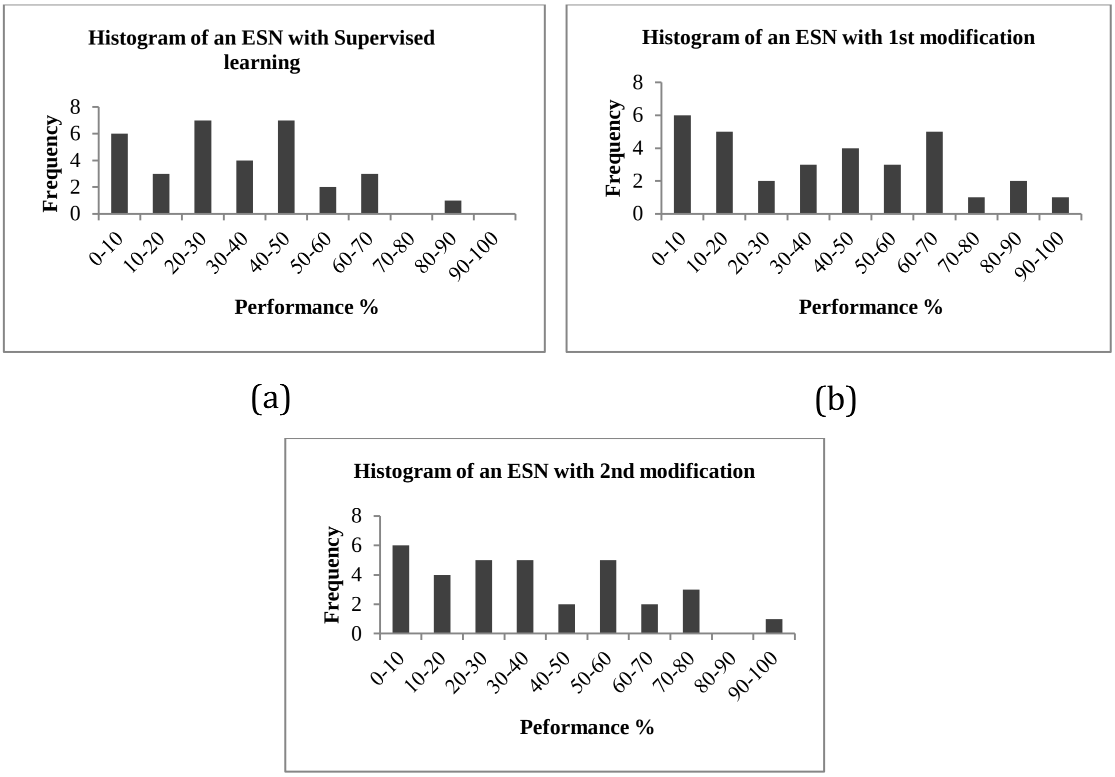
(a) (b)

(c)

Figure 7.10 - A histogram used for a comparison between the average results of ESN with different learning algorithms for the CSLU2002 training dataset with 13 MFCCs coefficients.

The main trend visible from the histogram of the training results is that the 1st modification greatly improved the performance on the training dataset compared to the supervised learning algorithm. The average performance results of the 33 Arabic phonemes of the CSLU2002 training dataset using this modification improved from 51.28% to 60.50%. On the other hand, the 2nd phase modified for the ESN had a slight effect on the recognition performance compared to the 1st modification on the ESN system. Only an additional 1.1% an improvement was obtained on the average recognition performance of the 33 Arabic phonemes. However, about 66.6% of the 33 Arabic phonemes were recognised with performance more than 60% using the deep learning (2nd modification).

Figure 7.11 shows the histogram of the testing dataset results.



(a) (b)

(c)

Figure 7.11 - A histogram used for a comparison between the average testing dataset results of ESN with different learning algorithms for CSLU2002 testing dataset and 13 MFCCs coefficient.

It is clear from Figure 7.11 that the average phoneme recognition results of the CSLU2002 testing dataset was improved using the adaptation (1st modification), where the average results of the recognition performance was enhanced by 4.7%. There is a significant difference in performance between the the baseline conventional ESN and the adaption (1st modification) results found when using a two-tailed Matched Pairs T-test significance test. However, the 2nd modification for the ESN did not improve the recognition performance for the CSLU2002 testing dataset.

The results with the same CSLU2002 corpus using the extended 39 coefficients MFCCs feature datasets with the three different ESN learning algorithms are compared together and with the 13 MFCCs features in Table 7.4.

Table 7.4 - A comparison between the CSLU2002 training and testing datasets framewise phoneme performance uses the ESN with supervised learning, combined supervised learning with adaption (1st modification) and deep combined supervised/forced supervised learning algorithms with 13 MFCC features and 39 MFCC features.

Network inputs	**Learning technique**	**Performance%**	
		Train	**Test**
13 MFCC coefficients	Supervised learning	51.28	33.50
	Combined Supervised /forced supervised (adaption)	60.50	**38.20**
	Deep Combined supervised/forced supervised	**61.06**	35.80
39 MFCC coefficients	Supervised learning	69.52	51.72
	Combined Supervised /forced supervised (adaption)	70.38	51.87
	Deep Combined supervised/forced supervised	**70.83**	**52.15**

It is obvious from Table 7.4 that the baseline ESN network with 39 MFCCs features vectors has the best overall framewise phoneme recognition performance. The performance of the 39 MFCCs features vector ESN was about 16.35% better than the baseline 13 MFCCs ESN system. The recognition performances on the CSLU2002 dataset was also considerably improved using the first and second modification for the supervised ESN when compared to the baseline ESN with the 13 MFCCs features. However, these modifications on the supervised ESN did not show comparable improvements over the baseline ESN performance with the 39 MFCCs features, where the performance results for the CSLU2002 dataset was increased by only about 1.31% for the training dataset and by only 0.43% for testing dataset. There is no significant difference between the baseline ESN system and the 1st and 2nd modified ESN trained on the CSLU2002 corpus with 39 MFCCs features using a the two-tailed Matched Pairs T-test significance test conducted with the null hypothesis that there is no performance difference between the conventional supervised ESN system and the 2nd and modified ESN system. The test failed to find differences at the level of p=0.05.

7.2.3.2 A comparison between the performance of the ESN and other related works on the CSLU2002 dataset

The results of the experiments of these systems cannot be directly compared to many related work found in the literature as these KAPD and CSLU2002 phoneme dataset are corpora that were manually segmented and labelled by the author. There is only one closely-related work (Alalshekmubarak, 2014). We collaborated with them and offered our CSLU2002 Arabic phonemes database to them in order to allow us to make a valid direct comparison between the systems. Thus, to evaluate the performance of the systems conducted on the CSLU2002 dataset, we compared our results to the work of (Alalshekmubarak, 2014) . They maintained the same reservoir size that we used in our work to ensure a valid comparison between the developed systems. This comparison is presented in Table 7.5, and the results show the superior performance of our developed system.

Table 7.5 - A direct comparison between the overall average test dataset performance of the supervised ESN and other related works on the CSLU2002 testing dataset.

Neural Network type	Performance	Reference
Combined Learning ESN- 13MFCCs	38.20 %	(Hmad and Allen, 2013)
ESN -39 PLP	44.67 %	(Alalshekmubarak, 2014)
RBM DNN-39 MFCCs	42.53%	This work Chapter 6
Deep Combined Learning ESN-39 MFCCs	**52.15%**	This work Chapter 7

The results, summarised in Table 7.5, show the performance of four models' experiments that were conducted on the CSLU2002 corpus trained on the two classification approaches, namely, DNNs and ESNs. Our developed system (combined learning ESN-based) shows superior performance when it is compared to the other systems recently reported in the literature that used the same CSLU2002 selected corpus. Generally, the ESN system outperformed the RBM DNN when using the extended 39 MFCCs feature vector as input. This promotes the adopting of ESN for Arabic speech recognition systems.

A significance test was conducted between the base line DNN's result and the ESN with combined learning result. The two-tailed Matched Pairs T-test significance test finds a significant difference at the level of p=0.05 that was conducted with the null hypothesis that there is no performance difference between the framewise phoneme recognition between the base line DNN's result and the ESN with combined learning result.

7.3 ESN acoustic modeling experimental results on the Levantine corpus

In order to obtain statistically relevant large vocabulary dataset results, the Levantine Arabic Corpus was also used to explore the potential gain in building an acoustic modelling Arabic speech recognition system using the ESNs model. This also allows a valid comparison to be made between our work and other developed systems presented in the literature.

7.3.1 Hyper-parameters Optimisation

The same selected dataset as described in Chapter five and used in Chapter Six for training and testing the RBM DNN acoustic models is also used in this experiment. The experiments on this corpus also used the same hyper-parameters values shown in Table 7.1. As the size of this dataset is huge, the online training, described in Section 7.1.3.2, is used for training with mini-batch size of 500 for stochastic gradient descent (SGD).

7.3.2 A comparison between the ESNs and DNNs acoustic models performances on the Levantine corpus

The main purpose of this experiment is to a make a direct comparison between the ESNs and the RBM DNN acoustic model performances. For the ESN experiments conducted on the Levantine corpus, we used a separation line after each utterance to represent the sequences that are fed to the ESN network. Table 7.6 shows results of the DNNs system performance presented in Chapter Six without the post process and the ESNs system performance trained on the same Levantine dataset for Arabic speech recognition.

Table 7.6 - A comparison between the average framewise phoneme recognition accuracies using the ESNs and DNNs acoustic models on the Levantine core test.

Model name	**Framewise phoneme recognition%**
DNNs-5frames context window	28.80
ESNs	**42.81**

It is obvious from Table 7.6 that a significant improvement on the recognition performance was achieved when using the ESN model compared to the baseline RBM DNN model's result. This was confirmed with a two-tailed Matched Pairs T-test significance test. The adoption of the ESNs model for acoustic modeling is shown to be more suitable than the adoption of the DNNs model for acoustic modeling speech recognition. This is because ESNs are recurrent models and support sequence models more than the RBM DNN models even with a contextual input window.

7.4 Comparison of ESN results with literature equivalents

Table 7.6 shows a comparison between the best performances achieved by the systems investigated in this project and systems in relative equivalent projects presented in the literature.

