Enhancing Sinhala Text-to-Speech System Using Deep Learning Techniques.

K.L.P.M. Senarath

2024

Enhancing Sinhala Text-to-Speech System Using Deep Learning Techniques.

A dissertation submitted for the Degree of Master of Computer Science

K.L.P.M. Senarath

University of Colombo School of Computing

2024

DECLARATION

Name of the student: K.L.P.M Senarath
Registration number: 2020/MCS/086
Name of the Degree Programme: Master of Computer Science
Project/Thesis title: Enhancing Sinhala Text-to-Speech System Using Deep Learning Techniques.

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.

3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.

4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.

5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.

6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
	**11/09/2024**

Certified by Supervisor

This is to certify that this project/thesis is based on the work of the above-mentioned student under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor
Name	**Dr. B.H.R. Pushpananda**
Signature	
Date	13 - 09 - 2024

I would like to dedicate this thesis to all users who are interested in this area of study.

ABSTRACT

Knowledge is an important asset to all human beings to lead a successful life. Normally people gain knowledge through several ways and resources. Visually impaired people face a lot of problems throughout their lifetime since normally they gain knowledge through word of mouth, audio books, and using braille systems. They need support or assistance to carry out even their day-to-day tasks. So, it is very important to initiate necessary steps to help them using the prevailing technologies. So that they can also lead a good life. Nearly a million in Sri Lanka suffer from blindness or from conditions that could lead to blindness. Blind people are unable to perform visual tasks. Most published printed works do not include braille or audio versions. There are some systems that use the OCR framework for recognition of its text, which is then synthesized through a process of TTS for languages such as English, Tamil, etc.

This study focused on enhancing Text-to-Speech (TTS) technology for the Sinhala language, aiming to improve accessibility for visually impaired individuals in Sri Lanka. It tackled the challenge of adapting TTS for a low-resource language by utilizing the VAENAR model, a strategy previously successful with English, in pursuit of creating a Sinhala TTS system capable of delivering natural and intelligible speech. Despite confronting substantial obstacles, including significant computational requirements and the inadequacy of the model to produce clear speech in both English and Sinhala, the research provided important directions for future TTS development.

The outcomes underscored the critical need for tailored deep learning approaches, enhanced linguistic data collection, and stronger collaborative networks within the academic and research communities. These elements are vital for crafting TTS technologies that are accessible and useful to visually impaired users and broadly beneficial across various linguistic groups. This work lays a foundation for future innovations in TTS systems, advocating for more inclusive and effective solutions for the Sinhala language and other low-resource languages, thereby offering significant contributions to the field and its potential impact on society.

ACKNOWLEDGEMENTS

I would like to express special thanks & gratitude to my research supervisor, Dr. B.H.R. Pushpananda Senior lecturer of the University of Colombo School of Computing who gave me a lot of ideas and help to work on this project on the topic of "Sinhala Text-to-Speech Using Variational Auto-Encoder based Non-Autoregressive Text-to-Speech Synthesis", which led me into doing a lot of Research which diversified my knowledge to a huge extent for which We are thankful.

I would also like to thank all my batch mates for giving comments, motivation, and their warm friendship. It is a great pleasure for me to acknowledge all the teachers, mentors, and people for the immense support they have given. Finally, to my parents and family for their immeasurable sacrifices and love they have given throughout this beautiful journey.

K.L.P.M Senarath
2020/MCS/086

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

DL - Deep Learning

G2P - grapheme-to-phoneme

ML - Machine Learning

MOS - Mean Opinion Score

TTS - Text-to-Speech

VAENAR – Variational Auto-Encoder based non-autoregressive

CHAPTER 1 - INTRODUCTION

1.1.Overview

Visually impaired people face a lot of problems throughout their lifetime. They need support or assistance to carry out even their day-to-day tasks. So, it is very important to initiate necessary steps to help them using the prevailing technologies. So that they can also lead a good life. Nearly a million in Sri Lanka suffer from blindness or from conditions that could lead to blindness (Xinhuanet.com, 2020). Blind people are unable to perform visual tasks. Most published printed works do not include braille or audio versions. There are some systems that use the OCR framework for recognition of its text, which is then synthesized through a process of TTS for languages such as English, Tamil, etc.

Speech synthesis is a popular area in which several research is being conducted with the purpose of helping disabled people. A Text-To-Speech (TTS) is an assistive technology that converts normal language text into speech. A computer that is used in this process is called a speech synthesizer. The success of a TTS system relies on naturalness, intelligibility, preference, and comprehensibility. This TTS system can help the visually impaired and people with reading difficulties. Various kinds of experiments have been done on TTS using different technologies. Also, we can see research related to Sinhala Language that has been done using various deep learning techniques (Weerasinghe, R. *et al, 2007* ;Kasthuri Arachchige, T.C. (1970)).

Using TTS allows us to develop a powerful chatbot system, which is another important benefit. It makes it possible for chatbots to interact with users who might struggle with reading or writing, like those who have dyslexia or visual impairments. TTS technology also enables chatbots to offer consumers hands-free information access, making it simpler for users to communicate with chatbots while taking part in other activities like driving or working out.

The approaches used for the development of Text to Speech synthesis include Concatenation synthesis (Hertz, S.R. (2002)), Formant synthesis (Hertz, S.R. (2002)), Articulatory synthesis (Hill, D.R., Taube-Schock, C.R. and Manzara, L. (2017)), HMM (Hidden Markov Model) (Jayaweera, A.J.P.M.P. and Dias, N.G.J. (2014) ; H. U. Mullah, F. Pyrtuh, and L. J. Singh, 2015) based synthesis, Sine wave synthesis (Rodríguez Crespo, M.Á. et al. (1997)) and Deep learning-based synthesis (Weerasinghe, R. *et al, 2007* ; Kasthuri Arachchige, T.C. (2023) ; Saba, R. *et al.* (2022)). Among these, Deep learning-based synthesis is the trending approach nowadays. This approach is used to overcome the inefficiencies of decision trees used in HMMs to model complex context dependencies. In this method Deep Neural Networks are employed to model the relationship between input and output in this approach. This approach has improved the naturalness and intelligibility of the speech output.

1.2.Motivation

So given the above context in this research, Focus is to find an optimized text-to-speech solution that can be give more accurate output in scope of Sinhala Language. This issue has been addressed and having discussions lately in research communities and trying to find a better solution with the latest technologies and techniques. By studying and analyzing the similar systems and technologies thoroughly, found that the making an optimized Sinhala TTS System would be valuable, especially for Visually impaired people whose mother tongue is Sinhala. Since these kinds of improved systems are currently available in English and some other foreign language, we try to fulfill this gap by addressing the research question, **"How can we utilize the existing deep learning techniques used for other languages to improve the existing Sinhala Text-to-Speech?"** which is the main research question that we are addressing through this research.

Accordingly, as sub research questions,

1. What are the current limitations of Text to Speech (TTS) systems, and how can they be improved to better mimic natural human speech?
2. How can machine learning and deep learning algorithms be used to enhance TTS systems, and what type of data is required for training these algorithms?
3. How can we finetune the identified approach to improve the quality of the Sinhala TTS?

have been discussed to provide the best solution for the identified problem.

1.3.Statement of Problem

While several TTS models exist for the English language, challenges persist when it comes to providing accurate and efficient solutions for low-resource languages like Sinhala. These limitations can hinder the accessibility and usability of TTS technology, particularly for the visually impaired members of the Sri Lankan community. Therefore, there is a pressing need to enhance Sinhala TTS systems to ensure greater accuracy and accessibility, ultimately benefiting those who rely on this technology within the community.

1.4.Research Aims and Objectives

The primary focus of the research is elaborated under the aims and objectives.

1.4.1 Aim

So, the primary aim of this research was to find and develop an optimized Text-to-Speech (TTS) system for the Sinhala language, aimed at enhancing accessibility for visually impaired individuals and enriching applications like chatbots.

1.4.2 Objectives

- Identify a suitable and novel DL approach for Sinhala Language.
- Identify suitable dataset with high quality data which can be used for DL.
- Identify gaps and limitation of the novel DL approach.

1.5.Scope of the Study

1.5.1. In Scope

The following will be covered under the scope of the research.

- **Deep Learning Model Implementation**: : This involves exploring deep learning frameworks, including Tacotron 2, VAENAR, and WaveNet, to capture and reproduce the distinctive phonetic and prosodic characteristics of the Sinhala language.
- **Dataset Compilation and Enhancement**: Curating a diverse and extensive dataset specific to Sinhala, which includes collecting high-quality speech recordings and corresponding text transcriptions to train the deep learning models effectively.

1.5.2. Out of Scope

Below areas do not cover under the scope of this project.

- This project is specifically tailored to enhancing Text-to-Speech (TTS) technology for the Sinhala language. It does not extend to improvements for other low-resource languages.

1.6. Structure of the Thesis

The first chapter of the dissertation outlines the background of the research area as well as research questions, aims, methodologies, and scope of the research. Chapter 2 of the dissertation critically reviews the research area and identifies the research gap clearly, further it discusses the importance of the problem using literature as well as possible avenues for solutions to the problem. Methods and tools that have been used in the proof of concept prototype are also included in Chapter 2. Chapter 3 focuses on research design. Further, it describes the evolution of the design and design choices made during the research process. Chapter 4 explains details about implementation and challenges that occurred during the implementation. Chapter 5 discusses the evaluation of the proposed model and the results. Finally, Chapter 6 provides discussions and conclusions about the positive and negative outcomes of the results as well as limitations and future avenues for the research.

CHAPTER 2 - LITERATURE REVIEW

This chapter would give essential background information referring to published material in research papers, magazine articles and similar literature related to the topic of Text-to-Speech Synthesis as well as to some of the specific tools selected to conduct this study.

TTS for English languages (Sasirekha & Chandra, 2012) has been able to achieve a high percentage of accuracy in conversion, but the TTS for Sinhala Languages is still lacking when taking the terms of accuracy. The plan is to address the accuracy issue of TTS systems for the Sinhala language through research. Existing TTS systems using machine learning and deep learning for the languages have been examined, and relevant research papers have been reviewed to gain insight and knowledge in the research area.

2.1.Machine Learning Literature

The research paper on "Text-to-Speech Synthesis System for Tamil Using HMM" (Jayakumari & Jalin, 2019) undertakes a comprehensive approach to develop a Tamil TTS system leveraging Hidden Markov Models (HMM). At the core of this study is the extraction and manipulation of speech features, pivotal for converting textual data into synthesized speech. The methodology begins with the utilization of Mel Frequency Cepstral Coefficients (MFCCs), a method known for its effectiveness in capturing the key acoustic properties of speech. These coefficients are essential for denoising and preparing the audio data for further processing. To enhance the speech signal, particularly its high-frequency components, a pre-emphasis filter is applied, which compensates for the natural attenuation of high frequencies during sound production. This step is crucial for achieving a balanced and clear speech signal.

