

Large Vocabulary Continuous Speech Recognition System for Persian Language Using Deep Neural Networks

by

Amir Lavasani

A Master's Thesis

Submitted to the Department of Computer and Electrical Engineering

AZAD UNIVERSITY, North Tehran Branch

Supervisor

Dr. Mohammad Mansori

Spring 2018

Table of Contents

Abstract.....	10
Introduction.....	11
Chapter 1: Thesis Outline.....	14
1.1 Introduction: Outline of the Research.....	14
1.2 Statement of the General Research Problem.....	14
1.3 The Significance and Necessity of the Research.....	15
1.4 The Aspect of Novelty and Innovation in Research.....	16
1.5 Specific Objectives of the Research.....	16
1.6 Main Research Questions.....	17
1.7 Research Hypotheses.....	17
1.8 Definition of Technical Abbreviations.....	17
1.9 Research Methodology.....	18
1.9.1 Research Method based on Purpose, Data, and Execution.....	18
1.9.2 The Investigated Variables and Description of Examination and Measurement.....	20
1.10 Data Collection Methods and Tools.....	22
1.11 Data Analysis Methods and Tools.....	22
1.12 Scope of Research: Thematic, Temporal, and Spatial Scope.....	22
1.13 Summary of Contents.....	23
Chapter 2: Theoretical Foundations and Literature Review.....	24
2.1 Introduction: Overview of Chapter Contents.....	24
2.2 Theoretical Foundations and Fundamentals of Speech Recognition.....	24
2.2.1 General Structure of a Speech Recognition System.....	24
2.2.2 Nature of Audio Data.....	26
2.2.3 Feature Extraction.....	29
2.2.3.1 Raw Signal as Input.....	30
2.2.3.2 Cepstral Frequency Coefficients.....	31
2.2.3.3 Spectrogram.....	32
2.2.4 Acoustic Modeling.....	33
2.2.4.1 Gaussian Mixture Models as a Phonetic Model.....	33
2.2.4.2 Deep Belief Neural Networks as a Phonetic Model.....	34
2.2.4.3 Convolutional Neural Networks as a Phonetic Model.....	35
2.2.4.4 Recurrent Neural Networks as a Phonetic Model.....	36
2.2.5 Theoretical Foundations of Speech Recognition System Architectures.....	37
2.2.6 Sounds and Linguistic Characteristics of Spoken Data.....	38
2.2.7 The Role of the Language Model in Speech Recognition.....	39
2.2.8 Linguistic Aspects of the Persian Language:.....	40
2.2.9 Advancements in Speech Recognition Systems: Research Findings.....	44
2.2.9.1 Evaluation Metrics: Word Error Rate (WER).....	44

2.2.9.2 Speech Corpora.....	45
2.2.9.3 LibriSpeech Dataset.....	46
2.2.9.4 Wall Street Journal Dataset.....	46
2.2.9.5 TED-LIUM Dataset.....	46
2.2.9.6 TIMIT Dataset.....	47
2.2.9.7 FarsDat Dataset.....	47
2.2.9.8 Word Error Rate in the Libri Speech Dataset.....	48
2.2.9.9 Word Error Rate in the Wall Street Journal Speech Dataset.....	48
2.2.9.10 Word Error Rate in the TED-LIUM Speech Dataset.....	49
2.2.9.11 Phonetic Error Rate in the TIMIT Speech Dataset.....	49
2.2.9.13 Phonetic Error Rate in the Farsdat Speech Dataset.....	50
2.2.10 Challenges in Speech Recognition Systems.....	51
2.3 Research Background and Literature Review.....	52
2.3.1 Deep Belief Network with Hidden Markov Model (DBN/HMM).....	53
2.3.2 Recurrent Neural Network Model (RNNs).....	54
2.3.3 Bidirectional Recurrent Neural Network Model (Bi-RNNS).....	54
2.3.4 Convolutional Neural Network Model (CNNs).....	54
2.3.5 End-to-End Deep Neural Network Model.....	55
2-4 Literature Review Summary.....	55
2-5 Summary of Contents.....	56
Chapter 3: Research Methodology.....	57
3-1 Introduction: Overview of Chapter Contents.....	57
2-3 FarsAva: Persian Speech Corpus.....	57
3-3 Collection of FarsAva Speech Dataset.....	58
3.3.1 Speech data collection methods.....	58
3.3.2 Sources of FarsAva Speech Data.....	60
3-3-3 Data Preprocessing and Text Normalization.....	61
3-3-3-1 The Challenge of Text Normalization in Persian Language.....	61
3-3-3-2 Normalization Procedures.....	62
3.3.4 Phonetic Dictionary.....	64
3.3.5 Language Model.....	66
3.3.6 Method of Preparing the Persian Language Model.....	67
3.3.7 Testing Datasets.....	68
3-4 Research Variables.....	69
3.5 Measuring Methods.....	70
3.6 Implementation and Training Approach.....	70
3.6.1 Data Prepration and Training of the GMM/HMM Model.....	70
3.6.1.1 Data Preparation.....	71
3.6.1.2 Feature Extraction.....	72
3.6.1.3 Monophone Model Training.....	72
3.6.1.4 Triphone Model Training.....	72
3.6.2 Data Prepration and Training of the DBN/HMM Model.....	73
3.6.2.1 Data Preparation.....	73
3.6.2.2 Deep Belief Neural Network Pre-training.....	74

3.6.2.3 Training and Fine-Tuning of the Deep Belief Network.....	74
3.6.3 Data Prepration and Training of an end-to-end RNN Model.....	75
3.6.3.1 Data Preparation.....	75
3.6.3.2 End-to-End Training of Deep Recurrent Neural Network.....	76
Chapter 4: Analysis of Research Findings.....	78
4.1 Introduction: Overview of Chapter Contents.....	78
4.2 Findings and Results (WER).....	78
4.2.1 First Part: WER Results on the First Test Dataset.....	79
4.2.2 Second Part: WER Results on the Second Test Dataset.....	80
4.2.3 Third Part: WER Results on the Third Test Dataset.....	82
4.2.4 Perplexity Results on Persian Language Model.....	83
4.2.5 WER Results based on Different Language Models.....	84
4.3 Hypotheses Testing and Research Questions.....	85
4.4 Summary of Findings.....	86
Chapter 5: Conclusion.....	87
5.1 Introduction: Overview of Chapter Contents.....	87
5.2 Research Summary.....	87
5.3 Summary of Findings.....	88
5.4 Discussion: A Comparative Analysis of Research Results.....	89
5.4.1 First Hypothesis.....	89
5.4.2 Second Hypothesis.....	90
5.4.3 Third Hypothesis.....	91
5.5 Conclusion: Key Research Message.....	91
5.6 Research Limitations.....	92
5.7 Future Research Suggestions.....	93
Appendix A - Theoretical Foundations of Language Models.....	94
Overview of Language Modeling.....	94
N-gram Language Models: Probability Estimation through Counting.....	95
Probabilistic Neural Network for Language Modeling.....	97
Probabilistic Neural Language Model based on Feed-Forward Neural Network.....	97
Probabilistic Neural Language Model based on Recurrent Neural Network.....	99
Appendix B - Gaussian Mixture Model.....	101
Gaussian Distribution.....	101
Multivariate Normal Distribution.....	101
Gaussian Mixture Models.....	102
Expectation Maximization Algorithm.....	103
Gaussian Mixture Model for Probability Distribution Function of Speech Features.....	104
Appendix C - Fundamentals of Deep Neural Networks.....	107
Appendix D - Fundamentals of Recurrent Neural Networks.....	109
Feed-Forward Recurrent Neural Networks.....	109
The Challenge of Long-Term Dependencies.....	110
Recurrent Neural Networks with Long Short-Term Memory.....	112
Appendix E - Connectionist Temporal Classification (CTC).....	115
References.....	121

List of Figures, Equations and Tables

Equation (0-1).....	12
Table (1-1) Definition of Technical Abbreviations.....	18
Equation (1-1).....	21
Equation (2-1).....	25
Figure (2-1) Probabilistic equation of speech recognition problem in detail.....	25
Figure (2-2) The General Structure of the Speech Recognition System.....	26
Equation (2-2).....	27
Figure (2-3) Depiction of a moving sound wave.....	27
Equation (2-3).....	28
Figure (2-4) Sampling of the continuous signal in the direction of time. Image from (Wikipedia, 2003).....	28
Figure (2-5) Quantization of a continuous signal along the signal range. Image from (Wikipedia, 2003).....	29
Figure (2-6) One second of the output of the WavNet. Image from (Oord et al., 2016)...	31
Equation (2-4).....	32
Figure (2-7) Short-Time Fourier Transform (STFT) for 20 milliseconds audio signal segment. Image from (Geitgey, 2016).....	32
Figure 2-8: Full spectrum of a speech signal containing the word "Hello". Image from (Geitgey, 2016).....	33
Figure (2-9) The structure of a speech recognition system that uses the Gaussian Mixture Model as the phonetic model. Image from (Maas, 2018).....	34
Figure (2-10) The structure of a speech recognition system that uses a deep belief neural network as phonetic model. Image from (Dahl et al., 2012).....	35
Figure (2-11) The structure of a speech recognition system that uses a convolutional neural network as a phonetic model. Image from (Mitra et al., 2017).....	36
Figure (12-2) Applying convolution layer and maximization layer. Image from (Zhang et al., 2017).....	36
Figure (2-13) Portrays the structure of the speech recognition system utilizing a recurrent neural network as the phonetic model. Image from (Amodei et al., 2016).....	37
Figure (2-14) Vowels sounds in Farsi.....	41
Table (2-1) Phonetic table of Farsi language. Table from (Sameti et al., 2011).....	43
Equation (2-5).....	45
Equation (2-6).....	45
Table (2-2) Word Error Rate on the LibriSpeech dataset in different articles.....	48
Table (2-3) Word Error Rate on the Wall Street Journal Speech Dataset in Various Articles.....	49
Table (2-4) Word Error Rate on the TED-LIUM Speech Dataset in Various Articles.....	49
Table (2-5) Phoneme Error Rate (PER) on the TIMIT Speech Dataset in Various Articles..	50
Table (2-3) Word Error Rate on the Wall Street Journal Speech Dataset in Various	

Articles.....	50
Figure (2-15) Challenging factors in speech recognition systems. Image from (Yu & Deng, 2014).....	51
Figure (3-1) time alignment of sounds in Timit and Farsdat speech data.....	59
Table (3-1) general characteristics of Farsava speech data.....	61
Figure (3-2) Depicts the many-to-many bidirectional recurrent neural network model utilized in this research. Image from (Yao & Zweig, 2015).....	65
Figure (3-3) Illustrates the symbols utilized to represent each of the Persian phonemes... 66	
Figure (3-4) Displays a section of the phonetic dictionary.....	66
Table (3-2) Statistics of Text Data.....	67
Figure (3-5) A part of the three-gram language model in the ARPA format.....	68
Table (3-4) Statistics of Primary Testing Dataset.....	69
Table (3-4) Statistics of News Program Testing Dataset.....	69
Table (3-5) Statistics of Third Testing Dataset.....	69
Figure (3-6) Hidden Markov Model based on monophonic model and triphonic model	73
Figure (3-6) Model error reduction diagram in different training epochs by Tensorboard 77	
Table (4-1) Word Error Rates on the First Test Dataset (460 hours of training data).....	79
Table (4-2) Word Error Rates on the First Test Dataset (1500 hours of training data)...	79
Table (4-3) Word Error Rates on the First Test Dataset (5000 hours of training data)...	79
Diagram (4-1) Overview of Word Error Rate on the first testing dataset.....	80
Table (4-4) Word Error Rates on the Second Test Dataset (460 hours of training data).	80
Table (4-5) Word Error Rates on the Second Test Dataset (1500 hours of training data)...	81
Table (4-6) Word Error Rates on the Second Test Dataset (5000 hours of training data)...	81
Diagram (4-2) Overview of Word Error Rate on the second testing dataset.....	81
Table (4-7) Word Error Rates on the Third Test Dataset (460 hours of training data)....	82
Table (4-8) Word Error Rates on the Third Test Dataset (1500 hours of training data).	82
Table (4-9) Word Error Rates on the Third Test Dataset (5000 hours of training data).	82
Diagram (4-3) of the error rate of words on the third experimental data set.....	83
Table (4-10) Perplexity result on 3-gram Persian Language Model with Different Smoothing Methods.....	83
Table (4-11) Perplexity result on 4-gram Persian Language Model with Different Smoothing Methods.....	84
Table (4-12) Perplexity result on 4-gram Persian Language Model with Different Smoothing Methods.....	84
Table (4-13) WER result on the second Test dataset based on using different Language Models (460 hours of training data).....	85
Table (4-14) WER result on the third Test dataset based on using different Language Models (460 hours of training data).....	85

Equation (A-1).....	95
Equation (A-2).....	96
Figure (A-1) General schematic of a probabilistic neural language model using a feed-forward neural network. Image adapted from (Bengio et al., 2000).....	98
Equation (A-3).....	98
Figure (A-2) Schematic of a language model based on recurrent neural network with long short-term memory neurons. Image from (Jin, 2017).....	99
Equation (B-1).....	101
Equation (B-2).....	101
Figure (B-1) Two-component Gaussian Mixture Model.....	102
Equation (B-3).....	102
Equation (B-4).....	103
Figure (B-2) Steps of the Expectation Maximization Algorithm.....	104
Equation (C-1).....	107
Figure (C-1) A restricted Boltzmann neural network comprising a visible layer and a hidden layer. Image source: (Restricted Boltzmann Machines, 2018).....	107
Figure (C-2) A deep belief neural network composed of Restricted Boltzmann Machines. Image source: (Deep Belief Networks, 2018).....	108
Figure (D-1) A simple recurrent neural network and its unfolded representation in time. Image source: (Karpathy, 2015).....	109
(Equation D-1).....	110
(Equation D-2).....	110
Figure (D-2) Short-term dependencies in a recurrent neural network. Image source: (Karpathy, 2015).....	111
Figure (D-3) Long-term dependencies in a recurrent neural network. Image source: (Karpathy, 2015).....	112
Figure (D-4) Recurrent module in a single-layer simple recurrent neural network. Image source: (Karpathy, 2015).....	113
Figure (D-5) Recurrent module in a single-layer Long Short-Term Memory network. Image source: (Karpathy, 2015).....	113
(Equation D-3).....	113
(Equation D-4).....	113
(Equation D-5).....	114
(Equation D-6).....	114
(Equation D-7).....	114
Figure (D-6) LSTM neuron. Image source: (Graves et al., 2013).....	114
Figure (E-1) Output of the Connectionist Temporal Classification after removing ϵ and merging repeated labels. Image source: (Hannun, 2017).....	116
Figure (E-2) Valid and non-valid aligned strings in the temporal alignment clustering algorithm. Image source: (Hannun, 2017).....	117
(Equation E-1).....	117
Figure (E-3) Merging Paths with the Same Output. Image source: (Hannun, 2017).....	118

(Equation E-2).....	118
(Equation E-3).....	118
(Equation E-4).....	118
Figure (E-4) Prefix search decoding with Dictionary $\{\epsilon, a, b\}$ and a extension value of three. Image source: (Hannun, 2017).....	119
Figure (E-5) Prefix search decoding modified by merging similar aligned atrings. Image source: (Hannun A., 2017).....	120
(Equation E-5).....	120

Abstract

Automatic speech recognition has been a formidable challenge for decades. With the advent of deep learning algorithms and access to vast amounts of speech data, coupled with the exponential growth of computational power over the last decade, we have come remarkably close to resolving this critical issue. Nevertheless, the realization of a language-independent speech recognition system remains a distant goal, and contemporary speech recognition systems continue to be trained specifically for recognizing particular languages.

This thesis endeavors to design and develop an automatic speech recognition system tailored to the Persian language. Language-dependent speech recognition systems present their unique linguistic challenges. Different languages are built upon distinct infrastructures, encompassing diverse phonemes, dictionaries, and grammatical structures, all of which contribute to the intricacy of a language-dependent speech recognition system. An additional challenge faced by such systems is the necessity for relevant training data in the target language.

The FarsAva dataset, a collection of Persian speech data, has been meticulously gathered and produced during the research conducted for this thesis. Utilizing this dataset, a speech recognition system based on deep recurrent neural networks with long and bidirectional short-term memory (LSTM) has been trained for the first time on the Persian language. Additionally, two models, the Gaussian Mixture Model - Hidden Markov Model (GMM/HMM), and the Deep Belief Neural network model - Hidden Markov Model (DBN/HMM), have also been trained to facilitate a comparative analysis of results using the FarsAva training data. Throughout this thesis, we have endeavored to examine each of these challenges thoroughly and propose effective methods to overcome these obstacles.

Introduction

The primary objective of a speech recognition system is to convert an audio signal from a speaker into corresponding text. For humans, speech recognition and transcription are innate and effortless abilities. However, despite several decades of engineering efforts, designing and implementing an efficient automatic speech recognition system remains a challenging task, and current systems are still far from achieving human-level accuracy in unrestricted and realistic environments.

Large vocabulary speech recognition (LVSR) pertains to a system equipped with an open or extensive dictionary. This concept also entails the system's independence from a specific speaker. A speech recognition system with a comprehensive lexicon should be capable of real-time recognition of both conversational and formal speech data, including official and written material. Moreover, these systems must exhibit resilience to various recording environments, both structured and unstructured noise, as well as background noise. Speech recognition with an extensive general dictionary finds applications ranging from basic tasks like converting audio content into text and transcription to more sophisticated applications, such as voice interfaces, voice-based search, conversational systems, speech data retrieval, and various other applications.

Acoustic variation in speech data encompasses three fundamental characteristics. The first aspect pertains to the variation in the spoken text itself. In large vocabulary speech recognition systems, it is impractical to collect data for every sentence or word within a language. The second aspect involves the inherent acoustic differences among various speakers. Each individual possesses a distinct voice, unique speaking style, and different accents. The third aspect involves variations in the environment and environmental noises. Anything beyond the original speech signal, such as background noise, overlapping speech, microphone hardware effects, etc., is considered noise.

The mathematical modeling of the speech recognition problem commences with a string of τ extracted features vectors derived from the audio signal $X = x_1 \dots x_\tau$, where τ represents the time steps. It is assumed that this string of extracted features serves as a mapping of T words as $W = w_1 \dots w_t$, and τ and t are not necessarily equal. This mathematical modeling necessitates a feature learning system and a search method to

identify the optimal sequence of words. More precisely, the objective of this modeling is to determine the most probable sequence of words based on a scoring function or probability distribution $S(X, W)$, which is estimated from data using machine learning techniques. Equation (1-0) represents the mathematical modeling equation of the speech recognition problem.

Equation (0-1)

$$\hat{W} = \operatorname{argmax} \{S(X, W)\}$$

Chapter 1 presents a thorough review of existing research and delves into the speech recognition problem, formulating it as a mathematical model. Furthermore, the chapter elaborates on the significance and necessity of this research, while describing the independent and dependent variables involved. The research assumptions, fundamental inquiries, and the chosen research methodology are also outlined in this initial chapter.

Chapter 2 serves as a comprehensive reference to background and theoretical aspects, encompassing the general structure of a speech recognition system, Gaussian mixture models, hidden Markov models, as well as various neural networks such as deep belief networks, recurrent networks, convolutional networks, and the deep learning algorithms to be employed in this thesis. Furthermore, within Chapter 2, a review of the advancements made globally in the field of speech recognition is presented, shedding light on the efforts undertaken in this domain and providing an in-depth literature review of the problem at hand.

Chapter 3 initiates by introducing and providing a detailed description of the collection of FarsAva speech data, which has been meticulously compiled in alignment with the objectives of this thesis. Notably, FarsAva stands as the first large-scale collection of Persian language speech, encompassing over five thousand hours of speech data. Within this chapter, we also delve into data collection and pre-processing methods, discussing the associated challenges, as well as outlining the methodology used to produce the phonetic dictionary and language model.

Continuing with the third chapter, we present a comprehensive exposition of the methods employed to train a speech recognition system utilizing diverse architectures, while also elaborating on the approach taken to conduct the experiments thoroughly.

Chapter 4 presents the results obtained from the experiments conducted in Chapter 3, focusing on the key research variables. These results encompass the system's word error rate concerning the research variables, along with the outcomes of the language model trained in Chapter 3.

Chapter 5 is dedicated to the examination and comparison of the obtained results. The research hypotheses are thoroughly scrutinized, and the conclusions and implications derived from this study are thoughtfully presented.

The appendices section encompasses the theoretical foundations and algorithms utilized in this thesis. These include the theoretical underpinnings of linguistic models, the structure of Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), along with the training algorithms applied to these models. Moreover, the architecture of Deep Neural Networks (DNNs) and Deep Recurrent Neural Networks (RNNs), as well as the Connectionist Temporal Classification (CTC) algorithm, are comprehensively elucidated.

Chapter 1: Thesis Outline

1.1 Introduction: Outline of the Research

This chapter provides a comprehensive overview of the thesis. The initial section offers a broad introduction to the speech recognition problem. The subsequent segment delves into the significance and necessity of this research. Sections 1.4 highlight the innovative aspects of this study, distinguishing it from previous research. Following this, the subsequent three sections are dedicated to elucidating the research objectives, fundamental research questions, and hypotheses. In the eighth part, a table of technical terms used in this thesis, along with their corresponding Persian synonyms, is provided.

Furthermore, the ninth section describes the research method and methodology, explicitly defining the independent and dependent variables of the research. The tenth part expounds upon the data collection method, while the subsequent segment discusses the statistical population and data volume. The final part of this chapter focuses on data analysis methods and tools.

1.2 Statement of the General Research Problem

The research problem at the core of this study is the challenge of speech recognition, which can be succinctly defined as the process of converting spoken language into its corresponding text form. While speech recognition is a commonplace and intuitive skill for individuals fluent in a particular language, the same task poses complex challenges when performed by a computer system, classifying it as a problem within the domain of intricate artificial intelligence.

The fundamental ambiguity underlying this problem stems from our limited understanding of how speech is identified and processed in the human mind. As a consequence, various models have been proposed to simulate this cognitive process.

Multiple variables significantly impact the effectiveness of a speech recognition system. These variables include the volume of data used for training, language nuances and grammar intricacies, speaking speed, variations in speech patterns attributable to diverse speakers, accents, and dialects, background noises and environmental sounds,

speech continuity, and the choice between a limited or extensive dictionary, among others.

The primary aim of this research is to emulate the speech recognition system in the Persian language, using the closest model to the human mind, namely neural networks, and subsequently compare its outcomes with other models available in the Persian language. The problem of speech recognition can be approached from a perspective of mathematical probabilities, as follows.

The input of a speech recognition system comprises a series of feature vectors τ , representing audio information extracted from an audio signal in the form $X = x_1 \dots x_\tau$. Here, the variable τ denotes time steps. The core assumption of the problem is that this sequence of feature vectors τ essentially corresponds to the word T in the form $W = w_1 \dots w_t$. It is essential to note that the lengths of the two aforementioned sequences, τ and t , may not necessarily be equal. This problem framework necessitates a learning model for the extracted features and a search method to identify the most optimal and probable string of words W . Alternatively, in more precise terms, the primary objective is to identify the most probable string of words based on a scoring function or, in essence, a probability distribution function, learned from speech data.

1.3 The Significance and Necessity of the Research

The speech recognition system is inherently language-dependent, and one crucial requirement of this research is to explore and implement the speech recognition system in the Persian language using deep neural networks. Notably, such an implementation has not yet been realized for Persian, resulting in a fundamental research gap in the field of artificial intelligence pertaining to this language.

The aspiration is that this research will serve as a scientific breakthrough in the domain of speech recognition for the Persian language, leveraging deep neural networks as a stepping stone to propel advancements in this system. From a practical standpoint, the existence of a robust speech recognition system in Persian will serve as a platform for new applications and revolutionize the way Persian speakers interact with computer systems.

Currently, many English programs, applications, and smart systems have integrated speech interfaces to provide efficient services to users, a luxury that has been

lacking for Persian speakers due to the limited progress of the Persian speech recognition system. The integration of speech interfaces can significantly enhance communication speed between humans and computer systems, increasing it up to four times. While the average typing speed is approximately forty words per minute, an individual can speak over one hundred and fifty words per minute. Moreover, speech interfaces offer the advantage of facilitating access and interaction with computer systems in situations where typing may not be feasible, such as while driving.

1.4 The Aspect of Novelty and Innovation in Research

This study encompasses various novel and innovative aspects that contribute to the advancement of speech recognition in the Persian language through the utilization of deep neural networks and deep recurrent neural networks. The distinctive features of the Persian language, such as additional phonetic variations, are incorporated to enhance the accuracy of the system.

The generation of Farsava, a comprehensive Persian speech dataset specifically curated to evaluate the proposed method, plays a vital role in the research's originality. This dataset serves as a valuable resource for assessing the system's performance and validating its effectiveness.

Additionally, a transliteration system of Persian words is devised using recurrent neural networks, enabling the transformation of Persian words into their equivalent phonetic representations. This transliteration system further contributes to the research's novelty, aiding in the alignment of phonetic information with text data.

Furthermore, an extensive phonetic dictionary encompassing over five hundred thousand transliterated Persian words is meticulously created. This dictionary forms a critical foundation for the research, enriching the understanding of phonetic structures and enabling improved recognition accuracy.

1.5 Specific Objectives of the Research

The primary objective in the realm of speech recognition systems is to attain systems that surpass the average human error in speech recognition accuracy.

In this research, the overarching aim is to enhance Persian speech recognition systems by utilizing Deep Recurrent Neural Networks and improve results compare to

previous GMM/HMM models. The ultimate goal is to achieve a significant reduction in recognition errors compared to previous systems.

Furthermore, a pragmatic objective of this research is to curate and prepare a large-scale dataset specifically tailored for Persian speech recognition. This dataset will serve as a valuable resource for the development and evaluation of robust speech recognition systems in Persian.

1.6 Main Research Questions

The research endeavors to address the following key questions:

- To what extent do deep neural network models improve the Word Error Rate (WER) of a speech recognition system compared to Gaussian and Markov models?
- Among the deep neural network models considered, which one exhibits the least error?
- How does the volume of data influence the accuracy of different models?

1.7 Research Hypotheses

The research hypotheses encompass the following assertions:

- Deep neural networks exhibit a significant 10% improvement in word error rate compared to Gaussian and Markov models.
- Among the models evaluated, the Deep Belief Network model demonstrates the lowest error rate, contingent on the volume of data considered.
- The data volume directly influences model accuracy. Various models demonstrate a reduction in error up to a certain threshold as the data volume increases, after which the error stabilizes.

1.8 Definition of Technical Abbreviations

Here are the abbreviations used in this thesis and what they stand for.

#	Technical Abbreviations	Definition
1	WER	Word Error Rate
2	PER	Phone Error Rate
3	LM	Language Model
4	HMM	Hidden Markov Model
5	GMM	Gaussian Mixture Model
6	DNN	Deep Neural Network
7	CNN	Convolutional Neural Network
8	DBN	Deep Belief Network
9	RNN	Recurrent Neural Network
10	LSTM	Long Short-Term Memory
11	CTC	Connectionist Temporal Classification
12	MMI	Maximum Mutual Information
13	MPE	Minimum Phone Error
14	SAT	Speaker Adaptive Training
15	MLLR	Maximum Likelihood Linear Regression
16	LDA	Linear Discriminant Analysis
17	MFCC	Mel Frequency Cepstral Coefficients

Table (1-1) Definition of Technical Abbreviations

1.9 Research Methodology

1.9.1 Research Method based on Purpose, Data, and Execution

The primary aim of this research is to investigate and implement a Persian speech recognition system using deep neural networks, with the goal of enhancing its performance compared to Gaussian models and Hidden Markov Models. The implementation process for this research encompasses the following steps:

1. **Literature Review:** Given the limited availability of scientific resources pertaining to the development of a Persian speech recognition system based on Deep Neural Networks, with most studies primarily focused on Gaussian and Markov models, this section will extensively review scientific articles related to speech recognition systems developed for other languages, particularly English. By carefully analyzing the structural distinctions and commonalities between English and Farsi, we can leverage solutions proposed by other researchers for common aspects and seek remedies for the identified differences.
2. **Creating a Phonetic Dictionary:** A fundamental aspect of speech recognition systems lies in having a comprehensive phonetic dictionary tailored to the target language. In the case of the Persian language, there are specific differences that require attention in this domain. Given that speech recognition relies on phonetic recognition, the dictionary must be designed to establish a coherent and accurate mapping between words and their corresponding phonetic representations. This phonetic dictionary can be generated through manual curation or by employing a learning model trained to transliterate words effectively.
3. **Data Preparation:** Speech recognition systems are essentially learning models that undergo training using data specific to a particular language, enabling them to recognize speech within that language. In the context of the Farsi language, the scarcity of such specialized data poses a challenge, as the existing corpus is limited in size. To address this limitation, we will undertake the task of collecting the necessary data for our research, a topic which will be expounded upon in the subsequent data collection section.
4. **Creation of the Baseline Model:** In line with the research objective of enhancing the performance of speech recognition systems compared to Gaussian and Markov models, this phase involves the development and training of a Persian speech recognition system using the aforementioned models and the collected data. Once the model has undergone training and its error on the test data is determined, this error is identified as the baseline error. To assess progress toward the goal, the improvement in results will be evaluated with respect to this baseline error. It is essential to note that the error of a speech recognition system is highly contingent on the data used for training. As a result, it is not feasible to directly compare the results with those of previously reported Gaussian and

Markov speech recognition models. Thus, this step is indispensable for accurate evaluation.