Table 7.7 - A comparison between the best performances were achieved for the proposed corpora of this project with relatively comparable projects.

Model name	No of Dialect	Database	Database Type (# of classes)	Training set	Validation set	Testing set	No of speakers	% Accuracy	Reference
				Phonemes					
ESN	Multi Arabic dialect	The CSLU2002 database	33	1894	1894	1908	34	52.15	This work
ESN	One - Saudi Arabia dialects	The KAPD Database	33	2376	2376	1970	6	73.40	This work
ESN	Multi Arabic dialect	The CSLU2002 database	33	1420	474	1908	34	44.67	(Alalshekmubarak , 2014)
RNN	One Arabia dialect	Self-created database	20	800	-	400	6	82.3	(Ismail and Ahmad, 2004)
hierarchical MLP	One - Iraq Arabia dialect	Self-created database	33	396	-	396	1	68.18	(Ali and Hwaidy, 2007)
MLP	One-Algerian dialect	Self-collected database	28	-	-	7393	6	68	(Selouani and Caelen, 1999)
HMM	Moroccan dialect	Self-created database	28	2240	-	560	10	87.66	(Hachkar et al., 2011)
ESN	Multi Arabic dialects	13 hours Levantine corpus	33	**Hours**			120	42.81	This work
				9.75	3.25	2.24			

It is obvious from the results in Table 7.7 that use of ESN models can improve the recognition performance of Arabic phoneme recognition systems. This finding is supported by comparing the ESN model results with the RBM DNN baseline model results found in Table 7.5 and Table 7.6. The performance of ESN model is very promising and encourages researchers to adopt this model for various applications. The most challenging issue for state-of-the-art speech recognition is the amount of data that is need for training any model. ESN models can provide good performance when only a limited size of corpus is available. ESNs are also simple with less computation compared to other NNs (MLP, RNN) techniques. Thus, training time is very competitive compared with other approaches.

7.5 Conclusion

The aim of this Chapter was to investigate Echo State Networks (ESNs) for phoneme classification and acoustic modelling. In order to achieve this aim, we have explored different ESN architectures.

In this chapter, we adopted ESNs for Arabic speech recognition. The KAPD Saudi Arabia dialect corpus and the CSLU2002 multi-dialect corpus were used to train and test an ESN with supervised and forced supervised learning algorithms. Furthermore, a novel combined supervised/forced supervised learning algorithm (unsupervised adaptation) was developed and tested on an optimised Arabic phoneme recognition datasets. In addition, the Levantine Arabic dataset is also used to train the mentioned systems as acoustic models.

We also investigated an unsupervised ESN adaptation approach to minimize classification errors. The main idea was to use simi-target outputs that act as a forced teacher for supervised training. Thus, a combined supervised/forced supervised learning was implemented (Unsupervised Adaption) to effectively produce teacher outputs before being fed back into the reservoir. We discussed the findings of the conducted experiments using this adaptation with the CSLU2002 corpus. The baseline performance of the 13 MFCCs input ESN was firstly explored and the adaptation approach was found to greatly improve the recognition performance. The performance 39 MFCCs feature vector ESN system was an about 16.35% better than the baseline 13 MFCCs. However, only slight improvement was gained when using this adaption with 39 MFCCs feature vectors. This was statistically not significant difference in the performance. A trivial enhancement on the recognition performance of the core testing dataset was obtained when we used deep ESN architecture for both type of features.

The results of the experiments of these systems cannot be directly compared to many related work found in the literature as these KAPD and CSLU2002 phoneme dataset are corpora that were manually segmented and labelled by the author. However, one closely-related work has been provided (Alalshekmubarak, 2014) that used our CSLU2002 Arabic phonemes database in order to allow us to make a valid direct comparison between the systems. We also compared the performance of four NN based models, conducted on the CSLU2002 corpus, with two classification approaches, namely, RBM DNNs and the ESNs. Our developed system (combined learning ESN-based) shows superior performance with 52.15% framewise phoneme recognition performance when compared to the other systems recently reported in the literature using the same CSLU2002 selected corpus. Generally, ESN outperformed the DNN under the considered the-state-of-the-art the 39 MFCCs features. This promotes the adopting of ESN in Arabic speech recognition systems.

The Levantine Arabic Corpus was also proposed in this Chapter to explore the potential gain in building large vocabulary acoustic models for Arabic speech recognition system using the ESNs model. A significant improvement on the recognition performance was achieved when the ESN model was implemented compared to the baseline RBM DNN model's result. The adoption of the ESNs model for acoustic modeling is seen to be more valid than the adoption of the DNNs model for acoustic modeling speech recognition. This is because ESNs are recurrent models and support sequence models better than the RBM DNN models even with the contextual input window.

Finally, comparisons between the best performances achieved by the systems investigated in this thesis and systems in relative equivalent projects presented in the literature used relatively equivalent models and different Arabic corpora that found in the literature were presented. The ESN model is shown to give improved recognition performance as an Arabic phoneme recognition system. The performance of ESN model is very promising and encourages researchers to adopt this model for various applications. ESN models can also provide good performance when only a limited size of corpus is available. ESNs are also simple with less computation compared to other NNs (MLP, RNN). Thus, training time is very competitive compared with other approaches.

Comparisons between the proposed models with others used the proposed corpora in the literature were not possible as the differences in the corpora size and type. Thus, we purposed to use well known corpus even that it is not Arabic to evaluate the proposed models and to make a valid comparison with other published work on the same corpus. We used The TIMIT English corpus for this goal in the next chapter.

Chapter Eight

Evaluation of the Deep Neural Networks (DNNs) and Echo State Networks (ESNs) on TIMIT corpus for acoustic modelling

Deep Neural Networks (DNNs) have been successfully applied for acoustic modelling of English language (Mohamed et al., 2009, Dahl et al., 2012, Hinton et al., 2012, Mohamed et al., 2012). Echo State Networks (ESNs) have also recently been applied to various applications with impressive success. ESNs models have proven to be highly successful in training the acoustic modelling of speech which is considered as a key component of most state-of-the-art speech recognition systems. This success encouraged us to be the first to attempt to investigate these types of networks for developing acoustic models for an Arabic speech recognition system. However, the most published work on the proposed Arabic corpora were various in the corpora types and sizes. Hence, it was very difficult to make a direct comparison on different corpora types and sizes to our work using DNNs and ESNs on the proposed Arabic corpora. Thus, we suggested evaluating DNNs and the ESNs models on the benchmark task of framewise phoneme classification and acoustic models using the TIMIT database, in order to make a reasonable and direct comparison between these models.

The aim of this Chapter is therefore to explore deep learning for framewise phoneme classification and acoustic modelling using Deep Neural Networks (DNNs) and Echo State Networks (ESNs) on TIMIT corpus. In order to achieve this aim, we have explored different experiments based on the work of Deep Neural Networks and on the work of the Echo State Networks (ESNs) introduced in Chapter Six and Chapter Seven respectively. This Chapter is organised as follows: in Section 8.1 the TIMIT database and the experimental setup used in this Chapter are described. In this section, TIMIT database, which is used for the experiments of this chapter, is first described. This is followed by Deep Neural Networks (DNNs) and the Echo State Network Experiments for phoneme classification and acoustic models and their results on the proposed corpus. A conclusion is finally drawn in Section 8.2.

8.1 Experimental Setup

In this Section the TIMIT[8] database and the experimental setup used in this Chapter are described.

8.1.1 TIMIT corpus

TIMIT corpus (Garofolo et al., 1993) is used in this thesis for evaluating the DNN and the combined learning ESN model for phone recognition and acoustic modelling experiments. The utterances of the corpus are spoken by speakers from a wide variety of American dialects. We used 3696 utterances from 462 speakers (3.14 hours) for training and 1344 utterances from 24 speakers (0.81 hours) as defined in (Mohamed, 2014) for testing. The sentences of this corpus are divided into three types: phonetically-compact (SX), phonetically-diverse (SI), and dialect (SA). The (SA) sentences were designed to detect the dialectical variation of the speakers, we removed all SA records for all speakers in the dataset similar to (Mohamed, 2014) as they are identical and could bias the results. More details for TIMIT corpus are in (Halberstadt, 1998).

We analysed the speech signal using a 25ms Hamming window and 10ms frame rate. The speech was represented by the 13 MFCCs and their first and second derivatives. We used 186 target class labels for the 62 phones in the corpus, i.e., 3 states for each phone. After decoding, the 62 phone classes were mapped into a set of 44 classes similar to (Graves and Schmidhuber, 2005) by making several identifications as following:

- The closures ‘dcl’, ‘bcl’, ‘gcl’, ‘kcl’, ‘pcl’,‘tck’ and ‘tcl’ were identified with the following stops (‘d’, ‘b’, ‘g’, ‘k’, ‘p’, ‘t’ and ‘ch’, respectively).
- The silence markers ‘sil’, ‘pau’, and ‘h#’ were considered equivalent and mapped to ‘sil’.

[8] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1.

- The flaps ‘nx’ and ‘dx’ were identified with ‘n’ and ‘d’ respectively.
- The nasals ‘eng’, ‘en’ and ‘em’ were identified with ‘ng’, ‘n’ and ‘m’.
- The semivowel/glides ‘hv’ and ‘el’ were identified with ‘hh’ and ‘l’.
- The vowels ‘ax-h’, ‘ux’, and ‘ix’ were identified with ‘ax’, ‘uw’, and ‘ih’.

8.1.2 Deep Neural Networks (DNNs) for phoneme classification and acoustic models Experiments

The aim of these experiments is to explore Deep Neural Networks, introduced in Chapter six, for framewise English phoneme classification and acoustic modelling using the TIMIT dataset. In order to achieve this aim, we have explored different network architectures based on the deep neural network. The 39 MFCCs feature vectors that were used for the experiments in this Section are normalized using Equation (6.5).