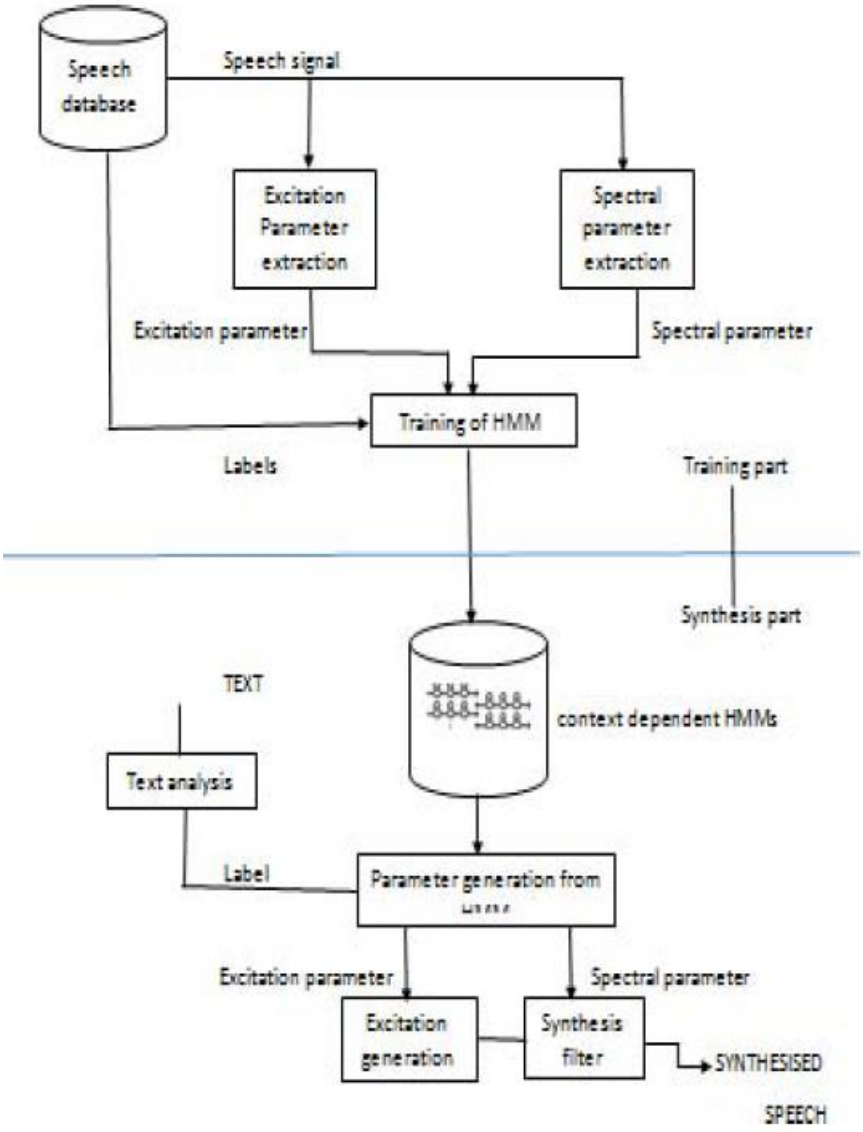


Figure 2.1 - Architecture of Tamil TTS with HMM

Following the initial processing, the methodology employs windowing techniques, segmenting the speech into manageable frames and applying a Hamming window to each. This reduces spectral distortion, ensuring that the analysis can accurately capture the nuances of the speech signal. The conversion of these time-domain signals into the frequency domain is accomplished through the Fast Fourier Transform (FFT), enabling the detailed analysis of the speech's frequency components. The application of the Mel filter bank further refines this process, emphasizing lower frequencies and smoothing the spectrum. Finally, the system is built using the festival framework, chosen for its language-independent capabilities and flexibility in integrating new modules for speech synthesis.

Despite the robustness of the methodologies and the use of advanced techniques, the paper acknowledges the challenges in fully capturing the naturalness of human speech. The research highlights the strides made in improving speech intelligibility and quality through these techniques, while also noting the ongoing need for enhancements to achieve truly lifelike speech synthesis in Tamil.

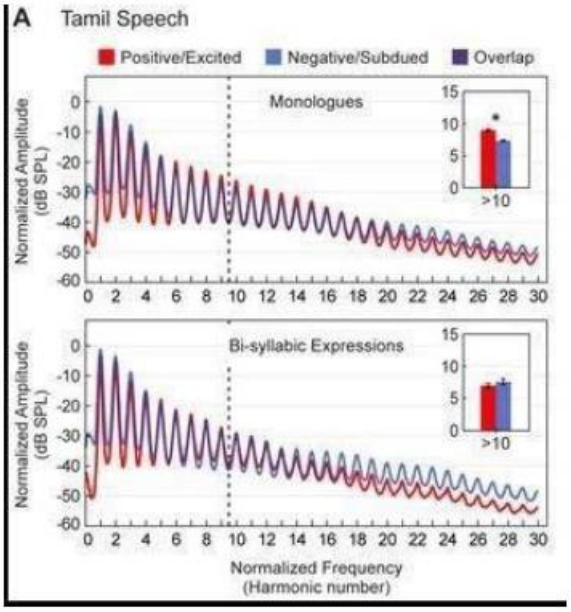


Figure 2.2- Tamil Speech Output

The paper "Text to Speech Synthesis System for Tamil" by (Sangeetha et al, 2013) outlines a comprehensive methodology for developing a corpus-driven Tamil Text-to-Speech (TTS) system, leveraging the concatenative synthesis approach. The authors focus on achieving naturalness and intelligibility, crucial qualities of synthesized speech, by using words and syllables as the basic synthesis units. The corpus consists of speech waveforms collected for frequently used words across various domains, with a speaker selected based on subjective and objective evaluations of both natural and synthesized waveforms. This meticulous approach ensures the synthesized speech closely resembles a natural human voice, highlighted by the system's utility in generating a high-quality Tamil text-to-speech WAV file.

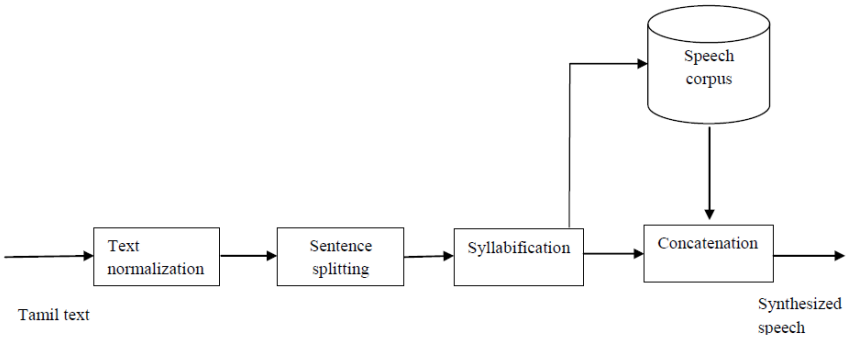


Figure 2.3- Tamil Text-to-Speech

The evaluation of the proposed system's output involved subjective tests, where human listeners ranked the quality of processed voice files on a Mean Opinion Score (MOS) scale ranging from 1 (Bad) to 5 (Excellent). This scale is a common method for assessing voice quality, providing a quantitative measure of the synthesized speech's intelligibility, naturalness, and overall quality. Tests conducted with a group of students in a laboratory environment yielded MOS scores that reflect the effectiveness of the system's methodology in producing intelligible and natural-sounding speech. The results, showing scores for sentences constructed with words both within and outside the speech corpus, demonstrate the system's capability and the success of the concatenative approach in preserving the naturalness and intelligibility of the synthesized speech.

A Human Quality Text to Speech System for Sinhala" (Nanayakkara et al. (2018)), This paper proposes an approach on implementing a Text to Speech system for Sinhala language using MaryTTS framework. In this project, a set of rules for mapping text to sound were identified and proceeded with Unit selection mechanism. The datasets used for this study were gathered from newspaper articles and the corresponding sentences were recorded by a professional speaker.

For this evaluation process they recorded the voice generated from the build Sinhala TTS for 15 selected sentences and then they facilitated to listen those pre-recorded 15 sentences for everyone in their testing sample and asked them to write down the sentence they can hear while ranking the speech quality and the naturalness of them according to the grey scale given. The responses were marked separately in the evaluation sheets and the analysis were made based on those responses. Based on the observation of re-written sentences the intelligibility of the Sinhala TTS system was measured, which was defined as follows,

$$Intelligibililty = Avarage(\sum_{n=1}^{20}(100\frac{X}{Y}))$$

Equation 1 - Intelligibility calculation

Where X = number of correctly identified words and Y = total number of words in the sentences Based on the results they calculate the quality of the TTS.

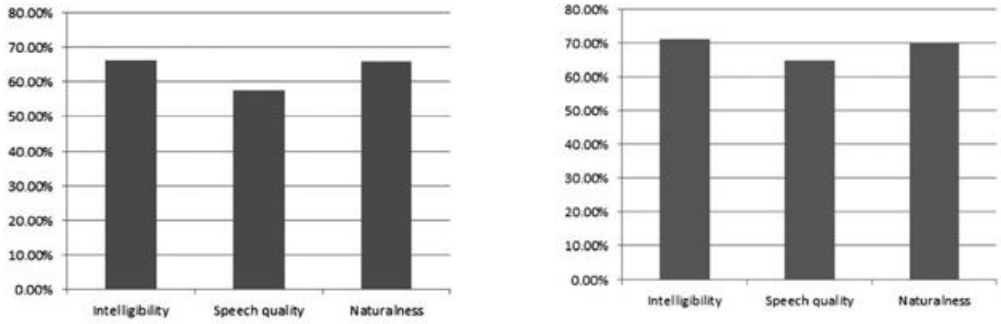


Figure 2.4 - Quality of the synthesis voice in visually impaired(left) and sited category (right)

They describe their outcomes performance in 3 aspects,

1. Intelligibility
2. Speech quality
3. Naturalness.

Also, they were able to be evaluated by the blind community and the sighted people separately. in Figure 2.4 we can see sited category has given higher values for the speech quality and the naturalness (70%) as well. May be the sited category have no previous experience on listening TTS voices, they may have heard the Sinhala TTS voice better than the visually impaired evaluators.

Festival-si: A Sinhala Text-to-Speech System (Weerasinghe, R. et al, 2007), In this research, they describe the implementation and evaluation of a Sinhala text-to-speech system based on the diphone concatenation approach. The festival framework was chosen for implementing the Sinhala TTS system. The festival-si was evaluated under intelligibility criteria since it's a general-purpose TTS tool which doesn't guarantee the naturalness of the speech. In this research the design of a diphone database and the natural language processing modules developed has been described.

They have used Modified Rhyme Test (MRT) which was designed to test TTS System. They have achieved 71.5%. According to the authors' knowledge, this is the only reported work in the literature describing the development of a Sinhala text-to-speech system, and

more importantly, the first Sinhala TTS system to be evaluated using the stringent Modified Rhyme Test.

2.2.Deep Learning Literature

TacoSi: A Sinhala Text to Speech System with Neural Networks (Kasthuri Arachchige, T.C. (2023)), In this research, A TTS (text-to-speech) system called TacoSi has been developed using an algorithm based on Tacotron, and it has been trained with pairs of raw text and audio. With raw text as input, TacoSi can produce speech in Sinhala that sounds like it was spoken by a human. Another advantage of TacoSi is that it can pronounce rare words that were not seen during training, and it can also comprehend common symbols, numerical values, and abbreviations used in written Sinhala. In here the model was trained using the "PathNirvana" dataset that is openly accessible. There are 3300 sentences in this collection totaling 7.5 hours of recordings.

In here, the text was preprocessed using methods that were used for training before synthesizing. The TensorFlow implementation of Griffin Lim (GRIFFIN, D. and LIM J, *IEEE*) algorithm is used to convert wav form from spectrograms. The synthesizing takes approximately 30 seconds on Collab environment which used for training.