5. **Data Preprocessing:** The intrinsic characteristics of audio data make it unsuitable for direct utilization in training speech recognition systems. Therefore, prior to model training, a series of crucial pre-processing steps are imperative to ensure data consistency and suitability. The following procedures are to be performed:
 - *Standardizing the Sampling Rate:* It is imperative to establish a uniform sampling rate for all data samples, maintaining a consistent value of 16,000 Hz throughout.
 - *Channel Normalization:* Ensuring uniformity in the number of audio channels, all input data must contain a single audio channel.
 - *Feature Vector Extraction:* Commonly adopted in speech recognition systems, the Mel Frequency Cepstral Coefficients (MFCCs) vector represents the prevalent feature vector used in this research for data transformation and processing.
6. **Deep Neural Network Model Training:** This section encompasses the implementation of Deep Neural Network models subsequent to data pre-processing and the establishment of a foundational model. The objective is to acquire the WER percentage on a standardized test dataset. Detailed elucidation of the tools employed for the implementation and training of these deep neural network models shall be provided in the ensuing sections.

1.9.2 The Investigated Variables and Description of Examination and Measurement

The primary independent variable under investigation is the volume of data utilized for training. Given the nature of the training data, comprising audio files paired with corresponding texts, data volume is quantified in hourly units. Another key independent variable pertains to the acoustic model or architecture employed in the speech recognition system. Additionally, a secondary independent variable is considered, namely, the language model utilized for speech recognition.

The principal dependent variable examined in this research is the error percentage of the speech recognition system when tested against experimental data.

The error is assessed based on the number of misidentified words detected by the system. This variable is commonly referred to as the Word Error Rate (WER), and its computation method is as follows.

Equation (1-1)

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

In the aforementioned equation, S represents the number of words requiring replacement to generate the correct string of words. D indicates the number of words to be removed for achieving the correct string, and I signifies the number of words to be included in the string to produce the accurate sequence of words. C represents the total count of words that are both correct and present in the correct string, while N denotes the overall word count in the context.

Additionally, there exists another independent variable referred to as the vocabulary size, which measures the number of words that the speech recognition system is capable of recognizing. This variable is determined by the count of words present in the dictionary utilized during the word identification process.

Various other variables also exert an influence on the performance of a speech recognition system. These factors include the rate of speech, variations in speech patterns attributed to different speakers, accents, dialects, environmental noises, and the continuity or interruption of speech. While acknowledging the significance of these variables, their precise quantification remains challenging. Therefore, efforts are made to maintain their consistency as much as possible during the data collection phase.

Another fundamental independent variable pertains to the model structure and the number of parameters employed in the neural network model. The choice of a specific deep neural network model, along with key parameters such as the number of layers, neurons per layer, and activation functions, significantly impacts the dependent variable of the problem, namely the word error rate. Given the tractability of these parameters, their determination and measurement can be performed with relative ease, and the findings should be accordingly documented concerning the state of these variables.

1.10 Data Collection Methods and Tools

The data for implementing, training, and testing a speech recognition system necessitates a specific structure. The data structure comprises a set of audio files, each accompanied by its corresponding transcribed text, collectively constituting a comprehensive dataset.

In this research endeavor, we meticulously collected and prepared the Farsava dataset to align with the research objectives. The third chapter elaborates extensively on the data collection methodology employed and the specialized tools harnessed for this purpose.

1.11 Data Analysis Methods and Tools

In this research, various tools have been utilized for data preprocessing and the implementation of deep neural network models. It is important to note that many of these custom-made tools are contingent on the language being processed. For instance, the text normalization tool is intricately tied to the language in question. Consequently, to meet the requirements of this thesis, several language-dependent tools have been purposefully designed and developed, and their comprehensive explanations can be found in the subsequent chapters. The tools employed in this research include:

Operating System: Linux Ubuntu 16.04

Data Preprocessing Tool: Sox software, ffmpeg

Gaussian and Markov Model Implementation Tool: CMUSphinx tool

Deep Belief Network Model Implementation Tool: Kaldi tool

Recurrent Deep Neural Network Model Implementation Tool: Tensorflow

Programming Languages: Python, C++, and Shell Scripting

1.12 Scope of Research: Thematic, Temporal, and Spatial Scope

This research is dedicated to the domain of speech recognition systems, with a specific focus on Persian language speech recognition systems. The temporal scope of this study encompasses the entire spectrum of historical attempts to produce speech recognition systems, from early endeavors to the latest achievements. As for the conducted

experiments, the most recent research advancements have been incorporated, ensuring relevance in terms of time. Geographically, the research and experiments have been confined to Iran; however, the outcomes of this study hold direct applicability for researchers in Farsi-speaking regions.

1.13 Summary of Contents

This chapter encompasses an overview of the research, beginning with the statement of the core problem and identification of the independent and dependent variables involved in the investigation. Moreover, it delves into the purpose and necessity of the research, emphasizing its innovative aspects. The research methodology is outlined briefly, providing an initial direction for the study. Additionally, the assumptions and fundamental research inquiries have been formulated, and poised for verification through experimentation. The chapter also offers a concise introduction to the data collection methods and data analysis tools that will be further elaborated upon in the third chapter.

Chapter 2: Theoretical Foundations and Literature Review

2.1 Introduction: Overview of Chapter Contents

This chapter delves into the theoretical underpinnings essential for our research and subsequently conducts a thorough review of relevant literature and prior studies concerning speech recognition systems. The theoretical and foundational section explores pivotal machine learning models directly applicable to speech recognition, particularly Gaussian Mixture Models, Hidden Markov Models, and various Neural Networks Models as Phonetic Models, which constitute key components employed in this thesis. Furthermore, it encompasses an elucidation of the general architecture of speech recognition systems, diverse feature extraction methods, and current advancements in the field. Given the specific focus on Persian speech recognition, a dedicated segment is allocated to delve into the cognitive linguistics of the Persian language and its distinctive phonetics.

The latter portion of this chapter presents an extensive literature review, examining prior research endeavors in the realm of speech recognition, both in a broader context and with a specific emphasis on studies conducted pertaining to Persian language speech recognition.

2.2 Theoretical Foundations and Fundamentals of Speech Recognition

2.2.1 General Structure of a Speech Recognition System

In this section, we provide a concise examination of the general structure of a speech recognition system. As previously stated in the thesis introduction, the core challenge of speech recognition involves seeking the most probable string of words based on the extracted features from the input speech data. Formally, we aim to find

$\text{argmax}_w P(W|O)$, where W represents a string of output words, O denotes a string of extracted and observed features, and more broadly, it represents the input audio signal.

Equation (2-1)

$$\text{argmax}_w P(W|O) = \text{argmax}_w P(O|W) P(W)$$

A more precise mathematical formulation is shown above with a probability expansion.

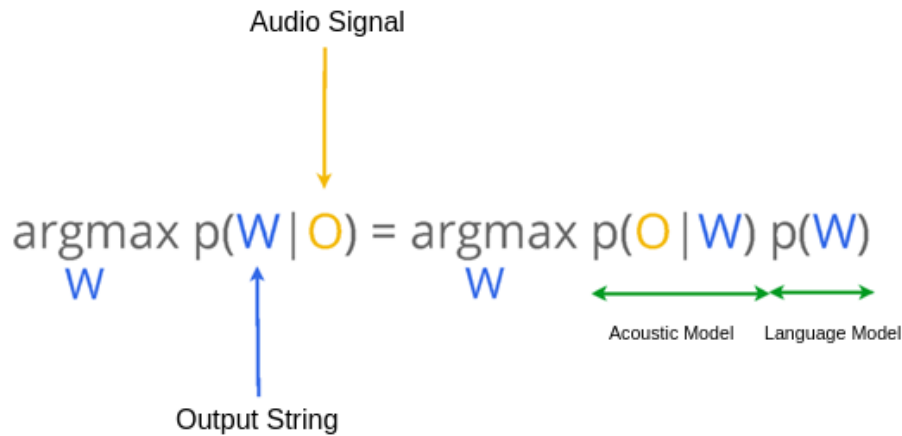


Figure (2-1) Probabilistic equation of speech recognition problem in detail

As evident from the formula and diagram provided above, $P(W|O)$ represents the probability of the output word string based on the input data, and $P(O|W)$ denotes the probability of encountering specific words given the observed features. The probability $P(O|W)$ is estimated through the acoustic model, while the likelihood $P(W)$ corresponds to the priori probability of that specific word string occurrence. This prior probability, or $P(W)$, is derived from modeling the linguistic connections, lexical dependencies, and the grammar and structure of the target language, which is commonly referred to as the Language Model (LM).

The general structure of a speech recognition system is illustrated in Figure (2-2). As depicted in the diagram, a speech recognition system can be subdivided into four main components. These components encompass input signal processing and feature extraction, indicated in blue, language modeling, acoustic modeling, and the final stage of searching for the most probable output string.

The primary objective of the first component, input signal processing and feature extraction, is to enhance the quality of the input signal by eliminating noise, appropriately segmenting the input, detecting periods of silence, and ultimately extracting the most representative features that characterize the input data.

The output of the first stage or the extracted features serves as the input of the next stage, which is the Acoustic Model. The Acoustic Model is generated by training a probabilistic model on the speech training data. After training, this model is used to estimate the most probable phoneme (Or character) based on the observed feature vector.

Next, the Language Model is applied to the previous output string. The task of the Language Model is to assign possible probabilities to each of the output hypotheses based on the correlation of words and the overall structure of the language in the target language.

The Language Model can estimate the probability $P(W)$ after training on a large text corpus. The decoding stage or searching for the best output by combining the probabilities of the Acoustic Model and the Language Model tries to produce the most probable output string of words.

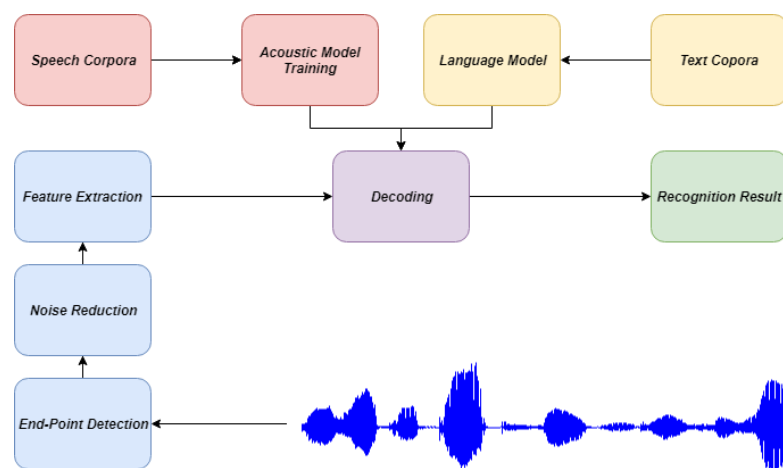


Figure (2-2) The General Structure of the Speech Recognition System

2.2.2 Nature of Audio Data

Sound waves are produced by the vibrations of an object, leading to the propagation of longitudinal waves in the surrounding medium, such as air or water. These propagating

waves are responsible for the sounds that we perceive. Sound waves consist of regions of low pressure (rarefactions) and regions of high pressure (compression).

Figure (2-3) illustrates a moving sound wave, where the top shaded bar represents the pressure variation within the wave. Brighter areas indicate low-pressure, progressive regions, while darker areas represent high-pressure, dense regions. The distance between two consecutive high-pressure regions or two consecutive low-pressure regions is known as the wavelength, highlighted in red.

This repetitive process continues indefinitely, with the typical wavelength of sound being around one meter. The wavelength, along with the speed of the wave, determines the pitch or frequency of the sound. The relationship between wavelength (λ), frequency (f), and speed (v) is given by the following equation:

Equation (2-2)

$$v = f * \lambda$$

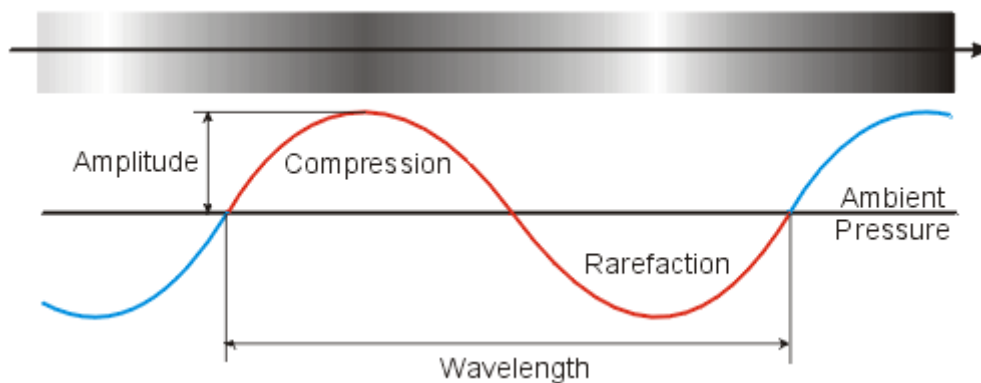


Figure (2-3) Depiction of a moving sound wave

In an environment with standard temperature and pressure, sound travels at a constant speed of 343 meters per second. Consequently, the frequency can be calculated by dividing the velocity by the wavelength. A longer wavelength results in a lower pitch of the sound, while the amplitude of the wave corresponds to its loudness. Larger amplitudes signify louder sounds.

To convert an analog signal into a digital one, discrete values are assigned to both the time and amplitude of the oscillation. This conversion process involves two main steps: sampling and quantization. Sampling is the process of representing a continuous-time signal with discrete values at specific time intervals. The resulting

representation is known as Pulse Amplitude Modulation (PAM), and all coding techniques used for pulse reconstruction are referred to as waveform coding.

The sampling is accomplished by measuring the amplitude value of the initial wave every T seconds, where T_s represents the sampling interval and $f_s = \frac{1}{T_s}$ denotes the sampling frequency. For example, a waveform with a sampling rate of 16 kHz means that an analog signal has been converted to digital with sampling intervals of 1/16000 seconds. This implies that the amplitude value has been sampled from the original analog signal once every 1/16000 seconds.

The Nyquist-Shannon sampling theorem stipulates that in order to perform sampling of an analog signal with a limited band without losing information, the sampling frequency must be greater than twice the maximum signal frequency component.

Equation (2-3)

$$f_s \geq 2 * f_{max}$$

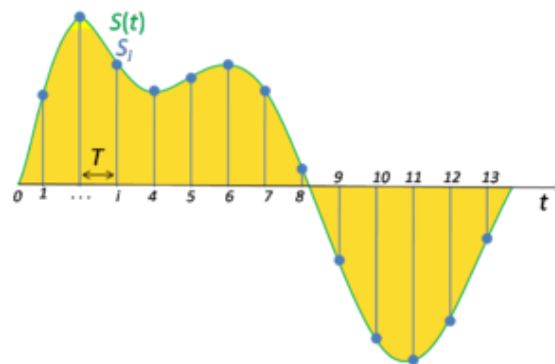


Figure (2-4) Sampling of the continuous signal in the direction of time. Image from (Wikipedia, 2003)

The second stage of the process involves quantization along the signal range. During this step, the continuous range of samples is encoded using discrete binary values, with each sample represented by w bits. This process is referred to as Linear Pulse Code Modulation or Lin-PCM. The continuous value of a sample is replaced by the nearest discrete value within the range of 2^w . The discrepancy between the original continuous signal and its digital representation is known as quantization error.

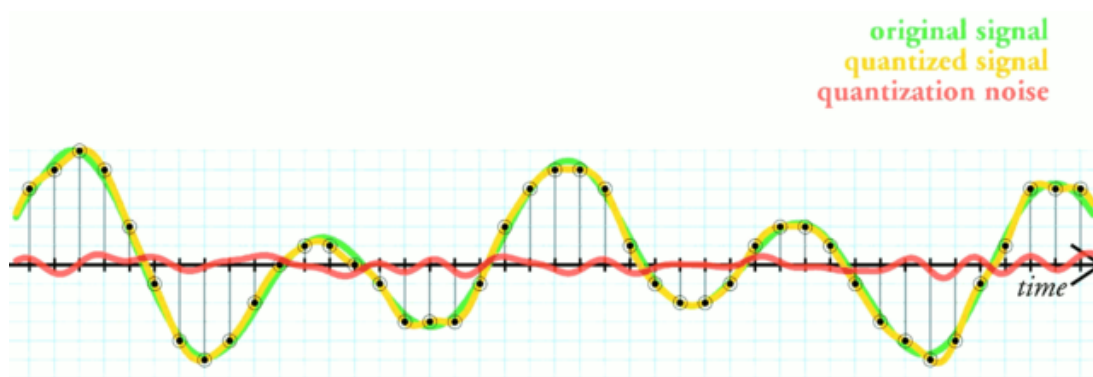


Figure (2-5) Quantization of a continuous signal along the signal range. Image from (Wikipedia, 2003)

2.2.3 Feature Extraction

The initial stage in a speech recognition system involves extracting pertinent features from the input speech data. Voice and speech data are rich in information, but not all of it is relevant to solving the speech recognition problem. The primary objective of feature extraction from speech data is to identify linguistic and speech-related aspects while disregarding extraneous information, such as background noise, ambient sounds, and speaker emotions (Davis & Mermelstein, 1980).

Numerous feature extraction methods have been proposed over the years to establish an optimal speech recognition system. Among these, Mel Frequency Cepstral Coefficients (MFCCs) stand out as one of the most renowned and extensively employed techniques in speech recognition systems. This distinctive feature for speech data extraction was introduced by Davis and Mermelstein in the 1980s, and it continues to be one of the most effective methods for feature extraction (Davis & Mermelstein, 1980). Prior to the advent of frequency coefficients as features, methods such as Linear Prediction Coefficients (LPCs) and Prediction Cepstral Coefficients (LPCCs) were the primary approaches used for feature processing and extraction, particularly in conjunction with Markov Hidden classifiers for speech data (Lyons, 2012).

With the advent of deep neural networks and advanced Acoustic Models, in addition to Frequency Coefficients (MFCCs), Linear Prediction Coefficients (LPCs), and Prediction Cepstral Coefficients (LPCCs), recent speech recognition systems have incorporated two other types of inputs. Leveraging the impressive achievements of Convolutional Neural Networks (CNNs) in image processing and computer vision tasks,

researchers have applied similar methodologies to phoneme classification and Acoustic Model training (Abdel-Hamid et al., 2012) (Sainath, 2013) (Deng et al., 2013). In this approach, the input for training the Acoustic Model is a feature known as the spectrogram, which can be viewed as a visual representation of an audio signal. This permits the utilization of methods akin to convolutional networks for phoneme classification.

The significant advancements in processing power, the availability of graphics card-based computing, and the exponential growth in training data have enabled the training of deeper and wider networks. Capitalizing on this potential and embracing the fundamental concept of representational learning in deep learning, researchers have begun moving towards end-to-end systems. As a result, some recent studies have employed the raw input signal to train the phonetic model (Oord et al., 2016) (Palaz & Collobert, 2015).

Various feature extraction methods, including Power Spectral Analysis (FFT), Mel Scale Cepstral Analysis (MEL), Relative Spectra Filtering of Log Domain Coefficients (RASTA), and First Order Derivative (DELTA) have been employed to derive relevant features from the audio signal. These methods are comprehensively elucidated in the work by (Shrawankar & Thakare, 2013).

2.2.3.1 Raw Signal as Input

Raw audio waveforms are signals with high time resolution, typically containing at least 16,000 samples per second. Notably, there have been highly successful endeavors to model probability distribution functions based on raw data using Neural Autoregressive Generative Models for images and text. In 2016, DeepMind Google introduced the WavNet, a generative neural network model specifically designed for audio data. This groundbreaking network is capable of directly producing audio signals, revolutionizing text-to-speech conversion systems. Moreover, the WavNet network can be effectively applied to the speech recognition problem by conditioning the input and output. This model relies on a type of neural network known as the Dilated Causal Convolution.



Figure (2-6) One second of the output of the WavNet. Image from (Oord et al., 2016)

While conventional speech recognition systems primarily rely on the features mentioned below for extraction, the possibility of utilizing raw audio data as input for certain neural networks should not be overlooked. Furthermore, one of the fundamental tenets of deep learning lies in the automatic extraction of features from raw data.

2.2.3.2 Cepstral Frequency Coefficients

Cepstral Frequency Coefficients, also known as Mel-Frequency Cepstral Coefficients (MFCCs), stands as the most widely adopted feature extraction method in speech recognition systems. Originally introduced by Davis and Mermelstein in the 1980s, MFCCs remain one of the most effective feature extraction approaches (Davis & Mermelstein, 1980). In signal processing, Mel-Frequency Cepstrum (MFC) represents a type of short-term spectral power signal representation. This representation is constructed using a linear cosine transform on the logarithm of the spectrum power, based on a non-linear mel scale. The distinctive characteristic of Cepstral Frequency Coefficients lies in the cumulative derivation of coefficients from the mel power of the frequency.

The procedure for obtaining Cepstral frequency coefficients is as follows:

1. Framing the signal into short segments, typically of 20 milliseconds duration.
2. Calculate the power spectrum (power) for each frame.
3. Summing up the energy of each filter through the application of the Mel filter bank on the spectrum power.
4. Computing the energies of all filter bank outputs.
5. Applying the discrete cosine transform to the logarithm of the filter bank energies.
6. Retaining the 2nd to 13th coefficients of the discrete cosine transform.

2.2.3.3 Spectrogram

A spectrogram serves as a visual representation of signal power across different frequencies over time in sound waves. In essence, it provides a graph of power or energy as a function of both time and frequency. Spectrograms exhibit the power of a signal at specific instances and frequencies, akin to other features utilized in speech recognition. They offer insights into how the power of a wave varies across different frequencies within a given time interval.

To compute the spectrogram of a signal, it is initially segmented into 20 millisecond intervals with overlapping segments. Subsequently, the squared value of the Short-Time Fourier Transform (STFT) is computed for each 20-millisecond audio segment, which can be represented by the equation below, where $s(t)$ denotes the signal and w represents the window width:

The output of this transformation on the audio components yields the signal power at various frequencies. Spectrograms are often visualized as two-dimensional graphs, with frequency depicted on the vertical axis and time on the horizontal axis. The intensity of colors in the spectrogram indicates the power of the signal at specific time-frequency points.

Equation (2-4)

$$spectrogram(t, w) = |STFT(t, w)|^2$$

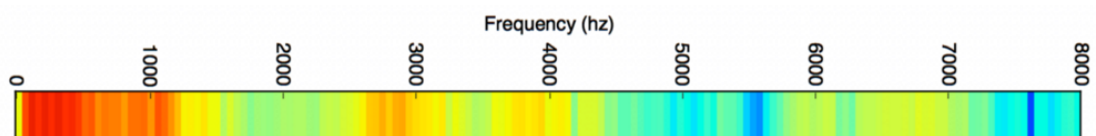


Figure (2-7) Short-Time Fourier Transform (STFT) for 20 milliseconds audio signal segment.

Image from (Geitgey, 2016)

In the process of calculating the Short-Time Fourier Transform for each 20-millisecond interval, the results are aggregated to form the complete spectrum of a signal. This process can be observed in Figure (2-8) below:

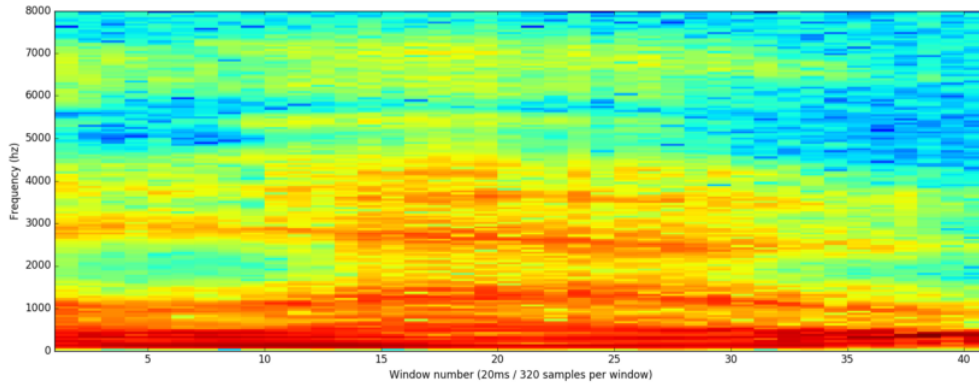


Figure 2-8: Full spectrum of a speech signal containing the word "Hello". Image from (Geitgey, 2016)

2.2.4 Acoustic Modeling

The acoustic, or phonetic model, serves as a crucial component within speech recognition systems, responsible for establishing the association between the audio signal and the language's sounds or phonetic elements that compose speech. This model undergoes training using data encompassing spoken audio and its corresponding textual or phonetic representation. By constructing a probabilistic model based on the provided data, the phonetic model establishes a link between the signal and the phonetic constituents of the spoken words.

As mentioned earlier, the primary objective of the phonetic modeler revolves around estimating the probability of $P(O|W)$. In this probabilistic formulation, O represents the extracted speech features, while W denoting the string of words or phonetic components within the speech data. The creation of a phonetic modeler can take various structural and architectural forms, which will be briefly explored in the ensuing sections.

2.2.4.1 Gaussian Mixture Models as a Phonetic Model

Gaussian Mixture Models (GMMs) stand as one of the earliest methods employed for constructing phonetic models. These models exhibit considerable efficacy in representing natural data through a combination of normal probability distribution functions. When employing GMMs to create a phonetic model, their primary task is to classify each feature vector of the audio signal into one of the language's sounds, or more precisely, into one of the three sub-categories associated with each sound.

In the general architecture of a speech recognition system that utilizes a Gaussian mixture model as its phonetic model, hidden Markov models (HMMs) are commonly employed to handle the temporal dependencies inherent in speech. Figure (2-9) below illustrates the structure of a speech recognition system that leverages Gaussian mixture models as its phonetic model.