8.1.2.1 RBM DNN Hyper-parameters Optimisation and experiment’s result

For the developed systems in this section, all conducted experiments followed a standard procedure. The stochastic gradient decent algorithm was used for pre-training phase with a mini-batch size of 100 training cases using learning rate 0.1, weight cost 0.0002 and momentum 0.9, these values are experimentally chosen. For the structure of the RBMs in the pre training stage, we used three Gaussian RBMs, and trained them using 100 epochs for the Gaussian RBMs and 0.1 for a learning rate with different number of nodes in each RBM layer. In the fine-tune stage, we used 5000 epochs due to the hardware availability. The inputs were decided to be 5frame input window based on the optimisation experiments that were conducted in Sections 6.2.2 and 6.2.3. However the structure of the network is slightly different due to the size of this dataset and the hardware availability. For this experiment we used 60, 40, and 60 nodes in the hidden layers. These numbers of nodes were experimentally chosen within limit number of nodes in each layer based on the hardware availability.

The baseline experiment was conducted for the TIMIT corpus using 5frames input windows that fed to the DNN network for framewise phoneme recognition. The result obtained for the system structure mentioned above is 42.79% framewise phoneme recognition.

8.1.3 Echo State Networks (ESNs) for phoneme classification and acoustic models experiments

This section describes the adoption of the Echo State Networks (ESNs) models for framewise English phoneme classification and acoustic modelling on the TIMIT dataset. The raw 39 MFCCs feature vectors were used for the experiments in this Section rather than the normalized featured based on the finding of the experiment in Section 7.2.1.1.1.

8.1.3.1 ESN Hyper-parameters Optimisation and Experiment's Result

The TIMIT dataset is also used in this experiment for evaluating the ESN acoustic modeling on the testing dataset to ensure a valid comparison of this acoustic modelling speech recognition system with other developed systems' published work (Graves A., & Schmidhuber J., 2005), (Chen and Jamieson, 1996), (Robinson, 1994), (Schuster, 1999). The experiments on this corpus used the hyper-parameters values that are shown in Table 7.1. Online training (described in Section 7.1.3.2) is used for training with mini-batch size of 500 for stochastic gradient descent (SGD). The result obtained for the system structure mentioned above is 62.74%.

8.1.4 A comparison between the ESNs and DNNs acoustic models performances on the TIMIT corpus

A direct comparison between the ESNs and the RBM DNN acoustic model performances presented in Section 8.1.2 and results of the ESN system performance presented in Section 8.1.3 without the post process is shown in Table 8.1. These systems trained on the same TIMIT dataset for speech recognition.

Table 8.1 - A comparison between the framewise phoneme recognition accuracies using the ESN and DNN acoustic models on the TIMIT core test.

Model name	**Framewise phoneme recognition%**
DNN-5frames context window	**42.79**
ESN	**59.74**

Our main findings are that ESN network outperform time-windowed RBM DNN ones. It is obvious from Table 8.1 that a significant improvement on the recognition performance was achieved when using the ESN model compared to the baseline RBM DNN model's result. This was confirmed with a two-tailed Matched Pairs T-test significance test that conducted with the null hypothesis that there is no performance difference between the framewise phoneme recognition between the base line DNN's result and the ESN result. This significance test finds a significant difference at the level of p=0.05. ESN is more accurate and also much faster than the time-windowed RBM DNN. This result confirmed the finding of the experiments of ESN on Arabic corpora presented in Chapters six and seven that ESN support sequence models more than the RBM DNN models even with a contextual input window.

8.1.5 A comparison between the performance of the ESN and other related works on the TIMT corpus

In this section, the best performances achieved by the systems investigated in this thesis are directly compared to closely- related published works in equivalent projects based on framewise phoneme recognition in all cases, the same corpus is used. This comparison is presented in Table 8.2.

Table 8.2 - A direct comparison between the best performances achieved by various Neural Networks types found in the literature.

Neural Network type	Training set	Validation set	Testing set	Performance on test set%	Reference
	utterances				
BLSTM (retrained)	3512	184	1344	70.20	(Graves and Schmidhuber, 2005)
RNN	3696	-	1344	74.20	(Chen and Jamieson, 1996)
RNN	3696	-	1344	65.30	(Robinson, 1994)
BRNN	3696	-	1344	65.10	(Schuster, 1999)
ESN-39 MFCCs	2772	924	1344	59.74	This work Chapter 8

The results, summarised in Table 8.2, show the framewise phoneme recognition performance of the most published works conducted on the TIMIT corpus and our best performance when it was trained on the ESNs. Our developed system ESN-based shows relatively lower performance when it is compared to the other systems recently reported in the literature that used the same corpus. This may be due to a number of reasons. First of all, increasing the reservoir size can increase the results. However, increasing the reservoir size increases the calculation and memory requirement. This completely dependent on the hardware availability that was relatively is not available for our research. In addition, speaker and noise adaption was not used to improve the results in this thesis as our aim is to investigate the proposed models for speech recognition and to make a direct comparison between these models.

8.2 Conclusion

The aim of this Chapter is therefore to explore deep learning for framewise phoneme classification and acoustic modelling using Deep Neural Networks (DNNs) and Echo State Networks (ESNs) on TIMIT corpus. In order to achieve this aim, we had explored different experiments to develop Deep Neural Network and Echo State Networks (ESNs) models introduced in Chapter Six and Chapter Seven respectively for speech recognition.

In this section, we discuss the findings of the conducted experiments using the proposed models through the TIMIT corpus. The performance of the ESN model outperformed the baseline RBM DNN model. ESN was found to be about 59.74% better than the base system trained with input window of 5frames. Statistically there was significant difference in performance between the two models.

Finally, comparisons between the best performances achieved by the systems investigated in this thesis and the published works in equivalent projects based on framewise phoneme recognition found in the literature and used the TIMIT corpus were presented. Our developed system ESN-based shows lower performance when it was compared to the other systems recently reported in the literature that used the same corpus. This due to the hardware availability and not applying speaker and noise adaption that can improve the results in this thesis as our aim is to investigate the proposed models for speech recognition and to make a direct comparison between these models.

Chapter Nine
Conclusion and Future Work

In this thesis we investigate phoneme and acoustic models based on Deep Neural Networks (DNN) and a Deep Echo State Networks for multi-dialect Arabic Speech Recognition. The TIMIT corpus with a wide variety of American dialects was also used to evaluate the proposed models.

9.1 Summary of Accomplishments

- **Arabic databases**

The phoneme is considered the basic unit of speech in much of speech processing. The phonemic representation of a given word is used in most speech recognizers to identify it. Thus, the availability of speech data that is time-aligned and labelled at phonemic level is a fundamental requirement for building speech recognition systems. In this thesis, we describe the segmentation and labelling of three Arabic corpora that were used to investigate the thesis's aim. In order for the accomplishment of this stage, a developed Arabic phoneme database (APD) was manually timed and phonetically labelled. This dataset was constructed from the KAPD database for Saudi Arabia dialect and the Centre for Spoken Language Understanding (CSLU2002) database for different Arabic dialects. This dataset covers 8148 Arabic phonemes. In addition, a corpus of 13 hours of Arabic speech randomly selected from the Levantine Arabic dialect database that is used for training and 2.4 hours for testing are revised and transcription errors were manually corrected. The selected dataset is labelled automatically using the HTK Toolkit. In order to achieve automatic alignment and labelling, a set of well-defined rules for phonetizing a written text was used to solve the pronunciation ambiguities in the text and to produce accurate phonemic transcriptions. Then a pronunciation dictionary was constructed to map words into one or more phonemic pronunciation variants. Multi pronunciations per word were typically used for training and testing (decoding) in Automatic Speech Recognition systems. So, the Arabic phonemic lexicon was built and the missed pronunciations were added manually. A total of 59k words were subsequently to create this phonemic dictionary and more than 4k words were added manually. In addition, timings suggested by the corpus are used to segment the audio data, and 13 Mel Frequency Cepstral Coefficients (MFCCs) features plus their first and second derivatives were extracted using 10 ms using a windowed speech signal of 25 ms., each conversation side was normalized using cepstral mean and variance normalisation as well as vocal tract length normalisation (VTLN). The Maximum Likelihood function was used to train all the models with context-dependence triphones. Each phone was modelled using left-to-right HMM with three states. A binary decision tree with phonologically phonetic questions were used to cluster the HMM states. A single Gaussian mixture component for each state with a total of 2083 clustered states was used. The HTK Hidden Markov Model toolkit was utilized to perform the automatic labelling using forced alignment mode. The labelled Arabic dataset thus produced to be used for acoustic models experiments.

- **Arabic speech segmentation and features extraction**

The Entropy-based algorithm was applied as an automatic speech segmentation process. The results of using this technique on 30 sentences from 9 speakers from the KAPD and the CSLU2002 databases with different rates of speech (ROS) showed that the best word segmentation accuracy, obtained when applying the pre-emphasis and low pass filter together on the speech signals, gave an error rate 32.95%. This error would be added to any errors in the following recognition stage. So, it was decided to move to segmentation-free phoneme recognition effectively using a trainable silence phoneme.

The MFCC and the Linear Predictive Code (LPC) algorithms were also compared as potential feature extraction techniques. The results showed that the MFCCs is the best for Arabic speech recognition with a performance of 73.40% compared to the LPC performance of 66.75% when using the ESN with supervised learning algorithm on the KAPD training database.

- **Deep Belief Networks (DNNs) for phoneme classification and acoustic modelling experiments on Arabic corpora**

The objective of the experiments was to explore the potential advantages of developing a RBM DNN-based system for Arabic speech recognition. In order to achieve this aim, several RBM DNN-based Arabic speech recognition systems were developed and evaluated across various Arabic corpora.

The performance of a state-of-the-art Shallow MLP based phoneme recognition system trained on the 13 MFCCs plus their first and second derivatives found to be about 0.35% better than the base system that used only the 13 MFCCs. Statistically there was no significant difference in performance between the two different MFCCs feature sizes. However, the state-of-the-art 39 MFCCs features were adopted over all experiments conducted in this thesis.

During this study, a direct comparison experiment was then conducted on the KAPD corpus between the results of the Shallow MLP model and the RBM Deep MLP (RBM DNN) model results and found that, the performance of the RBM DNN was about 2.13% better for Arabic phoneme recognition than the Shallow MLP NN.

