# Low-resource Sinhala Speech Recognition using Deep Learning

N.A.K.H.S.Karunathilaka

2015/CS/069

This dissertation is submitted to the University of Colombo School of Computing
In partial fulfillment of the requirements for the
Degree of Bachelor of Science Honours in Computer Science

University of Colombo School of Computing
35, Reid Avenue, Colombo 07,
Sri Lanka

July, 2020

# Declaration

I, N.A.K.H.S.Karunathilaka (2015/CS/069) hereby certify that this dissertation entitled "Low-resource Sinhala Speech Recognition using Deep Learning" is entirely my own work and it has never been submitted nor is currently been submitted for any other degree.

..... .....

Date Signature of the Student

I, Mr V. Welgama, certify that I supervised this dissertation entitled "Low-resource Sinhala Speech Recognition using Deep Learning" conducted by N.A.K.H.S. Karunathilaka in partial fulfillment of the requirements for the degree of Bachelor of Science Honours in Computer Science.

..... .....

Date Signature of the Supervisor

# Abstract

Automatic speech recognition has progressed considerably in the past several decades for most of the European languages, but still it is a prominent research area for most of the low-resourced languages. This project presents a study to build an automatic speech recognition (ASR) system using the Kaldi toolkit for the Sinhala language which is one of the low-resourced languages with a large lexical variety. We experiment on different deep neural architectures like pre-trained DNN, DNN, TDNN, TDNN+LSTM to enhance the acoustic modeling process and each of their performances is investigated in this document. A statistical model of GMM-HMM is also trained on the same data set keeping it as the baseline model for comparing the effectiveness of deep learning approach. For the language model, a corpus containing more than 20K sentences taken from UCSC LTRL is used to generate the 220K extended lexicon. The experiments are conducted using a phonetically balanced training data set consisting of twenty-five hours of speech data collected from fifty females and twenty males and tested on 1.6 hours of speech data. We present an overview of different architectures with their procedures and compare and contrast the performances of models with the statistical baseline approach. The results obtained show that Deep neural network architecture exceeds the statistical baseline model with a Word Error Rate (WER) of 7.48% on the test data set. The best observed lowest WERs are produced by the TDNN architectures.

**Keywords: Sinhala, Speech recognition, Deep neural network, Kaldi**

# Preface

This document has been written for the partial fulfillment of the requirements of the B.Sc. in Computer Science (Hons) Final Year Project in Computer Science(SCS4124). I was engaged in researching and writing this dissertation from January 2019 to February 2020.

This basis for the research originally stemmed from my passion for contributing to the researches relevant to my native language - Sinhala. In truth, it helps to contribute to the preservation of the language. After discussing with my supervisor, Mr.V.Welgama, we identified the problems that have emerged due to the unavailability of automatic speech recognition for the Sinhala language. As the world moves further into the digital age, the importance of automatic speech recognition rises, since it is able to develop natural interfaces for both literate and illiterate users by enabling hands-free technology and even aiding hearing-impaired people.

To the best of my knowledge, research work on ASR for the Sinhala language using deep learning approaches has not been carried out so far. First, we identified the main factors that make the ASR task challenging. They were the lack of resources and the morphological richness in the Sinhala language. As described in Chapter 3, we selected the DNN-HMM approach and designed the suitable model architecture in such a way that it gets the best out of resources. We first pre-processed the data set and trained the statistical baseline model. Thereafter I experimented on different deep neural networks one by one and based on the observations, I modified and tuned the architectures to obtain optimal performance. When analyzing the results it was observed that a rich text corpus along with more training data is necessary requirements for a robust ASR system.

With constant guidance and supervision of my supervisor and co-supervisor, more conclusions were drawn on evaluating and training the models. This piece of research would be a great source of knowledge for future research on Sinhala ASR systems.

# Acknowledgement

First and foremost, I would like to thank my university, (UCSC) University of Colombo School of Computing for giving this great opportunity to conduct a self research in which I have developed myself both academically as well as self confidently. I would like to express my sincere gratitude to Mr. W.V Welgama, my supervisor and Dr. H.N.D. Thilini,Dr. Ruvan Weerasinghe, my co supervisors at UCSC. They all gave me very in-time valuable instructions and gave an extensive guidance to make the research a success from the beginning to end. As they gave me in-time feedback after reviewing my work, I was able to complete the work successfully and ontime. I am really grateful for Dr. H.N.D. Thilini for her contribution in stimulating suggestions and encouragements. I also wish to express my gratitude to my family and friends for giving me helping hands throughout the year.

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| ASR | Automatic Speech Recognition |
| BLSTM | Bidirectional Long Short Term Memory |
| CNN | Convolutional Neural Network |
| CNTK | Microsoft Cognitive Toolkit |
| CD-DNN-HMM | Context Dependent-Deep Neural Network-Hidden Markov Model |
| DNN | Deep Neural Network |
| DNN-HMM | Deep Neural Network-Hidden Markov Model |
| DNN-GMM-HMM | Deep Neural Network-Gaussian Mixture Model-Hidden Markov Model |
| DBN | Deep Belief Network |
| GMM-HMM | Gaussian Mixture Model-Hidden Markov Model |
| LM | Language Model |
| LSTM | Long short-term memory |
| LTRL | Language Technology Research Laboratory |
| LVCSR | Large Vocabulary Continuous Speech Recognition |
| LDA+MLLT | Linear Discriminant Analysis + Maximum Likelihood Linear Transform |
| MFCC | Mel frequency cepstral coefficients |
| MLP | Multilayer perceptron |
| pGLSTM | prioritized Grid LSTM |
| PLSTM | Parallel Long Short-Term Memory |
| RNN | Recurrent Neural Network |
| RCNN | Region-based Convolutional Neural Networks |
| SER | Sentence Error Rate |
| TDNN | Time delay neural network |
| UCSC | University of Colombo School of Computing |
| WER | Word Error Rate |

# Chapter 1

# Introduction

## 1.1 Background to the Research

Over the past few decades, there has been a curiosity in making computers perform what only humans could perceive. The tremendous development in machine learning paradigms has helped to achieve unbelievable success in recognizing speech, understanding natural language, processing images, etc. Most of the researchers are interested in adopting speech recognition because it can be used effectively to build interactions between humans and robots. Furthermore, this helps in generating massive amounts of information that will eventually lead to a collection of a huge amount of ideas, memories, and unstructured data. Speech has been the most efficient and convenient way of communicating, a whole raft of applications including commercial products like Google Assistant, Siri from Apple and Alexa from Amazon are rapidly using this technology -Automatic Speech Recognition(ASR) (J.Arora and Singh, 2012),(Jurafsky and Martin, 2008). These virtual assistants are capable of voice interaction, music playback providing weather, traffic, sports, and much other real-time information retrievals, and they keep competing.

A typical ASR system's goal is to transform the acoustic input into a sequence of words. The internal process of an ASR consists of several steps: feature extraction which is transforming the audio signal into a series of vectors of acoustic features; acoustic modeling which converts speech to relevant phonemes and lastly language modeling which defines what kind of phoneme and word sequences are possible in the target language, and their probabilities.

Statistical approaches like GMM-HMM has been the state-of-art of speech recognition in the early days. Recently, Deep Neural Networks (DNN) has become a flagship and has proven it by improving the achieved results significantly (Hinton et al., 2012). Moreover, DNN has the capability of generalization and the ability to discover and learn complex structures (Deng et al., 2013),(Hinton et al., 2012). The use of deep learning (Du et al., 2016) neural architectures such as Deep Neural Networks(DNN), Convolutional Neural Networks(CNN), and Recurrent Neural Networks(RNN) for the speech recognition of English and European languages have shown significant improvement with compared to conventional GMM-HMM approach (Fohr et al., 2017),(Markovnikov et al., 2018).

## 1.2 Research Problem and Research Questions

Sinhala is one of the official and national languages which is used by a majority of Sri Lankans. According to the recent analytic ($wiki_Sinhala,\ n.d.$), the Sinhala language speaking population in SriLanka is 87%, making it over 16.6 million user base. Therefore, as Sri Lankans, there is a need to pay attention to the research area of recognizing Sinhala speeches as it will either directly or indirectly affect the beneficial.

Unavailability of an open-source speech recognizer that can transcribe Sinhala speeches accurately precludes the achievable performance of many local systems compared to other global contemporary systems of well-versed languages such as multilingual call centers and voice command-and-control systems. Besides, due to the geographical and technological shortcomings such as the low percentage of the global population that speaks and understands Sinhala and less contribution towards Sinhala, speech recognition development has also precluded the services from most commercial off-the-shelf (COTS) and open speech recognition software. Therefore, many organizations, including both local and global establishments, can make use of a Sinhala Automatic Speech Recognition (ASR) tool to optimize existing processes. One of the challenging tasks in Sinhala speech recognition is finding a large speech corpus since the model should have the capability to estimate the probability for all possible word sequences. Moreover, Sinhala been

a morphologically rich language, which makes it difficult to further as they may produce a vast number of word forms for a given root form.

The main research question that is addressed in the research is as follows.

- What deep neural architectures will perform well for Sinhala ASR with limited resources?

  There is a vast range of deep neural architectures extended for tasks related to speech recognition, voice detection, etc. But the question arises which structures would provide the most accurate transcriptions to achieve our goal?. When addressing this question, two more essential facts need to be considered; the limited data set and the complexity of the Sinhala language. With time constraints, it is often hard to collect a sufficient amount of resources. Hence, by addressing this research question, this work would be able to identify the specifications that a deep neural network model should possess to achieve the best possible accurate results with limited resources.

## 1.3 Justification for the research

Most of the research work (Amarasingha and Gamini, 2012),(Manamperi et al., 2018),(Nadungodage and Weerasinghe, 2011),(Gunasekara and Meegama, 2015) carried out so far for Sinhala speech recognition has followed only traditional approaches (i.e. GMM-HMM). However, the researches that have been conducted for other high-resource and low-resource language ASR, it is notified that DNNs have performed in a higher classification and generalization when compared with the statistical-based approach. Especially, this can be observed in high-resource languages. Thus, an exploration regarding the validity of deep learning techniques on the task of Sinhala ASR is an unexplored direction.

Moreover, the necessity of an open domain continuous speech recognizer for Sinhala is observable since most of the research work found is domain-specific (Amarasingha and Gamini, 2012),(Manamperi et al., 2018). But, the practicability of an ASR system highly depends on the number of domains it can be applied. Further, almost all the research work that has been conducted so far has used only

a minimal data set for training and testing. Thus the validity of them is again questionable.

There exist numerous deep architectures that could be applied for the task of speech recognition as the possibilities are almost endless. This project intends to implement a sufficient number of deep architectures for acoustic modeling in the intended ASR system. Thus, for those who are interested in this field can get an overview of the performances of different deep architectures and which architecture or the approach is the best suited for an under-resourced, morphological rich language like Sinhala. The project also intends to compare the performances of deep neural architectures with the existing standard statistical methods. Therefore, other researchers can come up with a more advanced ASR system for the Sinhala language.

## 1.4 Methodology

There are several approaches for developing an ASR system such as GMM-HMM, DNN-GMM-HMM, DNN-HMM (Pallavi Saikia and Open Learning, 2017), End-to-end DNN (Zhang et al., 2017). Among them two significant approaches are DNN-HMM: a hybrid architecture which uses deep neural networks for acoustic modeling along with a Hidden Markov Model(HMM); End-to-end DNN : which takes acoustic features as input and outputs its transcriptions directly, thus relying solely on a deep learning architecture.