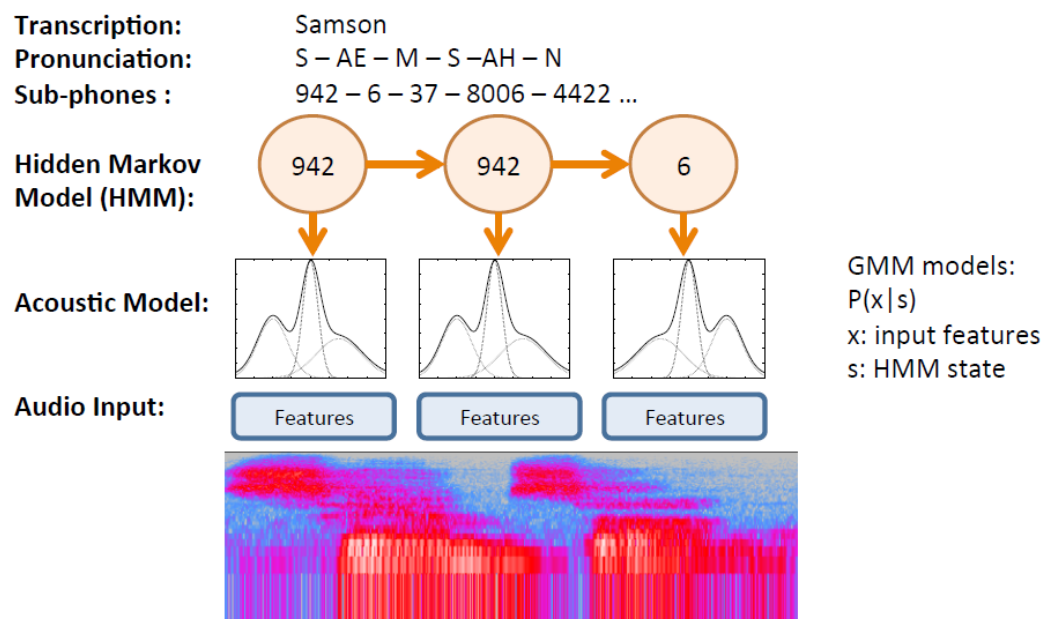


Figure (2-9) The structure of a speech recognition system that uses the Gaussian Mixture Model as the phonetic model. Image from (Maas, 2018)

2.2.4.2 Deep Belief Neural Networks as a Phonetic Model

Introduced by Dr. Geoffrey Hinton in 2006 (Hinton, 2006), deep neural networks have emerged as a potent tool for modeling natural data, including audio signals.

Deep Belief Networks (DBNs) have the potential to replace Gaussian Mixture Models within the architecture of a speech recognition system. Owing to their distinctive characteristics, parameter count, and training approach, deep neural networks have exhibited superior performance compared to Gaussian mixture models in phonetic modeling (Mohamed et al., 2012). As previously mentioned, deep neural networks can be seamlessly integrated into the structure of a speech recognition system in lieu of Gaussian Mixture Models, while Hidden Markov Models are employed to capture the temporal dependencies of speech features. Consequently, speech recognition systems that adopt this approach are referred to as "Deep Neural Network - Hidden Markov

Model" (DBN/HMM) systems. Figure (2-10) illustrates the generalized structure of a speech recognition system utilizing deep belief neural networks as its phonetic model.

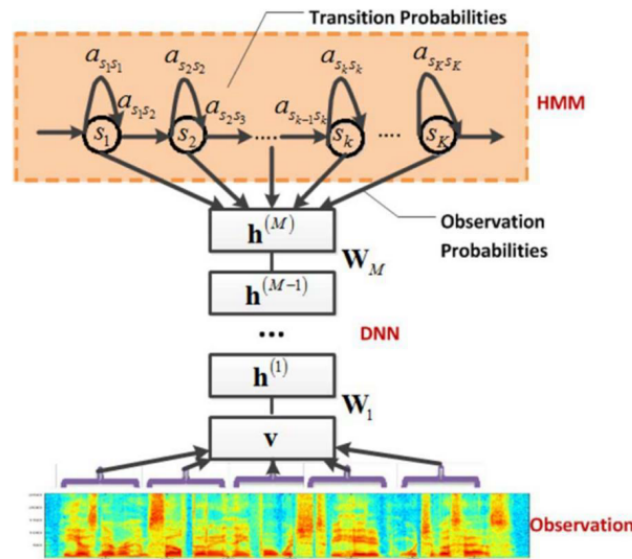


Figure (2-10) The structure of a speech recognition system that uses a deep belief neural network as phonetic model. Image from (Dahl et al., 2012)

2.2.4.3 Convolutional Neural Networks as a Phonetic Model

Convolutional Neural Networks (CNNs) have achieved remarkable success in tackling image processing and machine vision problems in recent years. Recognizing the similarity between audio data and images, both possessing two dimensions, namely time and frequency, researchers have sought to leverage CNNs by employing spectrogram-based feature extraction methods to address the task of phonetic modeling.

In various studies, novel approaches have been explored to integrate convolutional neural networks into the DBN/HMM models. One such endeavor involves creating a "Convolutional Hidden Markov Model," wherein a CNN replaces the DBN part in the DBN/HMM model (Abdel-Hamid et al., 2012) (Mitra et al., 2017). Additionally, researchers have endeavored to implement the temporal modeling aspect of the Hidden Markov Model using an end-to-end Convolutional Neural Network Model (Zhang et al., 2017). Figure (2-11) illustrates the generalized structure of an end-to-end speech recognition system incorporating convolutional neural networks as the phonetic model (Mitra et al., 2017). Furthermore, Figure (2-12) depicts the application of the

convolution layer and maximization layer in the network architecture, as described in (Zhang et al., 2017).

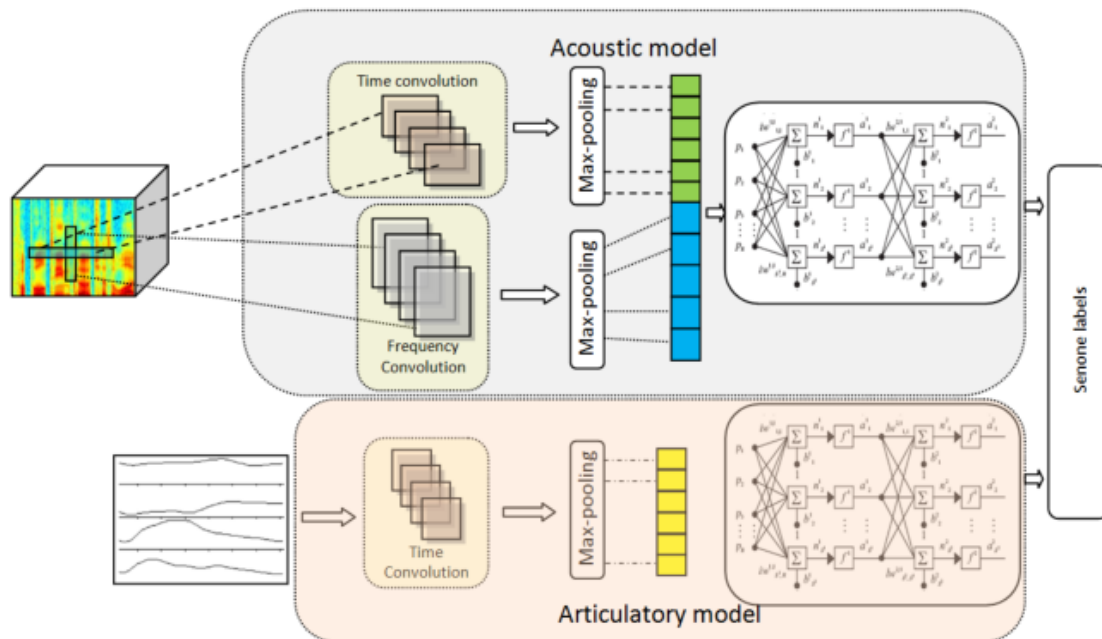


Figure (2-11) The structure of a speech recognition system that uses a convolutional neural network as a phonetic model. Image from (Mitra et al., 2017)

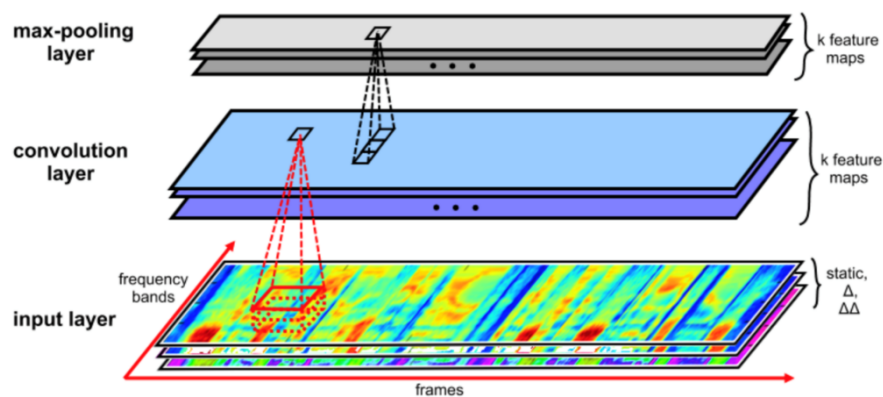


Figure (12-2) Applying convolution layer and maximization layer. Image from (Zhang et al., 2017)

2.2.4.4 Recurrent Neural Networks as a Phonetic Model

Recurrent Neural Networks (RNNs) hold a distinct advantage in handling problems governed by temporal dependencies due to their inherent recursive nature. These networks are proficient in capturing and learning temporal dependencies within the

data, rendering them well-suited for the intricacies of speech recognition tasks, which inherently exhibit a temporal structure.

In parallel with convolutional neural networks, researchers have explored the replacement of Deep Belief Neural Networks with Recurrent Neural Networks as the phonetic model within the structure of a DBN/HMM model (Graves et al., 2005). However, significant advancements were achieved after Alex Graves introduced the Connectionist Temporal Classification (CTC) in 2006 (Graves et al., 2006). Subsequently, numerous researchers embarked on developing comprehensive speech recognition systems using recurrent neural networks and their diverse variants, such as Long Short-Term Memory (LSTM) networks and Bi-directional Recurrent Networks (Graves & Jaitly, 2014). The schematic below illustrates an end-to-end speech recognition system employing recurrent neural networks (Amodei et al., 2016).

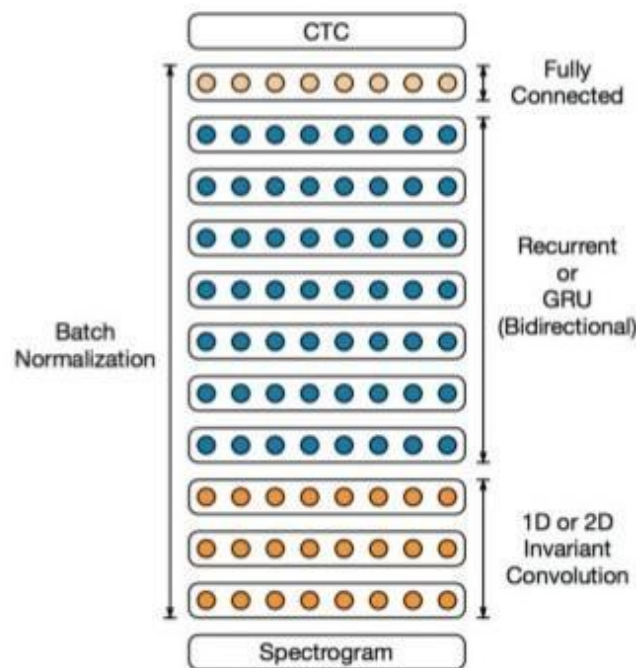


Figure (2-13) Portrays the structure of the speech recognition system utilizing a recurrent neural network as the phonetic model. Image from (Amodei et al., 2016)

2.2.5 Theoretical Foundations of Speech Recognition System Architectures

As delineated in the phonetic modeling section, this thesis delves into three principal acoustic models: Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) based on Gaussian distribution functions and Markov models. The comprehensive theoretical

underpinnings of the Gaussian Mixture Model are elucidated in Appendix B, while the Hidden Markov Model is elaborated upon in the reference (Rabiner & Juang, 1986).

The DBN/HMM model is based on the principles of the deep belief neural network. This model leverages the deep learning technique of Boltzmann machine layers, followed by network fine-tuning. The theoretical foundations of neural networks and deep neural networks, along with the training methodologies for these models, are meticulously elucidated in Appendix D.

Lastly, the end-to-end deep recurrent neural network model adheres to the theoretical foundations of recurrent neural networks. These foundational principles, along with in-depth explanations about long short-term memory neurons and bidirectional networks, are meticulously presented in Appendix E. Furthermore, Appendix F expounds on the Connectionist Temporal Classification (CTC) method employed in this structure.

2.2.6 Sounds and Linguistic Characteristics of Spoken Data

Phoneme, or "awa" in Persian, constitutes the smallest sound unit in speech. It is the elemental component that distinguishes one word from another. For instance, in words like "زرد" (z-a-r-d meaning yellow) and "سرد" (s-a-r-d meaning cold) in the Persian language, the sole dissimilarity lies in the sounds "z" and "s", which sets the two words apart.

Sounds serve as the fundamental building blocks of words. Although certain sounds may be shared across different languages, each language possesses its unique set of sounds. For instance, English comprises 44 sounds, despite having only 28 letters. In contrast, the Persian language encompasses 29 sounds and 32 letters.

Traditionally, the speech recognition problem has been framed as a phoneme classification challenge. Breaking down speech into its constituent sounds, categorizing them, and subsequently combining them to form words and sentences offers a highly rational approach for speech recognition systems.

This approach offers several advantages over direct word-based classification. Firstly, it trains a classifier with a limited number of categories (i.e., the sounds of the language, usually between 25 to 50 categories) rather than a vast number of word categories.

Additionally, this classifier's task involves establishing the relationship between extracted speech features and spoken sounds, which is comparatively simpler than the relationship between speech features and entire words. Sounds act as the foundational components of speech, establishing a lower-level relationship with extracted speech features than words do. Consequently, sounds and words lack one-to-one correspondence. For instance, the word "sister", "خواهر" in Persian, consists of the sounds "kh-A-h-a-r," but its letters do not map directly to these sounds.

To directly derive letters and words from speech-extracted features, the system must capture the relationship between phonemes and words in addition to the one between extracted speech features and phonemes. This additional complexity can introduce errors in the system.

The utilization of a phonetic dictionary and a search method to establish the relationship between phonemes and words helps simplify and enhance the phonetic acoustic model's accuracy.

Nevertheless, phonetic acoustic models present certain drawbacks. First, they necessitate a comprehensive phonetic dictionary containing variations of each word. Unfortunately, such a dictionary is not readily available for the Persian language. Thus, during this thesis, a specialized phonetic dictionary was compiled for implementing various phonetic models, as discussed in the third chapter. Another drawback of phonetic acoustic models is their dependence on the specific language.

2.2.7 The Role of the Language Model in Speech Recognition

In the context of speech recognition, the Language Model plays a crucial role in estimating the likelihood of the output string of words $P(W)$ in equation (2-1).

Language Models serve to assign higher probabilities to output strings that align with the characteristics and grammar of the target language, thereby enhancing the system's accuracy. For instance, consider the English sentences "I saw a van" and "eyes awe of an." Though they may share phonetic similarities, the first sentence is significantly more probable due to English language characteristics and grammar, and thus is more likely to be the correct corresponding text.

Additionally, Language Models aid in constraining assumptions during the search for the most fitting text. Speech recognition systems often employ the Beam Search method to convert phonemes into words and sentences. The Beam value in Beam Search

balances the trade-off between accuracy and speed. A larger beam value enhances accuracy but slows down the process.

Language Models assist in reducing the beam size while maintaining constant accuracy. They achieve this by choosing words and sentences that are more likely based on linguistic structure.

Chapter three will offer a comprehensive explanation of Language Models tailored to the Persian language, outlining their development in alignment with this thesis. For a comprehensive review of the theoretical and computational foundations of the n-gram Language Model or probabilistic neural network-based language models, refer to Appendix A.

2.2.8 Linguistic Aspects of the Persian Language:

The Persian language, spoken by Iranians, belongs to the Indo-Iranian branch of the European-Hindu languages. It serves as the official language for over 130 million people in Iran, Afghanistan, and Tajikistan, with a significant presence in Uzbekistan and to a lesser extent in Iraq, Bahrain, and Oman. The language has displayed remarkable stability since the 8th century, though it has undergone some influence from other languages, notably Arabic, which has contributed numerous words and phrases.

Despite the considerable impact of Arabic on Persian vocabulary, it has not altered the fundamental structure of the language. In essence, Persian has primarily adopted lexical elements from Arabic without affecting its syntax (grammar) and morphology. As a result, the patterns of Persian and Arabic diverge significantly, with distinct phonological structures leading to variations in their phoneme models. Consequently, the development of speech recognition systems for Arabic and Persian necessitates different acoustic and language models.

The grammar of Persian exhibits similarities to many contemporary European languages, often adhering to the "subject, object, verb" sentence structure. That is, sentences typically follow the sequence of subject, object, and verb, with the object often following the verb. However, beyond standard sentence patterns, Persian allows considerable flexibility in the arrangement of prepositions, conjunctions, and complements, resulting in expressions that may not follow a fixed order.

For instance, verbs in Persian may appear at the beginning, middle, or end of a sentence without altering the meaning. This inherent flexibility in word order poses challenges in extracting Persian grammar.

The writing system of Persian is right-to-left, utilizing a very similar alphabet to Arabic. Notably, short vowels are frequently omitted in the Arabic script, leading to potential ambiguities in pronunciation. The Persian alphabet comprises 23 fixed letters and 6 vowels. To facilitate a more detailed understanding of the vowel sounds in Tehrani Farsi, the following diagram illustrates their phonetic representations.

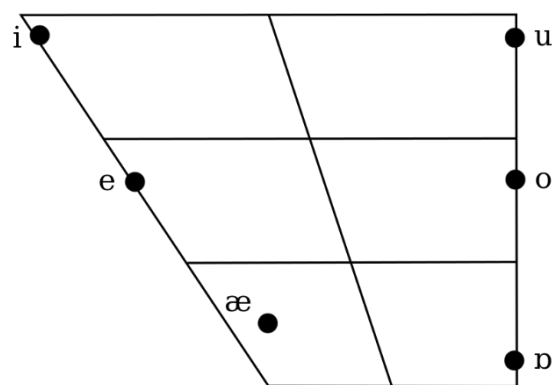


Figure (2-14) Vowels sounds in Farsi

The Persian language features three long vowels (/i/, /u/, /ɒ/) and the remaining vowels (/e/, /o/, /a/) are considered short or clear. It is worth noting that the categorization of vowels as long or short is primarily based on their manner of articulation, rather than solely their sound duration. The table below illustrates the pronunciations of these vowels in Persian, accompanied by the corresponding letters of the Persian alphabet, their respective codes, and symbols.

IPA	Char	Cod e	Farsi Letter	Phonetic Description
<i>i</i>	i	105	ای	high front unrounded
<i>e</i>	e	101	ئ	mid front unrounded
<i>a</i>	a	97	آ	low front unrounded

IPA	Char	Code	Farsi Letter	Phonetic Description
<i>u</i>	u	117	او	high back unrounded
<i>o</i>	o	111	اُ	mid back unrounded
ɒ	/	47	آ	low back rounded
p̥	\	92	پ	unvoiced bilabial plosive closure
<i>p</i>	p	112	پ	unvoiced bilabial plosive
b̥	'	96	ب	voiced bilabial plosive closure
<i>b</i>	b	98	ب	voiced bilabial plosive
t̥	-	45	ت، ط	unvoiced alveolar plosive closure
<i>t</i>	t	116	ت، ط	unvoiced dental plosive
d̥	=	61	د	voiced dental plosive closure
<i>d</i>	d	100	د	voiced dental plosive
c̥	@	64	ک	unvoiced palatal plosive closure
<i>c</i>	c	99	ک	unvoiced bilabial plosive
k̥	*	42	ک	unvoiced velar plosive closure
<i>k</i>	k	107	ک	unvoiced bilabial plosive
ɟ̥	!	33	گ	voiced palatal plosive closure
ɟ̥	;	59	گ	voiced palatal plosive
g̥	&	38	گ	voiced velar plosive closure
<i>g</i>	g	103	گ	voiced velar plosive
ɣ̥	^	94	ق، غ	voiced uvular plosive closure
<i>G</i>	q	113	ق، غ	voiced uvular plosive

IPA	Char	Code	Farsi Letter	Phonetic Description
ʔ	(40	ا،ؤ،ع	glottal stop closure
ʔ]	93	ا،ؤ،ع	glottal stop
tʃ̞	\$	36	چ	unvoiced alveopalatal affricate closure
tʃ	'	39	چ	unvoiced alveopalatal affricate
dʒ̞	#	35	ج	voiced alveopalatal affricate closure
dʒ	'	44	ج	voiced alveopalatal affricate
f	f	102	ف	unvoiced labiodental fricative
v	v	118	و	voiced labiodental fricative
s	s	115	س،ث،ص	unvoiced alveolar fricative
z	z	122	ز،ذ،ظ،ض	voiced alveolar fricative
ʃ	.	46	ش	unvoiced alveopalatal fricative
ʒ	[91	ژ	voiced alveopalatal fricative
x	x	120	خ	unvoiced uvular fricative
h	h	104	ح،ه	unvoiced glottal fricative
l	l	108	ل	lateral alveolar
r	r	114	ر	trill alveolar
m	m	109	م	nasal bilabial
n	n	110	ن	nasal alveolar
j	y	121	ی	approximant palatal

Table (2-1) Phonetic table of Farsi language. Table from (Sameti et al., 2011)

As previously mentioned, the Persian alphabet bears resemblance to the Arabic alphabet, with the addition of four extra letters in Persian, resulting in a total of 32 letters as opposed to the 28 letters found in Arabic.

Notably, these letters in Persian are represented by symbols that are not present in the Arabic alphabet. Specifically, the characters “گ”, “ز”, “پ”, and “چ” are used in the Persian alphabet. Additionally, the pronunciation of four letters in Persian differs from their pronunciation in Arabic.

Conversely, the Arabic language possesses its own set of pronunciations that are not applicable in the Persian language. In Farsi, a wide range of words, prepositions, roots, nouns, and adjectives is utilized. Furthermore, new words are often formed through combinations of nouns, adjectives, and infinitives, similar to the German language, where words are composed of other words.

Suffixes play a dominant role in the morphology of the Persian language, while the use of prefixes is relatively less. In Persian, verbs can convey a specific time-related context and are adjusted according to the subject's number, whether singular or plural. Notably, gender distinctions are absent in Persian, and pronouns are employed uniformly for both genders (Sameti et al., 2011).

2.2.9 Advancements in Speech Recognition Systems: Research Findings

In the past decade, speech recognition has been an active and dynamic area of research, prompting numerous efforts to enhance the performance of these systems. Researchers have explored diverse methods and architectures, ranging from traditional Gaussian Mixture models to cutting-edge deep neural networks, all with the aim of creating superior speech recognition systems. In this section, we shall delve into the endeavors undertaken in this domain and carefully scrutinize the outcomes of these researches across various datasets. Additionally, one of the crucial dependent variables in this research, namely the word error rate, will be thoroughly examined and discussed.

2.2.9.1 Evaluation Metrics: Word Error Rate (WER)

The word error rate (WER) serves as a fundamental metric for evaluating the performance of translation or speech recognition systems. However, calculating the

WER can be complex due to potential differences in length between the output word string of the system and the reference word string. WER is essentially a variation of the Levenshtein distance, providing a valuable and powerful scale to compare speech recognition systems and track their improvement. Nonetheless, while the WER effectively quantifies the number of vocabulary errors, it lacks the ability to shed light on the underlying nature of these errors, limiting its usefulness in guiding to system enhancements.

To compute the WER, the output word string is aligned dynamically with the reference string, yielding the following expression:

Equation (2-5)

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

Here, S represents the number of replaced words, D stands for the number of deleted words, I denotes the number of added words, C represents the count of correctly recognized words, and N denotes the total number of words in the reference string. Alternatively, the word accuracy W_{acc} is used as another performance measurement, calculated as the number of correctly recognized words:

Equation (2-6)

$$W_{acc} = 1 - WER = \frac{H-I}{N}$$

In the above formula, H signifies the number of correctly recognized words.

Notably, an alternative criterion for evaluating speech recognition systems is the Phonetic Error Rate (PER). Unlike WER, PER operates at the phonetic level and applies to datasets that are phonetically ordered, labeled, and aligned. For instance, the TIMIT corpus is an English dataset where this metric is commonly utilized. For additional examples and implemented code, refer to (Thoma, 2013).

2.2.9.2 Speech Corpora

In this section, we delve into the current accuracy rates in the realm of speech recognition systems and introduce renowned datasets for comparative analysis in both

the English and Persian languages. Specifically, the following datasets are presented: LibriSpeech Corpus, Wall Street Journal (WSJ) Speech Corpus, TED-LIUM Speech Corpus, and TIMIT Speech Corpus. These datasets stand as prominent examples in the English language, while the FarsDot Speech Corpus represents a notable speech corpus in the Persian language. Below, we provide a succinct overview of the characteristics of each of the aforementioned speech datasets.

2.2.9.3 LibriSpeech Dataset

The LibriSpeech dataset encompasses a substantial corpus of nearly 1000 hours of English speech, sampled at 16 kHz. It was meticulously curated by Vassil Panayotov with the assistance of Daniel Povey. The dataset comprises audiobooks from the LibriVox project, thoughtfully separated and precisely aligned. It is categorized into six distinct subsets, rendering LibriSpeech one of the most extensive freely available datasets in the English language.

2.2.9.4 Wall Street Journal Dataset

In 1991, the DARPA Speech Language Program initiated efforts to develop a comprehensive speech corpus to support continuous speech recognition systems with an extensive vocabulary. The first two corpora, WSJ0 and WSJ1, consisted of speech readings from Wall Street Journal news texts. As the collection evolved, additional texts from diverse sources, including North American business news, were incorporated to enhance its coverage.

The selected texts were designed to encompass between 5,000 to 20,000 words from the Wall Street Journal's written collection. Apart from text readings, the dataset includes dictation sections, where journalists dictated hypothetical articles for gathering spoken data.

Data collection was conducted using two microphones, one positioned near the speaker and the other with a variable position. Consequently, the dataset is available in three configurations: speech from the close-proximity microphone, speech from the distant microphone, and a combination of both. Each configuration is complemented with transcriptions, test sets, and comprehensive documentation. For further details about this speech corpus, refer to (Garofolo et al., 1993).

2.2.9.5 TED-LIUM Dataset

The TED-LIUM speech datasets encompass dialogues from TED programs in English, accompanied by their corresponding transcriptions, all sampled at a rate of 16 kHz. This comprehensive collection comprises approximately 118 hours of spoken data. The data was meticulously curated by Rousseau, Deléglise, and Esteve, who elaborated on their data collection methodologies and specifications in their article (Rousseau et al., 2012).

2.2.9.6 TIMIT Dataset

The TIMIT speech dataset is a compilation of read speeches, designed to cater to phonetic studies and the development and evaluation of automatic speech recognition systems. TIMIT comprises a speech database featuring 630 speakers representing eight main American English dialects. Each speaker enunciates ten sentences, carefully chosen for their rich array of sounds and pronunciations. The dataset includes sorted and transcribed texts, alongside 16-bit, 16 kHz speech files for each utterance. Notably, temporal alignment between the texts and speech files is meticulously established, and the dataset's accuracy is typically evaluated based on phonetic error rates. This endeavor was a collaborative effort between MIT University, SRI International Group, and Texas Institute.

The corresponding texts were meticulously reviewed manually, and the test and training subsets were carefully divided based on variations in sounds and accents. The dataset is presented in a computer-searchable tabulated format, complemented by recorded documents (Garofolo, 1993).

2.2.9.7 FarsDat Dataset

The FarsDat Persian speech corpus comprises speech recordings from 300 Iranian speakers, encompassing diverse attributes such as age, gender, education level, and regional accent or dialect. This comprehensive collection embraces ten different dialects from various regions of Iran, including Tehrani, Turkish, Isfahani, Southern, Northern, Khorasani, Balochi, Kurdish, Lori, and Yazdi. Each speaker uttered 20 sentences, contributing to two series of conversations, while an additional 100 speakers individually articulated 110 words. The 6000 utterances are categorized and meticulously tagged for phonetic analysis. Among the collection are 386 sentences, phonetically annotated according to the IPA standard.

The audio signal is saved in a standard wav file format, rendering it compatible with various software tools. The samples were captured at a frequency of 22.5 kHz, and the signal-to-noise ratio measures at 34 dB (Bijankhan et al., 1994).

2.2.9.8 Word Error Rate in the Libri Speech Dataset

In this section, we present a compilation of word error rates (WER) reported by various researchers in their respective articles on the Libri Speech dataset. The WER values are tabulated below for comparison and analysis.