MLP models cannot handle dynamic systems directly. Thus, a window of feature vectors (a time-window) was used to represent a sequence of frames which are used as input to the MLPs NN. As a consequence, one of the most important parameters that were optimized is the number of frames in the window input that fed to the network. A study was conducted on the segmented phonemes of the KAPD and CSLU2002 datasets to infer the suitable number of frames in the input window. The main finding from this study was that the average number of frames that represent a suitable input window is 5 frames. By analysing the performance results of experiments conducted on the KAPD and the CSLU2002 corpora, it was confirmed that 5 frames is more suitable than 1, 3, 7, 9, and 11 frames to represent the time-window for Arabic speech recognition. A post-processing enhancement was also applied to the DNN outputs in order to improve the framewise phoneme recognition accuracy and to find the accuracy at a phoneme level instead of the frame level. We found that applying the post-processing significantly improved the frame wise phoneme recognition and successfully constructed the phoneme level.

To evaluate the adoption of the DNN model in developing an acoustic modelling for Arabic speech recognition system instead of a phoneme-based system, the Levantine Arabic corpus was used to explore the potential gain in building an acoustic modelling Arabic speech recognition system using the DNN model. To the best of our knowledge, RBM DNN models have not been explored for any Levantine corpora. This encouraged us to investigate the DNN model for Arabic acoustic modelling on one of the proposed Levantine corpus. This allows us to claim priority for adopting this RBM DNN model for Levantine Arabic acoustic models. A post-processing enhancement was also applied to the DNN acoustic model outputs in order to improve the recognition accuracy and to obtain the accuracy at a phoneme level instead of the frame level. This was based on '*windowing'* the acoustic sequence followed by cancelling the repeated frame in the sequence. The '*windowing'* post process significantly improved the phoneme recognition performance. About 3.84% phoneme recognition was increased in the accuracy performance by applying a window size of 29 frames to the baseline which is the framewise phoneme recognition. However, applying the cancelling procedure, harmfully affected the accuracy performance due to the increase in the number of intersections in the sequence as a result of this procedure.

Since the most published work on the Levantine dialect were for various corpora and sizes. It was difficult to make a direct comparison between other work using these corpora and our work as they have different training and testing sizes and different data catalogue. Thus, we used the work presented in this chapter as a baseline for our work in Chapter seven in order to make direct comparison.

Despite the success of the RBM DNN, reported in this work, for phoneme classification and acoustic modeling, there is an inherent limitation in using feed-forward neural network, even when using a deep structure with multiple context frames. Fundamentally, feed forward NNs are not capable of handling dynamic data directly.

- **Echo State Networks (ESNs) for phoneme classification and acoustic models on Arabic corpora**

To investigate Echo State Networks (ESNs) for phoneme classification and acoustic modelling Arabic speech recognition, we explored different ESN architectures and experiments. The KAPD Saudi Arabia dialect corpus and the CSLU2002 multi-dialect corpus were used to train and test an ESN with supervised and forced supervised learning algorithms. Furthermore, a novel combined supervised/forced supervised learning algorithm (unsupervised adaptation) was developed and tested on an optimised Arabic phoneme recognition datasets. In addition, the Levantine Arabic dataset was also used to train the mentioned systems as acoustic modeling. For these experiments using the ESN-based model, we used the raw dataset instead of the normalized dataset based on the significant difference found when we compared the performance results of two systems experiments that used the normalized and the raw KAPD dataset using the two-tailed Matched Pairs T-test significance test.

For comparison between the ESN performance and the performance of Deep Multi-Layer Perceptron (MLP) for the KAPD database; we found that conventional ESN model outperformed the Deep MLP (DNNs) model, where The frame recognition performance of Arabic phonemes recognition were much improved (more than 23% difference) using the ESN with supervised learning model compared with the performance of the Arabic frame phoneme recognition when using the DNN model. The results clearly show that the ESN model has a better ability for recognizing phonemes sequences than the DNN model for small vocabulary size dataset.

We also investigated an unsupervised ESN adaptation approach to minimize classification errors. The main idea was to use adapted-target outputs that act as a forced teacher for supervised training. Thus, a combined supervised/forced supervised learning was implemented (Unsupervised Adaption) to effectively produce teacher outputs before being fed back into the reservoir. We discussed the findings of the conducted experiments using this adaptation with the CSLU2002 corpus. The baseline performance of the 13 MFCCs input ESN was firstly explored and the adaptation approach was found to greatly improve the recognition performance. The performance 39 MFCCs feature vector ESN system was an about 16.35% better than the baseline 13 MFCCs. However, only slight improvement was gained when using this adaption with 39

MFCCs feature vectors. This was statistically not significant difference in the performance. A trivial enhancement on the recognition performance of the core testing dataset was obtained when we used deep ESN architecture for both type of features.

The results of the experiments of these systems cannot be directly compared to many related work found in the literature as these KAPD and CSLU2002 phoneme dataset are corpora that were manually segmented and labelled by the author. However, one closely-related work has been provided (Alalshekmubarak, 2014) that used our CSLU2002 Arabic phonemes database in order to allow us to make a valid direct comparison between the systems. We also compared the performance of four NN based models, conducted on the CSLU2002 corpus, with two classification approaches, namely, RBM DNNs and the ESNs. This comparison was presented in Table 7.5, and the results showed the superior performance of our developed system (combined learning ESN-based) with 52.15% framewise phoneme recognition performance when it was compared to the other systems recently reported in the literature that used the same CSLU2002 selected corpus. Generally, ESN outperformed the DNN under the considered state-of-the-art 39 MFCCs features. This promotes the adoption of ESN in Arabic speech recognition systems.

The Levantine Arabic Corpus was also proposed to explore the potential gain in building large vocabulary acoustic models for Arabic speech recognition system using the ESNs model. A significant improvement on the recognition performance was achieved when the ESN model was implemented compared to the baseline RBM DNN model's result. Table 7.6 showed results of the DNNs system performance and the ESNs system performance trained on the same Levantine dataset for Arabic speech recognition. A significant improvement on the recognition performance was achieved when using the ESN model compared to the baseline RBM DNN model's result, where ESN was found to be about 14.01% better than the base system trained with input window of 5frames. This was confirmed with a two-tailed Matched Pairs T-test significance test. The adoption of the ESNs model for acoustic modeling is seen to be more valid than the adoption of the DNNs model for acoustic modeling speech recognition. This is because ESNs are recurrent models and support sequence models better than the RBM DNN models even with the contextual input window.

The performance of ESN model is very promising that encourages the researchers for adopting this model for various applications. ESN models can provide good performance with a limited size of corpus is available. ESNs are also simple with less computation compared to other NNs (MLP, RNN). Thus, training time is very competitive compared with other approaches.

We proposed to use well known corpus even that it is not Arabic to evaluate our proposed models and to make a valid comparison with other published work on the same corpus. We used The TIMIT English corpus for this goal.

- **Evaluating the Deep Neural Networks (DNNs) and Echo State Networks (ESNs) for acoustic modelling on TIMIT corpus**

To investigate deep learning for framewise phoneme classification and acoustic modelling using Deep Neural Networks (DNNs) and Echo State Networks (ESNs) on TIMIT corpus, we had explored different experiments to develop DNN and ESN models introduced in Chapter Six and Chapter Seven respectively for speech recognition.

The main findings of the conducted experiments using the proposed models through the TIMIT corpus was that, the performance of the ESN model outperformed the baseline RBM DNN model, where ESN was found to be about 16.95% better than the base system trained with input window of 5frames. This was a statistically significant difference in the performance of these two models.

Finally, comparisons between the best performances achieved by the systems investigated in this thesis and the published works in equivalent projects based on framewise phoneme recognition found in the literature and using the TIMIT corpus were presented. This comparison was presented in Table 8.2. All the resented research used the whole TIMIT testing dataset for testing their systems. However, the TIMIT training dataset has been divided into training and validation set as cleared in Table 8.2 by some research. In this research, we used the smallest training dataset (2772 utterances) compared to the other research (3696 utterances) to overcome with the hardware availability. Our developed system ESN-based showed lower performance when it was compared to the other systems recently reported in the literature that used the same corpus. This might due to the small size of training dataset used for this research, hardware availability and not applying speaker and noise adaption that can improve the results in this thesis as our aim is to investigate the proposed models for speech recognition and to make a direct comparison between these models.

It can be concluded that speech recognition is a big challenge due to the similarity between phoneme pronunciation and the effect of time varying frequency patterns for different phonemes. Moreover, different dialectics, limited examples of phonemes in the training set and noise from the environment and from the speakers has a significant degradation effect on the speech recognition system performance.

9.2 Future work

Further research can be done based on the work was done in this thesis to improve Arabic phoneme recognition and acoustic models. There are several potential points that would be worthwhile investigating.

- **Database**. A fundamental key point in this research is using a suitable database for the proposed task. The 41 hours of the Levantine corpora needs to be used for training the proposed models that will lead to more confident work. In addition, constructing a well prepared Libyan dialect database with a large number of speakers is needed for region research.
- **Input features**. More focus on the feature extraction methods needs to be directed towards in developing speech systems. Other features type could be investigated with ESNs for speech recognition task to improve the speech recognition performance such as cochlear data (hmidt et al., 2010), and the delayed rank order coding scheme (Tashan, 2012). Moreover, of the mel-frequency spectral coefficients (MFSC) has demonstrated to be input features more effective than the MFCCs features for the DBNs applied by (Mohamed, 2014) for acoustic models, this type of features needs to be examined as input features for the ESNs models. In addition, neural network features such as bottle-neck features and Tandem features model are another sucsseful ways of using ANNs with HMMs in hybrid systems. These features sould be examined with the NN/NN hybrid system.

- **Hybrid systems**. Using combination of different neural network types to construct Hybrid systems such as ESN/DNNs or DNN/ESN networks need to be investigated for phoneme classification problem and acoustic modelling systems. Also, ESN/HMM system needs to be used for acoustic models. These combined models could effectively improve the recognition performance of Arabic speech.

- **Adaptation algorithms.** Using various adaptation approaches that have been suggested to compensate mismatches between the training model conditions and the input conditions may occur due to differences between speakers, environmental noise, and differences in channels. Such as speaker identity vectors (i-vectors) (Dehak et al., 2011).

References

AHMED, A., YU, K., XU, W., GONG, Y. & XING, E. P. 2008. Training hierarchical feed-forward visual recognition models using transfer learning from pseudo tasks. *in Proceedings of the 10th European Conference on Computer Vision (ECCV'08),* 69–82.