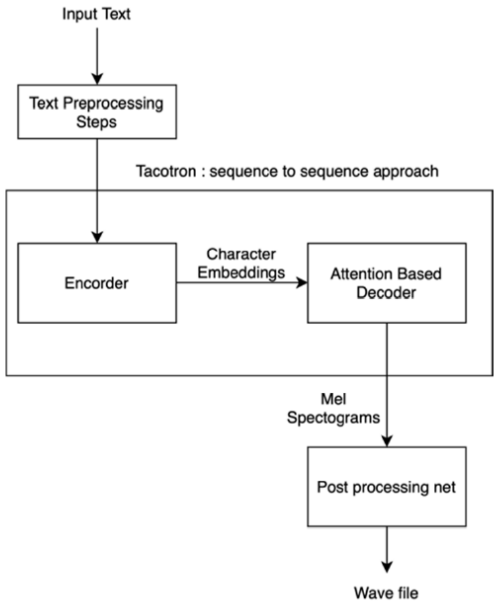


Figure 2.5 - TacoSi Training Algorithm

The base of the above training algorithm (Figure 2.5) is the Tacotron (Wang, Y. et al,

2017). The Tacotron was introduced by Google in 2017. Though Google haven't shared the source code of the implementation with the public yet, few individuals have made attempts to implement Tacotron described in the paper and published their source code with open-source licenses. Using the TensorFlow library the Tacotron model was implemented by referring to numerous online sources.To train the Tacotron model, a cloud environment (Google Colab) was used with upgraded RAM to 26.75 GB and Tesla T4 GPU.

They have evaluated using MOS Score (Vishwanathan, M, 2005;Nanayakkarara et al. (2018)) using below formula.

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N} \quad (1)$$

$$Inteligibility = \frac{\sum_{respondant=1}^{10} correct\ words}{(\sum_{sentence=1}^{10} \text{words}) * respontants} \quad (2)$$

Equation 2 - MOS and Intelligibility

The TacoSi obtained 4.39 MOS by equation (1) and obtained was able to achieve a 0.84% of intelligibility score according to equation (2). Also, we can see they have gained an improvement compared with work done by (Nanayakkarara et al. (2018))

Factor	Nanayakkarara et al. (2018)'s Stats	Evaluation of TacoSi
Intelligibility	70%	84%
Naturalness	70%	78.2%
Speech Quality	65%	86.4%

Table 2.1 - Comparison with previous work

Urdu Text-to-Speech Conversion Using Deep Learning (Saba, R. *et al.* (2022)), This Research aimed to create a system that takes Urdu textual content as input and then produces an audio version of the same textual content by using state-of-the-art techniques. In the proposed deep learning-based technique, Tacotron 2 with WaveGlow (Shen, J. et al. (2018)) is used. They trained on a preprocessed dataset before being tested on a dataset of 100 sentences.

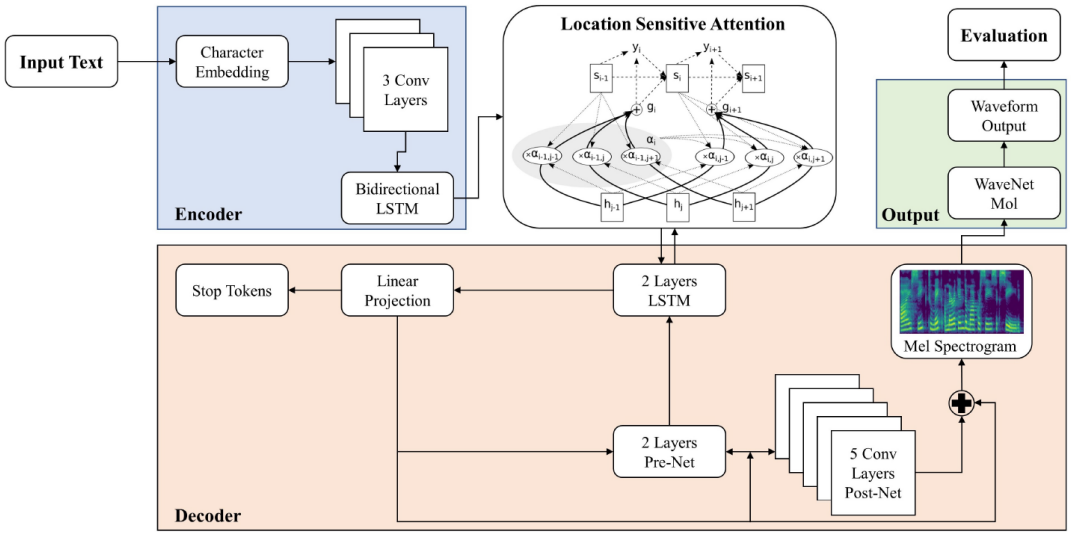


Figure 2.6 - Tacotron 2 architecture for Urdu TTS

The Urdu phonetic corpus is used in this study for training. The speech corpus consisted of the 70 minutes of reading speech and consisted of the 780 voraciously created sentences representing all the phonetics and tri-phonemic combination of the Urdu language. The input text is passed through an encoder which consist of three components (Mentioned in Figure 2.6), character embedding, three convolution layers and the Bidirectional LSTM. The output of the proposed system is generated in sound waves that sound waves are based on the Mel spectrogram. These sound waves are the speech/audio of the input text sequence.

The results are evaluated using the Mean Opinion Score (MOS), which is a standard performance evaluation measure in the TTS conversion domain. In this study fifty native Urdu language speakers are involved for evolution of output speech generated by Urdu TTS. The evaluation results of them show that the proposed approach outperformed the existing approaches, achieving a MOS of 3.76. Below table 2.2 shows the overall evaluation results against those evaluators.

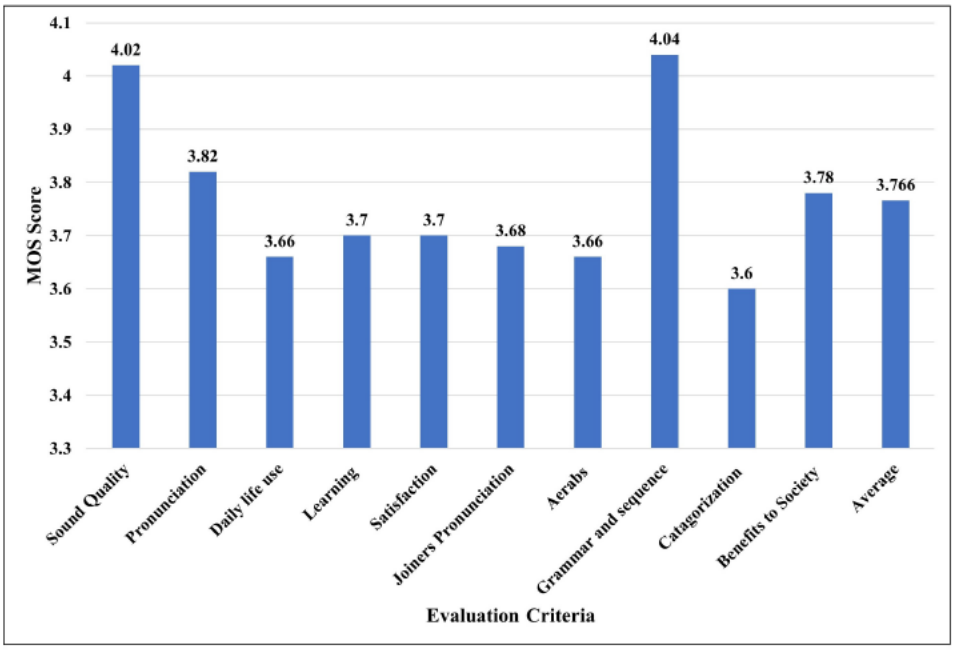


Table 2.2- Evaluation Results Based on Criteria

This study also compares the results of using Tacotron 2 with WaveGlow for Urdu TTS with some other languages that used the same technique. Below table 2.3 presents the comparison of MOS score for different languages.

Reference	Language		MOS Score	No of Evaluators
[26]	Japanese		4.13	15
[27]	Sanskrit		3.38	37
[28]	Mandarin		Naturalness 3.65, Prosody 3.86	5
[25]	English		3.88	20
[29]	Indian	Gujrati	4.41	8
		Tamil	3.54	19
This Study	**Urdu**		**3.76**	**50**

Table 2.3- Comparison of MOS Score for Different Languages

VAENAR-TTS: Variational Auto-Encoder based Non-Autoregressive Text-to-Speech Synthesis (Lu et al., 2021), In this research they have implemented a TTS with Non-Autoregressive model and achieved state-of-the-art synthesis quality. The autoregressive Text-to-Speech (AR-TTS) models produce high-quality speech but are slow due to sequential decoding. Non-autoregressive TTS (NAR-TTS) models offer faster parallel decoding but rely on complex phoneme-level durations for alignment, affecting naturalness. The proposed VAENAR-TTS model introduces an end-to-end solution,

eliminating the need for phoneme-level durations. Operating without recurrent structures, it uses a Variational Autoencoder (VAE) to encode alignment information in a latent variable. During decoding, attention-based soft alignment reconstructs the spectrogram. VAENAR-TTS achieves top-tier synthesis quality with a speed comparable to other NAR-TTS models, showcasing the effectiveness of VAE architecture in addressing TTS challenges.

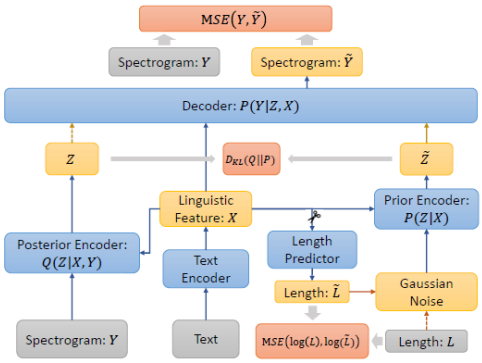


Figure 2.7 - Architecture of VAENAR-TTS

As shown in above Figure 2.7, It consists of a text encoder, a posterior encoder, a prior encoder, a length predictor, and a decoder. The text encoder aims to encode the raw character sequence into the context-aware linguistic feature X. Also, they have conducted MOS calculation for this model and they could achieve higher MOS values compared to other models specially with Tacotron 2 model which I have mentioned in earlier literature reviews. Those results are mentioned below in table 2.4.

Model	**MOS**	**RTF**(Sec)
Ground-Truth	4.56 ± 0.09	-
Hifi-GAN-Resyn	4.47 ± 0.10	-
Tacotron2	4.03 ± 0.12	1.35×10^{-1}
FastSpeech2	3.83 ± 0.14	$\mathbf{4.21 \times 10^{-3}}$
Glow-TTS	3.62 ± 0.13	9.39×10^{-3}
BVAE-TTS	3.16 ± 0.13	$\mathbf{4.21 \times 10^{-3}}$
VAENAR-TTS	$\mathbf{4.15 \pm 0.12}$	7.45×10^{-3}

Table 2.4 - Comparison between different TTS Models

This VAENAR-TTS implemented focusing on English language and another few low-level languages such as mandarin. So up to present day no one has try this approach for Sinhala language.

2.3.Sinhala G2P Approaches Based Literature

The paper "Sinhala G2P Conversion for Speech Processing" by Nadungodage et al. (2018). discusses the development of a rule-based method for converting Sinhala text strings into phonemic representations, addressing the challenge posed by the Sinhala writing system's lack of a direct correlation with its spoken form. The authors enhance an existing set of rules to achieve more accurate grapheme-to-phoneme (G2P) conversion, critical for speech processing applications. Their evaluation demonstrates the effectiveness of these sound pattern rules in improving the accuracy of Sinhala G2P conversion, highlighting the importance of rule-based approaches in processing under-resourced languages like Sinhala.