Research Paper	Paper Title	Test Clean (WER)	Test Other (WER)
(Amodei et al., 2016)	Deep Speech 2: End-to-End Speech Recognition in English and Mandarin	5.33%	13.25%
(Han et al., 2017)	The CAPIO 2017 Conversational Speech Recognition System	7.64%	3.19%
(Zeyer et al., 2018)	Improved training of end-to-end attention models for speech recognition	12.76%	3.82%
(Povey et al., 2016)	Purely sequence-trained neural networks for ASR based on lattice-free MMI	-	4.28%
(Peddinti et al., 2015)	A time delay neural network architecture for efficient modeling of long temporal contexts	-	4.83%
(Panayotov et al., 2015)	LibriSpeech: an ASR Corpus Based on Public Domain Audio Books	13.97%	5.51%
(Liptchinsky et al., 2017)	Letter-Based Speech Recognition with Gated ConvNets	14.5%	4.8%
Base	Kaldi - HMM-(SAT)GMM	22.49%	8.01%

Table (2-2) Word Error Rate on the LibriSpeech dataset in different articles

As depicted in the table, speech recognition systems employing the end-to-end recurrent neural networks have demonstrated superior performance compared to other models.

2.2.9.9 Word Error Rate in the Wall Street Journal Speech Dataset

In this section, we present the Word Error Rate (WER) reported by various researchers in their articles on the Wall Street Journal speech dataset. The following table (Table 2-3) displays the amount of word errors observed in different articles.

Research Paper	Paper Title	Test Set (92)	Test Set (93)
(Chan & Lane, 2015)	Deep Recurrent Neural Networks for Acoustic Modelling	3.47%	-
(Amodei et al., 2016)	Deep Speech 2: End-to-End Speech Recognition in English and Mandarin (Deep Speech Version 1)	4.94%	6.94%
(Amodei et al., 2016)	Deep Speech 2: End-to-End Speech Recognition in English and Mandarin	3.60%	4.98%
(Panayotov et al., 2015)	LibriSpeech: an ASR Corpus Based on Public Domain Audio Books	3.63%	5.66%
(Palaz et al., 2015)	Convolutional Neural Networks-based Continuous Speech Recognition using Raw Speech Signal	5.6%	-

Table (2-3) Word Error Rate on the Wall Street Journal Speech Dataset in Various Articles

As indicated in the table, the speech recognition systems employing the end-to-end recurrent neural networks have achieved superior results compared to other models.

2.2.9.10 Word Error Rate in the TED-LIUM Speech Dataset

This section presents the Word Error Rate (WER) reported by various researchers in their articles on the TED-LIUM dataset. The following table (Table 2-4) displays the amount of word errors observed in different articles.

Research Paper	Paper Title	Test Dataset
(Han et al., 2017)	The CAPIO 2017 Conversational Speech Recognition System	6.5%
(Povey et al., 2016)	Purely sequence-trained neural networks for ASR based on lattice-free MMI	11.2%
(Rousseau et al., 2012)	TED-LIUM: an Automatic Speech Recognition dedicated corpus	15.3%

Table (2-4) Word Error Rate on the TED-LIUM Speech Dataset in Various Articles

2.2.9.11 Phonetic Error Rate in the TIMIT Speech Dataset

In this section, we present the Phoneme Error Rate (PER) reported by various researchers in their articles on the TIMIT speech dataset. The following table (Table 2-5) provides details on the PER observed in different articles.

Research Paper	Paper Title	Test Dataset
(Tóth, 2015)	Phone recognition with hierarchical convolutional deep maxout networks	16.5%
(Vaněk et al., 2017)	A Regularization Post Layer: An Additional Way how to Make Deep Neural Networks Robust	16.5%
(Tóth, 2014)	Combining Time- and Frequency-Domain Convolution in Convolutional Neural Network-Based Phone Recognition	16.7%
(Lu et al., 2016)	Segmental Recurrent Neural Networks for End-to-end Speech Recognition	17.3%
(Chorowski et al., 2015)	Attention-Based Models for Speech Recognition	17.6%
(Graves et al., 2013)	Speech Recognition with Deep Recurrent Neural Networks	17.7%
(Zeghidour et al., 2018)	Learning Filterbanks from Raw Speech for Phone Recognition	18.0%
(Oord et al., 2016)	Wavenet: A Generative Model For Raw Audio	18.8%
(Mohamed et al., 2009)	Deep Belief Networks for Phone Recognition	23%

Table (2-5) Phoneme Error Rate (PER) on the TIMIT Speech Dataset in Various Articles

As depicted in the table, the speech recognition systems based on end-to-end Convolutional Neural Network (CNN) have demonstrated superior performance compared to other models. It is worth noting that the TIMIT dataset offers the advantage of containing time alignment information of phonemes with corresponding speech files, which enables the use of Phonetic Error Rate (PER) for result analysis.

2.2.9.13 Phonetic Error Rate in the Farsdat Speech Dataset

This section presents the Phoneme Error Rate (PER) reported by various researchers in their articles on the Farsdat dataset. The following table (Table 2-6) displays the amount of phoneme errors observed in different articles.

Resource	Kaldi Model	Test Set	Valuation Set
(BabaAli, 2014)	SGMM2 + MMI Training	19.7%	20.2%
(BabaAli, 2014)	Hybrid System (Karel's DNN), sMBR training	19.8%	20.1%

Table (2-3) Word Error Rate on the Wall Street Journal Speech Dataset in Various Articles

2.2.10 Challenges in Speech Recognition Systems

The efficacy of a speech recognition system is contingent upon the conditions under which it operates. Transitioning from controlled and constrained environments to real-world scenarios introduces a multitude of challenges. The accuracy of the system can be notably impacted by factors such as the size of the lexicon, speech quality, speaker accents, variable speaking rates, presence of background conversations, and the broader domain context. To comprehend the extent of these factors and their limitations on system performance, refer to the illustrative diagram below.

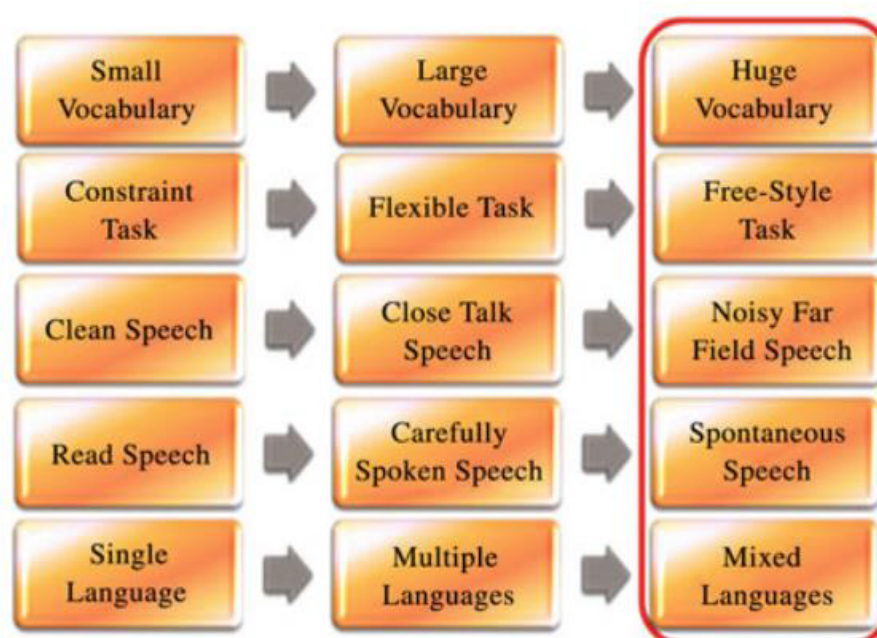


Figure (2-15) Challenging factors in speech recognition systems. Image from (Yu & Deng, 2014)

In the contemporary landscape of speech recognition, each of these factors represents a vast and promising realm for researchers to explore, seeking improved solutions and achieving more robust outcomes. Throughout the successive sections of this thesis, we will delve into the methodologies employed to surmount these challenges in a systematic manner.

Notably, within the context of the Persian language, the accent of the speaker and the contrast between colloquial and formal conversational styles stand out as prominent

hurdles. Given the inherent nature of this language, these factors pose significant challenges, and this thesis endeavors to address and overcome these obstacles.

2.3 Research Background and Literature Review

Speech recognition systems can be categorized based on various parameters, one of which is whether the speech is continuous or discrete. In discrete systems, the focus is on identifying a series of patterns to recognize individual words. These systems were prevalent during the emergence of speech recognition in the 1960s to 1980s, mainly used for voice-command systems (Salazar et al., 1998).

On the other hand, continuous speech recognition systems aim to comprehend the entirety of a speaker's speech without dividing it into discrete words. In recent decades, research efforts have predominantly focused on this category. Continuous speech recognition presents challenges, particularly in word boundary recognition and dealing with coarticulation. Unlike segmented speech, where words are separated by pauses, continuous speech lacks such clear boundaries, making it challenging to employ traditional signal processing methods to recognize word boundaries. Moreover, the phenomenon of coarticulation further complicates correct phoneme recognition and word identification in continuous speech systems (Chorowski et al., 2014) (Morgan & Bourlard, 1995).

Another challenge pertains to the conversational style and formality of speech. Conversational speech is more difficult due to its wide range of vocabulary and language models. Researchers have been actively investigating methods for conversational speech recognition, as exemplified by (Xiong et al., 2016).

A historical classification of speech recognition systems was based on the number of words they could recognize. Carnegie Mellon University's speech recognition department categorized this as follows:

- *Small dictionary*: Dozens of words, such as identifying commands or numbers
- *Medium dictionary*: Hundreds of words, for voice-command systems
- *Large dictionary*: Thousands of words
- *Very large dictionary*: Tens of thousands of words, which is the focus of most current research.

Initially, speech recognition systems with small dictionaries were developed, but over time, efforts were made to increase the vocabulary. (Lin et al., 2009) is an example of such endeavors.

Another research classification is based on speaker-dependent and speaker-independent models. Speaker-dependent models, simpler than their counterparts, were early attempts to identify unique speaker features and improve accuracy. Researchers explored this approach in (Unnikrishnan et al., 1991). Subsequently, the focus shifted towards speaker-independent or general models, addressed in (Lee, 1988).

A speech recognition system can be dissected into several fundamental components, each serving a specific purpose. To gain a comprehensive understanding of prior literature, it is necessary to review research efforts concerning each of these sections and the diverse methods explored within them. We will now introduce the different sections and proceed to examine significant research works pertaining to each of them. The Acoustic Model, tasked with recognizing phonemes, stands out as a critical component, along with Decoding, responsible for transforming phonetic strings into the most probable word sequence.

Another vital aspect of the language model involves the modeling of language grammar and structure. Apart from the aforementioned structural components, there exists another category of researchers focusing on end-to-end speech recognition systems, where the Acoustic Model encompasses the functionalities of the other two parts. In such systems, training occurs in a unified manner, leading to what is known as an end-to-end speech recognition system.

The research in the domain of Acoustic Models can be divided into two based on the research time. Preceding the emergence and advancement of deep neural networks, Gaussian hybrid models were the primary choice for the phonetic model section, while hidden Markov models were utilized for temporal modeling and word recognition (Huang et al., 1990).

However, after the rise of deep neural networks, various models based on these networks have been developed for phonetic modeling. Let us explore some of the models based on different types of neural networks, including deep neural networks.

2.3.1 Deep Belief Network with Hidden Markov Model (DBN/HMM)

In this model, a deep belief neural network (DBN) is employed for the phonetic modeling, while the decoding part still relies on the conventional Hidden Markov Model (HMM) technique. This model was initially proposed and implemented in 2012 by Abdel-rahman Mohamed and demonstrated a significant increase in the accuracy of speech recognition systems by more than ten percent on well-known datasets. The methods were further explored and elaborated upon in the works of Abdel-rahman Mohamed (Mohamed et al., 2012) and Dahl (Dahl et al., 2012).

2.3.2 Recurrent Neural Network Model (RNNs)

Recurrent Neural networks due to their structure are inherently suited for problems with temporal dependency such as speech recognition. Recurrent neural network models that employ Long Short-Term Memory neurons (LSTMs) have been used for Acoustic Modeling in many researches including (Hochreiter & Schmidhuber, 1997). The word finder section in this model is implemented using either the Hidden Markov Model resulting in a hybrid RNN/HMM model, or Connectionist Temporal Classification (CTC) algorithm that results in an end-to-end ASR (Graves, 2006). These models have been extensively discussed in the works of (Sak et al., 2015) and (Sak et al., 2014).

2.3.3 Bidirectional Recurrent Neural Network Model (Bi-RNNs)

The bidirectional recurrent network model closely resembles the recurrent neural network model, with the key distinction being that this model analyzes input data bidirectionally. In other words, it processes the input both from the beginning to the end and from the end to the beginning, and subsequently sends both output paths to the next layer. The original presentation of this model can be found in the work of Alex Graves (Graves et al., 2005), and ongoing research on this model is being conducted by various scholars, as seen in articles like (Graves et al., 2013) (Zeyer et al., 2017).

2.3.4 Convolutional Neural Network Model (CNNs)

Convolutional networks have shown remarkable performance in various image processing domains in recent years. Consequently, some researchers are exploring the

application of these networks to model speech recognition systems. In this approach, two-dimensional spectrogram of audio data are taken as input and treated as images.

These models can be categorized into three groups: frequency-direction convolution models, time-direction convolution models, and models that perform convolutions in both time and frequency directions, with the latter category yielding superior results. Relevant research on convolution models for speech recognition systems can be found in the works of (Sainath et al., 2015) (Abdel-Hamid et al., 2014), while (Tóth, 2014) investigates the combined time and frequency method. Additionally, (Zhang et al., 2017) and (Zhang et al., 2017, 4845-4849) explore the integration of convolutional neural network models with attention layers.

2.3.5 End-to-End Deep Neural Network Model

The end-to-end deep neural network model, as the name suggests, represents a newer approach where there are no distinct divisions as in other models. Instead, all components are trained under a single network and model. There is no separate decoding or language model, and all steps are carried out within the framework of a deep recurrent network. Relevant research on this end-to-end model can be found in the works of (Miao et al., 2015), (Song & Cai, 2015) (Graves & Jaitly, 2014). (Zhang et al., 2016) also examine the combination of the end-to-end deep neural network model with an attention mechanism.

In the context of the word finder section, apart from the traditional Hidden Markov models, an alternative approach known as the Connectionist Temporal Classification (CTC) has been introduced by (Graves et al., 2006), which many researchers have adopted since its proposal.

Reviewing the scientific literature on speech recognition systems reveals that the majority of the mentioned articles and research studies focus on the English language's structure and grammar. As a result, these systems are not equipped to recognize speech in the Persian language. However, some researchers, such as (Sameti et al., 2009) and (Zeinali et al., 2017), have explored speech recognition systems specifically for Persian using GMM/HMM models. Notably, phonetic models based on deep neural networks have not been employed yet on Persian language, despite being among the best-performing methods in this domain. Building on this research background, the

objective of this study is to develop a speech recognition system based on deep neural networks for the Persian language.

2-4 Literature Review Summary

This chapter begins with an overview of the categories of speech recognition systems. Numerous studies have been conducted on various subcategories, which can be broadly classified based on continuous or discrete speech, dictionary volume, speaker dependence, acoustic model, and feature extraction methods. The research within these categories has been reviewed to provide a comprehensive understanding of the field.

Speech recognition systems consist of several key components, including the acoustic model, language model, signal preprocessing section, and various word search algorithms. Researchers have diligently investigated and improved each of these components, and their findings have been examined in relation to the overall structure of speech recognition systems.

The acoustic model, being a principal variable in this research, receives particular attention, and diverse acoustic models and related studies have been thoroughly discussed.

2-5 Summary of Contents

This chapter delves into the fundamental theoretical underpinnings and the challenges of speech recognition. It introduces key structures of speech recognition systems, such as the GMM/HMMs, DBN/HMMs model, convolutional neural networks, and deep recurrent neural networks.

As the primary focus of this thesis is on speech recognition systems for the Persian language, a background on Persian language structure and phonology provided in this chapter. Prominent English datasets, along with the Farsdat dataset in Persian, are introduced, and the respective results achieved on these datasets are presented.

The chapter also addresses the main challenges in implementing speech recognition systems. Finally, an extensive literature review covering researchers' work on the design and implementation of speech recognition systems is presented.

Chapter 3: Research Methodology

3-1 Introduction: Overview of Chapter Contents

This chapter presents the research methodology employed in this study. It begins with a comprehensive description of the data collection process, followed by an introduction to the FarsAva speech dataset. Subsequently, we delve into the methods of data preprocessing and normalization for Persian texts. Additionally, the research variables are introduced, and the measurement metrics for evaluating results is presented. The chapter further elucidates the implementation of speech recognition systems through three distinct methods: GMM/HMM model, DBN/HMM model, and end-to-end recurrent deep neural networks.

2-3 FarsAva: Persian Speech Corpus

This section discusses the process of collecting and creating the FarsAva speech dataset, large scale Persian Speech corpora, which specifically designed and gathered for this thesis. The significance of training data in machine learning algorithms, particularly with the rise of deep learning techniques, cannot be overstated.

Among the Persian speech datasets available, the Farsdat corpus prepared by (Bijankhan et al., 1994) stands out as the most reliable source. However, these data are unfortunately not freely accessible to the public. The Farsdat corpus comprises 76 hours of Farsi speech, meticulously annotated and aligned at the phonetic level, forming the foundation for most research endeavors in Persian speech recognition. This corpus includes 386 sentences uttered by 300 Persian speakers from ten different dialects.

Another Farsi speech dataset, Voxforge multilingual project, contains less than an hour of speech as of this writing. While there are other Persian speech databases, none of them can rival the data volume available in other languages. For instance, the LibriSpeech dataset boasts a thousand hours of training data, and the Baidu Research Institute trained its English speech recognition system (Amodei et al., 2016) on a massive 11,940 hours of data.

Given the constraints of the existing Persian speech data and the growing demand for novel deep learning algorithms that require extensive training data, it became imperative for us to create a Persian speech corpus that aligns with our research objectives. To ensure meaningful comparisons of results, we undertook the task of constructing a substantial Persian speech dataset.

Thus, the FarsAva Persian speech corpus emerged, containing over 5000 hours of diverse speech data. With more than 6000 speakers, this dataset captures various Farsi dialects spoken in different contexts and environments. The richness in sources, dialects, and environmental conditions empowers this corpus to encompass the full spectrum of speech variations present in the Persian language.

3-3 Collection of FarsAva Speech Dataset

In the following sections, we will examine the methods of data collection, audio pre-processing, text pre-processing, preparing a phonetic dictionary and creating a language model in the preparation of FarsAva data.

3.3.1 Speech data collection methods

Different methods have been used to collect speech data in different years in the production of different datasets in the world. TIMIT's speech dataset was created by recording the speech of 630 speakers with different American accents who read ten sentences each. TIMIT's speech data includes information such as corresponding text, transliteration of the text, temporal alignment of the phonemes along with the audio file of the spoken sentence with a sampling rate of 16 kHz. 300 Farsi speakers each said 386 phonetically rich sentences to prepare Farsdat data. Farsdat data includes time alignment of sounds for each sample.

The main effort in creating a speech dataset in the past has been manual temporal alignment of sounds. Before the use of Hidden Markov model alignment or the advent of Connectionist Temporal Classification (CTC), researchers were faced with the approach of breaking the speech recognition problem into the phoneme classification problem and finding the best output string.

The figure below shows a time alignment for two sentences, one in TIMIT's speech data and the other in Farsdat's speech data. As shown in the figure, the temporal

alignment of the phonemes includes the start time and the end time of each pronounced phoneme in the sentence in milliseconds.

0 3050 h#	j	j	82000	89694	2
3050 4559 sh	n	n	89694	94014	2
4559 5723 ix	u	u	94014	101274	3
5723 6642 hv	h	h	101274	104964	2
6642 8772 eh	((104964	105654	2
8772 9190 dcl]]	105654	106884	2
9190 10337 jh	a	a	106884	109764	3
10337 11517 ih	z	z	109764	113034	2
11517 12500 dcl	=	=	113034	114624	2
12500 12640 d	d	d	114624	115584	2
12640 14714 ah	a	a	115584	119604	3
14714 15870 kcl	s	s	119604	123324	2
15870 16334 k	-	-	123324	124044	2
16334 18088 s	t	t	124044	125274	2
18088 20417 ux	e	e	125274	127132	3
20417 21199 q	\	\	127132	129398	2
21199 22560 en	p	p	129398	130914	2
22560 22920 gcl	e	e	130914	133044	3
22920 23271 g	s	s	133044	136794	2
23271 24229 r	a	a	136794	139464	3
24229 25566 ix	r	r	139464	140994	2
25566 27156 s	a	a	140994	144444	3
27156 28064 ix	.	.	144444	148252	2
28064 29660 w	=	=	148252	150532	2
29660 31719 ao	d	d	150532	151254	2
31719 33360 sh	e	e	151254	155394	3
33360 33754 epi	x	q	155394	159564	2
33754 34715 w	@	*	159564	160854	2
34715 36080 ao	c	k	160854	162924	2
36080 36326 dx	a	a	162924	171204	3
36326 37556 axr		r	171204	171204	2
37556 39561 ao		d	171204	171204	2
39561 40313 l	j	j	171204	174000	2
40313 42059 y					
42059 43479 ih					
43479 44586 axr					
44586 46720 h#					

Figure (3-1) time alignment of sounds in Timit and Farsdat speech data

After the emergence of automatic time alignment methods by the Hidden Markov model after the invention of the tConnectionist Temporal Classification (CTC) method for speech recognition systems that use recurrent neural networks, the need for phonetic time alignment in speech datasets perished.

For example, the LIBRISpeech and the TED-LIUM datasets are only composed of pairs of audio files and corresponding text of each sentence. Different datasets include the speaker of each sentence and the total number of speakers, which is generally not useful in the phonetic model training stage. Eliminating phonetic time alignment made it possible to obtain speech data with a larger volume and at a lower cost.

Another method that some datasets such as LibriSpeech have used is the use of recorded speech data such as audio books. LibriSpeech dataset uses recorded speech in

audio books in the Voxforge project. This method increases the speed of data collection. However, in the production of LibriSpeech, a two-step method has been used to segment and align the text and the corresponding audio file (Panayotov et al., 2015).

By carefully examining the methods of collecting data in other well-known databases of the world, a mixed method has been used to collect FarsAva. This mixed method consists of a combination of recording speech by speakers and using recorded speech. In the following, Farsava data sources are explained.

3.3.2 Sources of FarsAva Speech Data

As previously mentioned, the FarsAva speech dataset is a hybrid collection of data, combining both recorded speech by speakers and pre-recorded speech from various sources, such as radio, television, audio books, and social networks.

In order to ensure diversity in speech content, speakers, topics, and environments, audio files were manually selected from these sources. Subsequently, a Python program was designed and implemented to retrieve the identified resources from the relevant platforms. This process resulted in obtaining thousands of hours of audio data, comprising a mixture of speech and other audio elements like music, etc.

To distinguish speech from non-speech segments, a specialized classification program was developed. Given the immense volume of data, manual separation was impractical. Instead, an automatic two-class categorization system was trained on approximately 340 hours of speech data and 300 hours of non-speech data, leveraging the support vector learning algorithm.

After successfully separating speech from non-speech portions, the speech files remained quite long for training speech recognition systems. To address this, an audio segmentation program was implemented in Python, which utilized silence-based techniques to divide the speech files into segments ranging from one to fifteen seconds.

It is important to note that the obtained speech files exhibited substantial diversity, encompassing various speakers, accents, topics, environmental noise, speech types, and speaking speeds. Following data collection, each audio file was manually labeled for quality assessment. The labeling process involved tagging information, such as overlapping speech, incorrect start and end points, lack of understandable speech, background noise, and overall data quality.

The characteristics of the FarsAva speech data are summarized in the following table.

#	Characteristics	Amount
1	Dataset Volume in Hours	5160 Hours
2	Number of Speakers	6320 Speaker
3	Number of Speech Files	3,798,814
4	Text Volume in Words Tokens	17,786,578 Words
5	Number of Speech Files Containing Background Noise	497,735
6	Number of Speech Files With Phone Quality	192,972
7	Speech Files Duration	1 - 15 Seconds
8	Average of Speech File Duration	4.88 Seconds

Table (3-1) general characteristics of Farsava speech data

3-3-3 Data Preprocessing and Text Normalization

In the subsequent sections, the process of pre-processing Persian texts and the challenge of text normalization in Persian language have been elucidated. Furthermore, specific measures taken to address text normalization in the FarsAva dataset are discussed.

3-3-3-1 The Challenge of Text Normalization in Persian Language

Text normalization in Persian language presents significant challenges due to the language's inherent reliance on prefixes and suffixes for word formation. Consequently, written words exhibit substantial diversity, as the same word can be represented in multiple ways with varying prefixes and suffixes. For instance, the plural suffix "ها", pronounced "ha", manifests in three distinct forms in texts, leading to a single word being represented in three different ways and hampering system accuracy. As an illustrative example, consider the English word "tree", in Persian "درخت", which can be pluralized in Persian as "درخت‌ها" ("derakht-ha"), "درخت ها" ("derakht-ha"), or "درختها" ("derakht-ha"). Among these variations, only the form written with a Zero-width non-joiner is considered correct according to Persian grammar. Thus, one essential

pre-processing step involves converting words with prefixes and suffixes into a standardized form with a Zero-width non-joiner.

Another notable challenge in text processing for the Persian language pertains to the diversity in Unicode representation. This variation is primarily caused by the differences in different keyboards and the similarity between Arabic and Persian letters in these keyboards. Letters such as "ک", "ی", and "و" exhibit significant Unicode diversity. To establish a consistent word dictionary, it becomes necessary to remove this variation and convert all texts into a standard Unicode representation. Additionally, preprocessing involves converting numerals, symbols, and other characters into a standardized Unicode format.

The process of text preprocessing and normalization yields uniformity in Persian texts, facilitating the development of various tools and models required in speech recognition systems. By achieving a normalized form for Persian texts, the accuracy and effectiveness of the phonetic dictionary and language model are directly influenced, ultimately resulting in enhanced precision for the speech recognition system.

3.3.3.2 Normalization Procedures

As emphasized in the previous section, normalizing Persian texts holds significant importance. To address this crucial aspect and align it with the objectives of this thesis, a preprocessing and normalizing program for the Persian language was developed. The primary aim of this program is to not only normalize Persian texts but also enhance the quality and accuracy of the language model and phonetic dictionary constructed in this research.

The following aspects are covered by the normalizer program:

1. *Unicode Normalization*: Ensuring consistent Unicode representation throughout the texts. Examples are below:
 - Converting the Arabic “Kaf”, “Ye”, and other Arabic characters to their corresponding Persian equivalents.
 - Tai Tanith Conversion: Transforming Tai Tanith characters to their Persian equivalents.

2. *Number Normalization*: Standardizing the representation of numbers and replacing them with their Persian equivalents.
3. *Punctuation Handling*: Replacing English semicolons and commas with their Persian equivalents, correcting punctuation spacing, and substituting English quotation marks and percent signs with their Persian counterparts.
4. *Spacing Correction*: Replacing multiple spaces with a single space, Replacing Zero-width non-joiner with a single one, and removing Zero-width non-joiner before or after spaces.
5. *Prefixes and Suffixes Correction*: Ensuring that all prefixes and suffixes are separated by Zero-width non-joiner. Specific items that require correction include various suffixes and prefixes. Examples are below:

- Plural Suffixes: “ها” “های” “هایی”
- Suffixes: “ترین” “تر”
- Verb Prefixes: “می” “نمی”
- Possessive Pronouns Suffixes: “م” “ت” “ش” “مان” “تان” “شان”
- Colloquial Possessive Pronouns Suffixes: “مون” “تون” “شون”
- Possessive Pronouns Suffixes: “ام” “ات” “اش”
- Suffixes: “ی” “ای”
- Prefix: “به”
- Numbers Suffixes: “م” “مین” “امین”
- Suffixes acting as Possessive Pronouns:
“هایم” “هایت” “هایش” “هایمان” “هایتان” “هایشان”
- Suffixes acting as Possessive Pronouns:
“هام” “هات” “هانش” “هامان” “هامون” “هاتان” “هاتون” “هاشان” “هاشون”
- Noun Maker Suffixes: “اک” “دار” “گار” “انه” “نا” “بی” “گاه”

By comprehensively addressing these normalization procedures, the normalizer program ensures consistency and standardization in the Persian texts. As a result, it contributes significantly to the enhanced accuracy and effectiveness of the language model and phonetic dictionary, ultimately leading to improved speech recognition performance.