AL-MANIE, M. A., ALKANHAL, M. I. & AL-GHAMDI, M. M. Automatic Speech Segmentation Using the Arabic Phonetic Database. Proceedings of the 10th WSEAS International Conference on AUTOMATION & INFORMATION, 2006. 76-79.

AL-RADAIDEH, Q. A. & MASRI, K. H. 2011. Improving Mobile Multi-Tap Text Entry for Arabic Language. *Computer Standards & Interfaces,* 33 108-113.

AL-SHAREEF, S. & HAIN, T. 2011. An Investigation in Speech Recognition for Colloquial Arabic. *INTERSPEECH*, 2869-2872.

AL-SHAREEF, S. & HAIN, T. 2012. CRF-based Diacritisation of Colloquial Arabic for Automatic Speech Recognition. *Interspeech*.

ALALSHEKMUBARAK, A. 2014. *Towards A Robust Arabic Speech Recognition System Based On Reservoir Computing.* PhD, University of Stirling.

ALALSHEKMUBARAK, A. & SMITH, L. S. 2014. On Improving the Classification Capability of Reservoir Computing For Arabic Speech Recognition. *inWermter, S.,Weber, C., Duch,W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., Villa, A.E.P. (Eds.) , Artificial Neural Networks and Machine Learning-ICANN 2014, 24th International.*

ALI, A. A. & HWAIDY, I. T. 2007. Hierarchical Arabic Phoneme Recognition Using MFCC Analysis. *Iraq J. Electrical and Electronic Engineering,* 3 97-106.

ANWAR, M. J., AWAIS, M. M., MASUD, S. & SHAMAIL, S. 2006 Automatic Arabic speech segmentation system. *International Journal of Information Technology,* 12**,** 102-111.

APPEN 2007. Levantine arabic conversational telephone speech. Linguistic Data Consortium, Philadelphia, Sydney, Australia, 2007. Catalog No: LDC2007S01 & LDC2007T01. *In:* LTD, A. P. (ed.).

BENGIO, Y. 1991. *Artificial Neural Networks and Their Application to Sequence Recognition.* PhD thesis, McGill University.

BENGIO, Y. 2009. Learning deep architectures for AI. Found. TrendsMach. Learn., 2 (1) (2009), 1–127. *Foundations and Trends in Machine Learning,* 2**,** 1–127.

BENGIO, Y., BOULANGER, N. & PASCANU, R. 2013a. Advances in optimizing recurrent networks. *in Proc. ICASSP*.

BENGIO, Y., COURVILLE, A. & VINCENT, P. 2013b. Representation learning: a review and new perspectives. *IEEETrans.PatternAnal.Mach. Intell.,* 35**,** 1798–1828.

BENGIO, Y., LAMBLIN, P., POPOVICI, D. & LAROCHELLE, H. 2007. Greedy layer-wise training of deep networks. *in Advances in Neural Information Processing Systems 19 (NIPS'06)***,** 153–160.

BENGIO, Y., MORI, R. D., FLAMMIA, G. & KOMPE, F. 1991. Global optimization of a neural network – Hidden Markov model hybrid. *in Proc. Proc. Eurospeech*.

BENGIO, Y., SIMARD, P. & FRASCONI, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on, Neural Networks,* 5**,** 157–166.

BENZEGHIBA, M., MORI, R. D., DEROO, O., DUPONT, S., ERBES, T., JOUVET, D., FISSORE, L., LAFACE, P., MERTINS, A., RIS, C., ROSE, R., TYAGI, V. & WELLEKENS, C. 2007. Automatic Speech Recognition and Speech Variability: A Review. *Speech Communication* 49**,** 763-786.

BIADSY, F. 2011. *Automatic Dialect and Accent Recognition and its Application to Speech Recognition.* COLUMBIA UNIVERSITY.

BIADSY, F., HABASH, N. & HIRSCHBERG, J. 2009. Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. *in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL '09, (Stroudsburg, PA, USA)***,** 397-405.

BILLA, J., NOAMANY, M., SRIVASTAVA, A., LIU, D., STONE, R., XU, J., MAKHOUL, J. & KUBALA, F. 2002a. Audio Indexing of Arabic broadcast news. *Acoustics, Speech, and Signal Processing (ICASSP),* 1**,** I-5 - I-8

BILLA, J., NOAMANY, M., SRIVASTAVA, A., MAKHOUL, J. & KUBALA, F. 2002b. Arabic Speech and Text in TIDES OnTAP. *in Proceedings of the second international conference on Human Language Technology Research, HLT '02, (San Francisco, CA, USA)***,** 7-11 Morgan Kaufmann Publishers Inc.

BOURLARD, H. & MORGAN, N. 1993. *Connectionist Speech Recognition: A Hybrid Approach,* KLUWER ACADEMIC PUBLISHERS.

BROWN, P. 1987. *The Acoustic-Modeling Problem in Automatic Speech Recognition.* PhD thesis, Carnegie-Mellon University,USA,.

BRUGNARA, F., FALAVIGNA, D. & OMOLOGO, M. 1992. A HMM-based system for automatic segmentation and labeling of speech. *The Second International Conference on Spoken Language Processing, ICSLP 1992.* Banff, Alberta, Canada.

BRUGNARA, F., FALAVIGNA, D. & OMOLOGO, M. 1993. Automatic segmentation and labeling of speech based on hidden Markov models. *Speech Communication,* 12**,** 357–370.

BUCKWALTER, T. 2002. LDC Buckwalter Arabic Morphological Analyzer (BAMA). Version 1.0. *LDC Catalog No. LDC2002L49, ISBN: 1-58563-257-0.*

CAMPBELL, N. 1996. Autolabelling Japanese ToBI. *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference,* 4**,** 2399 - 2402.

CHEN, C. H. 1988. *Signal processing handbook,* New York.

CHEN, R. & JAMIESON, L. 1996. Experiments on the implementation of recurrent neural networks for speech phone recognition Proceedings of the thirtieth annual asilomar conference on signals. *systems and computers* 779–782.

COLLOBERT, R. & WESTON, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. *in Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)***,** 160–167.

COSI., P., D. FALAVIGNA & OMOLOGO, M. 1991. A Preliminary Statistical Evaluation of Manual and Automatic Segmentation Discrepancies *In EUROSPEECH***,** 693-696.

COX, S., BRADY, R. & JACKSON, P. 1998. TECHNIQUES FOR ACCURATE AUTOMATIC ANNOTATION OF SPEECH WAVEFORMS. *In Proceedings of ICSLP '98 (Sydney, Australia),* 5**,** 1947-1950.

DAHL, G., YU, D., DENG, L. & ACERO, A. 2011. Context-dependent DBN-HMMs in large vocabulary continuous speech recognition. *in Proc. ICASSP*.

DAHL, G. E., RANZATO, M., MOHAMED, A. & HINTON, G. E. 2010. Phone recognition with the mean-covariance restricted Boltzmann machine. *In NIPS'2010*.

DAHL, G. E., YU, D., DENG, L. & ACERO, A. 2012. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing,* 20**,** 33–42.

DAI, J., VENAYAGAMOORTHY, G. K. & HARLEY, R. G. 2010. An Introduction to the Echo State Network and its Applications in Power System. *IEEE Xplore*.

DAVIS, S. B. & MERMELSTEIN, P. 1980. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions On Acoustic, Speech, And Signal Processing,* ASSP-28**,** 357-366.

DEHAK, N., KENNY, P. J., DEHAK, R., DUMOUCHEL, P. & P.OUELLET 2011. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing,* 19**,** 788-798.

DEMPSTER, A., LAIRD, N. & RUBIN, D. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society,* 39**,** 1–38.

DENG, L., SELTZER, M., YU, D., ACERO, A., MOHAMED, A. & HINTON, G. 2010. Binary coding of speech spectrograms using a deep auto-encoder. *in Proc. Interspeech*.

DOYA, K. 1992. Bifurcations in the learning of recurrent neural networks. *In: Proceedings of IEEE International Symposium on Circuits and Systems,* 6**,** 2777-2780.

DUDA, R. O., HART, P. E. & STORK, D. G. 2001. *Pattern Classification*.

EL-IMAM, Y. A. 2004. Phonetization of Arabic: rules and algorithms. *In Computer Speech and Language,* 18**,** 339–373.

ELLIS, D. P. W., SINGH, R. & SIVADAS, S. 2001. Tandem acoustic modeling in large-vocabulary recognition. *in Proc. ICASSP*.

ELMAHDY, M., GRUHN, R. & MINKER, W. 2012a. Fundamentals. *Novel Techniques for Dialectal Arabic Speech Recognition.* Sphingers.

ELMAHDY, M., HASEGAWA-JOHNSON, M. & MUSTAFAWI, E. 2012b. A Baseline Speech Recognition System for Levantine Colloquial Arabic.

FRANZ, A. & MILCH, B. 2002. Searching the web by voice. *Proc. Computational Linguistics***,** 1213–1217.

FREUND, Y. & HAUSSLER, D. 1994. Unsupervised learning of distributions on binary vectors using two layer networks. *In:* REPORT, T. (ed.). University of California at Santa Cruz, Santa Cruz, CA, USA.

FURUI, S. 1986. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. Acoustics, Speech and Signal Processing***,** 52-59.

G.-DOMINGUEZ, J., L.-MORENO, I., MORENO, P. J. & G.-RODRIGUEZ, J. 2014. Frame-by-frame language identification in short utterances using deep neural networks. *Artical in press,Neural Networks*.

GALES, M. & YOUNG, S. 2007. The application of hidden Markov models in speech recognition. *Foundations and Trends in Signal Processing,* 1**,** 195–304.

GALES, M. J. F. 1998. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language,* 12**,** 75–98.

GALES, M. J. F. 2007. Discriminative Models for Speech Recognition. *in ITA Workshop, University San Diego, USA, February 2007.*

GAMMAS, D., JUDITH, N. & NORA, C. 2013. *Arabic Pod 101* [Online].

GAROFOLO, J. S., LAMEL, L. F., FISHER, W. M., FISCUS, J. G., PALLETT, D. S. & DAHLGREN, N. L. 1993. Darpa timit acoustic phonetic continuous speech corpus cdrom.