Consonant Character(s)	Pronunciation		Consonant Character(s)	Pronunciation
කබ/	k		ඳ	d̃
ගඝ/	g		පඵ/	p
ඞ/	ŋ		බභ/	b
ඟ	g̊		ම	m
චඡ/	c		ඹ	b̃
ජඣ/	ɟ		ය	j
ඤඳ/	ɲ		ර	r
ටඨ/	ʈ		ළළ/	l
ඩඪ/	ɖ		ව	v
ණ/න	n		ශෂ/	ʃ
ඬ	d̪		ස	s
තථ/	t		හඃ/	h
දධ/	d		ෆ	F

Figure 2.8 - G2P Mapping for Consonant Characters

Phoneme sequences	Diphthong	Example
/ivu/ /iv/	/iu/	‘කිවුවා’‘කිව්වා’
/i:vu/ /i:v/	/i:u/	‘රජිව්’
/evu/ /ev/	/eu/	‘පෙවුවා’‘පෙව්වා’
/e:vu/ /e:v/	/e:u/	‘සේව්’
/ævu/ /æv/	/æu/	‘නැවුවා’ ‘නැව්වා’
/æ:vu/ /æ:v/	/æ:u/	‘බැවුම්’ ‘ගැව්වා’
/ovu/ /ov/	/ou/	‘ඔවුන්’ ‘පොව්ව’
/avu/ /av/	/au/	‘කවුද’ ‘කව්පි’
/a:vu/ /a:v/	/a:u/	‘සාවුරුද්දක්’
/uyi/	/ui/	‘බතුයි’
/u:yi/	/u:i/	‘ළූයි’
/oyi/	/oi/	‘සුපොයි’
/o:yi/	/o:i/	‘රෝයි’
/ayi/	/ai/	‘කයි’
/a:yi/	/a:i/	‘මංයි’
/eyi/	/ei/	‘බැලෙයි’
/e:yi/	/e:i/	‘ගේයි’
/æyi/	/æi/	‘ඇයි’
/æ:yi/	/æ:i/	‘ඈයි’

Figure 2.9 - Sinhala Diphthongs Mapping with Examples

This strategy has proven effective, achieving an impressive approximate accuracy of 98% in their evaluations, underlining the potential of rule-based systems in handling the complexities of the Sinhala language for speech processing applications. So, I used a pretrained model of this to convert grapheme to phonetics during the preprocessing stage of my implementation.

All the above-mentioned studies focused on improving text-to-speech in various ways. If we consider the accuracy of the Sinhala TTS system, still we can improve output with deep learning techniques. So, the Study that will be conducted through this project will be focused on achieving that goal.

CHAPTER 3 - METHODOLOGY

3.1. Introduction

This chapter explains how research questions are addressed. Further, it describes the design choices and evolution of the model throughout the research and describes the design evolution and analysis of each design approach. Apart from that this chapter provides a higher-level architecture of the proposed approach and definitions and assumptions made during the research.

3.2. Design Evolution

3.2.1. Selecting a Proper Deep Learning Technique

The initial stage of my study involved a thorough review of Tacotron 2, the VAENAR approach, and additional significant deep learning research, aiming to identify opportunities for advancements in Sinhala Text-to-Speech (TTS) systems. This exploration was geared towards uncovering innovative techniques and methodologies that could potentially elevate the performance and naturalness of Sinhala TTS.

- One notable advancement explored is applying Tacotron 2 for Sinhala TTS, as discussed in the "Taco Si Research" by Kasthuri Arachchige, T.C. (2023). The study highlights the integration of the WaveNet architecture as a key enhancement, emphasizing the need for parameter optimization and the use of a multi-speaker dataset.
- Additionally, the VAENAR-TTS approach by Lu et al. (2021), though applied to English and Mandarin, has not yet been explored for Sinhala. The original VAENAR-TTS model revolutionizes English text-to-speech synthesis by integrating a Variational Auto-Encoder (VAE) within a non-autoregressive framework.

Given the above insights, it has led me to pursue the adaptation of the VAENAR approach for developing a Sinhala TTS system, addressing the identified gaps and leveraging the potential for enhanced synthesis quality and linguistic diversity.

3.2.1.1. Why VAENAR is a better approach compared to Tacotron 2?

The superiority of the VAENAR model over Tacotron 2 in synthesizing Sinhala TTS is substantiated by its innovative approach and the empirical evidence presented in the literature review section. As outlined in "VAENAR-TTS: Variational Auto-Encoder based Non-Autoregressive Text-to-Speech Synthesis" by Lu et al. (2021), the VAENAR model demonstrates a marked improvement in Mean Opinion Score (MOS) compared to Tacotron 2's implementation for English. This benchmark serves as a compelling justification for adopting the VAENAR model in my thesis, particularly for enhancing Sinhala TTS systems. This makes VAENAR particularly suited for languages like Sinhala, where tonal nuances and speech variability are important for naturalness.

3.2.2. Research High level Design

The high-level design for the Sinhala TTS system includes several preprocessing steps as well as core processes such as model training and audio synthesis. The key components of the research design, which are aimed at developing a Sinhala TTS system, are illustrated in Figure 3.1. This design outlines the workflow from the initial input of paired audio-text data, which is stored in a dataset repository, through to the cleaning and normalization of text data. It then details the extraction of mel-spectrograms from the audio data, which is a crucial step for feature representation. Subsequently, the model is trained using the Variational Autoencoder Non-Attentive Tacotron (VAENAR) architecture. The final stage involves the generation of synthesized speech in the form of output audio in WAV format, using the trained model and a vocoder, from new Sinhala text inputs.

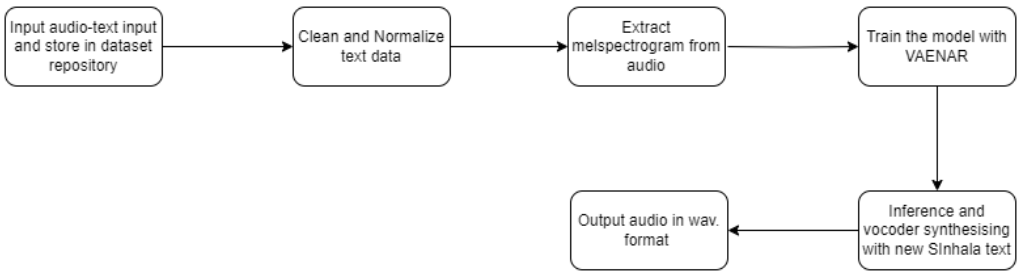


Figure 3.1 - High Level Research Design

3.2.3. Selecting a Proper Sinhala Dataset

Dataset - *Path Nirvana Sinhala TTS Dataset [5]*

This is a publicly available High Quality Multi Speaker Sinhala dataset for Text to speech algorithm training specially designed for *deep learning algorithms*. As I mentioned in section 2 this dataset was used in the "Taco Si Research" by Kasthuri Arachchige, T.C. (2023 as well.

Stats

- Number of Recordings: 6248
- Total Length: 13.7 hours
- Maximum Length: 15 seconds
- Minimum Length: 2 seconds
- Number of Unique Characters: 54 roman
- List of Roman Characters:
  !'(),.:;=?abcdefghijklmnoprstuvyæñāēīōśşūǣḍḥḷṁṅṇn̲ṛṟṭ
- Silences have been removed from both the beginning and the end of the recordings.
- Sample Rate 22050Hz and 16-bit PCM encoded similar to the *ljspeech dataset*.

Credibility of above-mentioned dataset:

- **Substantial Size and Length**: With 6248 sentences and 13.8 hours of recordings, this dataset provides a comprehensive and substantial collection of Sinhala language data. The length of the recordings ensures a diverse and representative dataset for training text-to-speech algorithms.
- **Multi-Speaker Representation:** Featuring two distinct speakers, Ven. Mettananda and Mrs. Oshadi, the dataset encompasses a variety of voice characteristics. This multi-speaker approach enhances the dataset's diversity, making it more adaptable for training algorithms to handle different speech styles and nuances.
- **Capture of Rarely Used Syllables**: An effort has been made to capture rarely used syllables in the Sinhala language, particularly those with Sanskrit and Pali origins. This attention to detail increases the dataset's richness, addressing potential challenges in synthesizing less common linguistic elements.

- **Detailed Documentation**: The dataset provides information about the origin of recordings, specifying that they were done during the second quarter of 2023. Additionally, acknowledgment of potential errors and an invitation for contributions to the repository demonstrate transparency and a willingness to improve the dataset.

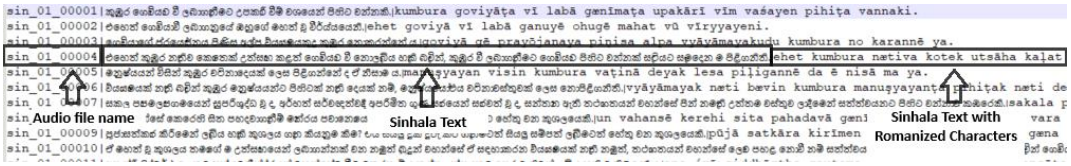


Figure 3.2 - audio text mapping in dataset

In summary, the Path Nirvana Sinhala TTS Dataset establishes credibility through its substantial size, multi-speaker representation, attention to rarely used syllables, detailed documentation, and an open contribution model. These factors collectively contribute to the dataset's reliability for training deep learning algorithms in the domain of Sinhala text-to-speech.

3.2.4. Preprocessing Phase

For adapting VAENAR to the Sinhala language, the preprocessing phase requires several key changes after identifying a suitable Sinhala dataset. First, it's essential to transcribe the dataset into phonetic representations that accurately reflect Sinhala's phonemic nuances.

During the preprocessing phase, audio files were transformed into linear and mel spectrograms using the *librosa*, *numpy* and *scipy* libraries. These spectrograms were stored as numpy array files (.npy format), and a metadata file (metadata.csv) was created linking the spectrograms to their corresponding textual utterances. To enhance system performance, specific text normalization methods for Sinhala were implemented, addressing the unique requirements of Sinhala TTS. To convert Sinhala graphemes to phonemes (G2P), I utilized a pre-trained model from the paper "Sinhala G2P Conversion for Speech Processing" by Nadungodage et al. (2018), as mentioned in the literature review section 2.3. Figure 3.3 illustrates how this G2P dictionary was integrated into the preprocessing logic.

```
def grapheme_to_phoneme(text, g2p, lexicon=None):
    """Converts grapheme to phoneme"""
    phones = []
    words = filter(None, re.split(r"([',:;.\-\?\!\s+])", text))
    for w in words:
        if lexicon is not None and w.lower() in lexicon:
            phones += lexicon[w.lower()]
        else:
            phones += list(filter(lambda p: p != " ", g2p(w)))
    return phones
```

LEXICONS FROM DICTIONARY

Figure 3.3 - Mapping Sinhala phoneme from dataset

3.2.5. Training Phase

The VAENAR-TTS model's training phase for Sinhala language adaptation involves a sophisticated architecture comprising multiple encoders, a length predictor, and a decoder. Initially, the text encoder processes raw Sinhala character sequences into context-aware linguistic features. The model employs both a prior and posterior encoder to handle the distributions of latent variables based on these features and the given spectrograms, enhancing the model's ability to capture the nuances of Sinhala speech. The length predictor plays a crucial role in determining utterance durations, vital for maintaining natural speech flow in the Sinhala language.

During training, the VAENAR-TTS model leverages a loss function that includes mean squared error (MSE) for spectrogram prediction accuracy and KL-divergence to minimize the difference between the prior and posterior distributions. This approach ensures the Sinhala TTS system not only generates high-quality speech but also maintains linguistic accuracy. The decoder's use of Transformer blocks aligns the linguistic features with the latent variables, crucial for synthesizing natural-sounding Sinhala speech. This phase is pivotal in refining the model's capability to produce clear, natural Sinhala speech, setting a foundation for high-quality TTS applications.