3.3.4 Phonetic Dictionary

As discussed in the second chapter, the phonetic dictionary encompasses not only the words of a language but also their corresponding phonetic representations, including various pronunciations. In the context of speech recognition systems, where the Acoustic Model recognize phonemes, and the decoding section searches for the best word sequence based on the detected phonemes, the phonetic dictionary plays a crucial role. It enables the system to associate sets of phonemes with specific words, determining the final output string of words.

The size of the phonetic dictionary significantly impacts the system's efficiency. Increasing the number of words expands the system's recognition capability, enhancing its generality. However, this expansion may also introduce challenges, such as phonetically similar words leading to potential misdiagnoses and reduced accuracy.

In many languages, like English, there are very complete phonetic dictionaries. Unfortunately, in the Persian language, a comprehensive phonetic dictionary is yet to be developed. The largest existing phonetic dictionary for Persian contains merely forty thousand words. Therefore, to ensure comprehensive coverage of the words in the FarsAva speech dataset and to complete the tests in this thesis, a substantial phonetic dictionary for Persian was created.

To construct the phonetic dictionary, distinct words were identified from a sizable text corpus of approximately 140 gigabytes, as described in the following section. After preprocessing and normalization, the distinctive words, around four hundred thousand in number, were extracted from the corpus. Words with low occurrences, typically misspellings, were excluded from consideration.

Transliteration the identified words can be achieved through manual transcription or automatic transcription oh phonemes using a phonetic dictionary of smaller volume. In this thesis, a recurrent neural network (RNN) model was employed for automatic transcription. The RNN model was trained using the existing forty-thousand-word phonetic dictionary in Persian. It achieved an accuracy of approximately 96% in correctly recognizing the phoneme sequence for words.

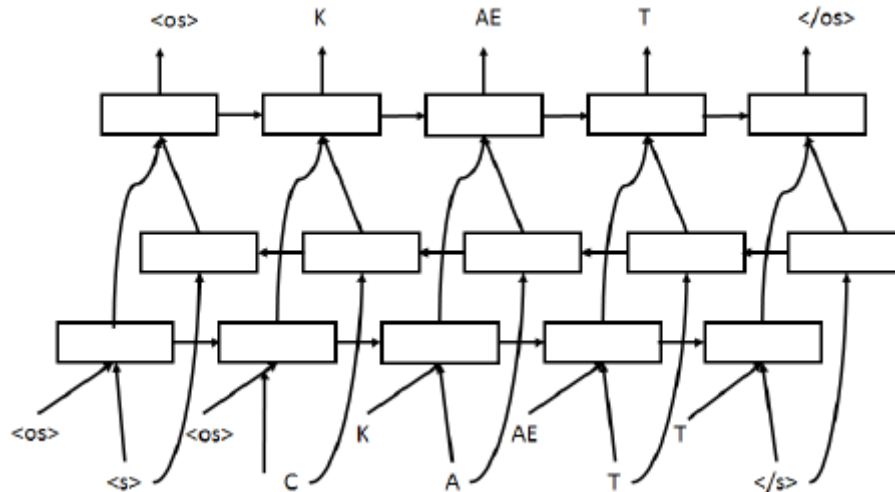


Figure (3-2) Depicts the many-to-many bidirectional recurrent neural network model utilized in this research. Image from (Yao & Zweig, 2015).

The aforementioned bidirectional recurrent neural network model was leveraged to create an automatic transliteration system, transcribing words to phoneme sequences, for words not present in the original dictionary.

A	فتحه
AA	آ
B	ب
CH	چ
D	د
E	کسره
F	ف
G	گ
H	ه
I	ای
J	ج
K	ک
L	ل
M	م
N	ن
NG	نگ
O	ضمه
P	پ
Q	ق
R	ر
S	س
SH	ش
T	ت
U	او
V	واو
W	پایه الف
X	خ
Y	ی
Z	ز
ZH	ژ

Figure (3-3) Illustrates the symbols utilized to represent each of the Persian phonemes.

ZH A R F	ژرف
ZH A R F I	ژرفی
ZH A R F AA	ژرفا
ZH A R F AA Y	ژرفای
ZH A R F AA Y I	ژرفایی
ZH A R F AA W A N D I SH	ژرفا اندیش
ZH A R F AA S A N J	ژرفا سنج
ZH A R F AA Y E SH	ژرفا یش
ZH A R F A N D I SH	ژرفا ندیش
ZH A R F A N D I SH I	ژرفا ندیشی
ZH A R F AA N E G A R	ژرفا نگر
ZH A R F AA N E G A R I	ژرفا نگری
ZH A R F B I N	ژرف بین
ZH A R F B I N I	ژرف بینی

Figure (3-4) Displays a section of the phonetic dictionary.

3.3.5 Language Model

As previously discussed, the language model's primary task is to estimate the probability of linguistic units, encompassing individual words and complete sentences. This estimation enables the generation of more accurate output strings that adhere to the structural and grammatical principles of the language in question. In recent years, the use of language models in natural language processing and speech recognition has sparked considerable research efforts aimed at improving language modeling techniques.

Two general approaches are employed for constructing language models. The first method is called count-based models using statistics, while the second relies on continuous-space methods or Neural Probabilistic Language Models. Although the latter exhibits better performance, it does suffer from drawbacks such as extended training times and contextual word limitations.

The older N-grams method or probabilistic model involves constructing an N-th order Markov assumption, estimating N-gram probabilities, and subsequently applying probability smoothing. Successful approaches within this category include the Kneser–Ney smoothing method and the Jelinek-Mercer smoothing method. More recently, researchers have introduced a new approach known as the continuous-space

language models. Sub-branches of this method, such as neural network language models and RNNs variations, were devised to address the issues of sparsity and discontinuity encountered in the N-gram method.

For the production and implementation of the speech recognition system in this thesis, the N-gram method was employed. A comprehensive explanation of this method can be found in Appendix A.

3.3.6 Method of Preparing the Persian Language Model

The language model utilized in this thesis is the N-gram language model. The N-gram language model is established through the counting and probability estimation of words within the text corpus. To construct a language model that accurately captures the characteristics and grammar of a given language, a large and diverse text corpus is required.

To prepare the language model, a vast and diverse text corpus was compiled. This corpus encompassed various sources of Persian textual references, including news sites, book texts, information sites spanning cultural, economic, political, and technological domains, as well as Persian content from social networks and blogs. Upon gathering these texts, a series of preprocessing and normalization steps were applied. The volume of the collected texts is presented in Table (3-2):

Volume in Gigabytes	Number of Sentences	Number of Unique Words
6.8 GB	37,435,450	316,678

Table (3-2) Statistics of Text Data

Following the pre-processing and normalization of the texts, the N-gram Language Model based on word occurrence probabilities and counting was constructed from the data. In the N-gram model, the value of 'N' determines the number of consecutive words considered for counting and probability estimation. For example, in a trigram model, all three-word sequences in the text are counted and ranked based on their repetition and probability.

To evaluate the performance of different language models, ten percent of the texts were randomly selected for testing. These test data were employed to assess various language models after training. The different tested models were formed by

combining the number of N-grams (three, four, and five) with different types of smoothing algorithms. Perplexity is used to measure the accuracy of the models. Lower perplexity values indicate more successful language probability estimation by the trained model.

In the fourth chapter, a comprehensive table presenting the results of different language models trained on the text corpus is provided. Below is a sample of a language model in the ARPA standard structure, specifically a part of the three-gram language model (Figure 3-5).

<s/> امروز یکشنبه	0.9703591-
امروز یکشنبه با	1.55099-
امروز یکشنبه بیست	1.155431-
امروز یکشنبه ششم	1.606492-
امروز یکشنبه نوزدهم	1.251106-
امروز یکشنبه هفتم	1.04297-
امروز یکشنبه چهاردهم	1.295994-
تا یکشنبه در	1.06602-

Figure (3-5) A part of the three-gram language model in the ARPA format

3.3.7 Testing Datasets

To assess the accuracy of the trained models, it is essential to evaluate their performance on an independent test data set, one that was not involved in the model training process. A common approach is to randomly withhold a portion of the original data to create the test dataset. In line with this methodology, five percent of the training data was randomly set aside for testing in this thesis.

In addition to the primary testing dataset, two additional datasets were prepared to ensure diversity within the testing datasets. Evaluating the model's accuracy on these two testing datasets would effectively gauge the generalization capability of the model. The first dataset consists of a half-hour news program, while the second data set is a combination of diverse sources, totaling four hours of audio data. The specifications of the test datasets are presented in the following tables.

After training, all the models have been tested on all three test datasets, and their accuracy has been obtained, and the results will be presented in the next chapter.

Specs	Volume
Data Volume in Hours	230
Number of Speech Samples	170,946
Data Source	Randomly Selected 5% of Training Data

Table (3-4) Statistics of Primary Testing Dataset

Specs	Volume
Data Volume in Minutes	30
Number of Speech Samples	184
Data Source	News Program

Table (3-4) Statistics of News Program Testing Dataset

Specs	Volume
Data Volume in Hours	4.2
Number of Speech Samples	3,033
Data Source	Different sources including news, TV shows, social media content, audio books and etc.

Table (3-5) Statistics of Third Testing Dataset

3-4 Research Variables

The research variables are as follows:

1. *Acoustic Model*: The acoustic model used in the speech recognition system.
2. *Training Data Volume*: It refers to the quantity of training data utilized for training the system, and its extent can be varied during experimentation.
3. *Type of Language Model*: This variable encompasses the various types of language models that can be employed, allowing for comparison and evaluation.

4. *Number of Dictionary Words:* The size of the phonetic dictionary can be modified, and this variable influences the diversity of words the system can recognize.

In the fourth chapter, the outcomes of the experiments will be meticulously presented, considering the aforementioned research variables.

3.5 Measuring Methods

As previously indicated in Chapter 2, the Word Error Rate (WER) serves as the primary metric for assessing the accuracy of speech recognition systems. Consistently, in this thesis, the WER has been employed to gauge the accuracy of the trained models. The number of vocabulary errors in each test was measured using the respective test data and two additional test datasets, as discussed in the preceding section.

3.6 Implementation and Training Approach

In this section, we delve into the implementation and training methodology employed for the speech recognition systems. As mentioned earlier in the chapter, this thesis encompasses the training of three distinct types of speech recognition models to facilitate a comprehensive accuracy comparison. These three models differ significantly in terms of their structure and acoustic model, namely: GMM/HMM Model, DBN/HMM Model, and finally, end-to-end RNNs.

The following subsections present an in-depth examination of the prerequisites for data preparation, the steps involved in training, and the challenges encountered during the implementation of each of the aforementioned structures.

3.6.1 Data Preparation and Training of the GMM/HMM Model

In this section, we discuss the implementation and training process of the Persian speech recognition system based on the Gaussian Mixture Model-Hidden Markov Model (GMM-HMM). In this model, the task of Acoustic Modeling is accomplished using the GMM, while the HMM is utilized to capture temporal dependencies among phonemes, words, and sentences.

The Language Model is employed to create a Finite State Transducer, which, together with the HMM parameters, helps determine the best output string.

For the implementation and training of this structure, the Kaldi toolkit has been utilized. The toolkit includes implementations of the HMM, GMM, and Finite State Transducer, all implemented in C++ language. We used linux as operating system for training our model using Kaldi and the pipelines are done using shell scripts.

3.6.1.1 Data Preparation

Before training the GMM/HMM model, data preparation and preprocessing are essential. This involves creating the files necessary for training in the Kaldi toolkit. The texts in the dataset are thoroughly normalized before generating the required files. The critical files include:

1. *Corresponding Transcription File*: This file contains the corresponding transcription of each audio file along with the audio file name. It is divided into two parts: the training dataset and the test dataset.
2. *Audio Files Path*: This file includes the name and full address of each audio file, enabling the Kaldi tool to access and perform feature extraction. It is also divided into two parts: the training dataset and the test dataset.
3. *Dictionary File*: To train the speech recognition system with the Kaldi, all words in the corresponding texts must have a phoneme representation. This file contains the phonetic representation of each word, and missing words are automatically transliterated using the previously discussed automatic transliteration system.
4. *Language Model File*: A language model file in ARPA format is required for training the ASR system. This file is prepared from the corresponding texts of the dataset and is crucial for rebuilding the Finite-State Transducer graph after training.
5. *Unique Phoneme List*: This file contains the set of used phoneme codes. The Kaldi tool replaces each phoneme with a number for efficiency during training.
6. *Settings File*: A configuration file that specifies the type of feature extraction, the number of frequency coefficients, search beam value, audio files sampling rate, utilization of power squared in frequency coefficients, and other relevant settings.

Additionally, other files such as the speakers specifications file, out-of-dictionary words list, phonetic ambiguities file, dictionary integer-coded file, silent phonemes file, etc., are also prepared during this stage. Once the files are completely prepared, the training process can commence.

3.6.1.2 Feature Extraction

Prior to training, feature extraction is performed on the data. In this experiment, the method of extracting frequency coefficient features with thirteen coefficients, along with the addition of the first and second derivatives and squared energy of the power and intervals of 20 milliseconds, is employed. The resulting feature vector contains 40 values for every 20 milliseconds.

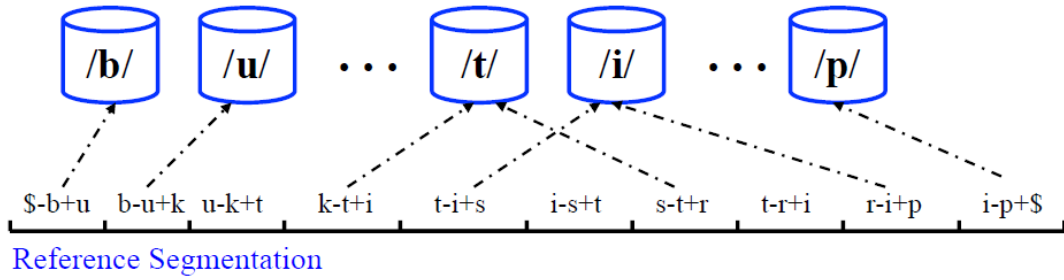
3.6.1.3 Monophone Model Training

The initial stage of modeling in the GMM/HMM based speech recognition system involves training the monophone model. In this model, each phoneme is recognized independently without considering any dependency information from consecutive phonemes. The expectation-maximization algorithm is used to train the model, and the forced alignment method is employed to align the feature vector and phoneme labels.

3.6.1.4 Triphone Model Training

Following the monophone model training, a triphone model is trained. In this model, phoneme triplets are taught to the system along with their relationships and pronunciations. The number of hidden Markov models in this section is the number of phonemes in the language to the power of three. The training process involves repeating the training of a three-phoneme model based on the alignment from the previous step and aligning the data again. This is done iteratively for three or more cycles. The figure below illustrates the distinction between monophone and triphone models.

Monophone HMMs



Triphone HMMs

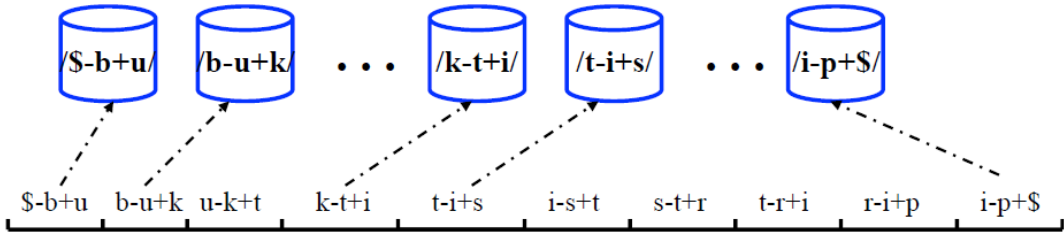


Figure (3-6) Hidden Markov Model based on monophonic model and triphonic model

3.6.2 Data Prepration and Training of the DBN/HMM Model

In this section, we will elucidate the implementation and training of the Persian speech recognition system based on the Deep Nelief Network - Hidden Markov Model (DBN/HMM). In this model, the Deep Belief Neural network is utilized for phonetic modeling, while the Hidden Markov Model is employed to capture temporal dependencies among phonemes, words, and sentences.

The same language model used in the previous structure is employed to create a Finite-State Transducer, which, in combination with the HMM parameters, facilitates the search for the best output string.

For the implementation and training of this structure, the Kaldi toolkit has been utilized. It is important to note that to train a deep neural network as a phonetic model using the Kaldi tool, a Gaussian Mixture Model must first be trained.

3.6.2.1 Data Preparation

Similar to the previous section, all files mentioned earlier are required to train this model. Additionally, a Gaussian Mixture Model is needed to train the DBN-HMM in the Kaldi tool.

After training the GMM/HMM model, this model gains the ability of forced alignment, meaning it can label the feature vectors extracted from the audio file with appropriate phonetic labels.

Phonetic alignment allows the dataset to be converted into a feature vector set with corresponding phonemes, replacing pairs of audio files and corresponding texts.

3.6.2.2 Deep Belief Neural Network Pre-training

The deep belief neural network is constructed by stacking multiple layers of Restricted Boltzmann Machine (RBM). Each layer of the Boltzmann neural network comprises two hidden and visible layers, categorized as an unsupervised generative model, trained solely from data without labels.

After training a layer of this network, the hidden layer can produce a higher-level representation of the data. The training of the deep belief network begins with the first layer, starting with the input, i.e., the extracted feature vectors. After training the first layer, its output becomes the input to the next layer, continuing this process until the last layer. The layer-by-layer training of Boltzmann networks in a deep belief neural network is referred to as pre-training.

For these experiments, a DBN network with six Boltzmann layers, each containing 1024 neurons, is employed. After the complete training of all layers, the second stage of training, is fine-tuning the model using the labels.

3.6.2.3 Training and Fine-Tuning of the Deep Belief Network

Following the training of all six Boltzmann layers in the previous stage, the second phase of training, Fine-tuning, commences with the use of labels.

At this stage, A Feed-Forward Fully Connected Neural Network is placed after the sixth Boltzmann layer. The output of this layer corresponds to the number of phonemes in the language, and the network learns to map the last output of the Boltzmann network to one of the phonemes in the language. This layer is known as the Softmax layer.

After training and fine-tuning, this neural network can classify the feature vectors extracted from the audio file into the phonetic classes of the language. Subsequently, a Hidden Markov Model is employed to find the best string of words by

utilizing the Language model and phoneme dependencies within words and the dictionary.

This structure is fully trained using the Kaldi tool and FarsAva speech dataset, and the results are presented in the fourth chapter.

3.6.3 Data Preperation and Training of an end-to-end RNN Model

In this section, we will implement and explore the third and final structure examined in this thesis. Recurrent Neural Networks (RNNs) are well-suited models for data with temporal dependencies, making them particularly suitable for speech data, as the feature vectors extracted from audio files exhibit a time sequence.

Two approaches to using RNNs in speech recognition systems were discussed in the previous chapter. The first approach involves employing a combined RNN-HMM Model, similar to the previous methods, except that the Acoustic Model is an RNN. The second approach, and the one used in this thesis, is a more modern technique known as the end-to-end method.

In the end-to-end method, a Connectionist Temporal Classification (CTC) algorithm layer is integrated into the neural network, thereby assigning the task previously performed by the Hidden Markov Model directly to the network itself. Consequently, a single, end-to-end neural network performs the speech recognition task. It is important to note that the implementation of the CTC enables updating the probabilities of the output string using a language model.

3.6.3.1 Data Preparation

Data preparation in this section differs slightly from the previous two sections. As this structure is an end-to-end one, it utilizes pairs of audio files and corresponding transcription for training. Similar to previous steps, all texts undergo pre-processing and normalization.

In this structure, by eliminating the middle layer of phoneme classification, the network directly recognizes the output characters and subsequently produces the best string of words using the Language Model. Therefore, a phonetic dictionary is not required for this model.

The extracted features comprise frequency coefficients, with the input of the network forming a sequence of these vectors. The labels consist of the characters from

the corresponding transcriptions. A file specifying the path of the audio files, the corresponding transcriptions, and the file sizes is prepared for training the model. Another file is also created to list all characters present in the transcriptions.

3.6.3.2 End-to-End Training of Deep Recurrent Neural Network

To train the end-to-end network, Tensorflow is utilized. In this thesis, a recurrent neural network with six layers of Long Short-Term Memory (LSTM) and 2048 neurons in each layer, along with a Connectionist Temporal Classification (CTC), is designed and implemented using Tensorflow.

During training, the input files are read and grouped in batches of 10. The corresponding transcriptions of these files are also read and assigned to expected labels for error calculation. The feature vectors are then extracted from these files. To increase the input speed of the network, they are processed in batches of ten. The batch error for each batch is calculated, and the network parameters are updated using the computed gradients.

Due to the large number of parameters in this network and the significant computational demands for training, the process is performed on graphics cards. Given the substantial volume of FarsAva speech data, training on a 1080 Ti graphics card would take approximately two months. To address this challenge, a parallelization option was implemented, enabling different graphics cards to process their respective inputs. After calculating the gradients, they are sent to a parameter server which updates the gradients and returns the results of the updated parameters to each graphics card. Consequently, the training time was reduced to two weeks by conducting training on four graphics cards.

The training process is monitored using a tool called Tensorboard. Below is the graph depicting error reduction in different training sessions.

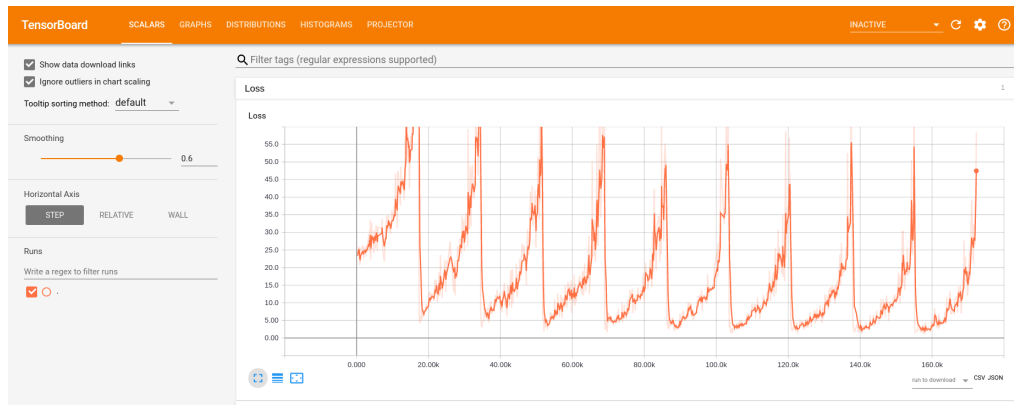


Figure (3-6) Model error reduction diagram in different training epochs by Tensorboard

Chapter 4: Analysis of Research Findings

4.1 Introduction: Overview of Chapter Contents

In this section, we present the findings and results derived from the preceding stages, and subsequently, we examine the research hypotheses. As mentioned in the first chapter, the primary independent variables of the research consist of the dataset volume and the Acoustic Model of the system. The independent sub-variables include the Language Model and the number of vocabulary words. The dependent variable, and the unit of measurement, is the Word Error Rate (WER).

The results presented in this chapter encompass the word error rate observed in the three test datasets introduced in the previous chapter. These sets are as follows: the first dataset, which is collected randomly from five percent of the training data; the second dataset, comprising half an hour of a news program; and the third dataset, comprising four hours of speech data sourced from various channels.

The aforementioned WER were obtained for each of the three models elucidated in the previous chapter. These models are respectively as follows: GMM/HMM Model, DBN/HMM model, and end-to-end Recurrent Neural Network Model.

In addition to presenting the direct results of this thesis, we also include the outcomes of training various Language Models (LMs), which will be discussed for clarification in the concluding chapter of this section.

4.2 Findings and Results (WER)

The table displaying the Word Error Rates (WER) is organized based on the training data collection, outlined in the following three sections. It is important to note that the collection of Farsava speech dataset involved a structured approach, conducted in three stages with varying volumes of data: the first stage encompassed 460 hours, the second stage consisted of 1500 hours, and the third stage entailed 5000 hours. Consequently, the evaluation of WER was performed at each stage, resulting in three distinct tables corresponding to the aforementioned steps.

4.2.1 First Part: WER Results on the First Test Dataset

Below are the tables representing the WER of three Acoustic Models on the first testing dataset, categorized by the amount of training data.

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	31.02
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	20.94
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	20.19

Table (4-1) Word Error Rates on the First Test Dataset (460 hours of training data)

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	30.93
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	18.04
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	18.02

Table (4-2) Word Error Rates on the First Test Dataset (1500 hours of training data)

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	30.98
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	17.90
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	16.12

Table (4-3) Word Error Rates on the First Test Dataset (5000 hours of training data)

Furthermore, the following diagram presents an overview of the word error rates on the first testing dataset:

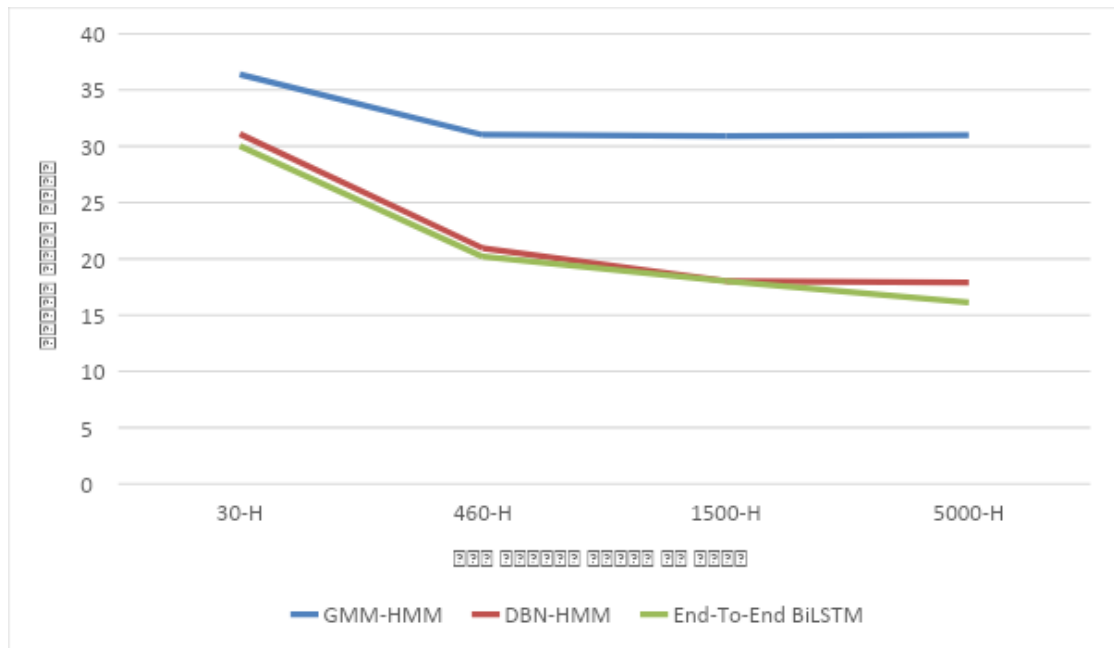


Diagram (4-1) Overview of Word Error Rate on the first testing dataset

4.2.2 Second Part: WER Results on the Second Test Dataset

The following tables present the WER of three Acoustic Models on the second testing dataset, categorized by the amount of training data.