GLOROT, X. & BENGIO, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. *in Proc. AISTAT*.

GOLDMAN, J.-P. 2011. EasyAlign: a friendly automatic phonetic alignment tool under Praat.

GR´EZL, F., KARAFIA´T, M., KONTA´R, S. & CERNOCKY, J. 2007. Probabilistic and bottle-neck features for LVCSR of meetings. *ICASSP'07.* Hononulu.

GRAVES, A. 2008. *Supervised Sequence Labelling with Recurrent Neural Networks.*

GRAVES, A. 2012. Supervised Sequence Labelling with Recurrent Neural Networks, volume 385 of Studies in Computational Intelligence. Springer, .

GRAVES, A., FERNANDEZ, S., GOMEZ, F. & SCHMIDHUBER, J. 2006. Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks. *in Proc. ICML*.

GRAVES, A. & JAITLY, N. 2014. Towards End-to-End Speech Recognition with Recurrent Neural Networks. *Proceedings of the 31 st International Conference on Machine Learning.* Beijing, China: JMLR: W&CP.

GRAVES, A., MOHAMED, A. & HINTON, G. 2013. Speech recognition with deep recurrent neural networks. *In Proc ICASSP 2013.* Vancouver, Canada.

GRAVES, A. & SCHMIDHUBER, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks,* 18**,** 602–610.

GRAVES, A. & SCHMIDHUBER, J. 2009. Offline handwriting recognition with multidimensional recurrent neural networks. *in Neural Information Processing Systems 21***,** 545–552.

GRE´ZL, F. & FOUSEK, P. 2008. Optimizing bottle-neck features for lvcsr. *Acoustics, Speech and Signal Processing. ICASSP 2008. IEEE International Conference on* Las Vegas, NV IEEE.

GRUBB, A. & BAGNELL, J. A. 2013. Stacked Training for Overfitting Avoidance in Deep Networks. *Appearing at the ICML 2013 Workshop on Representation Learning*.

HABASH, N. 2010. Introduction to Arabic natural language processing. *Synthesis Lectures on Human Language Technologies,* 3**,** 1–187.

HACHKAR, Z., MOUNIR, B., FARCHI, A. & ABBADI, J. E. 2011. Comparison of MFCC and PLP Parameterization in Pattern Recognition of Arabic Alphabet Speech. *Canadian Journal on Artificial Intelligence, Machine Learning & Pattern Recognition,* 2**,** 56-60.

HADSELL, R., ERKAN, A., SERMANET, P., SCOFFIER, M., MULLER, U. & LECUN, Y. 2008. Deep belief net learning in a long-range vision system for autonomous offroad driving. *in Proc. Intelligent Robots and Systems (IROS'08)***,** 628–633.

HAIN, T. 2001. *Hidden Model Sequence Models for Automatic Speech Recognition.* PhD thesis, University of Cambridge.

HAIN, T. & WOODLAND, P. C. 1998. CU-HTK acoustic modeling experiments. *The 9th Conversational Speech Recognition Workshop, .* MITAGS, Linthicum Heights, Maryland.

HALBERSTADT, A. 1998. *Heterogeneous acoustic measurements and multiple classifiers for speech recognition.* PhD thesis, Massachusetts Institute of Technology.

HAWKINS, P. 1988. *Introducing phonology,* Australia, Pty Ltd.

HEINTZ, I. 2010. *Arabic Language Modeling With Stem-Derived Morphemes For Automatic Speech Recognition.* PhD thesis, The Ohio State University.

HERMANS, M. & SCHRAUWEN, B. 2013. Training and Analyzing Deep Recurrent Neural Networks.

HERMANSKY, H. 1990. Perceptual linear predictive (PLP) analysis of speech. *Acoustical Society of America Journal,* 87**,** 1738–1752.

HERMANSKY, H., ELLIS, D. & SHARMA, S. 2000. Tandem connectionist feature extraction for conventional HMM systems. *ICASSP-2000, Istanbul***,** 1635–1638.

HERMANSKY, H. & SHARMA, S. 1998. TRAPS - Classifiers of Temporal Patterns. *In Proc. International Conference on Spoken Language Processing (ICSLP)***,** 1003–1006.

HINTON, G., DENG, L., YU, D., DAHL, G., MOHAMED, A., JAITLY, N., SENIOR, A., VANHOUCKE, V., NGUYEN, P., SAINATH, T. & KINGSBURY, B. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine,* 29**,** 82-97.

HINTON, G. & SALAKHUTDINOV, R. 2006. Reducing the dimensionality of data with neural networks.

HINTON, G. E. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation,* 14**,** 1771–1800.

HINTON, G. E., OSINDERO, S. & TEH, Y. 2006. A fast learning algorithm for deep belief nets. *Neural Computation,* 18**,** 1527-1554.

HMAD, N. & ALLEN, T. Biologically inspired Continuous Arabic Speech Recognition. *In:* PETRIDIS, M. B. A. M., ed. Research and Development in intelligent systems XXIX, 2012 Cambridge, UK. Springer London, 245-258.

HMAD, N. & ALLEN, T. 2013. Echo State Networks for Arabic phoneme recognition. *World Academy of Science, Engineering and Technology, International Journal of Computer, Control, Quantum and Information Engineering* 7.

HMIDT, S. P., WIERING, M. A., ROSSUM, A. C. V., ELBURG, R. A. J. V., ANDRINGA, T. C. & VALKENIER, B. 2010. Robust Real-TimeVowel Classification with an Echo StateNetwork. [Accessed 2010].

HOCHREITER, S. & SCHMIDHUBER, J. 1997. Long short-term memory. *Neural Computation,* 9**,** 1735–1780.

HOLZMANN, G. 2008. *Echo State Networks with Filter Neurons and a Delay&Sum Readout with Applications in Audio Signal Processing.* Master Master's Thesis, Graz University of Technology.

HOSOM, J. P. 2000a. *Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information.* Oregon Graduate Institute of Science and Technology.

HOSOM, J. P. 2000b. A Comparison of Speech Recognizers Created Using Manually-Aligned and Automatically-Aligned Training Data. *Technical Report ESE-00-002.* Beaverton: Center for Spoken Language Understanding (CSLU), Oregon Graduate Institute of Science and Technology (OGI).

HOSOM, J. P. 2009. Speaker-independent phoneme alignment using transition-dependent states. *Speech Communication,* 51**,** 352-368.

HUANG, X., ACERO, A., HON, H. W. & REDDY, R. 2001. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. *Prentice Hall PTR, 1st edition*.

ISMAIL, S. & AHMAD, A. B. 2004. Recurrent Neural Network with Backpropagation Through Time Algorithm for Arabic Recognition. European Simulation Multiconference, 2004 Graham Horton. SCS Europe.

JAEGER, H. 2001. The "echo state" approach to analysing and training recurrent neural networks. *In:* 148, T. R. G. R. (ed.). German National Research Center for Information Technology.

JAEGER, H. 2005. A Tutorial On Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF And The "Echo State Network" Approach. Fraunhofer Institute for Autonomous Intelligent Systems (AIS).

JAEGER, H. 2007. Echo state network. *Scholarpedia* 2**,** 2330.

JAEGER, H. 2012. Long Short-Term Memory in Echo State Networks: Details of a simulation study. *Technical Report 27* School of Engineering and Science: Jacobs University Bremen

JAEGER, H. & HAAS, H. 2004. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Telecommunication. *Science.*

JAEGER, H., LUKOSEVICIUS, M. & POPOVICI, D. 2007. Optimization and Applications of Echo State Networks with Leaky Integrator Neurons.

JALALVAND, A., TRIEFENBACH, F., VERSTRAETEN, D. & MARTENS, J. P. 2011. Connected digit recognition by means of reservoir computing. *In: Proceedings of Interspeech 2011***,** 1725-1728.

JEAGER, H. 2003. Adaptive Nonlinear System Identification With Echo State Networks.

JOU, S. C. S. 2008. *Automatic Speech Recognition on Vibrocervigraphic and Electromyographic Signals.* PhD, Carnegie Mellon University.

KINGSBURY, B., SAINATH, T. N. & SOLTAU, H. 2012. Scalable Minimum Bayes Risk Training of Deep Neural Network Acoustic Models Using Distributed Hessian-free Optimization *in Proc. interspeech*.

KIRCHHOFF, K., BILMES, J., DAS, S., DUTA, N., EGAN, M., JI, G., HE, F., HENDERSON, J., LIU, D., NOAMANY, M., SCHONE, P., SCHWARTZ, R. & VERGYRI, D. 2002. Novel Approaches To Arabic Speech Recognition. *the 2002 johns-hopkins summer workshop*.

KVALE, K. 1994. On the connection between manual segmentation conventions and "errors" made by automatic segmentation. *In Proceeding of ICSLP '94 (Yokohama, Japan)* 3.

LAVIE, A., WAIBEL, A., LEVIN, L., FINKE, M., GATES, D., GAVALDA, M., ZEPPENFELD, T. & ZHAN, P. 1997. Jansus-III: speech-to-speech translation in multiple languages. *Proc. ICASSP***,** 99–102.

LEE, H., GROSSE, R., RANGANATH, R. & NG, A. Y. 2009. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. *In Proceedings of the 26th International Conference on Machine Learning***,** 609-616.

LEE, L.-S. 1997. Voice dictation of Mandarin Chinese. *IEEE Signal Processing Magazine,* 14**,** 63–101.

LEE, S.-M., FANG, S.-H., HUNG, J.-W. & LEE, L.-S. 2001. Improved MFCC Feature Extraction by PCA-Optimized Filter Bank for Speech Recognition. *Automatic Speech Recognition and Understanding***,** 49-52.

LEGGETTER, C. J. & WOODLAND, P. C. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech & Language,* 9**,** 171 – 185.

LEUNG, H. & ZUE, V. W. 1984. A procedure for automatic alignment of phonetic transcriptions with continuous speech. *In Proceedings of ICASSP '84 (San Diego, California),* 9**,** 2.7.1-2.7.4.

LEVNER, I. 2008. *Data Driven Object Segmentation.* PhD Univesity of Alberta.