Loss Function:

$$L = \mathbf{MSE}(Y, \tilde{Y}) + \alpha \mathbf{D_{KL}}(Q(Z|X,Y)||P(Z|X)) + \beta \mathbf{MSE}(\log(L), \log(\tilde{L})),$$

Equation 3 - Formula for Equation calculation

- MSE (Y, Ý): This is the mean squared error between the predicted spectrogram, Ý and the actual spectrogram Y, indicating the fidelity of the synthesized audio to the true audio.
- αD_{KL} (Q (Z|X, Y) ||P(Z|X): This is the KL divergence, weighted by α, quantifying the difference

between the predicted posterior distribution Q(Z|X,Y) of the latent variable Z, based on the linguistic features X and spectrogram Y, and the prior distribution P(Z|X) of Z based solely on X.

- βMSE (log(L), log(Ĺ)): This term, weighted by β, represents the mean squared error between the logarithm of the predicted and actual utterance-level durations, log(Ĺ) and log(L) respectively, ensuring accurate prediction of speech timing.
- Overall, the hyperparameters α and β balance the contribution of each part in the loss function to optimize both acoustic and temporal aspects of speech synthesis.

3.2.4.1. Alignment Learning in VAENAR approach.

In the realm of Text-to-Speech synthesis, achieving precise alignment between linguistic features and the corresponding spectrogram is crucial for generating high-quality speech. The VAENAR-TTS model innovates by learning this alignment without relying on autoregressive components. It employs an annealing reduction factor which, by initially simplifying alignment via shortened sequences and subsequently refining detail as training progresses, facilitates both early learning and nuanced detail capture. Additionally, the model incorporates a causality mask within the self-attention mechanism, enhancing temporal feature focus and reducing repetition errors, thus bolstering the alignment's accuracy.

3.2.6. Environment Setup

In adapting the VAENAR-TTS model for Sinhala Text-to-Speech synthesis, I refined the optimization process using the Adam optimizer. Initially setting the KL-divergence weight at a precise 0.00001 and facing initial KL losses between 300-400, I incrementally adjusted this value during the training. By adjusting to values of 0.1 and 1, I managed to significantly mitigate the loss as mentioned in below figure 3.4.

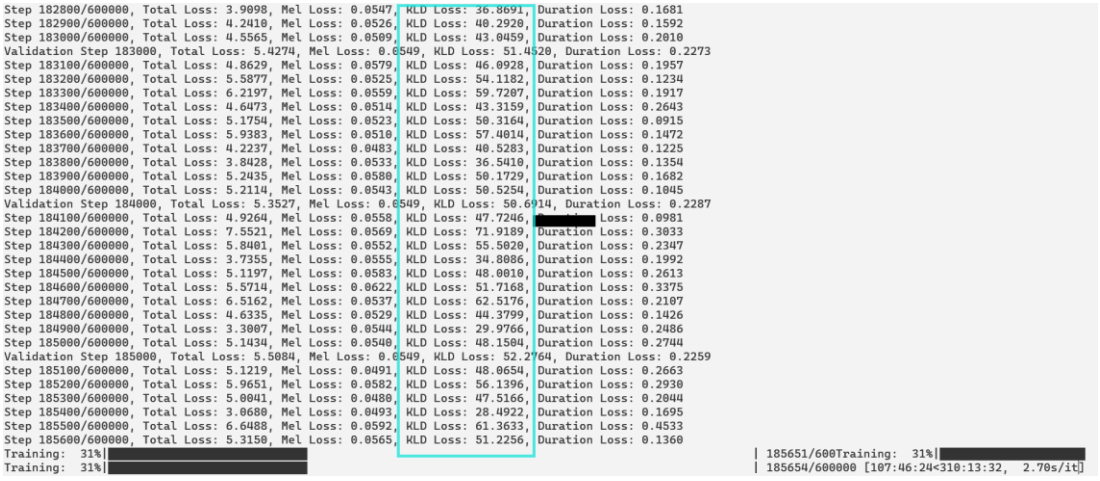


Figure 3.4 - KL Loss Reduction During Training

The reduction factor, r, is initially set at 5 to simplify the early stages of alignment learning and is methodically decreased by 1 every 150 epochs. This decrease continues until r reaches the value of 2, beyond which it is held constant, allowing the model to refine its understanding of the finer details in the later stages of training. The training proceeds for a total of 2000 epochs to thoroughly embed the complexities of Sinhala speech patterns into the model. This model was trained using varies GPU resources such as *AWS G instance, UCSC Deep learning box, google colab pro A100 GPU with higher RAM, GCP VM instances*.

CHAPTER 4 - IMPLEMENTATION

4.1. Introduction

This chapter explains the implementation of the proposed model. Further this will describe the evolution of the Proof-of-Concept implementation and it describes the technologies and software's used in the implementation and analysis of those tools.

4.2. Dataset alignment and preprocessing techniques

The preliminary stage of Sinhala Text-to-Speech synthesis development, logic needed to be implemented for reading the metadata of the Sinhala dataset. The decision had to be made between utilizing Sinhala characters directly or opting for Romanized characters. Due to the compatibility of the g2p model with Sinhala, the choice to use Romanized characters was made more straightforward, given their similarity to English characters. Furthermore, a text cleaning function was developed to ensure the text was prepared in a normalized form. For further enhancement of text normalization, such as handling abbreviations, the "re" library was utilized, allowing for the refinement of input text for the TTS system.

For audio processing, 'librosa' was used along with the 'scipy.io' module to manage WAV file operations. The process was initiated by reading a metadata CSV file, structured similarly to the LJ Speech Dataset, which included crucial mappings of audio file names to their textual utterances. The WAV files corresponding to these mappings were then processed; they underwent resampling to the required rate and were normalized to a consistent volume level, as dictated by the maximum WAV value specified in the configuration. This preparatory work ensured that the raw data was in an ideal state for alignment and further processing, laying a solid foundation for the TTS model's robust performance.

In the second stage of the implementation, a comprehensive preprocessing pipeline was developed, laying the groundwork for the Sinhala TTS system. The Python-based framework utilized libraries such as *tgt*, *librosa*, and *numpy*, with a focus on extracting and processing acoustic features from the raw audio data. This involved a process where each audio file underwent normalization, feature extraction, and a transformation from graphemes to phonemes using a pre trained Sinhala g2p dictionary done by

Nadungodage et al. (2018). The preprocessing also strategically handled data augmentation, speaker identification, and the division of data into training and validation sets to ensure robust model training.

```
# Write metadata
with open(os.path.join(self.out_dir, "train.txt"), "w", encoding="utf-8") as f:
    for m in out[self.val_size_:]:
        f.write(m + "\n")
with open(os.path.join(self.out_dir, "val.txt"), "w", encoding="utf-8") as f:
    for m in out[: self.val_size]:
        f.write(m + "\n")

return out
```

Figure 4.1 - Process of dividing data into validation and training

The output of this stage was a structured dataset, with metadata annotations facilitating subsequent modeling phases.

```
def process_utterance(self, speaker, basename, clip_norm=False):
    wav_path = os.path.join(self.in_dir, speaker, "{}.wav".format(basename))
    text_path = os.path.join(self.in_dir, speaker, "{}.lab".format(basename))

    # Compute mel-scale spectrogram from raw audio
    mel_spectrogram = self.audio_processor.get_mel_from_wav(wav_path, clip_norm=clip_norm)

    # Read raw text
    with open(text_path, "r", encoding="utf-8") as f:
        raw_text = f.readline().strip("\n")
    # Get phoneme
    phone = grapheme_to_phoneme(raw_text, self.g2p, self.lexicon)
    text = "{" + " ".join(phone) + "}"

    # Save files
    mel_filename = "{}-mel-{}.npy".format(speaker, basename)
    np.save(
        os.path.join(self.out_dir, "mel", mel_filename),
        mel_spectrogram.T,
    )

    return (
        "|".join([basename, speaker, text, raw_text]),
        mel_spectrogram.shape[1],
    )
```

Figure 4.2 - Method for preparing mel spectrogram and npy arrays.

Continuing from the initial setup of the preprocessing framework, the project advances into a next phase where individual audio files and their corresponding textual information are prepared. At this juncture, each audio file is transformed into a mel-spectrogram (Figure 4.2), which is an essential step as it converts the complex auditory data into a visual and quantifiable format that deep learning models can utilize effectively.

The outcome of this phase is a well-organized dataset, where each piece of processed data is carefully annotated with relevant metadata. This dataset is tailor-made to ensure seamless integration into the machine learning pipeline that follows. The preparation

conducted in this stage embodies the critical groundwork necessary for enabling the sophisticated algorithms of the TTS system to generate speech that closely mimics natural human intonation and rhythm.

This thorough and detailed preprocessing sets a strong foundation for the project, ensuring that the nuances of the Sinhala language are captured and represented, ready for the intricate process of speech synthesis. The quality of synthesized speech relies heavily on the precision and meticulousness of this stage, highlighting its importance in the overall success of the TTS system's development.

4.3. Preparing VAENAR Model for training

For the preparation, we built two main parts of our system using ***PyTorch***, which is great for developing complex algorithms that can learn from data. These two parts are the Transformer Encoder and Decoder. The Encoder's job is to read the input text and understand the context of each word and how each word relates to the others. It turns this understanding into a form that the computer can work with to produce speech. Then, the Decoder takes over. It uses the Encoder's output and additional information to generate the actual speech sounds, represented as mel-spectrograms.

One of the interesting parts of this implementation is its ability to pay attention to different parts of the sentence as it generates speech (self-attention mechanisms). This means it can focus more on certain words or sounds as needed, making the speech output more natural and similar to how humans speak.

```
class SelfAttentionBlock(nn.Module):
    def __init__(self, input_dim, attention_dim, attention_heads, attention_temperature,
                 ffn_hidden):
        super(SelfAttentionBlock, self).__init__()
        self.input_dim = input_dim
        self.attention_dim = attention_dim
        self.attention = MultiHeadScaledProductAttention(attention_dim=attention_dim,
                                                         input_dim=input_dim,
                                                         memory_dim=input_dim,
                                                         num_head=attention_heads,
                                                         temperature=attention_temperature)
        self.att_proj = LinearNorm(attention_dim + input_dim, input_dim)
        self.layer_norm = nn.LayerNorm(input_dim)
        self.ffn = FFN(in_features=input_dim, hidden1=ffn_hidden, hidden2=input_dim)

    def forward(self, inputs, memory, query_lengths, memory_lengths, causality=None):
        att_outs, alignments = self.attention(inputs=inputs, memory=memory,
                                              query_lengths=query_lengths,
                                              memory_lengths=memory_lengths,
                                              causality=causality)
        contexts = torch.cat([inputs, att_outs], dim=-1)
        att_outs = self.att_proj(contexts)
        att_outs = self.layer_norm(inputs + att_outs)
        ffn_outs = self.ffn(att_outs)
        return ffn_outs, alignments
```

Figure 4.3- Self attention handling code block

But there's more to speech than just the words and their order. The way someone says something like their tone or emotion, can change the meaning. To capture these subtleties, we introduced what are called latent variables. Think of these as hidden factors that influence how the speech sounds, such as pitch and emotion, which aren't directly mentioned in the text. We use two special networks, one focusing on the actual speech input and the other on the text input, to model these variables. This setup could have helped our system understand and generate the rich variety of human speech, even from just text.