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	33.5
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	27.8
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	24.7

Table (4-4) Word Error Rates on the Second Test Dataset (460 hours of training data)

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	33.4
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	25.3
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	21.4

Table (4-5) Word Error Rates on the Second Test Dataset (1500 hours of training data)

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	33.4
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	24.1
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	19.3

Table (4-6) Word Error Rates on the Second Test Dataset (5000 hours of training data)

Additionally, the following diagram presents an overview of the word error rates on the second testing dataset:

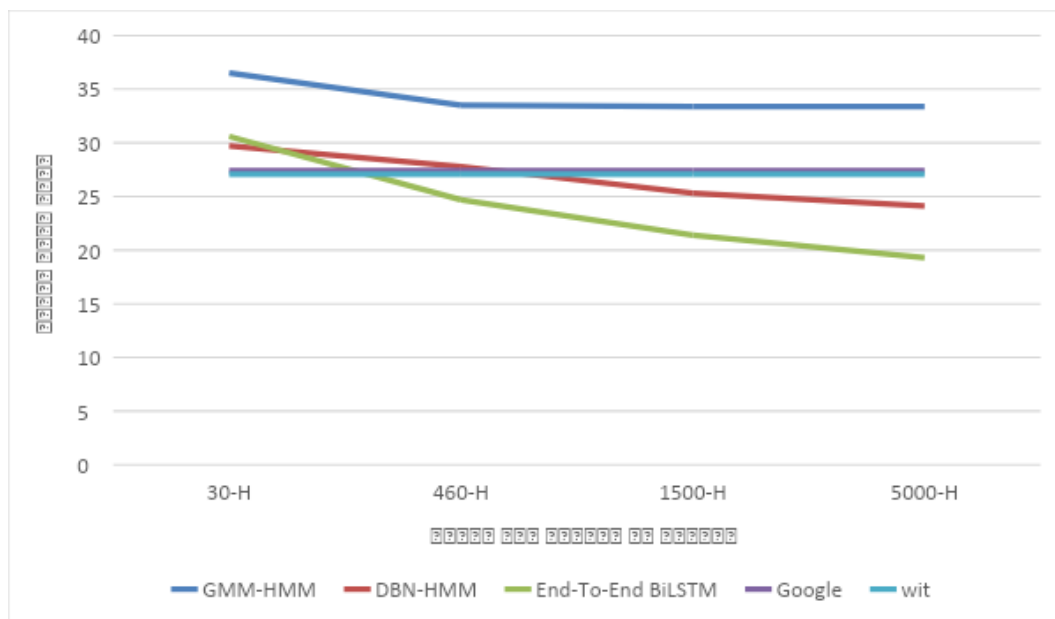


Diagram (4-2) Overview of Word Error Rate on the second testing dataset

4.2.3 Third Part: WER Results on the Third Test Dataset

The following tables present the WER of three Acoustic Models on the third testing dataset, categorized by the amount of training data.

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	39.4
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	33.1
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	31.2

Table (4-7) Word Error Rates on the Third Test Dataset (460 hours of training data)

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	39.2
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	33.6
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	26.4

Table (4-8) Word Error Rates on the Third Test Dataset (1500 hours of training data)

#	Acoustic Model	Word Error Rate (%)
1	Gaussian Mixture Model - Hidden Markov Model (GMM - HMM)	39.2
2	Deep Belief Network - Hidden Markov Model (DBN - HMM)	33.5
3	Recurrent end-to-end Deep Neural Network Model (LSTM)	23.2

Table (4-9) Word Error Rates on the Third Test Dataset (5000 hours of training data)

Additionally, the following diagram presents an overview of the word error rates on the third testing data set:

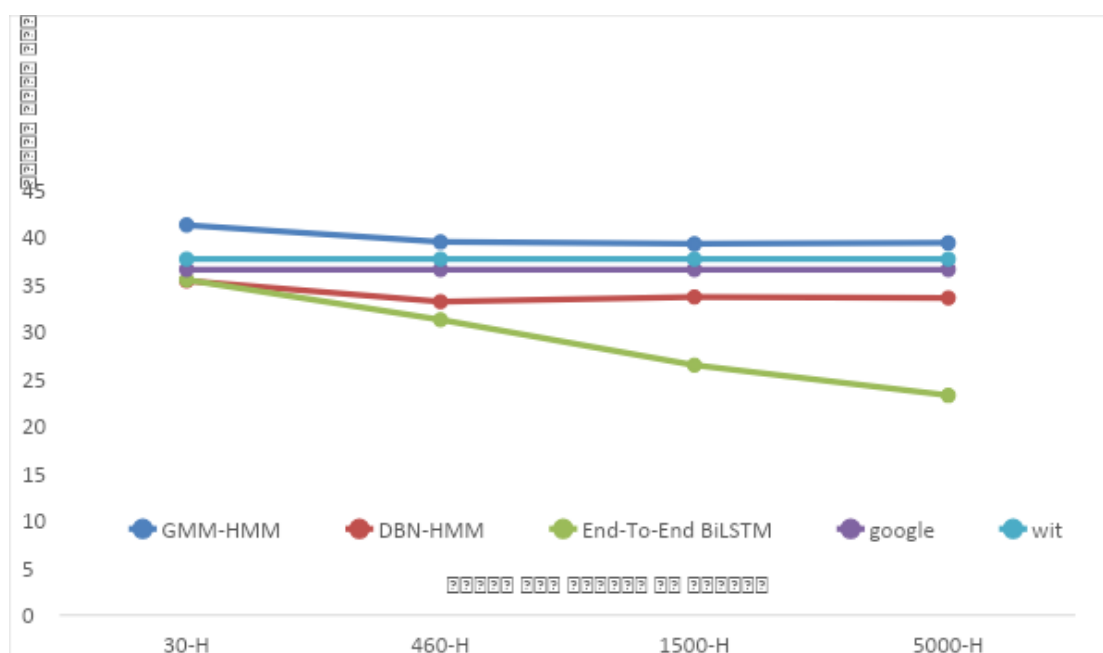


Diagram (4-3) of the error rate of words on the third experimental data set

4.2.4 Perplexity Results on Persian Language Model

As stated in the initial section of this chapter, presented below are the findings and results concerning the accuracy of the trained Language Models on Persian, considering the variable number of N in N-grams and the type of softening.

The methods of collecting text dataset on Persian language was thoroughly explained in chapter (3.3.6) and the statistics of the dataset is mentioned in table (3-2). These results are calculated using the test section of our text dataset.

#	Language Model (3-gram)	Perplexity
1	3-gram (without pruning and smoothing)	112.64
2	3-gram with Pruning	159.27
3	3-gram with Kneser–Ney smoothing	113.22
4	3-gram with Pruning and Kneser–Ney smoothing	164.60

Table (4-10) Perplexity result on 3-gram Persian Language Model with Different Smoothing Methods

#	Language Model (4-gram)	Perplexity
1	4-gram (without pruning and smoothing)	78.75
2	4-gram with Pruning	141.29
3	4-gram with Kneser–Ney smoothing	76.06
4	4-gram with Pruning and Kneser–Ney smoothing	166.63

Table (4-11) Perplexity result on 4-gram Persian Language Model with Different Smoothing Methods

#	Language Model (5-gram)	Perplexity
1	5-gram (without pruning and smoothing)	69.52
2	5-gram with Pruning	138.61
3	5-gram with Kneser–Ney smoothing	64.44
4	5-gram with Pruning and Kneser–Ney smoothing	171.91

Table (4-12) Perplexity result on 4-gram Persian Language Model with Different Smoothing Methods

4.2.5 WER Results based on Different Language Models

In addition to the aforementioned results, the table below illustrates the results of the accuracy of the speech recognition model based on using different language models. The table takes into account the architecture of the end-to-end recurrent neural network and various language models, incorporating the respective results.

#	ASR Model	Language Model	WER
1	End-to-end LSTM	Without Language Model	38.9
2	End-to-end LSTM	4-gram with Pruning	27.8

3	End-to-end LSTM	5-gram with Pruning	24.7
---	-----------------	---------------------	------

Table (4-13) WER result on the second Test dataset based on using different Language Models (460 hours of training data)

#	ASR Model	Language Model	WER
1	End-to-end LSTM	Without Language Model	41.9
2	End-to-end LSTM	4-gram with Pruning	33.4
3	End-to-end LSTM	5-gram with Pruning	31.2

Table (4-14) WER result on the third Test dataset based on using different Language Models (460 hours of training data)

4.3 Hypotheses Testing and Research Questions

As delineated in the introductory chapter, this research comprises three central questions and corresponding hypotheses:

- *First Hypothesis:* Deep neural networks demonstrate a 10% error reduction in comparison to Gaussian Mixture and Hidden Markov models, leading to enhanced system accuracy.
- *Second Hypothesis:* Depending on the volume of training data, the deep belief network model exhibits the lowest error rates.
- *Third Hypothesis:* The data volume is linked to model accuracy. Different models exhibit error reduction up to a certain threshold with increasing data volume, after which the error stabilizes.

Regarding the first hypothesis, the first, second, and third graphs in this chapter affirm that deep neural networks surpass Gaussian Mixture - Hidden Markov models by approximately 25% in relative improvement and 9% in absolute accuracy, thereby confirming the stated hypothesis.

Regarding the second hypothesis, while it was initially assumed that the Deep Belief Network model would perform optimally due to data volume being estimated at several tens of hours in the proposal, the results in the 30-hour dataset table indicate

that for data volumes below a few hundred hours, the deep belief network model yields the best results. However, as the data volume increases to several hundred hours, the recurrent deep neural network model demonstrates superior performance.

Regarding the first part of the third hypothesis, the graphs in this chapter reveal the significant impact of data volume on model accuracy. As data volume increases, model accuracy improves. However, for Gaussian Mixture Model - Hidden Markov Model, the accuracy rate trend shows a plateau from the 30-hour to the 460-hour. Similarly for Deep Belief Neural networks the improvements stops from the 460-hour to the 1500-hour. This indicates that Deep Belief Neural - Hidden Markov Models, due to their characteristics and parameter count, do not exhibit further accuracy improvement after reaching a certain data volume. Consequently, models with increased training capacity and more parameters are needed. In this thesis, the end-to-end Deep Recurrent Neural Networks using LSTMs effectively addresses this requirement.

4.4 Summary of Findings

This chapter presents the findings and results obtained from various experiments, categorizing them according to the main research variables: the training data volume and the acoustic model. Additionally, the language model is analyzed, and its accuracy results are presented based on the value of N in N-grams and the type of smoothing. The first to third graphs in this chapter present the results, corresponding to the aforementioned variables.

Following the presentation of the results related to the proposed assumptions in the introductory chapter and first part of this thesis, their validity has been demonstrated based on the numerical outcomes.

Chapter 5: Conclusion

5.1 Introduction: Overview of Chapter Contents

This chapter offers a comprehensive overview of the research by summarizing the contents of the third chapter, including the methodology, implementation, and test procedures. It proceeds with the presentation of key findings and results from the tests. Furthermore, a section is dedicated to comparing the research results with the initial assumptions and discussing the outcomes in light of the expectations.

Additionally, the general conclusion of this research is outlined, and the research's limitations are acknowledged. Finally, valuable suggestions arising from this study are provided to other researchers to encourage further investigation and advancement in this field.

5.2 Research Summary

The first chapter serves as an introductory foundation, addressing the research problem's context, importance, necessity, and innovation. It also outlines the research questions and assumptions, while presenting a table defining technical terms utilized in the thesis. The chapter subsequently delves into the research methodology, discussing data collection techniques in detail.

The subsequent section focuses on the language model and its significance within a speech recognition system. The creation of a diverse text corpus, data collection method, and the training of various language models are thoroughly explored to address the need for a language model.

The second chapter is divided into two main parts. The first part delves into the theoretical groundwork, examining machine learning models directly applicable to speech recognition, such as the Gaussian Mixture Model, Hidden Markov Model, and various neural network architectures used as phonetic models. This section also encompasses the general structure of speech recognition systems, various feature extraction methods, and current research results. Furthermore, the chapter dedicates a section to linguistic characteristics of the Persian language and its phonemes, in light of the thesis' focus on Persian speech recognition.

In the second part of this chapter, an extensive literature review on speech recognition research, both in general and within the context of the Persian language, is provided.

The third chapter commences with a presentation of the methodology used for collecting training data, including the examination of various speech datasets and the introduction of FarsAva Speech Dataset collected specifically for this thesis. Preprocessing and normalization methods for Persian texts are thoroughly investigated, and a normalization tool was designed and implemented due to the lack of suitable tools for these processes.

The concept of a phonetic dictionary is then explored, with a focus on the creation of a customized phonetic dictionary for experiments in this thesis. The preparation of this dictionary and the design and implementation of a recurrent neural network for creating phonetic representation of words are expounded.

The final part of the chapter elucidates the design and implementation of three Persian speech recognition systems: Gaussian Mixture Model - Hidden Markov Model (GMM/HMM), Deep Belief Neural Network Model - Hidden Markov Model (DBN/HMM), and End-to-End Recurrent Deep Neural Network Model using LSTMs. The chapter concludes with a comprehensive explanation of data preparation, pre-processing, feature extraction methods, and specifications for training each of the three models. Furthermore, it introduces the primary and secondary independent variables and the dependent variable of measurement, i.e., the word error rate.

5.3 Summary of Findings

In this section, a comprehensive overview of the fourth chapter is provided, encompassing the research findings and results. The presentation of results is structured around the main independent variables: the volume of training data and the acoustic model. The Word Error Rate is calculated based on three test datasets. Additionally, the chapter includes an evaluation of different language models accuracy and their respective effect on the speech recognition system accuracy.

Furthermore, the research hypotheses and key questions are thoroughly examined. The subsequent part of this chapter will involve a detailed comparison of the

results and an in-depth discussion of the positive and negative outcomes based on the findings.

5.4 Discussion: A Comparative Analysis of Research Results

This section is dedicated to a meticulous comparison of the results and a comprehensive discussion of the findings. As previously mentioned, the research hypotheses were closely examined, and this section delves into a detailed analysis of these three hypotheses and the corresponding results.

5.4.1 First Hypothesis

The first hypothesis posits that Deep Neural Networks have the capacity to substantially reduce word errors in speech recognition systems, thus enhancing their overall accuracy. A simple calculation reveals that in a speech recognition system based on the GMM/HMM model, there are approximately one million parameters, taking into account a feature vector length of 39, three phonetic subclasses for each phoneme, and a total of thirty thousand Gaussians. In contrast, a Deep Belief Neural Network, with six layers and 1024 neurons in each layer, comprises about 7 million parameters. Consequently, the Deep Belief Neural Network possesses a significantly larger parametric space compared to a Gaussian mixture model.

The increased number of parameters in DBNs allows them to learn more complex models through data training. This more complex model enables the discovery of higher-level relationships in the data, which in turn facilitates more accurate classification.

However, one drawback of this large number of parameters is the requirement for a substantial amount of training data. To fully leverage the potential of the Deep Belief Neural Network model, a substantial database, comprising tens or hundreds of hours of speech training data, is essential. This enables the model to more accurately categorize by extracting higher-level phonetic relationships.

Similar to DBNs, Recurrent Deep Neural networks possess a higher number of parameters than the Gaussian Mixture Model. Specifically, the deep recurrent neural network employed in this study comprises six layers of bidirectional recurrent neurons,

each layer containing 1024 neurons, and a fully connected feed forward layer utilized for character classification, resulting in approximately 9 million parameters.

In addition to their increased parameter count, recurrent deep neural networks are particularly suitable for modeling temporal data. This crucial aspect of recurrent neural networks has been thoroughly investigated in the appendices. In essence, each neuron in the recurrent network receives input from the lower layer and also obtains output from the previous time step. Consequently, the network learns the temporal relationships between the feature vectors at the current and previous time steps.

Moreover, the bidirectional feature of the network, also employed in this thesis, is of paramount importance. In a bidirectional RNN each layer consists of two sub-layers. The first sub-layer processes data in the forward temporal direction, while the second sub-layer handles data in the reverse temporal direction. As a result, during each data pass within the network, it can access information from both past and future time steps. Consequently, the network effectively processes feature vectors from start to end and then from end to start.

This bidirectional model of recurrent neural networks effectively learns temporal relationships from both temporal directions and ultimately combines this information to generate more accurate outputs.

Given these characteristics, the two-way recurrent deep neural network model yields exceptional results with an appropriate amount of training data, typically several thousand hours.

Consequently, the first hypothesis, suggesting that deep neural models yield superior results, is validated both theoretically and empirically.

5.4.2 Second Hypothesis

The second hypothesis posits that Deep Belief Neural Networks will excel with the right amount of data, specifically a few tens of hours. This assumption was made during the proposal preparation, where the expected training data volume was in the tens of hours range.

Theoretically, it is anticipated that DBNs, equipped with a higher number of parameters compared to the Gaussian Mixture Model, would yield improved results when trained on several tens of hours of speech data. However, in a limited data scenario, such as a few tens of hours, recurrent neural networks with their higher

parameter count and end-to-end model may outperform the deep belief network, which is a simpler model. The preliminary findings on the 30-hour dataset corroborate this observation. Nevertheless, as the data volume is increased to several thousand hours, the recurrent neural networks exhibit superior performance due to their enhanced training capacity and temporal characteristics.

5.4.3 Third Hypothesis

The third hypothesis posits that each model has a finite learning capacity, and beyond a certain data volume, increasing data does not improve the model's accuracy. This is because the model's parameters reach their full capacity and cannot comprehend more complex relationships with additional data. This hypothesis is validated through the experiments. For instance, the accuracy of Gaussian Mixture Networks increases up to several tens of hours, as observed in our experiments using 30-hours of data. Similarly, the accuracy of Deep Belief Neural Networks improves up to a data volume of 460 hours. After this point, the accuracy plateaus as these networks reach their training capacity. In contrast, recurrent neural networks continue to be trainable, and their accuracy shows continuous improvement up to 5000 hours of data. These findings reaffirm the validity of the third hypothesis and highlight the importance of considering model capacity concerning data volume.

Furthermore, the results from the previous section concerning the accuracy of language models indicate that accuracy rises with an increase in the value of N in N -grams. However, it is essential to acknowledge that this accuracy enhancement comes at the cost of balancing the final language model's volume and speed against its accuracy. The findings also reveal the positive impact of using Kneser–Ney smoothing in enhancing accuracy.

Additionally, based on the results from Tables 4-13 and 4-14, the direct influence of the language model perplexity on the accuracy of speech recognition systems is evident.

5.5 Conclusion: Key Research Message

This research has yielded significant insights into the factors influencing the accuracy of speech recognition systems. The findings and results emphasize the critical roles played

by both the volume of the training dataset and the architecture of the speech recognition system in determining its accuracy. Additionally, the language model employed also exerts a considerable influence on the overall accuracy of the speech recognition system.

The acoustic models employed in the research, each characterized by its unique structure and approach to learning feature relationships, demonstrate distinct capacities. It is important to recognize that solely increasing the training data volume is insufficient for achieving enhanced accuracy. Rather, achieving higher accuracy necessitates the use of a more comprehensive and suitable model tailored to the inherent complexities of speech data.

Apart from the volume of the Persian FarsAva training speech dataset and the architecture of the model, the Persian language model and certain pre-processing techniques specific to the Persian language significantly contribute to elevating the accuracy of Persian speech recognition systems.

In conclusion, this research underscores the significance of considering multiple factors in the pursuit of accurate speech recognition systems. The interplay of training data volume, model architecture, linguistic model, and language-specific pre-processing is crucial for achieving optimal performance in Persian speech recognition systems.

5.6 Research Limitations

Throughout this research, we encountered various challenges that we diligently addressed to advance the study. These challenges encompassed the scarcity of Persian speech dataset with adequate volume, the absence of Persian language models, the dearth of an ideally voluminous phonetic dictionary in the Persian language, and the lack of preprocessing tools tailored to Persian texts.

Moreover, like any research endeavor, this study is not exempt from limitations. The extensive training time necessitated the evaluation of only one reference structure for each of the models considered. Additionally, the training of recurrent neural networks was constrained by the limited availability of graphic cards, preventing the exploration of diverse network configurations.

5.7 Future Research Suggestions

Based on the findings and the superior performance of Deep Recurrent Neural Networks, it is highly recommended to explore diverse variations of these networks in future research. Such variations may entail investigating different architectural designs, incorporating attention mechanisms into the network, scaling up the number of layers and neurons in each layer, and conducting multiple iterations of the training phase.

In the domain of language models, it is possible to further leverage the models trained on the datasets with larger volume. Additionally, an intriguing suggestion in the context of language modeling is the adoption of probabilistic neural language models, which have recently demonstrated promising outcomes.

Appendices

Appendix A - Theoretical Foundations of Language Models

Overview of Language Modeling

As explained in the preceding chapters, the primary objective of a language model is to estimate the probability of linguistic units, encompassing words and sentences. By employing language models, more accurate output strings can be generated, adhering to the structural and grammatical principles of a given language. In recent years, the application of language models in natural language processing and speech recognition has prompted numerous researchers to delve into enhancing language modeling techniques. Two fundamental approaches are prevalent in constructing a language model: the counting-based method and the continuous-space method, also known as neural probabilistic language models. While the latter exhibits superior performance, it is not without drawbacks, such as prolonged training time and context word limitations.

The traditional N-gram model or probabilistic model operates by establishing an N-th order Markov assumption to estimate the n-gram probabilities and subsequently applying smoothing techniques to these probabilities. Successful approaches, such as the Kneser-Ney smoothing method and the Jelinek-Mercer smoothing method, have been used to improve the language model accuracy. In more recent years, researchers have introduced a novel approach called the continuous space method, comprising sub-branches such as feed-forward neural network language models and recurrent neural network language models. These advancements aim to mitigate the issues of sparsity and discontinuity inherent in n-gram methods.

N-gram Language Models: Probability Estimation through Counting

Employing a statistical probabilistic model for language modeling involves determining the joint probability of word sequences. One of the prevalent modeling methods is the n-gram language model, which builds upon the Markov assumption to derive this joint probability.

In the N-gram model, the probability estimation of a word sequence is decomposed into the probability estimation of each individual word at each step. The language model's probability for a word sequence, such as $P(w_1, w_2, \dots, w_n)$, is computed by multiplying the probability of each word given the preceding words. The number of preceding words influencing the probability of a specific word is typically limited. Mathematically, this limitation is represented by the parameter m , which denotes the number of history words or the words preceding the current word that impact its probability. The following formula depicts this concept:

Equation (A-1)

$$P(w_1, w_2, \dots, w_{n-1}) \approx P(w_{n-m}, \dots, w_{n-2}, w_{n-1})$$

The aforementioned formula signifies the restriction of the number of history words in a language model to m words. It also forms a Markov chain that represents the previous states or, in this context, the preceding words is the order of the Markov model.

The foundational idea behind n-gram-based language models lies in obtaining the joint probability of w_n and w_{n+1} by counting the number of co-occurrences of these two words and dividing it by the frequency of the word w_n . This two-gram model is referred to as bigram. If one only considers the relative frequency of word w_{n+1} in the entire text corpus, a one-gram language model is employed. The estimation of a three-gram language model, following the maximum likelihood estimation, is as follows. Generally, three-gram language models find more extensive application in natural language processing and speech recognition challenges compared to two-gram and one-gram language models.

Equation (A-2)

$$P(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\sum_w \text{count}(w_1, w_2, w_3)}$$

However, when an n-gram language model is employed to estimate the joint probability of a sentence, it often results in a probability of zero. This is due to the infrequent occurrence of certain word combinations in a text corpus, leading to zero probabilities being assigned to test sentences that lie outside the training corpus. It is evident that certain combinations of words not present in the training data may actually be quite likely, necessitating a language model that avoids assigning zero probability to such sentences.

This issue is known as the "discontinuity problem" in n-gram language models. While methods like backoff (Kneser & Ney, 1995) and smoothing (Stanley F. & Goodman, 1999) and (Goodman, 2001) have been proposed to address this problem, a perfect solution remains elusive.

Besides the aforementioned smoothing methods, one of the substantial challenges of n-gram language models, shared with various other machine learning algorithms, is the curse of dimensionality. In this context, the number of distinct states that an n-gram must handle becomes exceedingly large. In the case of language modeling, this number corresponds to the vast array of states represented by word sequences used to construct sentences. For example, considering a word sequence comprising ten words from a dictionary of 100,000 words, the number of distinct states reaches an overwhelming 10^{50} states.

N-gram language models also exhibit other limitations. For instance, they heavily rely on patterns of words and sentences, lacking a profound understanding of language from a linguistic perspective. Even replacing a word with a synonym can lead to a substantial difference in their probability estimation. Another drawback of Markovian language models, including n-grams, is their failure to account for word dependencies beyond the specified context window. In the estimation of a word's probability, an n-gram model only considers the probability of the previous $n - 1$ words, overlooking the influence of words beyond this window. In essence, the n-gram model cannot fully capture conditional probabilities under the Markov assumption, which contradicts human language processing. Humans take into account words that are far apart in the sentence when estimating the probability of the current word.

To address these limitations and seek better modeling of semantic and lexical dependencies in language, the adoption of deep learning and deep neural networks for building language models seems promising. Neural networks have the potential to learn language dependencies more effectively and mitigate the curse of dimensionality through enhanced generalization. While n-gram language models have been predominantly used in this thesis, the proposal stemming from this research is to incorporate neural network language models into speech recognition systems. Therefore, in the subsequent part of this appendix, neural network language models are elaborated upon, along with their potential applications.

Probabilistic Neural Network for Language Modeling

Continuous space language models, also known as probabilistic neural language models, constitute two primary types based on neural networks. The first type employs feed-forward neural networks to address the discontinuity problem encountered in n-gram language models. The second type employs recurrent neural networks to overcome the limitation of vocabulary context present in previous language models.

To tackle these challenges, language models based on neural networks have been proposed. Subsequent research in this area has explored neural network-based modeling at the sub-word layer and corpus-based language models employing recurrent neural networks, particularly those incorporating long short-term memory.

Probabilistic Neural Language Model based on Feed-Forward Neural Network

One prominent approach in the domain of probabilistic neural network-based language models is the feed-forward probabilistic neural network. This network is trained based on the conditional probability distribution of the next word given the preceding $n - 1$ words. First introduced in the work of (Bengio et al., 2000), this network comprises three forward layers. The general schematic of this network is illustrated in Figure A-1.

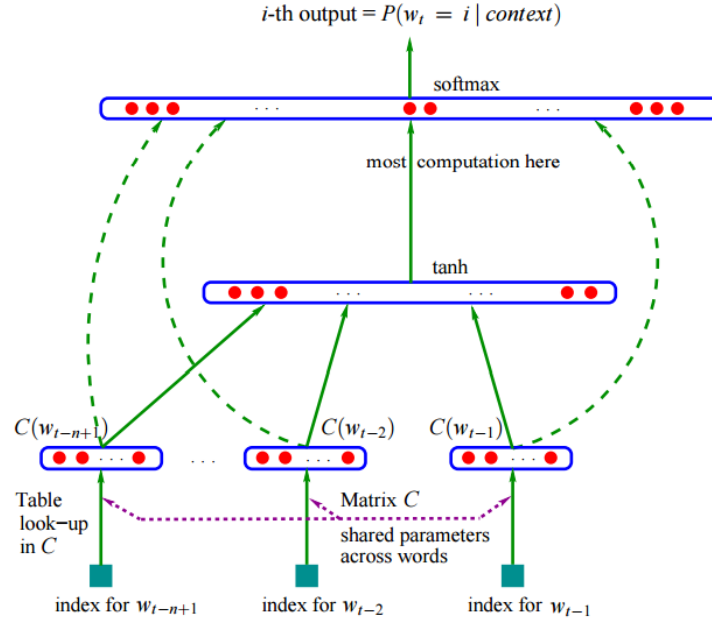


Figure (A-1) General schematic of a probabilistic neural language model using a feed-forward neural network. Image adapted from (Bengio et al., 2000)

In this architecture:

- The model establishes a mapping from the context (C) of each word " i " in the vocabulary (V) to a vector in the real space, represented as $C(i) \in \mathbb{R}^m$, where " m " denotes the dimension of the feature vector. C is a matrix with dimensions $|v| \times m$, where each row " i " corresponds to the feature vector $C(i)$ for word " i ".
- A function " g " on words maps the input word string's feature vector $(C(w_{(t-n+1)}), \dots, C(w_{(t-1)}))$ to a conditional probability distribution over words in the vocabulary V , creating w_t for the next word.
- Finally, the feature vectors of the words and the parameters of the second probabilistic model are jointly trained using a composite function " f " that encompasses two functions: C and g .

Equation (A-3)

$$f(i, w_{t-1}, \dots, w_{t-n+1}) = g(i, C(w_{(t-n+1)}), \dots, C(w_{(t-1)}))$$

In this model, every word in the vocabulary is assigned a continuous and distributed feature vector. The joint probability function of a sequence of words is then defined

based on these feature vectors. Consequently, the model is capable of learning both the feature vectors of the words and the joint probability function simultaneously.