The End-to-End (E2E ASR) is a single integrated approach with a much simpler training pipeline, and it reduces the training time and decoding time. However, current E2E ASR systems also suffer from limitations such as these systems need orders of magnitude more training data than hybrid ASR systems to achieve similar word error rate (WER). These limitations arise when the training data is limited; there is a propensity to overfit the training data. Thus, it becomes quite expensive as it to get a higher performance, a large number of speech data should be fed and also needs higher computing power. The DNN-HMM hybrid systems comprise an acoustic model, a language model, and a pronunciation model. For limited computation power and training data, this hybrid architecture can achieve better

results (Pallavi Saikia and Open Learning, 2017) (Fohr et al., 2017). Therefore, this project intends to follow the DNN-HMM hybrid approach (Pallavi Saikia and Open Learning, 2017), as it enriches with the strong learning power of DNNs and the sequential modeling of HMMs.

### 1.4.1 Speech Corpus

A speech corpora collected by Language Technology Research Laboratory(LTRL) of the University Of Colombo School Of Computing(UCSC) that has recordings from 50 females and 20 males, which would be roughly estimated to about 25 hours of speech will be used as training data.

### 1.4.2 Pre-processing Raw Data

The raw speech signals are first pre-processed into a vector of numeric values, which is also known as feature extraction. For this, the raw speech signal is divided into small portions of typically 25 ms frames shifted by 10ms each time. A transformation is then applied as the human hearing perceptron is not linear with frequency scales. This process can be performed with Mel-Frequency Cepstral Coefficients (MFCCs) or filter banks.

### 1.4.3 Deep Neural Networks(DNNs)

DNNs work similar to the neurons in our brain. They consist of highly interconnected units known as neurons and forms a data processing element. Mostly, this project intends to focus on deep neural networks that support supervised learning. Experiments with Deep Neural Networks(DNNs), Time-Delay Neural Networks (TDNN/CNN1-d), Long Short Term Memory(LSTM) Networks have been carried out for acoustic modeling. These models take a window of frames that includes real-valued acoustic features as inputs and estimate the likelihoods of phones. The output layer from these deep neural networks is then integrated with the Hidden Markov Model(HMM). With the observation probabilistic scores gained by the neural network, HMM maps them to a sequence of phones. The DNN models are trained and fine-tuned empirically by analyzing the speech accuracy of the valida-

tion data set. The analysis is performed quantitatively by calculating the Word Error Rate(WER) of the models.

### 1.4.4 Language Modeling

A language model consists of a large amount of text data, and it aims to compute the probability of the sequence of words to find the best word sequence of the acoustic model. In this research, the most well-known n-gram model technique is used to model the language. A corpus containing more than 20K sentences collected from phonetically balanced corpora were used to generate the 220K long grapheme lexicon.

## 1.5 Outline of the Dissertation

The thesis is organized as follows.
In Chapter 2, a comprehensive study in describing technologies, performances, and special characteristics observed by authors in previous researches will be presented. The research design, together with the high-level architecture for addressing the research question, is presented in Chapter 3. Later in Chapter 4, a comprehensive explanation of the implementation is carried out. Last but not least, experiments and results of every model are evaluated in Chapter 5. Finally, Conclusion and future work are discussed in Chapter 6.

## 1.6 Definitions

- GMM-HMM model is defined with several names such as ”baseline,” ”statistical,” and ”traditional” in this document.
- Pre-trained DNN is defined as a model that is subjected to a training process with a stack of Restricted Boltzmann machines at first, and later a DNN is trained on those weights.
- Non-pre-trained DNN is defined as a DNN model that is trained on randomly initialized weights.

## 1.7 Delimitations of Scope

The below listed are boundaries that are faced when addressing the research question,

- A limited number of appropriate deep neural network architectures will be experimented.
- Achieving 100% performance with the limited data set will not be guaranteed since deep neural networks inherently perform better with large data sets.

  Although a moderate vocabulary of phonetically balanced Sinhala corpus is used for training the models, it will not be sufficient to model all the occurrences in Sinhala speech.

## 1.8 Conclusion

This chapter laid the foundations for the dissertation. It introduced the general focus area and the more specific research problem and research question that are addressed in this research. Then the research was justified analyzing the significance of the study, the important factors in methodology was briefly described and justified, the dissertation was outlined, and the limitations were given. On these foundations, the dissertation can proceed with a detailed description of the research.

# Chapter 2

# Literature Review

## 2.1 Review on Sound characteristics and feature extraction

A speech signal is a physical representation of any waveform whose frequencies range is in the human audible range and contains information.
Extracting the information to achieve intended objectives by transforming the raw signals into a more informative format is known as the process of feature extraction. It is also a measure of competing for a compact numerical representation that can be used to characterize a segment of audio. The audio signals contain features like Mel Frequency Cepstral Coefficients (MFCC), Pitch, sampling frequency, loudness, volume, etc. Among them, the most widely used technique is the extraction of MFCCs.

## 2.2 Review on different approaches to ASR

Since from earlier times, specific approaches for pattern matching in speech recognition have been experimented. From them, template-based,knowledge-based, dynamic time warping-based, statistical-based, and neural network-based are some prominent approaches. Most of the above have been declined at present due to the limitations they suffer.
The template-based approach is the process of matching unknown speech against

a set of pre-recorded words or templates in order to find the best match (Saksamudre et al., 2015). Inefficiency in terms of both storage and computation power and tediously speaker dependence are the drawbacks of this method. The knowledge-based approach uses the information regarding phonetic, linguistic, and spectrogram (Saksamudre et al., 2015), but it suffers from the limitation that it requires expert knowledge on the language. The dynamic warping-based (DTW) approach measures the similarity between two sequences which may vary in time or speed, and this algorithm effectively works to cope with different vocalization speeds(Saksamudre et al., 2015). Generally, DTW works well for only isolated word recognition, which is a limitation. The most known approach, which has been state-of-the-art for speech recognition for several years, is the statistical-based approach. There, the most extended way is the traditional GMM-HMM hybrid system.

## 2.3 Review on Deep Neural Network algorithms for acoustic modeling

Recently, neural network-based (NN) acoustic models have significantly improved ASR performance over traditional Gaussian Mixture Models (GMMs). Different types of neural network architectures have been used for training the acoustic model of ASR systems in several attempts in speech recognition.

The following literature reviews attempt to demonstrate the usage of DNN on speech recognition and clarify the performances of the best models obtained from different practices.

In the research paper by Alexey Karpov (2017)(Markovnikov et al., 2018), they have described research of DNN-based acoustic modeling for Russian speech recognition using Kaldi toolkit (Povey et al., 2011). The DNN models are created with p-norm and tanh nonlinearities with a different number of hidden layers and units and are compared with the baseline GMM-HMM models. These hybrid models are trained using fMLLR-adopted features, the decision trees, and alignments obtained from the SAT-fMLLR GMM system. The best result (20.30% WER) is achieved with the p-norm DNN, which is obtained by a six-layer DNN with 1024 hidden units in each hidden layer. The author has observed after analyzing the results,

that the number of layers has only a slight influence on the speech recognition results. In interest, the results show that increasing the number of hidden units leads to increasing WER, which may be caused due to the limited amount of training data. The results showed that DNN based acoustic models have well performed compared with the baseline GMM-HMM models with a reduction of 5% of WER.

An overview of different architectures and training procedures for DNN-based acoustic models for the task of recognizing French speech is presented in a research paper done by Dominique Fohr et al. (2017)(Fohr et al., 2017). The network is implemented as an MLP with six hidden layers of 2048 neurons per layer. The experimented dataset consists of 300 hours of manually transcribed shows from French-speaking radio stations from which 250h recorded in a studio and 50h on the telephone. As in figure 2.1, the results showed a significant difference in performance between the baseline GMM-HMM model and the DNN hybrid model, which suggests that DNN-based acoustic models achieve better classification and generalization ability.

| Shows | # words | GMM-HMM | DNN-HMM |
|---|---|---|---|
| 20070707_rfi (France) | 5473 | 23.6 | 16.5 |
| 20070710_rfi (France) | 3020 | 22.7 | 17.4 |
| 20070710_france_inter | 3891 | 16.7 | 12.1 |
| 20070711_france_inter | 3745 | 19.3 | 14.4 |
| 20070712_france_inter | 3749 | 23.6 | 16.6 |
| 20070715_tvme (Morocco) | 2663 | 32.5 | 26.5 |
| 20070716_france_inter | 3757 | 20.7 | 17.0 |
| 20070716_tvme (Morocco) | 2453 | 22.8 | 17.0 |
| 20070717_tvme (Morocco) | 2646 | 25.1 | 20.1 |
| 20070718_tvme (Morocco) | 2466 | 20.2 | 15.8 |
| 20070723_france_inter | 8045 | 22.4 | 17.4 |
| Average | 41908 | 22.4 | **17.1** |

Figure 2.1: WER(%) for the 11 shows from GMM-HMM and DNN-HMM KATS systems (Fohr et al., 2017)

An in-depth overview of the usage of deep neural networks to achieve a successful model for speech recognition is presented by four research groups in the research article by Hinton et al. (2013)(Hinton et al., 2012). All the four research groups have managed to achieve the best accuracy results using a hybrid deep belief net (DBN)/HMM – DNNs experimented on TIMIT dataset(*TIMIT Dataset*, n.d.), which is a corpus of phonemically and lexically transcribed speech of Amer-

ican English speakers. The results showed that pre-training is much more helpful in deep neural nets than in shallow ones, mainly when limited amounts of labeled training data are available. Also, It has resulted in reducing over-fitting, and the time required for discriminative fine-tuning with back-propagation.

They have experimented with the AMUAV database(Samudravijaya et al., 2000), and the results demonstrate that CD-DNN-HMMs outperform the conventional CD-GMM-HMMs model and provide the improvement in a word error rate of 3.1% over the traditional model of triphone. Some recent studies(2018)(Kimanuka and BUYUK, 2018),(Deka et al., 2018),(Saurav et al., 2018) investigated on low resourced languages such as Turkish, Assamese, Bengali speech recognition, shows that the use of deep neural networks for acoustic modeling has resulted in lower WER compared with baseline GMM-HMM models.

The following literature reviews attempt to demonstrate the usage of different advance neural network architectures for speech recognition and clarify the best results obtained from them.

Recently, many pieces of research have been carried out investigating Recurrent Neural Networks(RNNs) for the task of speech recognition because of its ability to utilize dynamically changing temporal information. Although deep RNNs have been argued to be able to model temporal relationships at different time granularities, it suffers from vanishing gradient problems.