LIMA, A., ZEN, H., NANKAKU, Y., MIYAJIMA, C., TOKUDA, K. & KITAMURA, T. 2004. On the Use of Kernel PCA for Feature Extraction in Speech Recognition. *IEICE Trans. Inf. & Syst.,* E87-D**,** 2802-2811.

LIMA, A., ZEN, H., NANKAKU, Y., TOKUDA, K., KITAMURA, T. & RESENDE, F. G. 2005. Applying Sparse KPCA for Feature Extraction in Speech Recognition. *IEICE Trans. Inf. & Syst.,* E88-D**,** 401-409.

LJOLJE, A., HIRSCHBERG, J. & SANTEN, J. P. H. V. 1994. Automatic speech segmentation for concatenative inventory selection. *In SSW2***,** 93-96.

LJOLJE, A., HIRSCHBERG, J. & SANTEN, J. V. 1997. Automatic speech segmentation for concatenative inventory selection. *Progress in Speech Synthesis, Springer Verlag, New York***,** 305-311.

LLOYD, S. P. 1982. least Squares quantization in PCM. *In IEEE Transactions on Information Theory,* 28 129-137.

LOAN, C. V. 1992. *Computational Frameworks for the Fast Fourier Transform*.

MALFRÈRE, F. & DEROO, O. 1998. Phonetic alignment: speech synthesis based vs. hybrid HMM/ANN. *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference.* Sydney Convention Centre, Sydney, Australia.

MARTENS, J. 2010. Deep learning with Hessian-free optimization. *in Proc. ICML*.

MARTENS, J. & SUTSKEVER, I. 2011. Learning recurrent neural networks with Hessian-free optimization. *in Proc. ICML*.

MIAO, Y., ZHANG, H. & METZE, F. 2014. Towards Speaker Adaptive Training of Deep Neural Network Acoustic Models.

MIKOLOV, T., KARAFIAT, M., BURGET, L., CERNOCKY, J. & KHUDANPUR, S. 2010. Recurrent neural network based language model. *in Proc. ICASSP***,** 1045–1048.

MITRA, V., WANG, W., FRANCO, H., LEI, Y., BARTELS, C. & GRACIARENA, M. 2014. Evaluating robust features on Deep Neural Networks for speech recognition in noisy and channel mismatched conditions. *INTERSPEECH 2014.* Singapore: ISCA.

MNIH, A. & HINTON, G. E. 2009. A scalable hierarchical distributed language model. *in Advances in Neural Information Processing Systems 21 (NIPS'08)***,** 1081–1088.

MOHAMED, A.-R. 2014. *Deep Neural Network acoustic models for ASR.* PhD thesis, University of Toronto.

MOHAMED, A., DAHL, G. & HINTON, G. 2009. Deep belief networks for phone recognition. *in NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*.

MOHAMED, A., DAHL, G. & HINTON, G. 2012. Acoustic modeling using deep belief networks. *IEEE Trans. on Audio, Speech and Language Processing,* 20**,** 14–22.

MOHAMED, A., YU, D. & DENG, L. 2010. Investigation of full-sequence training of deep belief networks for speech recognition. *in Proc. Interspeech*.

MORGAN, N. 1990. The ring array processor (RAP): Algorithms and architecture. International Computer Science Institute.

MORGAN, N. 2012. Deep and wide: multiple layers in automatic speech recognition. *IEEE Trans.Audio Speech, Lang. Process,* 20**,** 7-13.

MORGAN, N., CHEN, B. Y., ZHU, Q. & STOLCKE, A. 2004. TRAPPING CONVERSATIONAL SPEECH: EXTENDING TRAP/TANDEM APPROACHES TO CONVERSATIONAL TELEPHONE SPEECH RECOGNITION. *in Proc. ICASSP*.

MORGAN, N. & FOSLER-LUSSIER, E. 1998. Combining Multiple Estimators Of Speaking Rate.

MORGAN, N., ZHU, Q., STOLCKE, A., SÖNMEZ, K., SIVADAS, S., SHINOZAKI, T., OSTENDORF, M., JAIN, P., H. HERMANSKY, ELLIS, D., DODDINGTON, G., CHEN, B., ÇETIN, Ö., BOURLARD, H. & ATHINEOS, M. 2005. Pushing the Envelope-Aside. *Signal Processing Magazine, IEEE,* 22**,** 81–88.

MOSA, G. S. & ALI, A. A. 2009. Arabic Phoneme Recognition using Hierarchical Neural Fuzzy Petri Net and LPC Feature Extraction. *Signal Processing: An International Journal (SPIJ),* 3**,** 161-171.

MURVEIT, H., BUTZBERGER, J., DIGALAKIS, V. & WEINTRAUB, M. 1993. Large-vocabulary dictation using SRI's DECIPHER speech recognition system: progressive search techniques. *Proc. ICASSP***,** 319–322.

NAHAR, K. M. O., ELSHAFEI, M., AL-KHATIB, W. G., AL-MUHTASEB, H. & ALGHAMDI, M. M. 2012. Statistical Analysis of Arabic Phonemes for Continuous Arabic Speech Recognition. *International Journal of Computer and Information Technology,* 01.

NOVOTNEY, S., SCHWARTZ, R. M. & KHUDANPUR, S. 2011. Unsupervised Arabic Dialect Adaptation with Self-Training. *in INTERSPEECH*, 541–544.

O'SHAUGHNESSY, D. 1987. *Speech communication - human and machine*.

OZTURK, M. C. & PRINCIPE, J. C. 2007. An Associative Memory Rreadout for ESNs with Applications to Dynamical Pattern Recognition. *Neural Networks* 20**,** 377–390.

P.S. GOPALAKRISHNAN, D. KANEVSKY, A. N´ADAS & NAHAMOO, D. 1991. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Trans. Information Theory*.

PEABODY, M. A. 2011. *Methods for pronunciation assessment in computer aided language learning.* PhD, Cambridge.

POVEY, D. 2004. *Discriminative Training for Large Vocabulary Speech Recognition.* Ph.D. thesis, Cambridge University.

POVEY, D. & WOODLAND, P. C. 2002. Minimum phone error and I-smoothing for improved discriminative training," ., , May . *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Orlando, FL*.

RABINER, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE,* 77**,** 257–286.

RABINER, L. R. & JUANG, B. H. (eds.) 1993. *Fundamentals of Speech Recognition,* Englewood Cliffs, New Jersey: Prentice Hall PTR.

RABINER, L. R. & SCHAFER, R. W. (eds.) 1978. *Digital Processing of Speech Signals,* New Jersey: Prentice-Hall.

RANZATO, M., BOUREAU, Y. & LECUN, Y. 2007. Sparse feature learning for deep belief networks. *in Proc. NIPS*.

RANZATO, M. & SZUMMER, M. 2008. Semi-supervised learning of compact document representations with deep networks. *in Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08),* 307**,** 792–799.

RAPP, S. 1995. Automatic phonemic transcription and linguistic annotation from known text with Hidden Markov Models. *In Proceeding of ELSNET Goes East and IMACS Workshop*.

ROBINSON, A. J. 1994. An application of recurrent nets to phone probability estimation *IEEE, Transactions on Neural Networks,* 5**,** 298 - 305.

SAINATH, T. N., KINGSBURY, B., MOHAMED, A.-R., DAHL, G. E., SAON, G., SOLTAU, H., BERAN, T., ARAVKIN, A. Y. & RAMABHADRAN, B. 2013. IMPROVEMENTS TO DEEP CONVOLUTIONAL NEURAL NETWORKS FOR LVCSR. *In ASRU*.

SAINATH, T. N., KINGSBURY, B., SAON, G., SOLTAU, H., MOHAMED, A.-R., DAH, G. & RAMABHADRAN, B. 2014. Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*.

SAKENAS, V. 2010. Distortion Invariant Feature Extraction with Echo State Networks. *School of Engineering and Science.* School of Engineering and Science Jacobs University Bremen gGmbH.

SALAKHUTDINOV, R. & HINTON, G. E. 2007a. Learning a nonlinear embedding by preserving class neighbourhood structure. *in Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS'07)*.

SALAKHUTDINOV, R. & HINTON, G. E. 2007b. Semantic hashing. *in Proceedings of the 2007 Workshop on Information Retrieval and applications of Graphical Models (SIGIR 2007)*.

SALAKHUTDINOV, R. & HINTON, G. E. 2008. Using deep belief nets to learn covariance kernels for Gaussian processes. *in Advances in Neural Information Processing Systems 20 (NIPS'07)***,** 1249–1256.

SALIMI, H., GIVEKI, D., SOLTANSHAHI, M. A. & HATAMI, J. 2012. Extended Mixture of MLP Experts by Hybrid of Conjugate Gradient Method and Modified Cuckoo Search. *International Journal of Artificial Intelligence & Applications (IJAIA), ,* 3.

SALMEN, M. & PLOGER, P. G. 2005. Echo State Networks used for Motor Control. *IEEE*, 1953-1958.

SANTEN, J. V. & SPROAT, R. 1999. High-accuracy automatic segmentation. *In Proc. EuroSpeech* Budapest, Hungary.

SAON, G., SOLTAU, H., NAHAMOO, D. & PICHENY, M. 2013. Speaker adaptation of neural network acoustic models using i-vectors. *in Proc. ASRU*, 55-59.

SCHAFER, R. W. & RABINER, L. R. 1975. Digital representations of speech signals. *Proc. of the IEEE,* 63, 662–677.

SCHRAUWEN, B. & BUSING, L. 2010. A Hierarchy of Recurrent Networks for Speech Recognition.

SCHUSTER, M. 1999. *On supervised learning from sequential data with applications for speech recognition.* PhD thesis, Nara Institute of Science and Technolog.

SEIDE, F., LI, G. & YU, D. 2011. Conversational speech transcription using context-dependent deep neural networks. *In Interspeech 2011*, 437–440.

SELOUANI, S. A. & CAELEN, J. 1999. Arabic Phonetic Features Recognition using Modular Connectionist Architectures.

SIEGLER, M. A. & STERN, R. M. 1995. On The Effects Of Speech Rate in Large Vocabulary Speech Recognition Systems.

SJÖLANDER, K. 2003. An HMM-based system for automatic segmentation and alignment of speech. *Umeå University, Department of Philosophy and Linguistics, PHONUM 9* 93-96.