This neural network-based approach effectively addresses the issue of discontinuity present in n-gram language models. Moreover, experimental results demonstrate its superior performance and improved perplexity compared to n-gram models in terms of generalization. Nevertheless, a significant drawback of this method is the extensive time required for model training and testing.

Probabilistic Neural Language Model based on Recurrent Neural Network

In contrast to the language model method based on feed-forward neural networks described in the preceding section, language models based on recurrent neural networks do not rely on a fixed and limited background or history of words. Instead, they utilize the recursive structure of the network, allowing the background information and word history to be dynamically maintained in the network for a proportional period.

The paper by (Mikolov et al., 2010) presents a more generalized approach to language models based on recurrent networks compared to the basic model. In this introduced network, neurons with long short-term memory (LSTMs) are used and all the previous sequence of words are taken into consideration for the probability of the next word rather than utilizing a fixed number of context words. The detailed depiction of this network is illustrated in the diagram below.

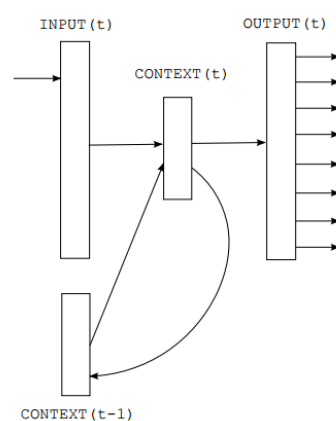


Figure (A-2) Schematic of a language model based on recurrent neural network with long short-term memory neurons. Image from (Jin, 2017)

Key Characteristics of the Model:

- The input layer w and the output layer y have the same dimensions, equal to the number of words in the dictionary.
- The hidden layer s is exponentially smaller and typically comprises about 50 to 100 neurons.
- The U matrix represents the weights between the input layer and the hidden layer, while the V matrix represents the weights between the hidden layer and the output layer.
- Without the recursive weights W , this model functions as a two-gram language model.

In the preceding sections, two language models were presented—one based on feed-forward neural networks and the other on recurrent neural networks. Below, we briefly highlight the main differences between these two models.

The language model based on feed-forward neural networks requires a fixed number of context words. On the other hand, neural networks based on recurrent neural networks theoretically make use of all context words, though practical observations have shown that the number of effectively used background words is limited in these networks too.

Nevertheless, language models based on recurrent neural networks offer the advantage of not requiring an explicit number of context words—a parameter that can be challenging to determine accurately and effectively.

Considering that recurrent neural networks are dynamic networks, certain challenges may arise during training and inference. One such issue is the possibility for the network to be exponentially affected by some sequence of inputs, which can have negative implications for other inputs.

Appendix B - Gaussian Mixture Model

Gaussian Distribution

The Gaussian distribution, also known as the normal distribution, is a Probability Density Function (PDF) commonly employed to describe physical and natural phenomena with unknown distributions.

The Gaussian mixture model finds extensive application in speech recognition systems as a statistical model for classifying features extracted from speech. A random variable x distribution is Gaussian if its probability density function is as follows:

Equation (B-1)

$$p(x) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] = N(x; \mu, \sigma^2)$$

Here, μ represents the mean value and σ^2 is the variance.

Multivariate Normal Distribution

The multivariate normal distribution, or multivariate Gaussian distribution, serves as a generalization of the one-dimensional Gaussian distribution to higher dimensions. For a random vector $x = (x_1, x_2, \dots, x_D)$, it is said to be normally distributed if it adheres to the following probability density function:

Equation (B-2)

$$p(x) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right] = N(x; \mu, \Sigma)$$

In this equation, Σ denotes the covariance matrix, and μ represents the mean value. The strength of this formulation lies not only in its favorable computational properties but also in its ability to model various natural data in the real world.

Gaussian Mixture Models

The GMM is a probabilistic model that assumes all data are generated from a combination of multiple Gaussian distributions, each with unknown parameters. This model can be seen as a generalized extension of the K-means model, incorporating both data covariance structure and Gaussian centers (Pedregosa et al., 2011).

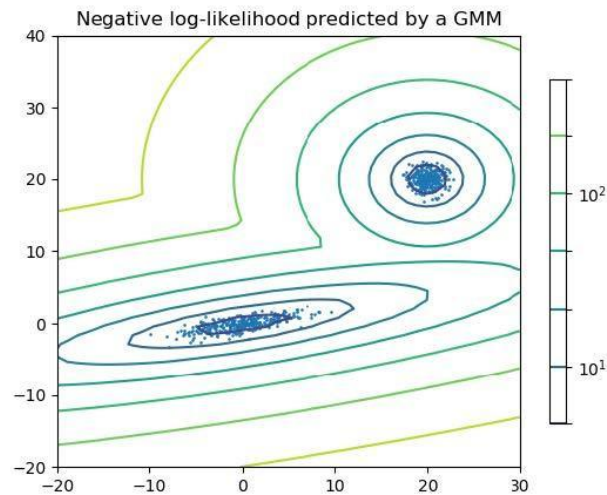


Figure (B-1) Two-component Gaussian Mixture Model

A true random variable adheres to a Gaussian mixture normal distribution if its probability density function follows the following equation:

Equation (B-3)

$$p(x) = \sum_{m=1}^M \frac{c_m}{(2\pi)^{\frac{1}{2}} \sigma_m} \exp \left[-\frac{1}{2} \left(\frac{x - \mu_m}{\sigma_m} \right)^2 \right] = \sum_{m=1}^M c_m N(x; \mu_m, \sigma_m^2)$$

where c_m represents the weight of the mixture, and M is the number of components. The weight of each mixture component is a positive value, and their sum equals one, i.e.,

$$\sum_{m=1}^M c_m = 1.$$

The Gaussian mixture distribution is a multimodal model, meaning it comprises M probability density functions. In contrast, the unimodal Gaussian model consists of only one such function. This multimodality enables the model to effectively represent various

natural data, including speech data, as natural data often exhibits multiple structural patterns.

As a result, the Gaussian mixture distribution is a powerful tool for modeling natural data, making it a fundamental component of traditional speech recognition systems. Furthermore, Equation B-3 can be generalized to accommodate a multivariate Gaussian mixture distribution, which features a probability density function as expressed below:

Equation (B-4)

$$p(x) = \sum_{m=1}^M \frac{c_m}{(2\pi)^{D/2} |\Sigma_m|^{1/2}} \exp \left[-\frac{1}{2} (x - \mu_m)^T \Sigma_m^{-1} (x - \mu_m) \right] = \sum_{m=1}^M c_m N(x; \mu_m, \Sigma_m)$$

Expectation Maximization Algorithm

The Expectation Maximization (EM) algorithm is an iterative procedure used for maximum likelihood estimation on unlabeled data. In this method, the model parameters are adjusted in a way that maximizes the probability of the observed data.

The EM algorithm falls under the category of soft clustering algorithms, in contrast to hard clustering methods like the K-means algorithm. Soft clustering assigns a sample to all clusters with specific probabilities, rather than exclusively to a single cluster. Figure B-2 illustrates the steps involved in the EM algorithm, with three iterative stages displayed. In each step, data points are initially labeled based on the model parameters, followed by updating the model parameters using maximum likelihood estimation.

The EM algorithm consists of two fundamental stages: the Expectation stage and the Maximization stage. In the Expectation stage, data is labeled according to the current model parameters. Subsequently, in the Maximization stage, the maximum likelihood estimation function is maximized based on the labeled data from the Expectation stage. These two steps are repeated iteratively.

During the initialization stage, the model parameters are randomly selected. The training process ensures convergence as the maximum likelihood estimation function is maximized in each iteration. While there is no universal rule for terminating the repetitive training steps, a threshold is set to measure the increase in the value of the

maximum likelihood estimation function. If the increase is below the desired threshold, the training is halted.

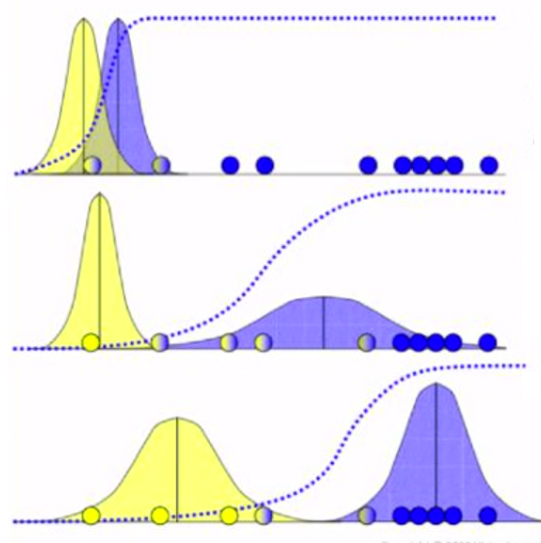


Figure (B-2) Steps of the Expectation Maximization Algorithm

The EM algorithm is widely used for estimating the parameters of Gaussian mixture models with a specified number of components. However, this method has some drawbacks. One disadvantage is the possibility of getting trapped in a local maximum during training. Additionally, the algorithm is sensitive to the choice of initial values. To mitigate the sensitivity to initial values, multiple training runs can be performed with different initializations, although this may incur additional computational cost.

Gaussian Mixture Model for Probability Distribution Function of Speech Features

Following the extraction of features from the speech audio file, applying the logarithm on the short-time Fourier transform on the signal power or related cepstra, the Gaussian mixture model discussed earlier has demonstrated its efficacy in modeling these features. However, it is crucial to note that this modeling does not account for time dependence information, making it suitable for single-frame speech feature modeling. The Gaussian mixture model utilizes the Gaussian probability distribution when representing the data.

Gaussian mixture models are widely employed in statistical classification for data modeling in speech research and other domains due to their ability to handle complex and diverse data distributions. In speech research, classifiers based on Gaussian mixture models find effective applications in tasks like speaker identification, noise removal during speech feature extraction, and speech recognition.

Speaker identification involves using Gaussian mixture models as a Universal Background Model (UBM) to represent the distribution of speech features from various speakers. Similarly, Gaussian mixture models serve as prior distributions for noise removal purposes. For speech recognition tasks, Gaussian mixture models are combined with Hidden Markov Models (HMMs) to enable accurate speech recognition.

When temporal information of speech is taken into consideration, Gaussian mixture models lack a temporal modeling structure. Hidden Markov Models are well-suited for temporal modeling in such cases. The combination of Gaussian Mixture Models with Hidden Markov Models proves beneficial, as the former can accurately model the probability distribution of speech features at specific time instances when the Hidden Markov Model is in a particular state.

Several advantages make Gaussian Mixture Models suitable for modeling the probability density function (PDF) of extracted speech features in each Hidden Markov Model state. Firstly, with an adequate number of components, Gaussian mixture models can accurately model probability density function. Secondly, the expectation maximization training algorithm facilitates fast training of the model on the data.

Numerous efforts have been made to increase the evaluation speed of Gaussian Mixture Models and optimize their flexibility ratio and data requirements to avoid overfitting. These efforts have led to the development of various Gaussian mixture models, such as the parameter-tied GMMs, semi-tied GMMs, and subspace GMMs.

Beyond using the expectation maximization algorithm to learn the parameters of a Gaussian Mixture Model, the accuracy of speech recognition systems based on these models has significantly improved by employing discriminative training of parameters after initial training using the expectation maximization generative learning algorithm. Discriminative learning has proven to be highly effective, especially when the optimization objective function is defined based on the error rate of phonemes or words.

Gaussian mixture models have enjoyed considerable success in modeling features extracted from speech over the years. However, their supremacy was challenged in 2011 when deep neural network models emerged, demonstrating superior accuracy in comparison.

Despite the successes of Gaussian mixture models in modeling speech feature extraction over the years, their limitations should not be overlooked. Gaussian mixture models are statistically inefficient in modeling data located near or on the edge of a manifold in the data space. For instance, representing data on a sphere in 3D space can be easily accomplished with a geometrical model, while using a Gaussian mixture model would require numerous distribution functions. For speech data, which is produced by a limited number of parameters in the human voice system, it is expected that other models capable of extracting superior features from speech will yield better results than Gaussian mixture models.

Appendix C - Fundamentals of Deep Neural Networks

In a seminal paper by (Hinton et al., 2006), it was demonstrated that Restricted Boltzmann Machines (RBMs) can be effectively stacked together and trained using a layer-by-layer greedy method. This novel arrangement, known as the first deep neural network, laid the foundation for Deep Belief Neural Networks—a class of hybrid generative graphical models capable of extracting hierarchical representations from training data.

Deep Belief Neural Networks model the joint probability distribution of the input vector x and the l hidden layers h^1 to h^l as shown below:

Equation (C-1)

$$p(x, h^1, \dots, h^l) = \left(\prod_{k=0}^{l-2} P(h^{k+1}) \right) P(h^{l-1}, h^l)$$

In the above equation, x corresponds to the input data and $P(h^{k+1})$ represents the conditional probability of the visible vector relative to the hidden vector in layer k of the Restricted Boltzmann Machine. Likewise, $P(h^{l-1}, h^l)$ denotes the conditional probability of the visible vector relative to the hidden vector in the final layer of the deep belief network. The figure below illustrates a layer of the Restricted Boltzmann Machine, and Figure (C-2) showcases a Deep Belief Network consisting of multiple layers of Restricted Boltzmann Machine.

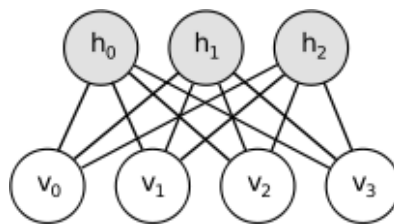


Figure (C-1) A restricted Boltzmann neural network comprising a visible layer and a hidden layer. Image source: (Restricted Boltzmann Machines, 2018)

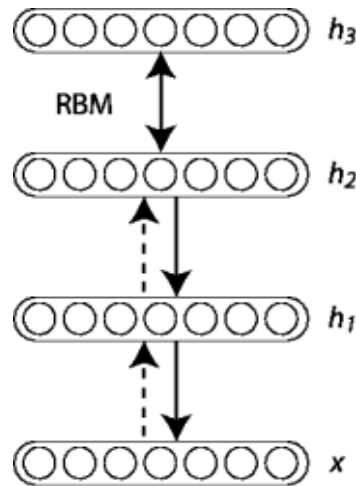


Figure (C-2) A deep belief neural network composed of Restricted Boltzmann Machines. Image source: (Deep Belief Networks, 2018)

The training procedure, based on the works by (Hinton et al., 2006) can be summarized as follows:

1. Training the first layer of the deep belief neural network, which consists of a Restricted Boltzmann Machine with input data as the visible layer $x = h^0$.
2. Utilizing the output or hidden layer from the first Boltzmann Machine as the input or visible layer for the second layer.
3. Training the third layer as another Boltzmann Machine and obtaining the output or hidden layer from this layer.
4. Repeating steps 2 and 3 to train all subsequent layers.
5. Optimizing all network weights using a supervised learning algorithm.

The layer-by-layer greedy method enables the extraction of rich hierarchical representations, making Deep Nelief Neural Networks a powerful tool in various applications, including speech recognition and image processing. These networks have proven to be highly effective in modeling complex data distributions and learning informative representations from large datasets.

Appendix D - Fundamentals of Recurrent Neural Networks

The inherent sequential and temporal nature of audio data makes recurrent neural networks (RNNs) an apt choice for modeling acoustic structures. RNNs possess feedback connections that enable them to retain past inputs and leverage the information embedded in sequential data. This characteristic is particularly beneficial when dealing with real-world data, such as speech and video, that exhibit temporal dependencies.

Feed-Forward Recurrent Neural Networks

Unlike feed-forward neural networks, which lack memory of previous inputs, RNNs use recurrent connections to remember past inputs and exploit temporal information. Conceptually, an RNN behaves like a feed-forward neural network that unfolds in time, processing input information over successive time steps. However, a significant challenge with simple RNNs is their limited ability to capture long-term dependencies, as they typically focus on only a few previous time steps.

The figure below illustrates a simple RNN with both its closed recurrent form and its unfolded representation in time.

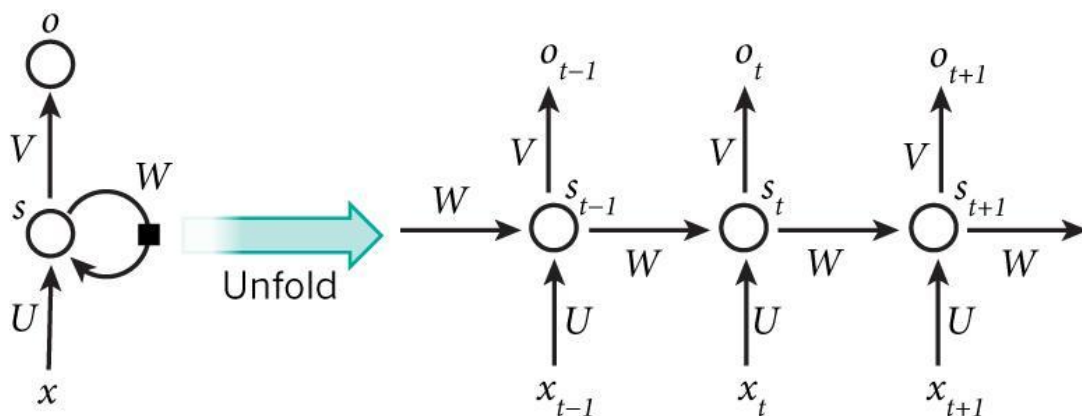


Figure (D-1) A simple recurrent neural network and its unfolded representation in time. Image source: (Karpathy, 2015)

Mathematically, considering the input vector $x = (x_1, \dots, x_T)$, a simple RNN computes the hidden vector $h = (h_1, \dots, h_T)$ and the output vector $y = (y_1, \dots, y_T)$. The

output vector is iteratively calculated from time step $t = 1$ to T based on the following equations:

(Equation D-1)

$$h_t = H(W_{xh} x_t + W_{hh} h_{t-1} + b_h)$$

(Equation D-2)

$$y_t = W_{hy} h_y + b_y$$

In these relations, W represents the weight matrix, b is the network bias vector, and H denotes the activation function used in the hidden layer. Typically, the activation function is implemented as a sigmoid function (Graves & Jaitly, 2014).

Recurrent neural networks have demonstrated great potential in modeling sequential data, making them a crucial component in speech recognition systems and various other applications involving temporal data analysis. Nevertheless, it should be noted that simple RNNs have limitations in handling long-term dependencies. These limitations led to the development of more sophisticated RNN variants, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which have shown superior performance in capturing and preserving long-term temporal information (Hochreiter & Schmidhuber, 1997) (Chung et al., 2014).

The Challenge of Long-Term Dependencies

One appealing aspect of simple recurrent neural networks lies in their potential to relate past information to the current context. For instance, they may use the frame of a previous image in a video to better understand the current frame. However, it is essential to acknowledge that these networks might not effectively capture such relationships in all situations.

In certain cases, accurate categorization of the current input can only be achieved by focusing on the most recent information. An illustration of this is evident in two-gram language models, where predicting the current word primarily relies on the previous word. For example, in the sentence "clouds are in the sky," predicting the word "sky" requires minimal background information. When relevant temporal information is in

close proximity to the current time step, simple recurrent neural networks can indeed utilize this short-term temporal dependence, as depicted in the figure below.

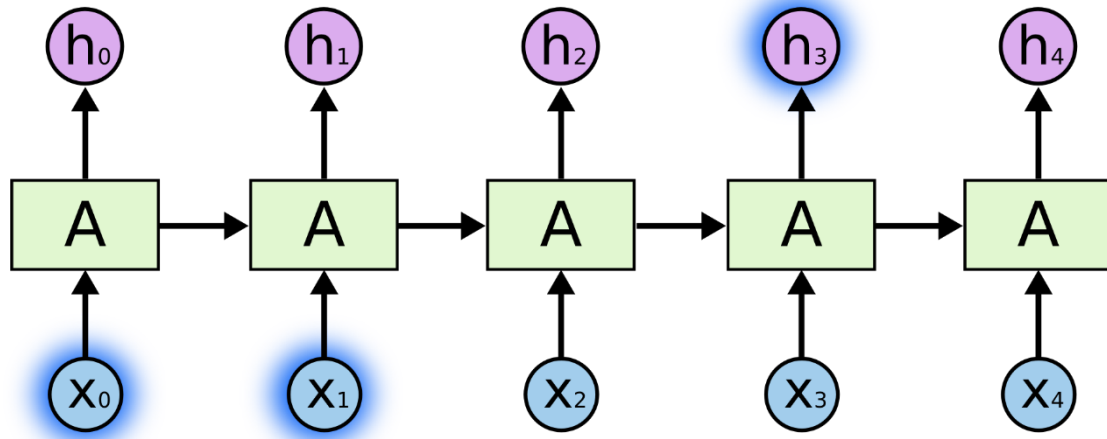


Figure (D-2) Short-term dependencies in a recurrent neural network. Image source: (Karpathy, 2015)

However, there are scenarios where more extensive contextual information is necessary. For instance, consider the sentence "I grew up in France... I speak French fluently". While attempting to predict the subsequent words, the available context suggests that the next word is likely to be the name of a language. Yet, to precisely determine the language, one needs to reference the word "French" from the initial part of the sentence. The gap between relevant information and its required position in the sequence can be substantial. As shown in the figure below, the hidden vector at time step $t + 1$ may necessitate information from a much later time, such as at time x_0, x_1 . Regrettably, as this temporal gap widens, simple recurrent neural networks struggle to establish these connections, leading to the problem of long-term dependency.

In theory, recurrent neural networks possess the capacity to learn long-term dependencies accurately, provided that their parameters are well-chosen. In some straightforward problems, they can indeed learn such dependencies. However, practical experiments conducted by researchers have revealed that in general, simple recurrent neural networks often fall short in learning these relationships. This issue has been thoroughly investigated by Hochreiter (Hochreiter & Schmidhuber, 1997) and Bengio (Bengio et al., 1994), and the fundamental reasons behind this limitation have been

expounded in their respective works. On the other hand, recurrent neural networks equipped with long short-term memory (LSTM) have demonstrated the ability to effectively learn long-term dependencies. Thus, this type of recurrent network serves as a powerful tool for modeling temporal data.

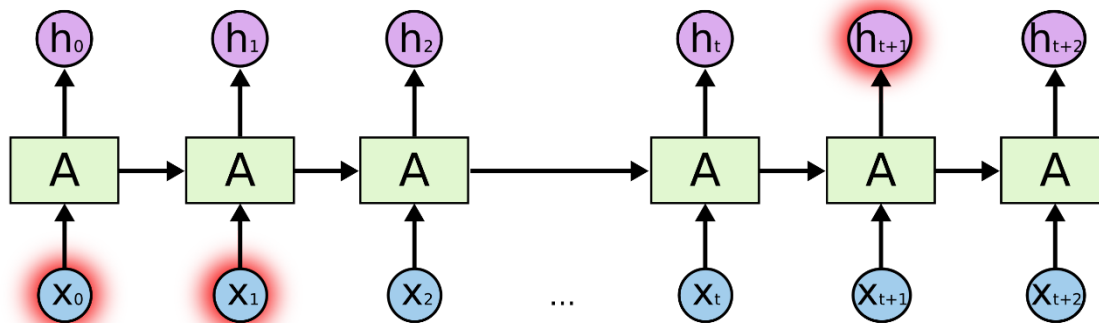


Figure (D-3) Long-term dependencies in a recurrent neural network. Image source: (Karpathy, 2015)

Recurrent Neural Networks with Long Short-Term Memory

Recurrent Neural Networks (RNNs) equipped with Long Short-Term Memory (LSTM) are a specialized variant designed to tackle the issue of learning long-term dependencies in data. Initially introduced by Hochreiter and Schmidhuber in their seminal paper (Hochreiter & Schmidhuber, 1997), these networks have since undergone numerous refinements by various researchers, delivering impressive results in diverse machine learning tasks.

LSTMs are explicitly crafted to overcome the long-term dependence problem that hampers conventional RNNs. These networks inherently grasp and utilize information concerning long-term temporal dependencies. In the architecture of any RNN, a repeating chain of neural network modules is formed. In the case of a simple RNN, this recurrent module is uncomplicated, consisting of a single layer with either a tangent or sigmoid activation function, as depicted in the figure below.

However, LSTMs diverge from this simplicity, featuring a distinctive recurrent module with four interconnected and specialized activation functions, as illustrated in the figure D-5.

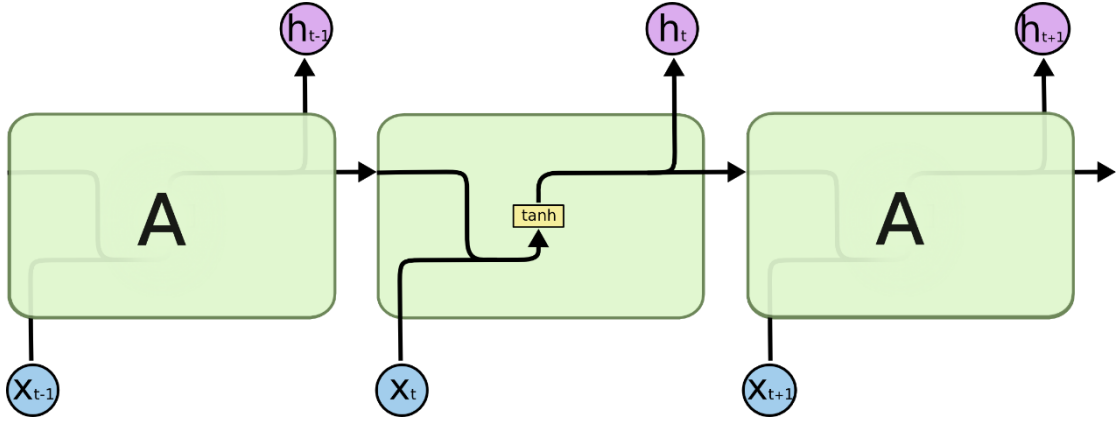


Figure (D-4) Recurrent module in a single-layer simple recurrent neural network. Image source: (Karpathy, 2015)

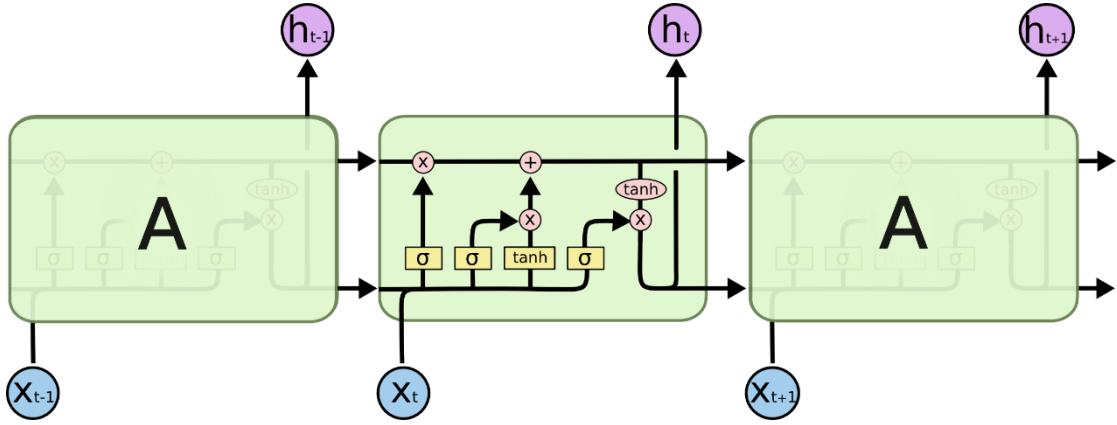


Figure (D-5) Recurrent module in a single-layer Long Short-Term Memory network. Image source: (Karpathy, 2015)

Within LSTM networks, three gates play a crucial role: the input gate, output gate, and forgetting gate. By leveraging these gates, an LSTM neuron effectively functions as a memory cell. The output of an LSTM network is computed using the following equations:

(Equation D-3)

$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i)$$

(Equation D-4)

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f)$$

(Equation D-5)

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c)$$

(Equation D-6)

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o)$$

(Equation D-7)

$$h_t = o_t \tanh(c_t)$$

In the above equations, the sigmoid function is denoted as σ . Additionally, i , f , o , and c represent the input gate, forgetting gate, output gate, and neuron activation vector, respectively. All of these vectors share dimensions with the output vector h .