An exploration of novel approaches to constructing deep long short-term memory (LSTM) based deep recurrent neural networks are presented in the research work by Xiangang Li (2015)(Li and Wu, 2014). The evaluations of different LSTM networks have been done on a large vocabulary Mandarin Chinese conversational telephone speech recognition task. The results, as shown in figure 2.2, reveals that constructing deep LSTM architecture outperforms the standard shallow LSTM networks and DNNs.

| Model Descriptions | CER(%) |
|---|---|
| LSTM | 40.28 |
| LSTM-IP | 39.09 |
| LSTM-OP | 35.92 |
| 3-layer ReLU + LSTM | 37.31 |
| 2-layer Conv + 2-layer ReLU + LSTM | 36.66 |
| LSTM + 3-layer ReLU | 37.16 |
| Stack of LSTM (3-layer) | **35.91** |

Figure 2.2: Speech recognition results of different strategies of constructing deep LSTM networks. (Li and Wu, 2014)

In the research study by Wei-Ning Hsu et al. (2016)(Hsu et al., 2016), they have experimented deep grid LSTMs (Kalchbrenner et al., 2015) on four different speech corpora(AMI, HKUST, GALE Mandarin, Arabic MGB) using the Kaldi toolkit to generate Mel-scale log filter bank coefficients along with first and second derivatives and tri-gram language model. The Computational Network Toolkit (CNTK) has been used for the rest of neural network training. They have used the prioritized Grid LSTM (pGLSTM) model to prioritize the depth dimension over the temporal one to provide more updated information for the depth dimension. The performances of baseline models and proposed models are summarized in figure 2.3. The results obtained show that the different grid LSTM architectures have outperformed the vanilla LSTM model and suggest that prioritizing the depth dimension is essential for achieving better performance.

| Model | #layers | #params | AMI | HKUST | GALE | MGB |
|---|---|---|---|---|---|---|
| BHLSTMP [11] | 3 | 11M | **48.3** | - | - | - |
| Stacked maxout LSTMPs [31] | 3 | - | - | 33.89 | - | - |
| Highway CLDNN [12] | 11 | 44M | - | - | 22.41 | - |
| LSTM | 3 | 12M | 50.7 | 33.29 | 23.96 | 23.56 |
| HLSTM | 3 | 14M | 50.4 | 32.86 | 23.33 | 23.32 |
| HLSTM | 5 | 24M | 50.7 | 32.40 | 22.63 | 23.12 |
| pGLSTM | 3 | 25M | 49.8 | 32.06 | 22.54 | 22.36 |
| pGLSTM | 5 | 44M | 48.6 | **31.36** | **22.33** | **22.18** |

Figure 2.3: Results of deep grid LSTMs on four different speech corpora(AMI, HKUST, GALE Mandarin, Arabic MGB).(Hsu et al., 2016)

In the research paper by Nikita Markovnikov et al. (2018) (Markovnikov et al., 2018), demonstrates a cluster of hybrid speech recognition systems cooperating deep neural networks with Hidden Markov Models and Gaussian Mixture Models

for recognizing Russian speech. The acoustic models of their proposed work are implemented as CNNs, modifications of LSTM, Residual Networks, and Region-based Convolutional Neural Networks (RCNNs). They have experimented with the models on more than 30h of Russian speech. The authors have done a significant job, analyzing each of the models and their results, to come up with a best-suited model for Russian speech recognition. The results on the Kaldi baseline model using a multi-layer perceptron (MLP), as in figure 2.4, reveals that the best outcome is achieved when the activation function is set to p-norm.

| | 1 model | 2 model | 3 model | 4 model |
|---|---|---|---|---|
| Layers | 3 | 6 | 6 | 6 |
| Dimensions | $450 \times 3$ | $2048 \times 6$ | $512 \times 6$ | $512 \times 6$ |
| Activation function | sigmoid | sigmoid | tanh | $p$-norm ($p = 2$) |
| Iterations | 20 | 20 | 18 | 18 |
| WER | 25.54% | 25.32% | 24.96% | 24.26% |

Figure 2.4: The results on the Kaldi baseline model using a multi-layer perceptron (MLP).(Markovnikov et al., 2018)

They have compared the results obtained from their LSTM model with a Bidirectional Long Term Short Memory (BLSTM) model, which has used nnet3 Kaldi's configurations. The performance of BLSTM over LSTM can be viewed in figure 2.5. According to the results, except CNN model (24.96%), other models which are LSTM (23.32%), BLSTM (23.08%),PLSTM(24.12%),ResNet(22.17%) and RCNN(22.56%) have surpassed the baseline model (24.26%). A new model is implemented after analyzing the above best models, including RCNN + residual unit + max-pooling + BLSTM, which has obtained the lowest WER of 22.07% and with a reduction of 7.5% WER compared to Kaldi baseline.

| Model | WER |
|---|---|
| Kaldi baseline | 26.62% |
| Kaldi + DBN baseline | 23.96% |
| Kaldi nnet3 | 22.80% |
| MLP-3-sigmoid | 25.54% |
| MLP-6-sigmoid | 25.32% |
| MLP-6-tanh | 24.96% |
| MLP-6-p-norm | 24.26% |
| LSTM | 23.32% |
| PLSTM | 24.12% |
| BLSTM | 23.08% |
| CNN | 24.92% |
| RCNN | 22.56% |
| ResNet | **22.17%** |
| RCNN + CL + BLSTM | 22.34% |
| RCNN + max-pooling + BLSTM | 22.28% |
| RCNN + residual unit + max-pooling + BLSTM | **22.07%** |

Figure 2.5: The performance of BLSTM over LSTM and other models.(Markovnikov et al., 2018)

The experimental results have revealed that, although ResNet shows the best results, it has been the slowest out of all.

A combination of CNN-BLSTM architectures for acoustic modeling is presented in (2018)(Markovnikov et al., 2018) for the task of Microsoft's conversational speech recognition system for the Switchboard and CallHome domains. They have applied 3 CNN on the acoustic features at a time t and then applied 6 BLSTM layers to the resulting time sequence. Unlike in an original BLSTM model, they have included the context of each time point as an input feature in the model.

The introduction of Time Delay Neural Networks (TDNNs) was presented in the research work by Daniel Povey et al. (2015)(Peddinti et al., 2015), where the researchers achieved significant improvements in the field of speech recognition. Their proposed TDNN architecture, models long term temporal dependencies with training times comparable to standard feed-forward DNNs. During the training phase, researchers have used the sub-sampling technique to reduce computation power. The work presents results on several LVCSR tasks to show the effectiveness of the TDNN architecture in learning broader temporal dependencies in both small and large data scenarios. The results obtained on the Switchboard task show a relative improvement of 6% WER over the baseline DNN model, which is significant. The performance of TDNNs over different LVCSR tasks is indicated in figure 2.6.

| Database | Size | WER | | Rel. |
|---|---|---|---|---|
| | | DNN | TDNN | Change |
| Res. Management | 3h hrs | 2.27 | 2.30 | -1.3 |
| Wall Street Journal | 80 hrs | 6.57 | 6.22 | 5.3 |
| Tedlium | 118 hrs | 19.3 | 17.9 | 7.2 |
| Switchboard | 300 hrs | 15.5 | 14.0 | 9.6 |
| Librispeech | 960 hrs | 5.19 | 4.83 | 6.9 |
| Fisher English | 1800 hrs | 22.24 | 21.03 | 5.4 |

Figure 2.6: Baseline vs TDNN on various LVCSR tasks with different amount of training data. (Peddinti et al., 2015)

From figure 2.6, it is shown that the Resource Management medium-vocabulary task has not gained any performance using TDNNs due to its limited amount of training data. As the authors suggest, this could be due to the slight increase in parameters in the TDNN architecture when processing broader input contexts. Nevertheless, they have gained an average relative improvement of 5.52% over the baseline DNN architecture through the use of TDNN architecture to process broader contexts.

In a comparative study between different neural network architectures for speech recognition by Mohamed Maher Zenhom et al. (2018), the authors have shown that a combination of TDNN and LSTM architectures exceed by a large margin, the performance of deep and convolutional neural networks. The work presents a comprehensive study on building an automatic speech recognition (ASR) system using the KALDI toolkit for the Arabic language, which presents many challenges related to a large lexical variety of the language. They have used a grapheme based lexicon generated with more than 478K entries and a dataset consisting of 90 hours of Modern Standard Arabic (MSA) broadcast news. The best WER of 8.09% from the TDNN-LSTM model is obtained from 2048 neurons per layer for six hidden layers after performing six epochs, and 40 frames in Chunk left Context in TDNN-LSTM. Also, to get better frame accuracy and lower WER, the authors suggest using higher gram LM or RNN in building the language model instead of using the probabilistic model and also using deeper architectures in TDNN-BLSTM with more neurons in each layer and more samples per iterations if required computa-

tional resources meet.

In the research paper by Irina Kipyatkova in 2017(*Subasa*, n.d.), they have studied an application of time-delay neural networks (TDNNs) in acoustic modeling for large vocabulary continuous Russian speech recognition and compared it with baseline DNN model with p-norm activation functions implemented according to Dan's implementation in Kaldi. Training of acoustic models has been carried out on a Russian speech corpus containing phonetically balanced phrases with a duration of 30h. They have created several TDNNs with a diverse number of hidden layers, different temporal contexts, and splice indexes. The TDNN has achieved the lowest WER of 19.04% with five hidden layers and a time context of $[-8, 8]$. The usage of the models with larger temporal times has led to increasing in WER that also can be caused by over-training. The results show that the TDNN model has surpassed the results obtained by the baseline DNN with a relative WER reduction of 9%.

## 2.4 Summary

As a summary of the literature review presented in Chapter 2, the following points can be notified.

- Different neural network architectures for acoustic modeling has recorded in greater accuracy than the vanilla DNN architecture. Specifically, LSTM, TDNN, and combinations such as TDNN-LSTM have dramatically improved the model performances in most of the research works. However, computational resources should meet up to experiment with these types of deeper architectures.

- There is a major possibility of the higher number of hidden layers resulting in higher WER, which signifies that it is better to keep a moderate number of layers in between 4 -10 in NNs.

# Chapter 3

# Design

As stated before, speech recognition systems tend to characterize the acoustic information of a given audio signal and recognize its text version. Thus, the recognizer needs to segment the audio signal into successive frames where each frame outputs corresponding phone and then transcribe the recognized phones into a text. This research design involves developing a method that takes the utterances of speakers as audio signals and produces the texts corresponding to those.

Figure 3.1 demonstrates the high-level architecture of the proposed solution for the speech recognition of the Sinhala language. The design process of the research is described in detail below.