SKOWRONSKI, M. D. & HARRIS, J. G. 2006. Minimum Mean Squared Error Time Series Classification Using an Echo State Network Prediction Model. Island of Kos, Greece: IEEE International Symposium on Circits Systems,.

SKOWRONSKI, M. D. & HARRIS, J. G. 2007. Automatic Speech Recognition Using a Predictive Echo State Network Classifier. *Neural Networks,* 20, 414-423.

SMOLENSKY, P. 1986. *Information processing in dynamical systems: foundations of harmony theory*, Parall. distrib. process.

SOLTAU, H., SAON, G., KINGSBURY, B., KUO, J., MANGU, L., POVEY, D. & ZWEIG, G. 2007. The IBM 2006 Gale Arabic ASR System *Acoustics, Speech and Signal Processing. ICASSP 2007. IEEE International Conference,* 4, IV-349 - IV-352

SRIVASTAVA, G. H. N., KRIZHEVSKY, A., SUTSKEVER, I. & SALAKHUTDINOV, R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR,abs/1207.0580*.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I. & SALAKHUTDINOV, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research,* 15, 1929-1958.

SUTSKEVER, I., MARTENS, J. & HINTON, G. 2011. Generating text with recurrent neural networks *in Proc. ICML*.

SVENDSEN, T. & KVALE, K. 1990. Automatic alignment of phonemic labels with continuous speech. *The First International Conference on Spoken Language Processing, ICSLP 1990.* Kobe, Japan.

TASHAN, T. 2012. *Biologically Inspired Speaker Verification.* PhD thesis, Nottingham Trent University.

TAYLOR, G. & HINTON, G. 2009. Factored conditional restricted Boltzmann machines for modeling motion style. *in Proceedings of the 26th International Conference on Machine Learning (ICML'09)*, 1025–1032.

TAYLOR, G. W., FERGUS, R., LECUN, Y. & BREGLER, C. 2010. Convolutional learning of spatio-temporal features. *In Eurpean Conference on Computer Vision*.

TEBELSKIS, J. 1995. *Speech Recognition using Neural Networks.* PhD, Carnegie Mellon University.

TOLEDANO, D. T., GOMEZ, L. A. H. & GRANDE, L. V. 2003. Automatic phoneme segmentation. *IEEE Trans. Speech and Audio Proc.,* 11, 617-625.

TONG, M. H., BICKETT, A. D., CHRISTIANSEN, E. M. & COTTRELL, G. W. 2007. Learning Grammatical Structure with Echo State Networks. *Neural Networks* 20, 424–432.

TRIEFENBACH, F., DEMUYNCK, K. & MARTENS, J.-P. 2012. Improving large vocabulary continuous speech recognition by combining GMM-based and reservoir-based acoustic modeling. *Spoken Language Technology Workshop (SLT), IEEE* 107-112

TRIEFENBACH, F., JALALVAND, A., DEMUYNCK, K. & MARTENS, J. 2013. Acoustic Modeling With Hierarchical Reservoirs. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING,* 21**,** 2439-2450.

TRIEFENBACH, F., JALALVAND, A., SCHRAUWEN, B. & MARTENS, J. P. 2011. Phoneme recognition with large hierarchical reservoirs. *In: Advances in Neural Information Processing Systems 23 (NIPS 2010), MIT Press, Cambridge***,** 2307-2315.

VENAYAGAMOORTHY, G. K. 2007. Online Design of an Echo State Network Based Wide Area Monitor for a Multimachine Power System. *Neural Networks* 20**,** 404–413.

VERGYRI, D. & KIRCHHOFF, K. 2004. Automatic diacritization of Arabic for Acoustic Modeling in Speech Recognition. *COLING 2004 Computational Approaches to Arabic Script-based Languages***,** 66–73.

VERGYRI, D., KIRCHHOFF, K., GADDE, R., STOLCKE, A. & ZHENG, J. 2005. Development of a Conversational Telephone Speech Recognizer for Levantine Arabic. *In Proceedings SRI Publication*.

VERGYRI, D., MANDAL, A., WANG, W., STOLCKE, A., ZHENG, J., GRACIARENA, M., RYBACH, D., GOLLAN, C., SCHLUTER, R., KIRCHHOFF, K., FARIA, A. & MORGAN, N. 2008. Development of the SRI/Nightingale Arabic ASR system. *In Proceedings of Interspeech***,** 1437–1440.

VERSTRAETEN, D., SCHRAUWEN, B., D'HAENE, M. & STROOBANDT, D. 2007. An experimental unification of reservoir computing methods. *Neural Networks,* 20 391–403.

VESEL´Y, K., GHOSHAL, A., BURGET, L. & POVEY, D. 2013. Sequence-discriminative training of deep neural networks. *interspeech*.

VETTER, R., VIRAG, N., RENEVEY, P. & VESIN, J.-M. 1999. Single Channel Speech Enhancement Using Principal Component Analysis and MDL Subspace Selection. *Eurospeech*.

WAGNER, M. 1981. Automatic labelling of continuous speech with a given phonetic transcription using dynamic programming algorithms. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '81,* 6.

WAHEED, K., WEAVER, K. & SALAM, F. M. 2002. A Robust Algorithm for Detecting Speech Segments Using an Entropic Contrast. *In proc. of the IEEE Midwest Symposium on Circuits and Systems. Lida Ray Technologies Inc.,* 45.

WAHLSTER, W. 2000. Verbmobil: foundations of speech-to-speech translation. *Springer-Verlag Berlin Heidelberg*.

WERBOS, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78**,** 1550-1560.

WESTON, J., RATLE, F. & COLLOBERT, R. 2008. Deep learning via semi-supervised embedding. *in Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)***,** 1168–1175.

WIGHTMAN, C. & TALKIN, D. 1997. The Aligner: Text to speech alignment using Markov Models. *Progress in Speech Synthesis, Springer Verlag, New York***,** 313-323.

WILLIAMS, R. J. & ZIPSER, D. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation,* 1**,** 270-280.

WITT, S. & YOUNG, S. 1997. Language learning based on non-native speech recognition. *Proc. Eurospeech***,** 633–636.

WOODLAND, P. C., LEGGETTER, C. J., ODELL, J. J., VALTCHEV, V. & YOUNG, S. J. 1995. The 1994 HTK large vocabulary speech recognition system. *In Proceedings of ICASSP '95 (Detroit, MI)*.

WOODLAND, P. C. & POVEY, D. 2002. Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech & Language,* 16**,** 25-47.

X.D. HUANG, A. ACERO & HON., H. W. 2001. *Spoken Language Processing*.

XU, Y., GOLDIE, A. & SENEFF, S. 2009. Automatic question generation and answer judging: a Q&A game for language learning. *Proc. SIGSLaTE*.

YOUNG, S., EVERMANN, G., GALES, M., HAIN, T., KERSHAW, D., LIU, X., MOORE, G., ODELL, J., OLLASON, D., POVEY, D., VALTCHEV, V. & WOODLAND, P. (eds.) 2006. *The HTK book, version 3.4*.

YOUNG, S. J., EVERMANN, G., HAIN, T., KERSHAW, D., MOORE, G. L., J.ODELL, J., OLLASON, D., POVEY, D., VALTCHEV, V. & C.WOODLAND, P. 2003. *The HTK Book. ,* Cambridge, U.K., Cambridge Univ. Press.

YOUNG, S. J. & WOODLAND, P. C. 1994. State clustering in hidden Markov model-based continuous speech recognition. *Computer Speech and Language,* 8**,** 369-383.

YUAN, J. & LIBERMAN, M. 2008. Speaker identification on the SCOTUS corpus. *Proceedings of Acoustics 2008***,** 5687-5690.

YUAN, J., RYANT, N., LIBERMAN, M., STOLCKE, A., MITRA, V. & WANG, W. 2013. Automatic Phonetic Segmentation using Boundary Models. *in INTERSPEECH, 2013***,** 2306–2310.

YUK, D. 1999. *Robust Speech Recognition Using Neural Networks and Hidden Markov Models-Adaptations Using Non-Linear Transformations.* Doctor of Philosophy, The State University of New Jersey.

ZAVAGLIAKOS, G., MCDONOUGH, J., MILLER, D., EL-JAROUDI, A., BILLA, J., RICHARDSON, F., MA, K., SIU, M. & GISH, H. 1998. The BNN Byblos 1997 large vocabulary conversational speech recognition system," in Proceedings of ICASSP, 1998. *Acoustics, Speech and Signal Processing. IEEE,* 2**,** 905 - 908.

ZAVALIAGKOS, G., SCHWARTZ, R. & MAKHOUL, J. 1995. Batch, incremental and instantaneous adaptation techniques for speech recognition. *Proc. ICASSP-95* 676-679.

ZUE, V., SENEFF, S., GLASS, J., POLIFRONI, J., PAO, C., HAZEN, T. & HETHERINGTON, L. 2000. JUPITER: a telephone-based conversational interface for weather information. *IEEE Trans. on Speech and Audio Processing,* 8**,** 100–112.

Appendixes

Appendix A - (Characters of Arabic Script, Buckwalter Transliteration, and Arabic pronunciation in romans letters)

Table A.1 - Characters of Arabic Script, Buckwalter Transliteration, and Arabic pronunciation in romans letters.

Alphabet							
د d /d/	خ x /kh/	ح H /h/	ج j /jh/	ث v /th/	ت t /t/	ب b /b/	ا A /ae:/
ط T /tt/	ض D /dd/	ص S /ss/	ش $ /sh/	س s /s/	ز z /z/	ر r /r/	ذ * /dh/
م m /m/	ل l /l/	ك k /k/	ق q /kq/	ف f /f/	غ g /gh/	ع E /ai/	ظ Z /zh/
		ي y /y/	و w /w/	ه h /hh/	ن n /n/		
Additional letters							
ء ' /q/	آ | /æ:/	أ > /q/	ؤ & /q/	إ < /q/	يء } /q/	ى Y /æ:/	
Diacritic Characters							
◌َ a /æ/	◌ُ u /uh/	◌ِ i /ih/	◌ْ 0 null	◌ّ ~ Double letter	◌ً F /an/	◌ٌ K /un/	◌ٍ N /in/