By sampling these latent variables in a specific way, our system can produce diverse and realistic speech sounds. This means that even with the same text, VAENAR can generate speech that sounds slightly different each time, much like how people might say the same sentence in various ways depending on their mood or context. By this part We can bridge the gap between written Sinhala text and spoken language, making it possible for machines to produce speech that feels natural and lifelike. By understanding the context of words, focusing on different parts of the text, and capturing the subtleties of speech, we're making digital content more accessible and engaging for Sinhala speakers.

To ensure the speech not only flowed naturally but also resonated with the clarity and richness of human conversation, we implemented a dual-quality check. One part reviews the speech's fidelity, ensuring the generated sounds are a faithful representation of human speech, while the other ensures the model's learned patterns are meaningful and varied. This dual approach helps in refining the speech output to sound more lifelike and engaging.

4.4. Training Stage of the model

In the process of training our Sinhala Text-to-Speech model, we adopted a detailed and iterative method. The model's task was to learn the conversion of text into speech. Throughout this process, we conducted evaluations using specific criteria to ensure the model was processing information correctly (such as KL divergence).

```
if step % log_step == 0:
    losses = [l.item() for l in list([total_loss, mel_l2, kl_divergence, length_l2])]
    message1 = "Step {}/{}, ".format(step, total_step)
    message2 = "Total Loss: {:.4f}, Mel Loss: {:.4f}, KLD Loss: {:.4f}, Duration Loss: {:.4f}".format(
        *losses
    )

    with open(os.path.join(train_log_path, "log.txt"), "a") as f:
        f.write(message1 + message2 + "\n")

    outer_bar.write(message1 + message2)

    log(train_logger, step, losses=losses, kl_weight=kl_weight)
```

Figure 4.4 - Training Monitoring

To facilitate learning, we included steps for the model to adjust its approach based on feedback, akin to fine-tuning. Regular checkpoints were made to gauge the model's ability to produce speech and to provide a recovery point for any interruptions in training.

Adjustments to the model's parameters were made as needed, with the goal of improving its grasp of speech nuances. This was part of an ongoing effort to organize and manage the data effectively.

This training regimen was designed with the intention of achieving a model capable of generating natural-sounding speech. However, it's important to note that the final output did not meet the intended goal.

4.5. Handling Synthesize Stage

In developing the Sinhala Text-to-Speech system, we carefully designed how the program handles Sinhala text to create spoken words. We used two different approaches for handling the text. One was for dealing with lots of sentences at once (batch processing), which is like preparing a big meal by chopping all the vegetables at the same time it's more efficient. The other was for taking it one sentence at a time (single processing), which is more like focusing on making just one dish taste perfect by carefully adding each spice.

```
# Check source texts
if args.mode == "batch":
    assert args.source is not None and args.text is None
if args.mode == "single":
    assert args.source is None and args.text is not None

# Read Config
preprocess_config = yaml.load(
    open(args.preprocess_config, "r"), Loader=yaml.FullLoader
)
model_config = yaml.load(open(args.model_config, "r"), Loader=yaml.FullLoader)
train_config = yaml.load(open(args.train_config, "r"), Loader=yaml.FullLoader)
configs = (preprocess_config, model_config, train_config)
audio_processor = Audio(preprocess_config)

# Get model
model = get_model(args, configs, device, train=False)
```

Figure 4.5 - Handling Batch and Single Texts

For the batch approach, we gathered lots of Sinhala texts and organized them neatly so that our system could work on many sentences simultaneously, saving time. For single sentences, we put extra care into making sure each word was ready to be spoken by the system, paying close attention to the specific sounds of the Sinhala language.

The main part of turning text into speech happened in a function we designed, which was supposed to take our prepared text and turn it into a visual pattern representing how the speech should sound. Another part of the system would then take this pattern and try to turn it into actual sound.

We also tried to add some variety to how the speech sounded so it wouldn't be too repetitive or robotic. We hoped this would make our system's speech sound more like a real person talking, with all the natural ups and downs in the voice.

Lastly, we have used a vocoder that take our patterns and turn them back into clear, understandable speech. The goal was to make sure that when someone listened to the text that had been spoken by our system, it would be as clear as reading the text themselves.

Despite adhering to the proven VAENAR research approach that showed promise in English, we must acknowledge that the final output for the Sinhala Text-to-Speech system did not live up to our expectations. The system was unable to produce clear and natural-sounding speech. This underscores the intricacy of TTS technology and the specificity required when adapting methodologies across languages with different phonetic and linguistic complexities. Although we mirrored the strategies that were successful for English, the results for Sinhala were not in quality.

4.6.Summary of the chapter

In summary, the implementation chapter detailed the development process of the Sinhala TTS system, explaining the use of advanced neural network architectures. It highlighted the systematic approach taken to preprocess text, model acoustic features, and attempt synthesis of speech from textual inputs. Despite the sophisticated integration of various stages and the potential shown by deep learning technologies in this domain, the project encountered significant challenges in realizing high-quality speech output for the Sinhala language.

CHAPTER 5 - EVALUATION AND FINDINGS

5.1. Introduction

Results and evaluation chapter describes the complete evaluation of the research and the results obtained. This chapter describes the evaluation of the implemented Sinhala TTS System, comparison with previous English TTS implemented with same VAENAR approach. This chapter also explains the reasons for the drawbacks of the application identified in the evaluation phase.

In the evaluation of the implemented Sinhala Text-to-Speech (TTS) system using the VAENAR approach, an attempt was made to replicate the methodology successfully applied to English, incorporating adjustments to cater to the Sinhala language's unique characteristics. Despite these efforts and the theoretical soundness of applying the VAENAR model across languages, the evaluation revealed a significant challenge: the approach struggled to produce accurate and intelligible speech outputs for Sinhala. This outcome points to an inherent incompatibility or limitation of the model when tasked with synthesizing speech for a language as structurally and phonetically distinct as Sinhala.

The assessment involved a detailed comparison between the outcomes for Sinhala and English, using the same VAENAR framework. Surprisingly, similar patterns of limitations were observed in both languages, indicating that the issues might not solely lie with the language-specific adaptations but potentially with the underlying model’s ability to handle the complexity of natural speech patterns in diverse linguistic contexts. Despite utilizing a wide range of training resources aimed at enhancing model performance, the synthesized speech failed to meet the expectations of naturalness and intelligibility across multiple trials and varied batch sizes.

5.2. Result of Implemented Sinhala TTS

Below table showing the outcome against different batch sizes and different resources.

Batch size	Used Resource	Duration for Training (Estimated)	Outcome
8	UCSC ant pc	2 days (200,000steps)	Completed the training
	AWS G Instance (g4dn.2x.large)	7 days (200,000steps)	
	Local machine (32GB Ram, GTX 1650 Ti)	13 days (200,000steps)	
16	UCSC ant pc	5 days (150,000steps)	Completed the training
	AWS G Instance (g4dn.2x.large)	12 days (150,000steps)	Didn't execute due to cost
	Local machine (32GB Ram, GTX 1650 Ti)	15 days (150,000steps)	Got CUDA out of memory error and timed out at 6200 steps.
32	UCSC ant pc	14 days (550,000steps)	Got CUDA out of memory error and timed out at 17100 steps.
	AWS G Instance (g4dn.2x.large)	4 weeks (550,000steps)	Task automatically got killed due to high gpu usage
	GCP VM instance with different gpus (Nvidia L4, Nvidia T4, Nvidia tesla P4)	~14 days (550,000 steps)	Got CUDA out of memory error and timed out in the range of 10,000-20,000 steps
	GCP VM instance with highest gpus (Nvidia A100 80GB, Nvidia tesla P100)	~10 days (550,000 steps)	Ran this for only 2 to 3 hours due to high cost.

Table 5.1 - Evaluation training results of Sinhala TTS

5.3. Result of Previous English TTS

Below table 5.2 showing the outcome against different batch sizes and different resources I have done for English TTS.

Batch size	Used Resource	Duration for Training (Estimated)	Outcome
8	UCSC ant pc	2 days (200,000steps)	Completed the training
	Local machine (32GB Ram, GTX 1650 Ti)	13 days (200,000steps)	
32	UCSC ant pc	12 days (550,000steps)	Got CUDA out of memory error
	AWS G Instance (g4dn.2x.large)	4 weeks (550,000steps)	Task automatically got killed due to high gpu usage
	GCP VM instance with different gpus (Nvidia L4, Nvidia T4, Nvidia tesla P4)	~14 days (550,000 steps)	Got CUDA out of memory error and timed out in the range of 10,000-20,000 steps

Table 5.2 - Evaluation training results of English TTS

5.4. Comparison of English and Sinhala TTS.

So, as I mentioned in above training results it takes higher computation power to train this VAENAR model. However, I could complete the training with batch size 8. So, I have compared the two models with that. This provided a common ground for comparing the performance of the two models, both trained with 200,000 steps.

For the Sinhala VAENAR model, the training, while computationally intensive, demonstrated the feasibility of model completion within the constraints of available resources. However, when synthesizing simple sentences in Sinhala and English using the trained models, the resulting audio files consisted merely of humming sounds without any discernible speech. This indicates a fundamental issue with the model's ability to generate intelligible voice output from the training it underwent.

Despite encountering challenges with the synthesized speech output, efforts were made to consult with researchers who had previously implemented the VAENAR model for English. I reached out to them through emails and LinkedIn messages, seeking guidance or insights that might address the issues encountered. Regrettably, these attempts to connect did not yield a response. Documentation of this correspondence has been included in Appendix B of this thesis, providing transparency to the efforts made to seek expert advice and potentially improve the Sinhala TTS system's performance.

5.5. Summary of the chapter

The training outcomes for the Sinhala VAENAR TTS model indicate potential problems with the VAENAR approach itself. Issues may lie in various areas, including how the model is set up, the quality of the training data, or challenges such as overfitting or underfitting, where the model learns too much or too little from the data, affecting its ability to perform well. The resulting unintelligible speech output from the model suggests a significant hurdle: it was able to learn from the training phase but could not effectively use that learning to generate clear and accurate speech. This reflects a deeper issue with the VAENAR approach that may require reevaluation for its application in synthesizing Sinhala speech.

CHAPTER 6 - CONCLUSION

6.1. Introduction

This chapter focuses on the conclusions drawn upon the completion of the research. The research aim stated in Section 1.4 has been accomplished by using the technologies and approaches that are mentioned in 4th chapter and methodologies that are mentioned in 3rd Chapter.

6.2. Conclusions about Research Questions

6.2.1. Reflections of Main Research Question

In addressing the central research question, *"How can we utilize the existing deep learning techniques used for other languages to improve the existing Sinhala Text-to-Speech?"*, several insights have been gleaned from the evaluations conducted on the Sinhala VAENAR TTS system. The journey to adapt and apply this approach, proven effective when we look at the theoretical aspect. But when I tried it practically it seems it does not work for English as well.

The evaluation of the Sinhala TTS system, particularly through the VAENAR approach, has revealed critical issues and significant challenges, indicating that solutions previously effective for English also proved ineffective in this context. Despite employing a theoretically robust model and utilizing substantial training resources, the output failed to produce intelligible speech in both Sinhala and English. This outcome suggests that while deep learning techniques hold universal potential, their successful application requires careful adjustment to accommodate the linguistic complexities of each target language.

The issues encountered during the project, ranging from the computational limitations reflected in the inability to train with larger batch sizes without encountering memory errors, to the absence of coherent speech output, underscore the complexity of TTS systems. They highlight that deep learning models are not one-size-fits-all solutions and that improvements to TTS systems for languages like Sinhala must consider the language's unique attributes from the outset.