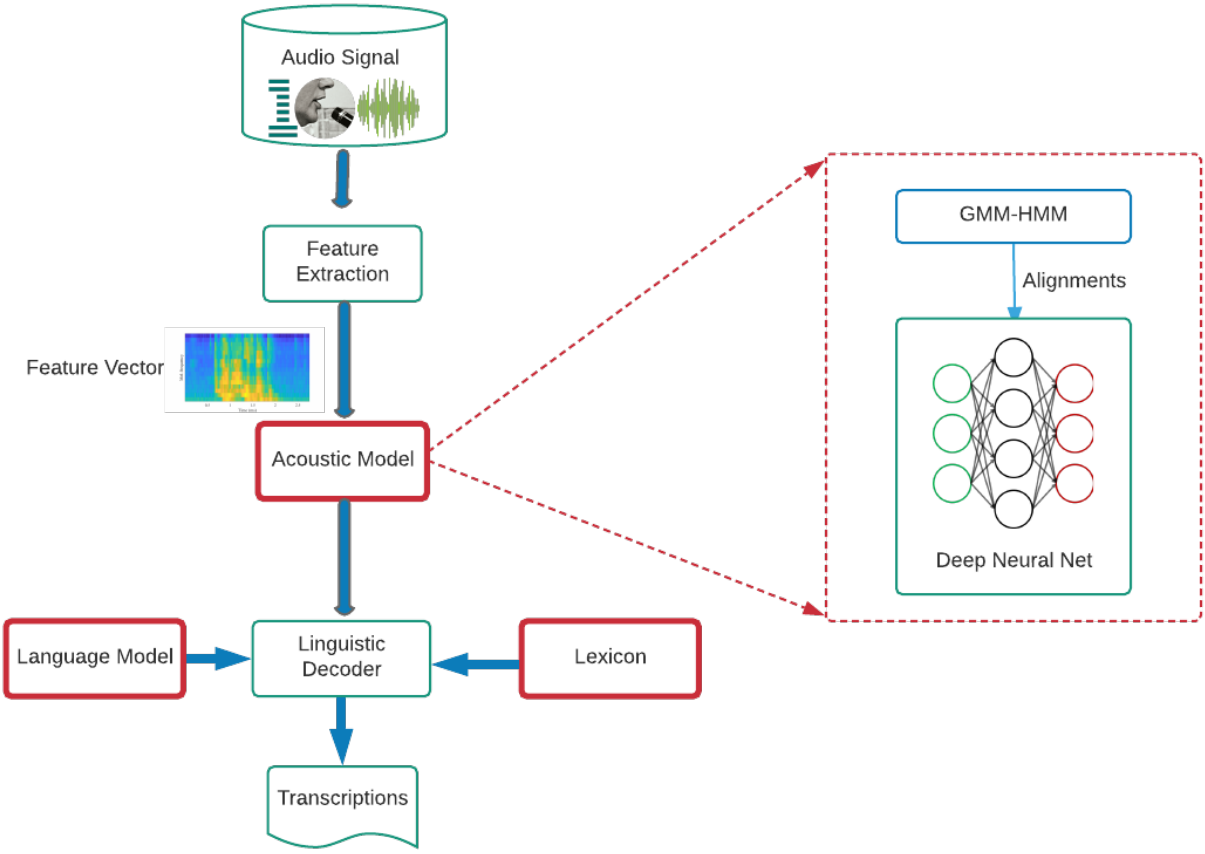


Figure 3.1: High-level architecture of the research design

The main design of the speech recognizer consists of

1. acoustic model
2. lexicon
3. language model

## 3.1 Acoustic Model (AM)

Considering the mathematical formulation of the task of speech recognition, let the speech signal in a time S(t) be represented by a set of acoustic observations $O = o_1, o_2, ..o_n$ and its corresponding word sentence $W = w_1, w_2, ...w_n$. Then $P(W|O)$ denotes the probability that the words $W$ were spoken, given the evidence $O$ observed from the sound waves. Using Bayer's rule, the most probable outcome can be computed by maximizing the following equation,

$$W^* = max_w \frac{P(O|W)P(W)}{P(O)} \tag{3.1}$$

The acoustic model is responsible for computing $P(O|W)$ the probability of an output sentence given a set of words.

The training data that is used in this research are audio files collected from 50 females and 20 males in the University of Colombo School of Computing LTRL, and it is approximately 25 hours of speech. This data is divided into training, validation, and test where the hours of speech will be 22,1.5, and 1.5, approximately. The splitting of data is done in a way that makes sure there are no overlapping speakers and sentences and gender distribution inequality in the train, validation, and test sets and thus ensuring the validity of the experiments.

The steps involved in training the acoustic model of the speech recognition system are described in the below subsections.

### 3.1.1 Feature Extraction

First, the raw audio files are pre-processed to obtain a vector of numeric values, which are often referred to as 'Mel frequency spectrograms,' which contains the acoustic information. The speech features are extracted by Mel Frequency Cepstral Coefficients(MFCC) features after normalizing by a standard 13- dimensional cepstral mean-variance. For that, the raw audio data are segmented into 25 ms of frames shifted by 10ms each time. Then for each frame, a windowing function is applied to extract the data. As the next step, a Fourier Transform is applied to convert the samples from the time domain to the frequency domain, which helps to compute the power spectrum.

### 3.1.2 Getting alignments from GMM-HMM model

To tackle the problem of limited data available for the Sinhala language, training of DNNs immediately from utterance level transcriptions is not performing since DNN requires a good initial approximation. Even with useful data, DNN training is tricky because it's not guaranteed to converge to an optimal point. Therefore,

as the first phase of the acoustic modeling, a GMM-HMM model is trained on the same data set to generate the alignments for the audio signals, as in the figure 3.2. Then, the DNN training is bootstrapped using the labeled frames (phoneme-to-audio alignments), which were generated by the GMM-HMM system. Thus, when the audio frames are fed into the input layer of the DNN model, the net will assign a phoneme label to a frame, and it will be compared with the phoneme label obtained from the GMM-HMM alignments.

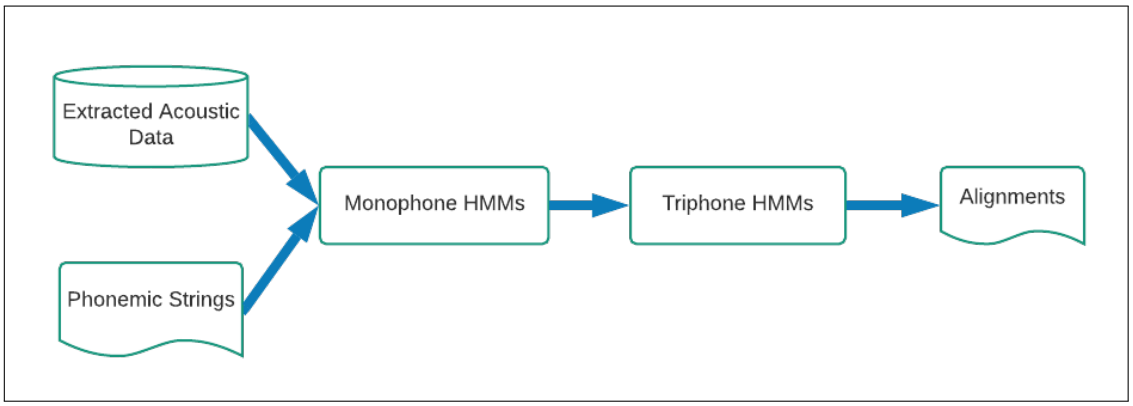


Figure 3.2: Acoustic modeling process in a GMM-HMM

The final alignments of the GMM-HMM model are taken after generating monophone HMMs and thereafter triphone HMMs.

- Monophone model training and alignment

  This model will also be the building block for the following triphone models. Monophone training depends on the contextual information of a single phone. Thus, monophone models do not give a good result, usually as phones sound different in different contexts.

- Triphone model training and alignment

  The phoneme variants in the context of the two phonemes, typically the preceding and following phones are considered in the training of the triphone model, thus ensuring a better prediction of alignments.

A pass of the alignment process is repeated after each training process to optimize the correct predictions between the text transcriptions and audio and also to make sure to have the proper latest alignments for the latest model in each

stage. Standard delta+delta-delta and LDA+MLLT training algorithms were used to obtain better alignments.

**Training algorithms for GMM-HMM model**

- Delta+delta-delta

  This algorithm computes the delta and double delta features that represent the first and second derivatives of the features, respectively. These delta features are computed on the window of the original MFCC features while the double delta features are computed on that of the delta features computed.

- LDA+MLLT

  LDA+MLLT term stands for Linear discriminant analysis – Maximum likelihood Linear Transform. LDA builds HMM states for the feature vectors, but with reduced feature space for all data. MLLT takes that reduced feature space output by LDA and derives a unique transformation for each speaker. This process is considered as a step for speaker normalization to minimize the differences among speakers.

**Determining hyper-parameters for triphone models**

Training of triphone models takes into consideration two parameters; the number of leaves in the decision tree(HMM states) and the total number of Gaussians across all states in the model for fine-tuning the model for the best alignments. The number of leaves parameter sets the maximum number of leaves in the decision tree while the number of Gaussians sets the maximum number of Gaussians distributed across the leaves. The number of Gaussians per leaf can be calculated by dividing the number of total Gaussians by the number of leaves. Since, 47 Sinhala phonemes are used in this project, it will require 47*47*47; nearly 100,000 models if a separate model is used for each triphone which will be computationally infeasible.

These parameters were set using the information retrieved after literature reviewing and then fine-tuning for our model, changing one parameter at once. In triphone training, the current phone, the preceding, and the following phone are considered as well. Thus, it requires at least 47*3 HMM states to model the con-

textual variation in the triphone model. Therefore,training of models were started with HMM states beginning from 250.

### 3.1.3 DNN training

As the final stage of the acoustic model, the resulting alignments and features are passed through different deep learning architectures such as a feed-forward network(DNN), Time Delay Neural Network(TDNN), and hybrid architectures of TDNN and LSTM(TDNN-LSTM). Since DNNs inherently perform well with large datasets, for our dataset, different techniques such as mini-batch Stochastic Gradient Descent, RBM pre-training,early-stopping, and dropouts were applied to enhance the performances of the models without getting over-fitted.

#### Mini-batch Stochastic Gradient Descent

The data set is divided into several n small batches, and the model error is calculated to update the model coefficients. Using this gradient descent method, the models enrich with the robustness of stochastic gradient descent and also the efficiency from training in small batches.

#### Dropouts

Dropouts are applied to overcome the problem of overfitting;the system performs well on trained data but performs very poorly on data that has never seen. Overfitting is a major problem especially when trained on a limited data set.
In TDNN, LSTM-TDNN training, the dropouts are applied layer-wise by introducing a new hyper-parameter that specifies the probability at which outputs of the layer are dropped out.

#### RBM Pre-training

According to Kaldi nnet1 setup(Hinton et al., 2012), before training the feed-forward network, an unsupervised pre-training process is applied to the training data. The resulting pre-trained Deep Belief Network is then passed into DNN training. With the use of pre-training, it adds a robustness to the system by giving a better generalization consistently.

## 3.2 Lexicon

The primary forms of lexicons found in natural language processing are phoneme and grapheme lexicons. The phoneme based lexicon takes into consideration the different pronunciation for each grapheme word; thus it has different sequences for each word, while the grapheme lexicon doesn't take the pronunciation into account. The proposed lexicon for the research is of type - grapheme. The grapheme lexicon that is used in this work contains over 220K entries with one unique grapheme sequence per word. This lexicon was created using the transcripts of the UCSC LTRL phonetically balanced corpus. Extracted Sinhala words were encoded to English letters and modified according to the rules in Sinhala transliteration. For this, the "Subasa" Sinhala transliteration software that has been developed by UCSC LTRL was used by modifying its Java scripts accordingly to prepare a lexicon in the Kaldi standard format. The figure 3.3 depicts the Sinhala transliteration scheme, which includes 28 distinct consonants, 19 distinct vowels where one of them is represented by a consonant character,17 distinct modifiers where vowel characters represent 15 of them. Therefore, altogether the length of the distinct phoneme set is 47 characters.

| | | | |
|---|---|---|---|
| ත t^*c | ක k*c | ◌ඃ h*v | ◌ා a:*m |
| ථ t^*c | ඛ k*c | ඇ a*v | ◌ැ ae*m |
| ද d^*c | ග g*c | ආ a:*v | ◌ෑ ae:*m |
| ධ d^*c | ඝ g*c | ඈ ae*v | ◌ි i*m |
| න n*c | ඞ x*c | ඉ i*v | ◌ී i:*m |
| ඳ nd^*c | ඟ ng*c | ඊ i:*v | ◌ු u*m |
| ප p*c | ච c^*c | උ u*v | ◌ූ u:*m |
| ඵ p*c | ඡ c^*c | ඌ u:*v | ◌ෘ ru*m |
| බ b*c | ජ j*c | ඍ ri*v | ◌ෲ ru:*m |
| භ b*c | ඣ j*c | ඎ ri:*v | ◌ෟ i^*m |
| ම m*c | ඤ cn*c | ඏ i^*v | ◌ෟ් i^:*m |
| ඹ mb*c | ඦ jn*c | ඐ i^:*v | ◌ෙ e*m |
| ය y*c | ඥ nj*c | එ e*v | ◌ේ e:*m |
| ර r*c | ට t*c | ඒ e:*v | ◌ෛ ai*m |
| ල l*c | ඨ t*c | ඓ ai*v | ◌ො o*m |
| ව w*c | ඩ d*c | ඔ o*v | ◌ෝ o:*m |
| ශ s^*c | ඪ d*c | ඕ o:*v | ◌ෞ au*m |
| ෂ s^*c | ණ n*c | ඖ au*v | |
| ස s*c | හ h*c | | |
| ඬ nd*c | ළ l*c | | |
| ෆ f*c | | | |

Figure 3.3: Sinhala transliteration scheme

## 3.3 Language Model

Modeling the way the words are connected to form sentences is done in language modeling. A tri-gram language model was built using the training corpus, provided lexicon, silence phones, and non-silence phones. For the validity of the experiments, the transcriptions relevant to the test data set were excluded from the text corpus as it would bias the performances of models. The SRILM language modeling tool(Stolcke, 2004) was used to create our own Sinhala language model. Kaldi framework also supports various language modeling tool kits.