The weight matrices connecting the gates to the neuron's inputs are all diagonal matrices, meaning that the m th element in the gate vectors is solely influenced by the m th element in the input (Graves et al., 2013). The figure below provides a visual representation of an LSTM neuron, incorporating the abbreviated notations from the equations mentioned above.

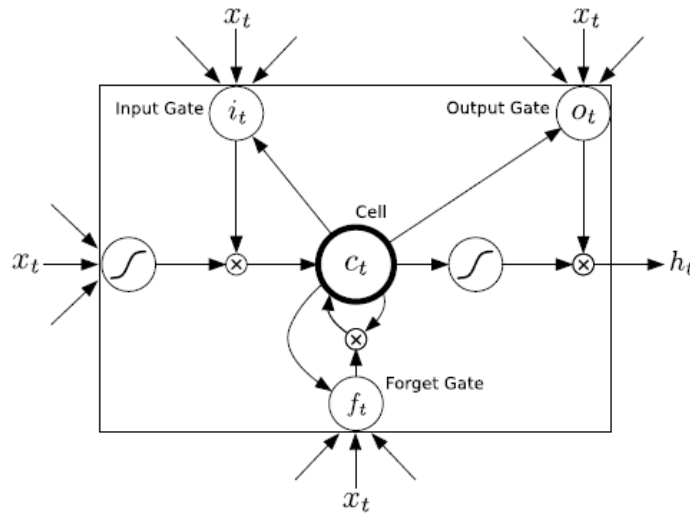


Figure (D-6) LSTM neuron. Image source: (Graves et al., 2013)

Appendix E - Connectionist Temporal Classification (CTC)

In 2006, Alex Graves introduced a groundbreaking technique known as Connectionist Temporal Classification (Graves et al., 2006). This classification algorithm revolutionized research in sequence-to-sequence problems and allowed for the handling of unsegmented voluminous datasets. Recurrent Neural Networks are highly effective in modeling data with sequential structures. However, prior to the introduction of this category, sequential data had to be partitioned, limiting the efficiency of recurrent neural networks due to the labor-intensive and time-consuming nature of manual data segmentation.

In this thesis, Connectionist Temporal Classification is harnessed to train deep recurrent neural networks using thousands of hours of unsegmented data. Herein, we outline the time segmentation algorithm.

Drawing from the mathematical literature of speech recognition, the algorithm is explained. In speech recognition, the objective is to establish a correct mapping from the audio input sequence $X = [x_1, x_2, x_3, \dots, x_t]$ to the output word sequence $Y = [y_1, y_2, y_3, \dots, y_u]$. Challenges such as variable length of X and Y , disparate ratios of X and Y , and the lack of exact alignment between X and Y render conventional methods and traditional machine learning algorithms ineffective in solving this problem. By employing the Connectionist Temporal Classification, we can compute the distribution of all possible outputs for a given input. This distribution calculation enables us to infer probable outcomes or assess the probability of one occurrence. Essentially, the Connectionist Temporal Classification addresses the problem of intractability of the objective function and the feasibility of inference.

For the optimal calculation of the objective function, the conditional probability $p(X)$ needs to be determined. Additionally, this function must be differentiable to apply the gradient descent algorithm. Solving the equation $Y^* = \operatorname{argmax}_y p(X)$ is crucial for deriving the output. The Connectionist Temporal Classification facilitates computable solutions for both the objective function and the inference equation.

As previously mentioned, the Connectionist Temporal Classification algorithm does not necessitate aligning the input and output data. Instead, it computes the objective function by summing the probabilities of all potential alignment states between the input and output. To generate all feasible alignment states, the CTC utilizes a softmax layer, containing one additional neuron beyond the number of output space labels L .

Activation of the first L neurons in the softmax function corresponds to observing the corresponding label at a specific time in the input. Activation of the additional neuron signifies the absence of a label or an empty label, commonly denoted as epsilon (ϵ). The aligned strings in the CTC algorithm are of length X . Moreover, this algorithm enumerates all the aligned strings after removing ϵ and merging repeated labels, ultimately representing the output Y and enabling the determination of the objective function's error rate.

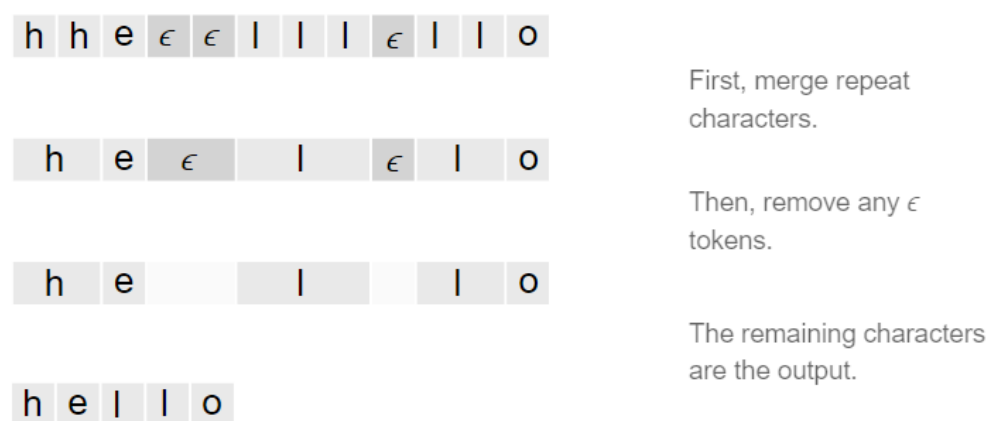


Figure (E-1) Output of the Connectionist Temporal Classification after removing ϵ and merging repeated labels. Image source: (Hannun, 2017)

Aligned strings in the temporal alignment clustering algorithm exhibit three distinct characteristics. Firstly, the possible aligned strings between input and output are one-to-one. Secondly, the aligned strings from X to Y are many-to-one, implying that there is only one output for every one or more inputs. Lastly, the length of the output string is less than or equal to the input length.



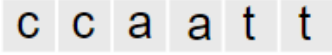
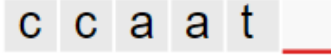

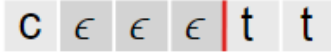
Valid Alignments	Invalid Alignments	
		corresponds to $Y = [c, c, a, t]$
		has length 5
		missing the 'a'

Figure (E-2) Valid and non-valid aligned strings in the temporal alignment clustering algorithm.

Image source: (Hannun, 2017)

The objective function for an input and output pair is calculated as follows:

(Equation E-1)

$$p(X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^T p_t(X)$$

The joint probability of the Connectionist Temporal Classification is denoted as $p(X)$,

while $\sum_{A \in A_{X,Y}}$ performs normalization on all valid aligned strings. Furthermore, $p_t(X)$ represents the output probabilities of the neural network at each time step.

The aforementioned objective function is computationally intensive, posing significant time constraints. To address this, the time series algorithm employs dynamic algorithms to calculate the objective function much faster. The core idea of the time series algorithm involves merging strings that have reached the same output at the same time step. This approach renders the probability summation of all aligned strings computationally feasible. Through this merging process, the calculations become significantly faster, significantly reducing computational time.

Once the above objective function has been efficiently calculated, gradients can be computed, and model parameters can be trained using unaligned data. The objective function of the Connectionist Temporal Classification is differentiable with respect to the time steps.

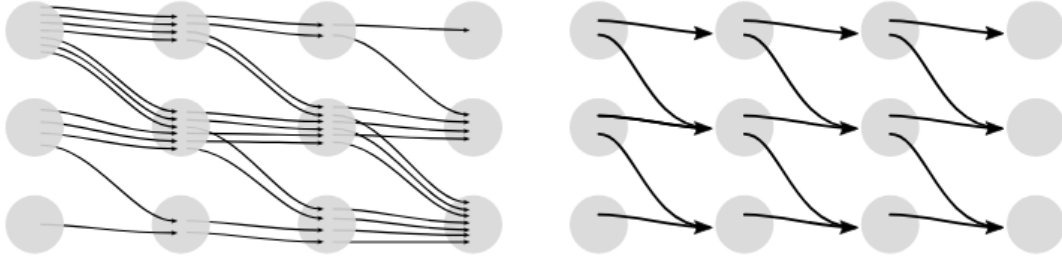


Figure (E-3) Merging Paths with the Same Output. Image source: (Hannun, 2017)

Once the above objective function has been efficiently calculated, gradients can be computed, and model parameters can be trained using unaligned data. The objective function of the Connectionist Temporal Classification is differentiable with respect to the time steps.

When training on a dataset D , model parameters are typically optimized to minimize the negative logarithm of the objective function, rather than directly maximizing the objective function.

(Equation E-2)

$$\sum_{(X,Y) \in D} -\log \log p(X)$$

To achieve the aforementioned inference, the following equation needs to be solved.

(Equation E-3)

$$Y^* = \operatorname{argmax}_y p(X)$$

In his article, Alex Graves proposed a heuristic method for inference, which involves selecting the most probable label at each time step. This method infers an output string based on the most likely label selected at each time step.

(Equation E-4)

$$A^* = \operatorname{argmax}_A \prod_{t=1}^T p_t(X)$$

While this heuristic method often produces accurate outputs, especially when the probabilities overwhelmingly favor a single output string, it is a greedy approach and may not always generate the most probable string. This limitation occurs, particularly when the probabilities of the correct output string are spread across several aligned strings.

To address this, the prefix search decoding method can be employed as an alternative approach. The prefix search decoding method does not guarantee finding the correct output but, with more computations, it increases the likelihood of obtaining a more accurate string.

A standard prefix search generates several new hypotheses or aligned strings at each time step. These hypotheses are created by adding the probabilities of all labels at the current time step to the string generated up to that point. Subsequently, only the top hypotheses with the highest probabilities are retained. The figure below illustrates a standard prefix search with a dictionary $\{\epsilon, a, b\}$ and a extension value of three.

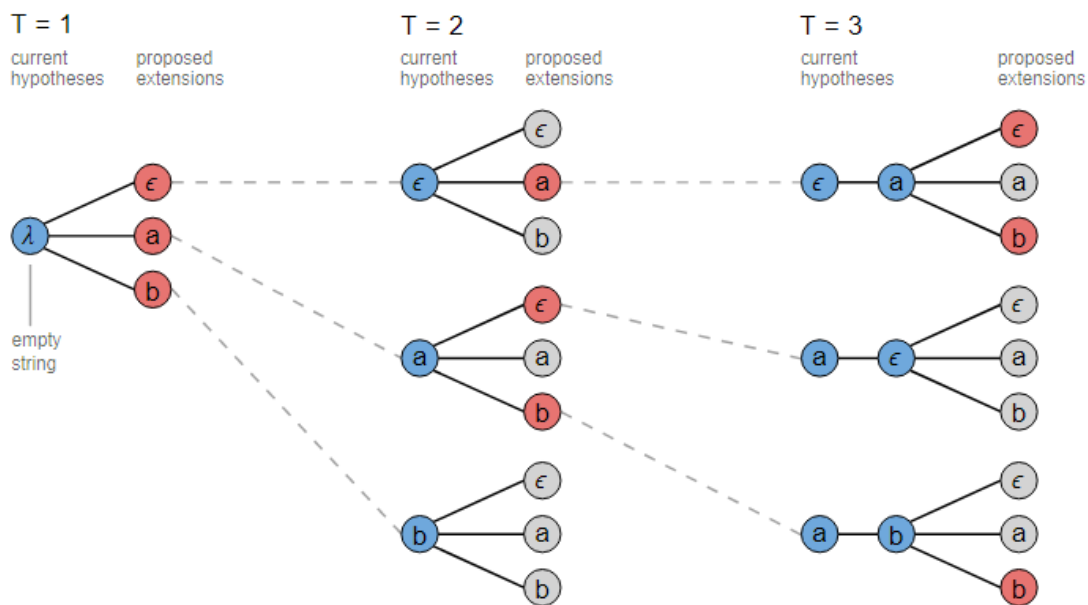


Figure (E-4) Prefix search decoding with Dictionary $\{\epsilon, a, b\}$ and a extension value of three.

Image source: (Hannun, 2017)

With minor adjustments, the prefix search decoding can be modified to merge the probabilities of similarly aligned strings at each time step, after combining the labels and removing epsilon. This modification involves retaining pre-strings instead of all aligned strings. These pre-strings are evaluated based on the sum of probabilities from all their constituent aligned strings.

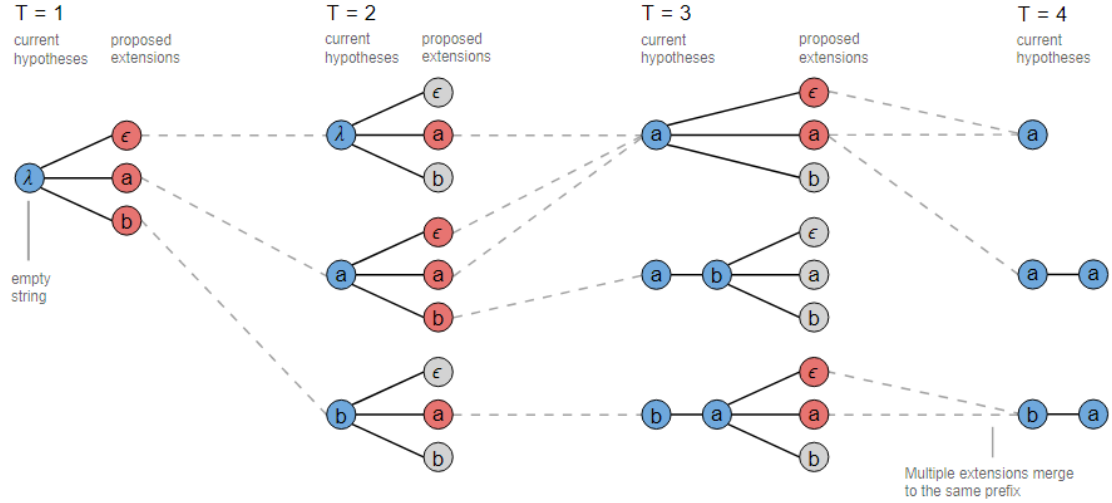


Figure (E-5) Prefix search decoding modified by merging similar aligned atrings. Image source: (Hannun A., 2017)

The modified prefix search decoding algorithm has been implemented and described in the reference (Hannun A., 2017).

In the explanation of the Connectionist Temporal Classification algorithm, the literature of the speech recognition problem has been employed. Notably, the integration of a Language Model with the CTC algorithm has shown to enhance the accuracy of speech recognition systems. The language model can be incorporated during the model's inference using the following formula.

(Equation E-5)

$$Y^* = \underset{y}{\operatorname{argmax}} p(X) \cdot p(Y)^\alpha \cdot L(Y)^\beta$$

In this formula, $p(Y)$ represents the probabilities of the language model, and $L(Y)$ denotes the increase in the probability of inserting the word. Parameters α and β are typically determined through validation data.

References

- Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014, July). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10), 1533 - 1545.
10.1109/TASLP.2014.2339736
- Abdel-Hamid, O., Mohamed, A., Jiang, H., & Penn, G. (2012). Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1(1), 4277-4280. 10.1109/ICASSP.2012.6288864
- Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., & Casper, J. (2016, June). Deep speech 2: End-to-end speech recognition in english and mandarin. *International conference on machine learning, PMLR 48*, 173-182.
- BabaAli, B. (2014). *Kaldi Results on Farsdata Dataset*. github. Retrieved 2013, from <https://github.com/kaldi-asr/kaldi/blob/master/egs/farsdat/s5/RESULTS>
- Bengio, Y., Ducharme, R., & Vincent, P. (2000). A Neural Probabilistic Language Model. *Advances in neural information processing systems*, 13.
- Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *NIPS*, 19, 153–160.
- Bengio, Y., Simard, P., & Frasconi, P. (1994, March). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 2(5), 157-166. 10.1109/72.279181
- Bijankhan, M., Sheikhzadegan, J., & Roohani, M. R. (1994). FARSDAT-The speech database of Farsi spoken language. *PROCEEDINGS AUSTRALIAN CONFERENCE ON SPEECH SCIENCE AND TECHNOLOGY*.
- Chan, W., & Lane, I. (2015). Deep recurrent neural networks for acoustic modelling. *arXiv, preprint arXiv:1504.01482*. 0.48550/arXiv.1504.01482
- Chorowski, J., Bahdanau, D., Cho, K., & Bengio, Y. (2014, December). End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. *arXiv, preprint arXiv:1412.1602*. 10.48550/arXiv.1412.1602

- Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014, December). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS - Deep Learning and Representation Learning Workshop*. 10.48550/arXiv.1412.3555
- Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012, January). Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1), 30 - 42. 10.1109/TASL.2011.2134090
- Davis, S., & Mermelstein, P. (1980, August). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4), 357 - 366. 10.1109/TASSP.1980.1163420
- Deng, L., Abdel-Hamid, O., & Yu, D. (2013). A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1(1), 6669-6673. 10.1109/ICASSP.2013.6638952
- Garofolo, J. S. (1993). Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium*.
- Garofolo, J. S., Graff, D., Paul, D., & Pallett, D. (1993). *CSR-I (WSJ0) Complete - Linguistic Data Consortium*. LDC Catalog. Retrieved August 7, 2023, from <https://catalog.ldc.upenn.edu/LDC93s6a>
- Geitgey, A. (2016, December 23). *Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning*. Medium. Retrieved August 7, 2023, from <https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a>
- Goodman, J. T. (2001, October). A bit of progress in language modeling. *Computer Speech & Language*, 15(4), 403-434. 10.1006/csla.2001.0174
- Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural

- networks. *Proceedings of the 23rd international conference on Machine learning*, 369-376.
- Graves, A., Fernández, S., & Schmidhuber, J. (2005, September). Bidirectional LSTM networks for improved phoneme classification and recognition. *International conference on artificial neural networks*, Springer, 799-804.
10.1007/11550907_163
- Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. *PMLR - International conference on machine learning*, 32(2), 1764-1772. <https://proceedings.mlr.press/v32/graves14.html>
- Graves, A., & Jaitly, N. (2014, June). Towards end-to-end speech recognition with recurrent neural networks. *International conference on machine learning*, PMLR 32(2), 1764-1772.
- Graves, A., Jaitly, N., & Mohamed, A. (2013). Hybrid speech recognition with deep bidirectional LSTM. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 273-278. 10.1109/ASRU.2013.6707742
- Graves, A., Mohamed, A., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645-6649. 10.1109/ICASSP.2013.6638947
- Han, K. J., Chandrashekar, A., Kim, J., & Lane, I. (2017). The CAPIO 2017 conversational speech recognition system. *arXiv, preprint arXiv:1801.00059*.
10.48550/arXiv.1801.00059
- Hannun, A. (2017). Sequence modeling with ctc. *Distill*, 2(11). distill.pub:
<https://distill.pub/2017/ctc/>
- Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006, November). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527-1554.
10.1162/neco.2006.18.7.1527
- Hochreiter, S., & Schmidhuber, J. (1997, November). Long short-term memory. *Neural computation*, 9(8), 1735-1780. 10.1162/neco.1997.9.8.1735
- Huang, X. D., Ariki, Y., & Jack, M. A. (1990). *Hidden Markov Models for Speech Recognition*. Edinburgh University Press.
- Jin, K. (2017, September). Language Model: A Survey of the State-of-the-Art Technology. *Synced*.

- <https://medium.com/syncedreview/language-model-a-survey-of-the-state-of-the-art-technology-64d1a2e5a466>
- Karpathy, A. (2015, August 27). *Understanding LSTM Networks*. colah's blog. Retrieved August 6, 2023, from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Kneser, R., & Ney, H. (1995). Improved backing-off for M-gram language modeling. *IEEE, International Conference on Acoustics, Speech, and Signal Processing, 1*, 181-184. 10.1109/ICASSP.1995.479394
- Lee, K.-F. (1988, December). On large-vocabulary speaker-independent continuous speech recognition. *Speech communication, 7*(4), 375-379. 10.1016/0167-6393(88)90053-2
- Lin, H., Deng, L., Yu, D., Gong, Y., Acero, A., & Lee, C.-H. (2009, April). A study on multilingual acoustic modeling for large vocabulary ASR. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4333-4336. 10.1109/ICASSP.2009.4960588
- Liptchinsky, V., Synnaeve, G., & Collobert, R. (2017). Letter-based speech recognition with gated convnets. *arXiv, preprint arXiv:1712.09444*.
- Lu, L., Kong, L., Dyer, C., Smith, N. A., & Renals, S. (2016). Segmental recurrent neural networks for end-to-end speech recognition. *arXiv, preprint arXiv:1603.00223*.
- Lyons, J. (2012). *Mel Frequency Cepstral Coefficient (MFCC) Tutorial*. Practical Cryptography. Retrieved July 31, 2023, from <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>
- Maas, A. (2018). *Spoken Language Processing*. CS224S. Retrieved 10 8, 2018, from <https://web.stanford.edu/class/cs224s/lectures/224s.17.lec7.pdf>
- Miao, Y., Gowayyed, M., & Metze, F. (2015, December). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, 167-174. 10.1109/ASRU.2015.7404790
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010, September). Recurrent neural network based language model. *Interspeech, 2*(3), 1045-1048.
- Mitra, V., Sivaraman, G., Nam, H., Espy-Wilson, C., Saltzman, E., & Tiede, M. (2017, May). Hybrid convolutional neural networks for articulatory and acoustic information

- based speech recognition. *Speech Communication*, 89, 103-112.
10.1016/j.specom.2017.03.003
- Mohamed, A., Dahl, G., & Hinton, G. (2009). Deep belief networks for phone recognition. *Nips workshop on deep learning for speech recognition and related applications*, 1(9), 39.
- Mohamed, A., Dahl, G. E., & Hinton, G. (2012, January). Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, 20(1), 14 - 22. 10.1109/TASL.2011.2109382
- Morgan, N., & Bourlard, H. (1995, May). Continuous speech recognition. *IEEE Signal Processing Magazine*, 12(3), 24 - 42. 10.1109/79.382443
- Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016, September). Wavenet: A generative model for raw audio. *arXiv, preprint arXiv:1609.03499*. 10.48550/arXiv.1609.03499
- Palaz, D., & Collobert, R. (2015). Analysis of CNN-based speech recognition system using raw speech as input. *Idiap, EPFL-REPORT-210039*.
- Palaz, D., Doss, M. M., & Collobert, R. (2015). Convolutional neural networks-based continuous speech recognition using raw speech signal. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4295-4299. 10.1109/ICASSP.2015.7178781
- Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015, April). Librispeech: an asr corpus based on public domain audio books. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 10.1109/ICASSP.2015.7178964
- Peddinti, V., Povey, D., & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. *Sixteenth annual conference of the international speech communication association*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825--2830. <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>
- Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., & Khudanpur, S. (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. *Interspeech*, 2751-2755.

- Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1), 4 - 16. 10.1109/MASSP.1986.1165342
- Rousseau, A., Deléglise, P., & Esteve, Y. (2012, May). TED-LIUM: an Automatic Speech Recognition dedicated corpus. *LREC*, 125-129.
- Sainath, T. N. (2013). Improvements to Deep Convolutional Neural Networks for LVCSR. *IEEE Workshop on Automatic Speech Recognition and Understanding*, 1(1), 315-320. 10.1109/ASRU.2013.6707749
- Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015, April). Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 10.1109/ICASSP.2015.7178838
- Sak, H., Senior, A., Rao, K., & Beaufays, F. (2015, July). Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv, preprint arXiv:1507.06947*. 10.48550/arXiv.1507.06947
- Sak, H., Senior, A. W., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling.
- Salazar, G. A., Haynes, D. S., & Sommers, M. J. (1998, June). Real-time reconfigurable adaptive speech recognition command and control apparatus and method. *U.S. Patent 5,774,841*.
- Sameti, H., Veisi, H., Bahrani, M., Babaali, B., & Hosseinzadeh, K. (2009). Nevisa, a persian continuous speech recognition system. *Advances in Computer Science and Engineering: 13th International CSI Computer Conference, CSICC 2008 Kish Island, Iran, March 9-11, 2008 Revised Selected Papers, Springer Berlin Heidelberg*, 485-492.
- Sameti, H., Veisi, H., Bahrani, M., Babaali, B., & Hosseinzadeh, K. (2011). A large vocabulary continuous speech recognition system for Persian language. *EURASIP Journal on Audio, Speech, and Music Processing*, 1-12.
- Shrawankar, U., & Thakare, V. M. (2013, May). Techniques for feature extraction in speech recognition system: A comparative study. *arXiv, preprint arXiv:1305.1145*. 10.48550/arXiv.1305.1145
- Song, W., & Cai, J. (2015). End-to-end deep neural network for automatic speech recognition. *Stanford CS224D Reports*, 1-8.

- Stanley F., C., & Goodman, J. (1999, October). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4), 359-394. 10.1006/csla.1999.0128
- Thoma, M. (2013, November 15). *Word Error Rate Calculation · Martin Thoma*. Martin Thoma. Retrieved August 7, 2023, from <https://martin-thoma.com/word-error-rate-calculation/>
- Tóth, L. (2014, July). Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 190-194. 10.1109/ICASSP.2014.6853584
- Tóth, L. (2015). Phone recognition with hierarchical convolutional deep maxout networks. *EURASIP Journal on Audio, Speech, and Music Processing*, 1-13.
- Unnikrishnan, K. P., Hopfield, J. J., & Tank, D. W. (1991). Connected-digit speaker-dependent speech recognition using a neural network. *IEEE Transactions on Signal Processing*, 39(3), 698-713.
- Vaněk, J., Zelinka, J., Soutner, D., & Psutka, J. (2017, September). A regularization post layer: An additional way how to make deep neural networks robust. *Statistical Language and Speech Processing: 5th International Conference, SLSP 2017*, 204-214.
- Wikipedia. (2003). *Quantization (signal processing)*. Wikipedia. Retrieved July 31, 2023, from [https://en.wikipedia.org/wiki/Quantization_\(signal_processing\)](https://en.wikipedia.org/wiki/Quantization_(signal_processing))
- Wikipedia. (2003). *Sampling (signal processing)*. Wikipedia. Retrieved July 31, 2023, from [https://en.wikipedia.org/wiki/Sampling_\(signal_processing\)](https://en.wikipedia.org/wiki/Sampling_(signal_processing))
- Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., & Zweig, G. (2016). The Microsoft 2016 conversational speech recognition system. *arXiv, preprint arXiv:1609.03528*.
- Yao, K., & Zweig, G. (2015). Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv, preprint arXiv:1506.00196*. 10.48550/arXiv.1506.00196
- Yu, D., & Deng, L. (2014). *Automatic Speech Recognition: A Deep Learning Approach*. Springer London. 10.1007/978-1-4471-5779-3
- Zeghidour, N., Usunier, N., Kokkinos, I., Schaiz, T., Synnaeve, G., & Dupoux, E. (2018, April). Learning filterbanks from raw speech for phone recognition. *IEEE*

- international conference on acoustics, speech and signal Processing (ICASSP)*, 5509-5513. 10.1109/ICASSP.2018.8462015
- Zeinali, H., Sameti, H., & Burget, L. (2017). HMM-based phrase-independent i-vector extractor for text-dependent speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(7), 1421-1435. 10.1109/TASLP.2017.2694708
- Zeyer, A., Doetsch, P., Voigtlaender, P., Schlüter, R., & Ney, H. (2017). A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2462-2466. 10.1109/ICASSP.2017.7952599
- Zeyer, A., Irie, K., Schlüter, R., & Ney, H. (2018). Improved training of end-to-end attention models for speech recognition. *arXiv, preprint arXiv:1805.03294*. 10.48550/arXiv.1805.03294
- Zhang, S. X., Chen, Z., Zhao, Y., Li, J., & Gong, Y. (2016, December). End-to-end attention based text-dependent speaker verification. *2016 IEEE Spoken Language Technology Workshop (SLT)*, 171-178. 10.1109/SLT.2016.7846261
- Zhang, Y., Chan, W., & Jaitly, N. (2017, March). Very deep convolutional networks for end-to-end speech recognition. *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4845-4849. 10.1109/ICASSP.2017.7953077
- Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, Y., & Courville, A. (2017). Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv, preprint arXiv:1701.02720*. 10.48550/arXiv.1701.02720