Furthermore, the lack of engagement from the broader research community, which

possesses knowledge about VAENAR that could be beneficial to others seeking assistance, suggests a need for increased collaborative efforts in the field, especially for low-resource languages. This situation underscores a wider implication for future research: to enhance TTS systems for languages like Sinhala, communities specializing in AI and linguistics must collaborate more closely to share knowledge, resources, and innovations.

6.2.2. Reflections of Sub Research Question 1

Addressing the sub-research question regarding *the limitations of Text to Speech (TTS) systems and their improvement for mimicking natural human speech*, In addressing the sub-research question related to the limitations of Text-to-Speech (TTS) systems and their capacity for improving the mimicry of natural human speech, my exploration and implementation with the VAENAR approach for Sinhala provided insightful outcomes, despite not achieving the intended success. The evaluation of the Sinhala VAENAR-TTS system unveiled significant hurdles, notably in generating intelligible speech, which was a critical aspect of this research endeavor.

The results from deploying the VAENAR model for Sinhala revealed a crucial limitation: the inability to produce clear and natural-sounding speech. This was an unexpected outcome, given the model's prior success with languages like English. The synthesized speech, rather than being intelligible, resulted in sounds lacking coherent linguistic content, a stark deviation from the anticipated improvement in natural speech mimicry.

In response to these findings, several proactive steps were undertaken to navigate these challenges. Recognizing the complexity of the task and the potential for the VAENAR model to contribute to the field of TTS for low-resource languages like Sinhala, I reached out to the broader research community for insights and guidance. This involved attempts to contact researchers with expertise in VAENAR through emails and LinkedIn, seeking advice that might illuminate paths to overcoming the observed limitations. Despite these efforts, the lack of response from the community underscored a broader challenge in collaborative engagement and knowledge sharing, particularly for advancing TTS technologies in less commonly spoken languages.

The encounter with these limitations and the subsequent outreach to the research community reflect a comprehensive approach to addressing the sub-research question. They highlight not only the technical and linguistic obstacles inherent in adapting deep learning TTS models to new languages but also the importance of collaboration and open dialogue within the scientific community. While the immediate outcomes may not have aligned with the initial objectives, the process undertaken to address these challenges contributes valuable insights to the field, emphasizing the need for continued research, innovation, and cooperation to enhance the capabilities of TTS systems for all languages.

6.2.3. Reflection of Sub Research Question 2

Addressing the sub-research question on *the utilization of machine learning and deep learning algorithms to enhance TTS systems*, my work with the VAENAR model for Sinhala taught me a lot. I attempted to apply techniques known to work for languages like English to enhance Sinhala TTS, aiming to make the voice sound more natural and realistic.

However, I learned that using these advanced computer techniques isn't straightforward. For Sinhala, a language that doesn't have as many resources as English, there were big challenges. One major issue was not having enough good data to teach the model how to mimic human speech properly. This is really important because the model needs to learn from a lot of examples to get good at its job.

Also, trying to apply a model that worked for one language directly to another showed me how tricky it can be. Every language is different and has its own rules, which means a model that works for English might not work the same way for Sinhala without some changes.

Because of these challenges, the model didn't produce the clear and understandable speech I was hoping for. But trying to solve these problems taught me that we need more focused work. We need to collect better data for Sinhala and maybe change the model to fit the language better.

Even though I didn't get the results I wanted, this project shows how important it is to

keep trying different ways to improve TTS systems, especially for languages that don't have as much support. It also shows we need to work together more, sharing ideas and resources to tackle these tough problems. This experience has opened up new questions and paths to explore, making it a valuable step toward better TTS technology for Sinhala and other similar languages.

6.2.4. Reflection of Sub Research Question 3

Addressing the sub-research question, *"How can we finetune the VAENAR approach to improve the quality of the Sinhala TTS?"* involves examining key aspects that contribute to the effectiveness and appeal of Text-to-Speech (TTS) systems. Through my experience with implementing the VAENAR model for Sinhala, several pivotal factors influencing TTS quality were identified, highlighting areas for optimization to enhance user satisfaction.

Factors Impacting TTS Quality:

- *Naturalness of Speech*: The extent to which the synthesized speech sounds like a natural human voice is crucial. Users prefer TTS outputs that mimic the fluidity and tone variations of human speech, making the listening experience more pleasant and less robotic.
- *Intelligibility*: The clarity with which users can understand the spoken words directly affects their satisfaction. Speech that is muddled or difficult to decipher can lead to frustration and disengagement.
- *Language and Accent Accuracy*: For multilingual users or those speaking different dialects, the ability of TTS systems to accurately reflect language nuances and regional accents is significant.

Optimization Strategies:

In the course of developing the Sinhala VAENAR-TTS system, strategies for optimizing these factors emerged, even though challenges in achieving clear and natural speech were encountered.

- *Enhanced Data Quality and Diversity*: Investing in the collection and curation of high-quality, diverse training datasets is essential. This includes capturing a wide range of speech patterns, emotions, and accents to train more adaptable and sensitive models.

- *Model Fine-Tuning and Customization*: Adapting the model to better capture the nuances of the target language and its phonetic characteristics can improve naturalness and intelligibility. For Sinhala, specific attention to the language's unique prosodic features was necessary.
- *User Feedback Integration*: Continuously incorporating user feedback into the development process ensures that the TTS system evolves in line with user preferences and needs, addressing any shortcomings in real-time.

Despite the initial setbacks with the VAENAR model in producing intelligible Sinhala speech, the exploration of these factors and optimization strategies has laid a foundation for future enhancements. It has underscored the complexity of developing TTS systems that meet high standards of quality and user satisfaction, especially for less-resourced languages. The lessons learned point towards a continued need for innovation, collaboration, and user-centered design in the quest to perfect TTS technology. This reflection not only answers the sub-research question but also charts a course for making TTS systems more effective, enjoyable, and satisfying for users across diverse linguistic and cultural backgrounds.

6.4.Conclusions about Research Problem

Reflecting on the research problem and the insights gathered from our evaluations, it becomes evident that addressing the challenges of developing Text-to-Speech (TTS) systems for low-resource languages like Sinhala requires a nuanced and resource-intensive approach. The journey of applying the VAENAR model to Sinhala TTS has illuminated not just the linguistic and technical complexities inherent in this task but also the substantial computational resources required to train such sophisticated models.

The application of advanced deep learning models like VAENAR, while promising in theory, encountered significant obstacles in practice. Among these, the high computational power needed for training emerged as a critical limitation. The attempt to produce intelligible Sinhala speech using these models highlighted that free or low-cost computational resources are insufficient for tasks of this complexity and scale. This requirement for high computational resources underscores the broader issue of accessibility and feasibility for researchers and developers working on TTS systems for languages with limited support and resources.

Moreover, the evaluation of the VAENAR model's performance in synthesizing Sinhala speech, though not yielding the desired outcomes, has shed light on the pressing need to enhance the accuracy and accessibility of Sinhala TTS technologies. This is particularly crucial for the visually impaired members of the Sri Lankan community, who stand to benefit significantly from advancements in this area. The research undertaken, despite its challenges, has laid a foundational understanding of the factors that must be addressed to move forward—highlighting the importance of not only linguistic and model-specific considerations but also the practical aspects of computational requirements.

In conclusion, while the research problem posed a significant challenge, the work carried out provides valuable lessons for future endeavors in this field. To advance Sinhala TTS systems towards greater accuracy and accessibility, a concerted effort is needed. This includes investing in the necessary computational infrastructure, exploring innovative data collection and model training techniques, and fostering a collaborative ecosystem among researchers, developers, and community stakeholders. By addressing these multifaceted requirements, we can move closer to developing TTS technologies that are both inclusive and effective, ultimately enhancing the lives of those who rely on them within the Sinhala-speaking community.

6.5.Limitations

The implementation of the Sinhala VAENAR-TTS system represents a notable step forward in Text-to-Speech (TTS) technology for the Sinhala language. However, recognizing the limitations encountered during this research is essential to understand its scope fully and the challenges that lie ahead. This acknowledgment sets the stage for a detailed examination of specific areas where the project faced obstacles, providing a foundation for future efforts to address these challenges.

6.5.1. High Computational Demand

A significant limitation encountered in this study was the high computational demand required for training models like VAENAR. The sophisticated architecture of these deep learning models necessitates extensive computational resources, which are not always accessible, especially in settings with limited funding or infrastructure. This constraint affected the project's capacity to conduct thorough model iterations and optimizations, essential for refining the TTS system to meet the desired outcomes.

6.5.2. Limited Engagement from the Research Community

Efforts to engage with the broader research community to seek insights and support for overcoming the project's challenges faced limitations. Attempts to connect with experts through emails and LinkedIn did not yield the anticipated collaborative opportunities. This lack of engagement highlights a gap in the research ecosystem for novel TTS technology like VAENAR, emphasizing the need for more robust networks and platforms to facilitate knowledge exchange and support.

6.5.3. Data Availability and Diversity

Another significant limitation arises from the challenges associated with low-resource languages like Sinhala. Despite efforts to compile and augment datasets, the availability and diversity of data for Sinhala remain limited when compared to more widely spoken languages. Additionally, finding a lexicon, or in other words, a dictionary with Sinhala phonetic mapping, poses a difficult task. This scarcity of resources can hinder the system's ability to learn and reproduce the full range of speech nuances, especially in less common dialects or sociolects within the Sinhala-speaking community.

6.6.Implications for Future Research

Viewing the limitations encountered in this research as opportunities for learning and growth sets a constructive path for future endeavors in Text-to-Speech (TTS) technology, particularly for low-resource languages like Sinhala. The problems we saw with the VAENAR model, like needing a lot of computer power and not having enough language data, show us important areas we need to work on and improve.

Enhanced Data Collection and Augmentation

The shortage of high-quality, diverse datasets for languages like Sinhala highlights the need for concerted efforts in data collection and augmentation. Future projects could leverage technology to generate synthetic datasets or employ crowd-sourced methods to gather and annotate speech data from a wider cross-section of the community. This would not only improve model training but also ensure that TTS systems can accommodate a broader range of dialects and speaking styles.

Fostering Collaborative Research Networks

The limited engagement from the broader research community experienced during this

project points to the need for more robust networks and platforms that facilitate collaboration. Establishing partnerships across academic institutions, industry, and language communities worldwide can accelerate progress by pooling resources, sharing knowledge, and tackling common challenges collectively. Such collaboration can also extend to interdisciplinary fields, combining insights from linguistics, cognitive science, and computer science to enrich TTS research.

Prioritizing Inclusivity and Accessibility

Ensuring that future TTS technologies are inclusive and accessible to all users, including those from linguistically diverse backgrounds and individuals with disabilities, must be a guiding principle for research. This involves not only technical advancements but also considerations of usability, integration with assistive technologies, and user-centered design principles.

Reflection and Adaptation

Reflecting on the lessons learned from the limitations of this project, future research in TTS technology must remain adaptable and responsive to new insights and challenges. Continuous evaluation, user feedback, and iterative development are essential to refining TTS systems to meet the evolving needs and expectations of users.

In summary, the future research implications stemming from the current study of the Sinhala VAENAR-TTS system are vast and varied. By building on the foundation laid by this work, future efforts have the potential to significantly advance the field of TTS technology, making speech synthesis more natural, inclusive, and accessible for users around the world.

BIBLIOGRAPHY

D. Sasirekha, E. Chandra, "Text to Speech: A Simple Tutorial", IJSCE, Volume 2, Issue 1, March 2012. ISSN: 2231-2307.

Sangeetha, J., Jothilakshmi, S., Sindhuja, S., & Ramalingam, V., 2013. Text to Speech Synthesis System for Tamil. In: International Conference on Information Systems and Computing (ICISC-2013), India.