## 3.4 Summary

In this chapter, the high level architecture and the overall design for addressing the research question were discussed in detail. The main components of the design and

each of their contributions to the ASR system were stated in this chapter. Next chapter, Chapter 4 will discuss the implementation of these components in detail.

# Chapter 4

# Implementation

The following sections will present the implementations performed in each step. Mainly, the experiments were carried out in the Kaldi speech recognition toolkit (Povey et al., 2011), which is freely available under the Apache License.

- Data Preparation
- Implementing the baseline model - (GMM-HMM)
- Implementing DNN models

## 4.1 Data Preparation

The data preparation stage requires a certain amount of time as the rest of the ASR pipeline highly depends on the consistency and integrity of the data preparation step. Kaldi requires data to be organized in a way that it supports all Kaldi underlying programming constructs.

The collected recordings from UCSC LTRL, which have a total of 15095 records. From the 15095 audio files, 700 of them were removed from test and validation sets as the speakers were reading the same script/sentences repeatedly and since it would bias the performances of the models by giving a too-low word error rate. The primary five files Kaldi requires for configuring the preparing stage are text, utt2spk, wav.scp, spk2gender, and segments. The format of the text and wav files that were used for the research are represented in 4.1 and 4.2, respectively. This lexicon is used for all the experiments conducted throughout the study.

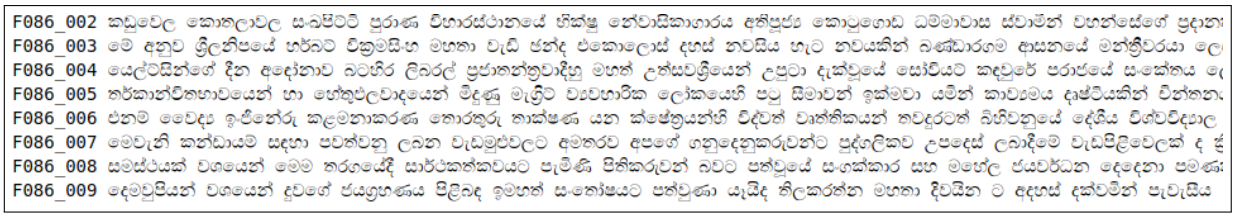


Figure 4.1: Part of the generated text file


Figure 4.2: Part of the generated wav.scp file

The figure 4.3 shows a part of the lexicon which was generated according to the Kaldi format by modifying the 'Subasa Transliteration Software' as stated under the design chapter. SIL and UNK refer to the silence and spoken noise, respectively.

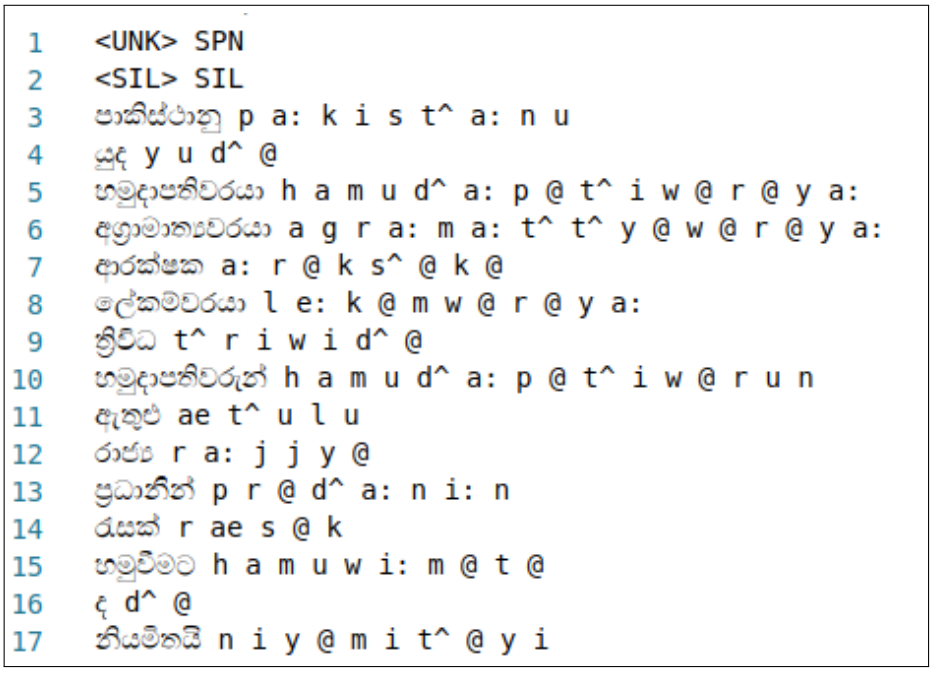


Figure 4.3: Part of the generated Sinhala lexicon

## 4.2 Implementing the baseline model - (GMM-HMM)

A detailed theoretical description of the steps involved in the implementation of the GMM-HMM model was presented in Chapter 3. Figure 4.4 presents the first part of the implementation steps of the monophone and triphone passes in the GMM-HMM model in an abstract way.

```
 98  echo "===== MONO DECODING ====="
 99  echo
100  utils/mkgraph.sh --mono data/lang exp/mono exp/mono/graph || exit 1
101  steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test exp/mono/decode
102  echo
103  echo "===== MONO ALIGNMENT ====="
104  echo
105  steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/mono exp/mono_ali || exit 1
106  echo
107  echo "===== TRI1 (first triphone pass) TRAINING ====="
108  echo
109  steps/train_deltas.sh --cmd "$train_cmd" 250 25000 data/train data/lang exp/mono_ali exp/tri1 || exit 1
110  echo
111  echo "===== TRI1 (first triphone pass) DECODING ====="
112  echo
113  utils/mkgraph.sh data/lang exp/tri1 exp/tri1/graph || exit 1
114  steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test exp/tri1/decode
115  echo
116  echo "===== TRI1 (first triphone pass) ALIGNMENT ====="
117  echo
118  steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/tri1 exp/tri1_ali || exit 1
119  echo
120  echo "===== TRI2 (Second triphone pass - [a larger model than TRI1]) TRAINING ====="
121  echo
122  steps/train_deltas.sh --cmd "$train_cmd" 300 30000 data/train data/lang exp/tri1_ali exp/tri2 || exit 1
123  echo
124  echo "===== TRI2 (Second triphone pass) DECODING ====="
125  echo
126  utils/mkgraph.sh data/lang exp/tri2 exp/tri2/graph || exit 1
127  steps/decode.sh --nj $nj --cmd "$decode_cmd" --config conf/decode.config exp/tri2/graph data/test exp/tri2/decode
128  echo
129  echo "===== TRI2 (Second triphone pass) ALIGNMENT ====="
```

Figure 4.4: Part of monophone and triphone passes

## 4.3 Implementing DNN models

This section presents different deep neural architectures that were implemented using the Kaldi toolkit to find out the architectures that perform well for the Sinhala speech recognition. In our Sinhala ASR system, the DNNs are applied on the top of the MFCC acoustic features extracted from Sinhala audio files with a variant number of hidden layers, hidden dimensions, activation functions, initial weights and other network parameters such as dropout schedules and number of epochs to compare and contrast the efficiency of the models. The following subsections give brief information about the DNN models that are implemented in the research.

### 4.3.1 Pre-trained DNN model

Layer-wise, pre-training is a still used technique that helps neural nets to converge faster and better. If the pre-training process is done rightly, it can put the model into better spots in function space that allow for better generalization, regularizing the architecture in a local, dataset dependent way. According to (Hinton et al., 2012), the supervised optimization from pre-trained weights consistently yields better performances rather than from randomly initialized weights in a neural network. The only difference from the standard neural network training is that its starting point in parameter space will be obtained after unsupervised pre-training.

Karel's nnet1(Hinton et al., 2012) sample setup present in the Kaldi toolkit which has been implemented according to (Hinton, 2010) was used for the pre-training process. The pre-training was done unsupervised manner on the training data set using a stack of Restricted Boltzmann machines, which is also known as a "Deep Belief Network."

### 4.3.2 Non pre-trained DNN model

Pre-training DNN models need extra hyper-parameters, and the computational time is also more substantial than training a general multilayer feed-forward network. Moreover, the results do not clearly show how the unsupervised generative model affects the final performance of our targeted supervised model. Because of this reason, the implementation of a regular deep neural network with randomly initialized weights was implemented using the same Karel's nnet1 sample setup.

### 4.3.3 TDNN models

After training with DNN models, the next model selection was a Time Delay Neural Network as they represent a mapping between past and present values. Although the same memory capture can be achieved through RNNs, when learning long term dependencies with RNNs, the "vanishing/exploding" gradient problem occurs, which means that as the error signals are propagated backward through the network's structure they tend to vanish or explode.

In recent researches such as (Liu et al., 2019),(Peddinti et al., 2015) and (Huang

et al., 2019), time-delay neural networks with sub-sampling have been proposed for effective modeling of long temporal contexts of speech. In a TDNN, the upper layers deal with information from a wider temporal context and thus can learn wider temporal relationships.

Two TDNN network types were experimented on the Kaldi toolkit. The first network has an asymmetric left and right context spliced frames while the second one is a factorized form of the TDNN with symmetric left and right contexts(time-strides), which has been introduced in the paper (Povey et al., 2018). The significant difference in the factorized TDNN is that it uses the resnet-type skips rather than skip-splicing. The experiments related to the first network type are carried out, changing the number of relu-renormalized layers(Rectified Linear Units-re-normalized), varying the left and right contexts of tdnn layers, and hidden dimensions in each layer.

Both the network types were trained on the 40-dimensional MFCC acoustic features. In addition to that, i-vectors of audios that have 100-dimensions were also extracted according to the run_ivector_common script provided in wall street journal Kaldi recipe. I-vector is a mapping from a variable-length speech segment to a fixed-dimensional representation that captures the long-term characteristics of the audio, such as the speaker characteristics or recording device. In ASR, it provides an additional input along with the MFCC acoustic features to the TDNN acoustic models, which helps the network learn to be robust to speaker and channel variations.

The next choice of the neural net architecture was a TDNN network followed by an LSTM layer. In recent literature, the experiments conducted for Russian (Markovnikov et al., 2018) and Mandarin Chinese (Li and Wu, 2014) speeches using combinations of LSTMs have resulted in higher accuracies. The experiments of these hybrid architecture were conducted using both TDNN and Factored TDNN network settings.

Implementation steps related to TDNN model trainings are depicted in the figure 4.5.

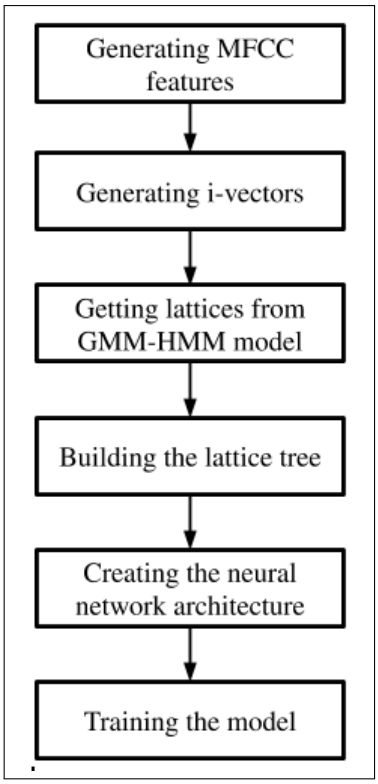


Figure 4.5: Implementation steps related to TDNN model trainings

## 4.4 Research Tools

- Kaldi

  Kaldi is an open-source toolkit for speech recognition written in C++ and licensed under the Apache License v2.0. It aims to provide software that is flexible and extensible. Kaldi tools support CUDA processing and other distributed parallel processing such as Grid Engine.

- "Antpc" server

  Training of all deep neural architectures and the decoding of the models were carried out on a single GPU - GeForce RTX 2080 Ti of 10.8GB provided by the Antpc server. A GPU-based instance is used to access to NVIDIA GPUs, thereby accelerate the deep learning training process by leveraging CUDA.

## 4.5 Summary

Throughout this chapter, the technical aspects related to implementation of the baseline model and four other deep neural networks were discussed. In addition to that, the tools and GPU specifications that enabled the implementation of the experiments were addressed in brief. Chapter 5 will address the results and evaluate the effects of these implementations for Sinhala ASR.

# Chapter 5

# Results and Evaluation

The first section of this chapter will discuss the evaluation metrics that are used in this research. In the second part, a detailed evaluation of the results experimented on the baseline model and DNN models will be discussed.

## 5.1 Evaluation Metric

The performance of this Sinhala speech recognition system is evaluated in terms of accuracy on the recordings taken in a quiet environment. This accuracy can be obtained by calculating either Word Error Rate (WER) or Sentence Error Rate (SER). WER is the number of words that are wrongly identified out of the total number of words in the audio sample used for recognition. SER is the number of sentences that are improperly identified out of the total number of sentences. Mostly, the WER is used in discrete speech recognition, whereas SER is used in continuous speech recognition where whole sentences are uttered. However, the standard measurement to assess the performance of an ASR system is the so-called WER.

### 5.1.1 Word Error Rate (WER)

Technically, WER is a minimum edit-distance measure produced by applying a dynamic alignment between the output of the ASR system and a reference transcript. It defines the following errors that can be distinguished in the alignment process,

- substitutions (sub)

- deletions (del) and
- insertions (ins)

Thus, the WER measure can be defined as follows,

$$WER = \frac{100 * (sub + ins + del)}{n} \tag{5.1}$$

$n$ refers to the total number of words in reference transcript.

Mainly, the WER measure will be used to evaluate the results obtained from both baseline and deep neural network models.

## 5.2 Experiments and results

### 5.2.1 Data set

Training the models involves a total data set from 70 speakers where 50 are female, and 20 are males. The data set is split in the ratio 8:1:1 for train, validation, and test data sets approximately. The training data set has audio recordings from 40 females and 16 males speakers, and the total utterances are 12295 sentences, which is 25h of speech data. As the validation data set,1050 speech utterances from five females and two male speakers are taken for fine-tuning the models. Testing the models involve a data set from five female speakers and two male speakers where they utter 1050 speech sentences altogether. Each of the validation and testing data set is 1.6 hours long. The overall details about the data sets are given in table 5.1.

Table 5.1: Details of train,validation and test data sets

| Data set | Female speakers | Male speakers | Hours |
|---|---|---|---|
| Train | 40 | 16 | 22h |
| Validation | 5 | 2 | 1.6h |
| Test | 5 | 2 | 1.6h |

### 5.2.2 Results of GMM-HMM model

As described in Chapter 3, a GMM-HMM model was trained on the same training data set to generate the alignments for the audio signals to bootstrap the DNN training. Thus getting correct alignments from GMM-HMM model highly affects the performances of DNN models.
After the monophone training process, the alignments are passed to three other triphone passes to fine-tune the models by modifying HMM states and the number of Gaussians. It was observed that cycling through training and alignment phases can better optimize the process.

After the observations made by decoding the triphone models, the lowest WER results were shown using the model configurations depicted in table 5.2.

Table 5.2: Results of GMM-HMM model

| Training Pass | WER% Valid set | WER% Test set |
|---|---|---|
| Monophone | 4.67 | 48.07 |
| Triphone Pass 1 | 3.81 | 42.88 |
| Triphone Pass 2 | 3.94 | 42.69 |
| Triphone Pass 3 | 3.80 | 42.64 |

### 5.2.3 Results of DNN models

As stated in previous chapters, these experiments were conducted by substituting the Gaussian Mixture Model by a Deep Neural Network algorithm. The MFCC features and CMVN were both used along with the alignments produced by the last triphone phase in the GMM-HMM to train the same training data set using each neural network algorithm.

The results generated using the RBM pre-trained deep neural network with different neural net configurations are shown in the table 5.3. The number of epochs were set to a constant value of 20 after experimenting several rounds.

Table 5.3: Results of pre-trained DNN models

| # hidden layers | Activation function | # epochs | #hidden units per layer | WER% Valid set | WER% Test set |
|---|---|---|---|---|---|
| 2 | Sigmoid | 20 | 256 | 3.54 | 40.04 |
| | | | 512 | 3.50 | 40.08 |
| | | | 1024 | 3.54 | 40.06 |
| 3 | Sigmoid | 20 | 256 | 3.61 | 39.92 |
| | | | 512 | 3.50 | 40.33 |
| | | | 1024 | 3.55 | 40.01 |
| 4 | Sigmoid | 20 | 256 | 3.46 | 40.39 |
| | | | 512 | 3.41 | 40.50 |
| | | | 1024 | 3.63 | 40.41 |
| 5 | Sigmoid | 20 | 256 | 3.52 | 40.59 |
| | | | 512 | 3.45 | 40.62 |
| | | | 1024 | 3.65 | 40.68 |
| 6 | Sigmoid | 20 | 256 | 3.50 | 40.33 |
| | | | 512 | 3.59 | 40.84 |
| | | | 1024 | 3.47 | 40.72 |
| 7 | Sigmoid | 20 | 256 | 3.61 | 41.60 |
| | | | 512 | 3.60 | 40.58 |
| | | | 1024 | 3.52 | 41.29 |

The results display the improvement gained over the baseline GMM-HMM model by lowering a WER of 2% or more in every model in the pre-trained DNN. The lowest or the best WER is observed to be 39.92%, which is 2.72% lower than the best WER of the baseline model. However, the results do not clearly show how the unsupervised generative model affects the final performance of our targeted supervised model. Because of this reason, as described in Chapter 4, a regular deep neural network with randomly initialized weights was implemented. The results generated for the non-pre-trained models with the same network configurations as in the pre-trained models are depicted in the table 5.4.

Table 5.4: Results of non pre-trained DNN models

| # hidden layers | Activation function | # epochs | #hidden units per layer | WER% Valid set | WER% Test set |
|---|---|---|---|---|---|
| 2 | Sigmoid | 20 | 256 | 3.54 | 40.52 |
| | | | 512 | 3.68 | 40.32 |
| | | | 1024 | 3.76 | 40.28 |
| 3 | Sigmoid | 20 | 256 | 3.60 | 40.41 |
| | | | 512 | 3.78 | 39.81 |
| | | | 1024 | 3.77 | 39.92 |
| 4 | Sigmoid | 20 | 256 | 3.59 | 40.31 |
| | | | 512 | 3.65 | 40.09 |
| | | | 1024 | 3.78 | 40.15 |
| 5 | Sigmoid | 20 | 256 | 3.61 | 40.06 |
| | | | 512 | 3.61 | 40.05 |
| | | | 1024 | 3.59 | 40.21 |
| 6 | Sigmoid | 20 | 256 | 3.52 | 40.22 |
| | | | 512 | 3.59 | 39.69 |
| | | | 1024 | 3.78 | 40.15 |
| 7 | Sigmoid | 20 | 256 | 3.58 | 41.60 |
| | | | 512 | 3.59 | 40.58 |
| | | | 1024 | 3.76 | 41.29 |

The results observed from the DNN models without pre-training are slightly better when compared with the pre-trained models under the same network configurations. The best WER was found to be 39.69%, which is only 0.23% lower than the best pre-trained DNN result. However, both the results from the DNN models surpass the performance of the baseline model. Another set of experiments was conducted by changing the activation function of these models from sigmoid to Tanh function. But the results observed were higher WERs for the same network configuration except the learning rate.

Since the results gained so far remain in a higher WER, the next experiments

were conducted on a different deep network structure named TDNN, as described in detail in Chapter 4. The results experimented on the two network types of TDNNs are summarized in table 5.5.

Table 5.5: Results of two network settings of TDNN

| TDNN network type | # of TDNN layers | layer dimensions | WER% Valid set | WER% Test set |
|---|---|---|---|---|
| 1 | 6 | 128 | 3.42 | 40.49 |
| | | 512 | 3.52 | 39.27 |
| | | 1024 | 4.71 | 40.36 |
| 2 | 9 | 1024 | 5.36 | 35.48 |
| 2 | 13 | 256 | 5.90 | 35.16 |
| | | 512 | 5.09 | 35.44 |
| | | 1024 | 4.75 | 35.45 |

As depicted in table 5.5, the lowest WER so far was observed from the TDNN network type 2, which is 35.16%. With compared to baseline, pre-trained, and non-pre-trained models, the WER has been decreased by 7.48%, 4.76%, and 4.53%, respectively.

Another variation of TDNN network was experimented by layering an LSTM on the top of TDNN layers. Experiments were conducted by layering the LSTM layer on both the TDNN and factored TDNN layers. An experiment was conducted by layering an LSTM on top of the best scored(35.16%) TDNN model. However, the WER was increased up to 35.87%. The overall results observed from these experiments are shown in table 5.6.

Table 5.6: Results of TDNN+LSTM models

| # of TDNN layers | # of LSTM layers | layer dimensions | WER% Valid set | WER% Test set |
|---|---|---|---|---|
| 4 | 1 | 512 -TDNN<br>384 -LSTM | 4.43 | 36.23 |
| | | 256 -TDNN<br>384 -LSTM | 4.80 | 36.33 |
| 13 | 1 | 256 - TDNNF<br>384 - LSTM | 4.35 | 35.87 |

The results from TDNN+LSTM does not surpass the performance of TDNN only models. This may due to the increase in the complexity of the models when adding an LSTM layer. The number of parameters blows up a lot since LSTMs makes four different projections from its input. Therefore, further experiments adding lstm layers were not conducted. However, there is a possibility of slightly changing the WERs when tuned with different network configurations.