J. Jayakumari, A. Femina Jalin, "An Improved Text to Speech Technique for Tamil Language Using Hidden Markov Model", 2019 7th International Conference on Smart Computing & Communications (ICSCC), pp.1-5, 2019.

Weerasinghe, R. et al. (no date) "Festival-si: A Sinhala text-to-speech system," Text, Speech and Dialogue, pp. 4702–479.

Nanayakkara, L., Liyanage,C., Tharaka Viswakula, P., Nagungodage, T., Pushpananda,R., Weerasinghe, R.: A Human Quality Text to Speech System for Sinhala," no. February 2019, pp. 157–161, (2018).

Wang, Y. et al. (2017) "Tacotron: Towards end-to-end speech synthesis," Interspeech 2017 [Preprint]. Available at: https://doi.org/10.21437/interspeech.2017-1452.

Saba, R. et al. (2022) "Urdu text-to-speech conversion using Deep Learning," 2022 International Conference on IT and Industrial Technologies (ICIT) [Preprint].

Nadungodage, T., Liyanage, C., Perera, A.H., Pushpananda, R., Weerasinghe, R. (2018). Sinhala G2P Conversion for Speech Processing. SLTU 18: 6th Workshop on Spoken Language Technologies for Under-resourced languages, Gurugram, India.Speech synthesis (2023) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Speech_synthesis (Accessed: December 9, 2023).

Shen, J. et al. (2018) "Natural TTS synthesis by conditioning wavenet on Mel Spectrogram predictions," 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) [Preprint]. Available at: https://doi.org/10.1109/icassp.2018.8461368.

Language Technology Research Laboratory - speech initiative. Available at:

https://ucsc.cmb.ac.lk/ltrl/projects/si/devutils.htm (Accessed: March 14, 2023).

Xinhuanet.com. 2020. Nearly 1 million In Sri Lanka Suffer from Blindness: Health Officials - Xinhua | English.News.Cn. [online] Available at:<http://www.xinhuanet.com/english/2018-10/09/c 137520410.htm> [Accessed 1March 2024].

Jayaweera, A.J.P.M.P. and Dias, N.G.J. (2014) "Hidden markov model-based part of speech tagger for Sinhala language," International Journal on Natural Language Computing, 3(3), pp. 9–23. Available at: https://doi.org/10.5121/ijnlc.2014.3302.

Hill, D.R., Taube-Schock, C.R. and Manzara, L. (2017) "Low-level articulatory synthesis: A working text-to-speech solution and a linguistic tool," Canadian Journal of Linguistics/Revue canadienne de linguistique, 62(3), pp. 371–410. Available at: https://doi.org/10.1017/cnj.2017.15.

Hertz, S.R. (no date) "Integration of rule-based formant synthesis and waveform concatenation: A hybrid approach to text-to-speech synthesis," Proceedings of 2002 IEEE Workshop on Speech Synthesis, 2002. [Preprint]. Available at: https://doi.org/10.1109/wss.2002.1224379.

Rodríguez Crespo, M.Á. et al. (1997) "On the use of a sinusoidal model for speech synthesis in text-to-speech," Progress in Speech Synthesis, pp. 57–70. Available at: https://doi.org/10.1007/978-1-4612-1894-4_5.

Arachchige, T.K. and Weerasinghe, R. (2023) 'Tacosi: A Sinhala text to speech system with Neural Networks', 2023 3rd International Conference on Advanced Research in Computing (ICARC) [Preprint]. doi:10.1109/icarc57651.2023.10145749

GRIFFIN, D. and LIM J., "Signal estimation from modified shorttime fourier transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2), pp. 236–243, 1984.

VISHWANATHAN, M and VISHWANATHAN, M., "Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (MOS) scale", Computer Speech & Language, 19 (1), pp. 55-88, 2005.

H. U. Mullah, F. Pyrtuh, and L. J. Singh, "Development of an HMM based speech synthesis system for indian english language," in 2015 International Symposium on Advanced Computing and Communication (ISACC), pp. 124–127, IEEE, 2015.

Pnfo/sinhala-TTS-dataset: High quality Sinhala dataset for text to speech algorithm training - specially designed for deep learning algorithms, GitHub. Available at: https://github.com/pnfo/sinhala-tts-dataset (Accessed: March 23, 2023).

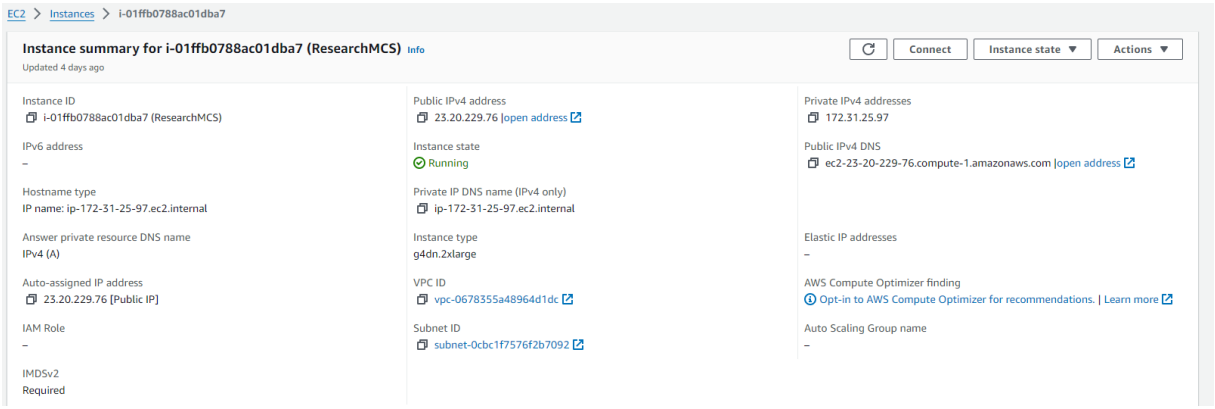
M. Rashad, H. M. El-Bakry, and I. R. Isma'il, "Diphone speech synthesis system for arabic using mary tts," International Journal of Computer Science & Information Technology, vol. 2, no. 4,2010

Lu, H. et al. (2021) 'Vaenar-TTS: Variational auto-encoder based non-autoregressive text-to-speech synthesis', Interspeech 2021 [Preprint]. doi:10.21437/interspeech.2021-2121.

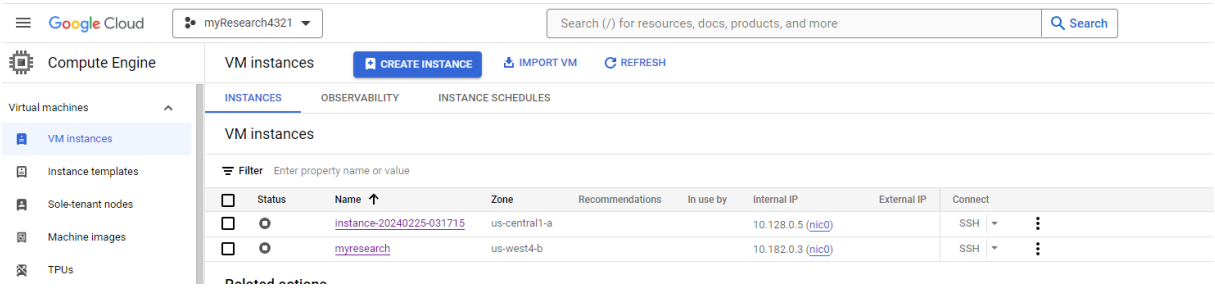
D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015.

APPENDIX A: TRAINING INSTANCES USAGE

AWS EC2 G Instance

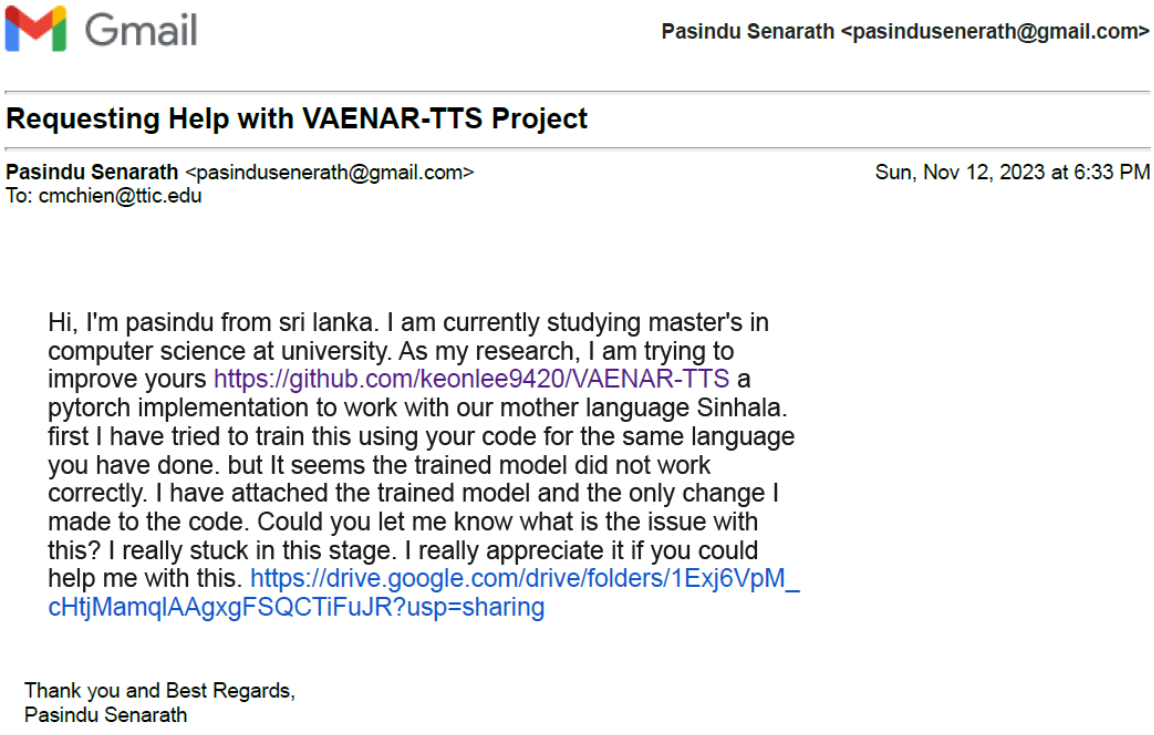


GCP VM Instance



APPENDIX B: EVIDENCE OF REACHING PREVIOUS RESEARCHERS

Through Emails



Gmail

Pasindu Senarath <pasindusenerath@gmail.com>

Requesting Help with VAENAR-TTS Project

Pasindu Senarath <pasindusenerath@gmail.com> Sun, Nov 12, 2023 at 6:33 PM
To: cmchien@ttic.edu

Hi, I'm pasindu from sri lanka. I am currently studying master's in computer science at university. As my research, I am trying to improve yours https://github.com/keonlee9420/VAENAR-TTS a pytorch implementation to work with our mother language Sinhala. first I have tried to train this using your code for the same language you have done. but It seems the trained model did not work correctly. I have attached the trained model and the only change I made to the code. Could you let me know what is the issue with this? I really stuck in this stage. I really appreciate it if you could help me with this. https://drive.google.com/drive/folders/1Exj6VpM_cHtjMamqlAAgxgFSQCTiFuJR?usp=sharing

Thank you and Best Regards,
Pasindu Senarath

Through LinkedIn