### 5.2.4 Evaluation of results

In this research, four variations of deep neural networks and a statistical baseline model were experimented for the task of speech recognition of the Sinhala language. A comprehensive evaluation of these models will be conducted in this section.

Table 5.7 shows a summary of the best performance or the lowest WER obtained from each model for the test data set.

Table 5.7: Summary of the best WERs obtained from all the models

| DNN model | WER% Test set |
|---|---|
| Baseline GMM-HMM | 42.64 |
| Pre-trained DNN | 39.92 |
| Non pre-trained DNN | 39.69 |
| TDNN | 35.16 |
| TDNN+LSTM | 35.87 |

According to table 5.7, the TDNN network has shown the lowest WER, which means it is the best network setting observed from the experiments conducted in this research. Even the hybrid architecture of TDNN+LSTM shows a lower WER than the regular DNNs, which highlights the factor that TDNNs perform much better in speech recognition tasks.

To clearly distinguish how the performances of models have practically affected the test audio files, a comparison of four translated sentences have been made in figure 5.1.

1. **Test sentence :** එම කටයුතු භාරවූයේ ඇඹරුම්හල් හිමියකුගේ පුතකු වන කල්දේරෝටය

| | |
|---|---|
| Baseline GMM-HMM | එම කටයුතු භාරවූයේ ඇහලුම් මළ හිමිය ක් උගේ පුතකු නොවන කරදදී රුවට ය |
| Pre-trained DNN | එම කටයුතු භාරවූයේ ඇහලුම් කල්හි මිය ගියේත් පුතකු නුවණ කලදේ රූපය |
| Non pre-trained DNN | එම කටයුතු භාරවූයේ ඇහලුම් කල්හි මිය ගියේත් පුතකු නුවණ කලදේ රූපය |
| TDNN | එම කටයුතු භාරවූයේ ඇඹරුණේ වහල් හිමියාගේ පුතකු වන කළ දේ රෝහලටය |
| TDNN+LSTM | එම කටයුතු භාරවූයේ ඇඹරුණේ වහල් හිමියාගේ උගේ පුතකු වන කලදේ රෝහලටය |

2. **Test sentence :** චිත්තිකාරයෝ තුන් දෙනෙක් බිහිවෙන්නෙ එහෙමයි

| | |
|---|---|
| Baseline GMM-HMM | බිත්ති කාරයෝ තුන්දෙනෙක් බිහිවෙන්නේ එහෙමයි |
| Pre-trained DNN | බිත්ති කාරයෝ තුන්දෙනෙක් බිහිවෙන්නේ එහෙමයි |
| Non pre-trained DNN | බිත්ති කාරයෝ තුන්දෙනෙක් බිහිවෙන්නේ එහෙමයි |
| TDNN | චිත්ති කාරයෝ තුන්දෙනෙක් බිහිවෙන්නේ එහෙමයි |
| TDNN+LSTM | චිත්ති කාරයෝ තුන්දෙනෙක් බිහිවන්නේ එහෙමයි |

3. **Test sentence :** ඊළහට පැමිණි යල් කන්නයේ දිගටම දැඩි නියහය පැවැති වැව පතුලටම හිඳිණි

| | |
|---|---|
| Baseline GMM-HMM | ඊළහට පැමිණි යළි කන්නේ දිගටම දැඩි නියහය පැවතියේ මැවූ පතුල ටැම්ල් වීමෙනි |
| Pre-trained DNN | ඊළහට පැමිණෙන්නේ යළි කන්නයේ දිගටම දැඩි නියහය පැවතියේ වැව පතුල පැමිණිනි |
| Non pre-trained DNN | ඊළහට පැමිණෙන්නේ යළි කන්නයේ දිගටම දැඩි නියහය පැවතියේ වැව පතුල පැමිණිනි |
| TDNN | ඊළහට පැමිණෙන්නේ යල් කන්නයේ දිගටම දැඩි නියහයක් පැවතියේ වැව පත් තුළටම හිඳිනු |
| TDNN+LSTM | ඊළහට පැමිණි යළි කන්නයේ දිගටම දැඩි නියහයක් පැවැතියේ වැව පවුලටම හිඳියි |

4. **Test sentence :** ඉන්දියාව යාපනයට පරිජ්පු ගෝනි උඩින් අතහැරියේ අපේ ස්වාධිපත්‍යය හැල්ලුවට ලක් කරමිනි

| | |
|---|---|
| Baseline GMM-HMM | ඉන්දියාව යාපනයට පරිජ්පු ගෝනි උඩින් අතහැරිය අපේ ස්වාධිපත්‍යයට ඇල්ලු වටලා කරමිනි |
| Pre-trained DNN | ඉන්දියාව යාපනයට පරිජ්පු ගෝනි උඩින් අත්හැරියේ අපේ ස්වාධිපත්‍යයට යාළුවන්ට ලක් කරමිනි |
| Non pre-trained DNN | ඉන්දියාව යාපනයට පරිජ්පු ගෝනි උඩින් අතහැරිය අපේ ස්වාධිපත්‍යයට බැලුවද ලක් කරමිනි |
| TDNN | ඉන්දියාව යාපනයට පරිජ්පු ගෝනි උඩින් අත්හැරියේ අපේ ස්වාධිපත්‍යයට ඇල්ලට ලක්කරමින් ය |
| TDNN+LSTM | ඉන්දියාව යාපනයට පරිජ්පු ගෝනි උඩින් අත්හැරියේ අපේ ස්වාධිපත්‍යයට සැහැල්ලුව ලක් කරමිනි |

Figure 5.1: Four translated example sentences based on baseline GMM-HMM, pre-trained DNN, non-pre-trained DNN, TDNN, and TDNN+LSTM. Phrases in the bold green text show the exact matching compared to the correct test sentence. Phrases in the bold red text show the words that are incorrectly translated by the models while the phrases highlighted in yellow shows the word segmentation issues and slight deviations

When considering the test sentence 1, all the models have failed to correctly translate the words "ඇඹරුම්හල්, "කල්දේරෝටය." When analyzing the sentence, it was found, these words are not used in the corpus, although they are included in the lexicon. Therefore, the models have failed to find any relationship between the words and have lead to incorrect translations. This signifies that the richness of the text corpus, along with the lexicon, is an essential factor for developing a robust speech recognition system. However, the TDNN type models has been able to slightly translate those words, which is a noteworthy feature. This same scenario has happened in the test sentence four also, where all the models have failed to identify the word "හැල්ලුවට ".

Significant performance is shown from TDNN and TDNN+LSTM models in the test sentence 2, as they have been able to identify the word "විත්ති" correctly. In the text corpus, the phrase " බිත්ති" is not followed by any suffix that is relevant to this sentence. However, the phrase "විත්ති" is followed by suffixes such as " යක්"," කරු","කරුවන්ගේ "," කරුගේ" in several times. The baseline, pre-trained, and non-pre-trained models have wrongly identified the word as "බිත්ති". This is because these models don't have the capability to learn long-term dependencies of speech as they only focus on the current context. The TDNN and TDNN+LSTM models have correctly identified the word since they are capable of modeling information from a wider temporal context and can find wider temporal relationships.

When analyzing the translated sentences, most of the sentences that are translated from the pre-trained and non-pre-trained DNNs take the same form as their best WERs also has only a small difference. This can be seen in test sentences 1,2, and 3 in figure 5.1.

## 5.3 Summary

The results from each experimented model were presented in this chapter in detail. Together with the results, an analysis of the WER scores and translated sentences, comparing different experiments, was also discussed. The potential conclusions that can be drawn from these results are presented in Chapter 6.

# Chapter 6

# Conclusions

## 6.1 Introduction

This thesis is on developing an automatic speech recognition system for the Sinhala language by using deep learning techniques. Initially, this research started with an in-depth look at the literature of the Automatic Speech Recognition(ASR). In chapter 2, a comprehensive literature review was conducted to figure out the gaps in this area of study. It becomes evident that less research has been done in the field of Sinhala speech recognition. The reason for this is the limited resources to continue a research study.

This chapter provides an overall picture of the conclusions drawn from the whole research work conducted by us.

## 6.2 Conclusions about research questions and objectives

The main research objectives of this comprehensive study are developing a Sinhala ASR system using deep learning techniques and comparing the performances of the experimented DNN models with the statistical baseline model, which is GMM-HMM. For this, we are addressing the research question, "What deep neural architectures will perform well for Sinhala ASR with limited resources?".

Initially, we started developing the statistical model, which has been the state-of-the-art method for speech recognition for several years. To tackle the problem of

fewer resources, the training of deep neural networks were performed on the alignments generated using the baseline model rather than training immediately from utterance level transcriptions.

In this research, we have trained four types of deep neural networks, namely pre-trained DNN, non-pre-trained regular DNN, TDNN, and TDNN, followed by LSTM (TDNN+LSTM). When evaluating the results, the highest WER was observed from the baseline GMM-HMM model, which is 42.64%. This result is used for evaluating the performance of deep neural networks over the statistical approach. It was observed that the results obtained from pre-trained and regular DNN are comparatively similar. As in table 5.3, the best WER from the pre-trained model, which is 39.92% can be obtained using three hidden layers with 256 hidden units, 0.008 of the learning rate, and Sigmoid activation function. When experimented under the same learning rate and activation function, the best WER of the regular DNN was observed as 39.69% when trained on six hidden layers with 512 hidden units. However, these two DNN models have a small improvement of 2.72% and 2.95% when compared with the baseline model. While in table 5.5, it can be observed that the TDNN model is dramatically better than the previous three models, including the baseline model. This proves the efficiency of TDNN network settings towards the task of speech recognition. The best WER achieved from the TDNN model, which is 35.16% can be obtained from the factored TDNN setting with 13 TDNN layers, each having 256 hidden units. The performance of the best TDNN model over the baseline and previous DNN models is 7.48%, 4.76%, and 4.53%, respectively. Later, when trained with the hybrid architecture - TDNN+LSTM as in table 5.6, the WER got from TDNN was increased slightly by 0.71% . This may due to the model complexity getting increased compared to the limited data set.

A significant improvement of the WERs can be observed, especially when using the factored form of TDNN layers where the architecture uses the resnet-type skips rather than skip-splicing. In Chapter 5, as analyzed the performance by bringing out example sentences from each model, it was observed that if the models can learn broader temporal contexts, they are likely to identify the correct words for the given audio frame. Thus, we could identify the Time Delay Neural Network(TDNN) as one of the deep neural architectures that perform well for the Sinhala speech

recognition even with a limited data set of about 25hours. The results obtained show that the Deep neural network architectures exceed the baseline - GMM-HMM performance model with a maximum WER of 7.48% on the test data set.

## 6.3 Conclusions about research problem

According to the results concluded in section 6.2, this research has been able to find a DNN model that will averagely perform better for speech recognition of the Sinhala language. Chapter 5 showed the lowest WERs observed from each DNN model and the baseline GMM-HMM model. In Section 5.2.4, it can be noted that if the text corpus is extended further along with more training data, the models are able to identify most of the words correctly.

When compared with previous work (Amarasingha and Gamini, 2012),(Manamperi et al., 2018),(Nadungodage and Weerasinghe, 2011),(Gunasekara and Meegama, 2015), that has been done on Sinhala ASR, this research has used relatively an extensive data set and have come up with a DNN solution that outperforms the statistical model results. Thus, this study contributed to the domain of speech recognition by exploring deep neural architectures that give better results for the Sinhala language with low resources.

## 6.4 Limitations

A moderate vocabulary of Sinhalese speech data is used for all the training conducted in the research. As this research aims in developing an ASR system that works for the general domain, it needs to have an adequate amount of speech data as well as a rich text corpus. However, the used data set was created by UCSC LTRL, and it was extended further by collecting recordings from time to time. Therefore, the training of models had to be repeated several times with the extended data set, which required a lot of time and effort.

Initially, the research experiments had to be conducted using the GCP personal account with minimal computational power until the University provides the computational facility. However, with the demand increases for computational resources, the jobs had to be kept in the waiting queue for several days.

Since it is impractical to train on every possible network configuration related to a particular model, the hidden layers were incrementally increased from 2 to 7, observing 256,512 and 1024 hidden dimensions in DNNs. In TDNN networks,the number of layers and hidden units were changed randomly based on the results. These network configurations were mostly based on the literature related to DNN training.

Understanding the Kaldi coding style, I/O internals, data structures, and process communication styles is a tedious task compared to other frameworks that are written in higher-level languages such as Java. It took several months to get familiar to the framework thoroughly. The reason for this is the Kaldi core is written in C++, and executable programs are consolidated in bash scripts, which are from little to no readable.

## 6.5 Implications for further research

Many investigations can be carried out to develop an improved robust model that will deliver lower WER and better frame accuracy. Since the lowest WER was observed from TDNN models, experimenting on the TDNN with a different numbers of layers, contexts, epochs, and other configurations can be conducted as future work. A deep error analysis on the outputs gained by different network types and network configurations can be carried out as a future enhancement. It would help to identify the types of words that can be easily recognized by the developed system. Since the size of the speech corpus affects the results of the models, another future work is creating a more extensive corpus with a phonetically balanced vocabulary can be done. In addition to that, with a larger corpus, experimenting on TDNN networks concatenating CNN, LSTM, and BLSTM layers is another future enhancement to this research.

# References


Amarasingha, W. and Gamini, D. (2012), Speaker independent sinhala speech recognition for voice dialling, pp. 3–6.

Deka, B., Nirmala, S. and K., S. (2018), Development of assamese continuous speech recognition system, pp. 215–219.

Deng, l., Hinton, G. and Kingsbury, B. (2013), New types of deep neural network learning for speech recognition and related applications: An overview, pp. 8599–8603.

Du, X., Cai, Y., Wang, S. and Zhang, L. (2016), Overview of deep learning, *in* '2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)', pp. 159–164.

Fohr, D., Mella, O. and Illina, I. (2017), New Paradigm in Speech Recognition: Deep Neural Networks, *in* 'IEEE International Conference on Information Systems and Economic Intelligence', Marrakech, Morocco.
**URL:** *https://hal.archives-ouvertes.fr/hal-01484447*

Gunasekara, M. K. H. and Meegama, R. G. N. (2015), Real-time translation of discrete sinhala speech to unicode text, *in* '2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)', pp. 140–145.

Hinton, G. (2010), 'A practical guide to training restricted boltzmann machines (version 1)'.

Hinton, G., Deng, l., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. and Kingsbury, B. (2012), 'Deep neural

networks for acoustic modeling in speech recognition: The shared views of four research groups', *Signal Processing Magazine, IEEE* **29**, 82–97.

Hsu, W.-N., Zhang, Y. and Glass, J. (2016), A prioritized grid long short-term memory rnn for speech recognition, pp. 467–473.

Huang, X., Zhang, W., Xu, X., Yin, R. and Chen, D. (2019), 'Deeper time delay neural networks for effective acoustic modelling', *Journal of Physics: Conference Series* **1229**, 012076.

J.Arora, S. and Singh, R. (2012), 'Automatic speech recognition: A review', *International Journal of Computer Applications* **60**, 34–44.

Jurafsky, D. and Martin, J. (2008), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Vol. 2.

Kalchbrenner, N., Danihelka, I. and Graves, A. (2015), 'Grid long short-term memory'.

Kimanuka, U. and BUYUK, O. (2018), 'Turkish speech recognition based on deep neural networks', *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi* **22**, 10.

Li, X. and Wu, X. (2014), 'Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition', *CoRR* **abs/1410.4281**.
**URL:** *http://arxiv.org/abs/1410.4281*

Liu, B., Zhang, W., Xu, X. and Chen, D. (2019), 'Time delay recurrent neural network for speech recognition', *Journal of Physics: Conference Series* **1229**, 012078.

Manamperi, W., Karunathilake, D., Madhushani, T., Galagedara, N. and Dias, D. (2018), Sinhala speech recognition for interactive voice response systems accessed through mobile phones, pp. 241–246.

Markovnikov, N., Kipyatkova, I., Karpov, A. and Filchenkov, A. (2018), Deep neural networks in russian speech recognition, pp. 54–67.

Nadungodage, T. and Weerasinghe, R. (2011), 'Continuous sinhala speech recognizer'.

Pallavi Saikia, Assistant Professor, G. U.-I. o. D. and Open Learning, Assam, I. (2017), *International Journal of Development Research* **07**.

Peddinti, V., Povey, D. and Khudanpur, S. (2015), A time delay neural network architecture for efficient modeling of long temporal contexts, *in* 'INTERSPEECH'.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M. and Khudanpur, S. (2018), Semi-orthogonal low-rank matrix factorization for deep neural networks, pp. 3743–3747.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G. and Vesel, K. (2011), 'The kaldi speech recognition toolkit', *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* .

Saksamudre, S., Shrishrimal, P. and Deshmukh, R. (2015), 'A review on different approaches for speech recognition system', *International Journal of Computer Applications* **115**, 23–28.

Samudravijaya, K., Rao, P. and Agrawal, S. (2000), Hindi speech database., pp. 456–459.

Saurav, J., Amin, S., Kibria, S. and Rahman, M. (2018), Bangla speech recognition for voice search, pp. 1–4.

Stolcke, A. (2004), 'Srilm — an extensible language modeling toolkit', *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)* **2**.

*Subasa* (n.d.), `http://transliteration.sinhala.subasa.lk/`.

*TIMIT Dataset* (n.d.), `https://catalog.ldc.upenn.edu/LDC93S1`.

*wiki_Sinhala*(*n.d.*), .

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Laurent, C., Bengio, Y. and Courville, A. C. (2017), 'Towards end-to-end speech recognition with deep convolutional neural networks', *CoRR* **abs/1701.02720**.
**URL:** *http://arxiv.org/abs/1701.02720*

# Appendices

# Appendix A

# Model Specifications

The network specifications of the pre-training process are depicted in table A.1.

Table A.1: Network specifications of pre-trained DBN

| # RBM hidden layers | #hidden units per layer | RBM learning rate | Lower RBM learning rate |
|---|---|---|---|
| 6 | 2048 | 0.4 | 0.01 |

The network specifications of the best TDNN model are represented in table A.2.

Table A.2: Network specifications of the best TDNN model

| Network type | TDNN layer information | # Dimensions | # Epochs |
|---|---|---|---|
| 2 | relu-batchnorm-layer name=tdnn1<br>tdnnf-layer name=tdnnf2 time-stride=1<br>tdnnf-layer name=tdnnf3 time-stride=1<br>tdnnf-layer name=tdnnf4 time-stride=1<br>tdnnf-layer name=tdnnf5 time-stride=0<br>tdnnf-layer name=tdnnf6 time-stride=3<br>tdnnf-layer name=tdnnf7 time-stride=3<br>tdnnf-layer name=tdnnf8 time-stride=3<br>tdnnf-layer name=tdnnf9 time-stride=3<br>tdnnf-layer name=tdnnf10 time-stride=3<br>tdnnf-layer name=tdnnf11 time-stride=3<br>tdnnf-layer name=tdnnf12 time-stride=3<br>tdnnf-layer name=tdnnf13 time-stride=3 | 256 | 10 |

The network configurations of the best TDNN+LSTM model is depicted in the figureA.3.

Table A.3: Network specifications of TDNN+LSTM model

| layer information | # TDNN dimensions | # LSTM dimensions | # Epochs |
|---|---|---|---|
| relu-batchnorm-layer name=tdnn1<br>tdnnf-layer name=tdnnf2 time-stride=1<br>tdnnf-layer name=tdnnf3 time-stride=1<br>tdnnf-layer name=tdnnf4 time-stride=1<br>tdnnf-layer name=tdnnf5 time-stride=0<br>tdnnf-layer name=tdnnf6 time-stride=3<br>tdnnf-layer name=tdnnf7 time-stride=3<br>tdnnf-layer name=tdnnf8 time-stride=3<br>tdnnf-layer name=tdnnf9 time-stride=3<br>tdnnf-layer name=tdnnf10 time-stride=3<br>tdnnf-layer name=tdnnf11 time-stride=3<br>tdnnf-layer name=tdnnf12 time-stride=3<br>tdnnf-layer name=tdnnf13 time-stride=3<br>lstm-layer name=lstm3 decay-time=20 delay=-3 | 256 | 384 | 10 |

# Appendix B

# Decoded text

**Test sentence :** ඒවාවලින් තහවුරු වුවොත් පමණයි අවලංගු කරන්නේ

| | |
|---|---|
| Baseline GMM-HMM | ඒවා බලෙන් තහවුරුවි පමණයි අවලංගු කරන්නේ |
| Pre-trained DNN | ඒ සේවාවලින් තහවුරුවි පමණයි අවලංගු කරන්නේ |
| Non pre-trained DNN | ඒවා වලින් තහවුරු වුවොත් පමණයි අවලංගු කරන්නේ |
| TDNN | ඒවා වලින් තහවුරුවද පමණයි අවලංගු කරන්නේ |
| TDNN+LSTM | ඒවා වලින් තහවුරුව පමණයි අවලංගු කරන්නේ |

**Test sentence :** රූපියල් හාර ලක්ෂ පනස් තුන් දහස් එකසිය විස්සයි

| | |
|---|---|
| Baseline GMM-HMM | රූපියල් හාර ලක්ෂ පනස් තුන් දහස් එකසිය විස්සයි |
| Pre-trained DNN | රූපියල් හාර ලක්ෂ පනස් තුන් දහස් එකසිය විස්සයි |
| Non pre-trained DNN | රූපියල් හාර ලක්ෂ පනස් තුන් දහස් එකසිය විස්සයි |
| TDNN | රූපියල් හාර ලක්ෂ පනස් තුන් දහස් එකසිය විස්සයි |
| TDNN+LSTM | රූපියල් හාර ලක්ෂ පනස් තුන් දහස් එකසිය විස්සයි |

**Test sentence :** සිංහලය අපූරු භාෂාවකි

| | |
|---|---|
| Baseline GMM-HMM | සිංහලයා අපූරු භාෂාවකි |
| Pre-trained DNN | සිංහලය අපූරු භාෂාවකි |
| Non pre-trained DNN | සිංහලයා අපූරු භාෂාවකි |
| TDNN | සිංහලයා අපූරු භාෂාවකි |
| TDNN+LSTM | සිංහලය අපූරු භාෂාවකි |

Figure B.1: Three other translated sentences based on baseline GMM-HMM, pre-trained DNN, non-pre-trained DNN, TDNN, and TDNN+LSTM. Phrases in the bold green text show the exact matching compared to the correct test sentence. Phrases in the bold red text show the words that are incorrectly translated by the models while the phrases highlighted in yellow shows the word segmentation issues and slight deviations