Sinhala Speech-to-Speech Chatbot Using Deep Learning Approaches

T.V.R.Weerakoon

K.K.S.Nayanathara

L.I.L.Harischandra

2025

Sinhala Speech-to-Speech Chatbot Using Deep Learning Approaches

T.V.R.Weerakoon
Index No: 20002009

K.K.S.Nayanathara
Index No: 20001207

L.I.L.Harischandra
Index No: 20000715

Supervisor: Dr. Randil Pushpananda

May 2025

Submitted in partial fulfillment of the requirements of the B.Sc. (Honours) in Computer Science Final Year Project

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: T.V.R. Weerakoon

Date: 30/06/2025

.....

Signature of Candidate

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: K.K.S. Nayanathara

Date: 30/06/2025

.....

Signature of Candidate

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: L.I.L. Harischandra

Date: 30/06/2025

.....

Signature of Candidate

This is to certify that this dissertation is based on the work of

Mr. T.V.R. Weerakoon
Ms. K.K.S. Nayanathara
Ms. L.I.L. Harischandra

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Principle/Co-Supervisor's Name: Dr. Randil Pushpananda

Date:30 - 06 - 2025

Signature of Supervisor

Abstract

This research presents the development of an advanced Sinhala speech-to-speech chatbot designed to bridge the gap in digital accessibility for native Sinhala speakers. Despite the rapid advancements in conversational AI systems, low-resource languages like Sinhala remain underrepresented, limiting the ability of native speakers to interact with technology in their own language. Addressing this critical gap, this study proposes an end-to-end solution that seamlessly integrates Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), and Text-to-Speech (TTS) synthesis, enabling real-time, voice-based communication in Sinhala.

The system leverages state-of-the-art deep learning techniques to achieve high accuracy and robustness. For ASR, transfer learning is employed to fine-tune the Wav2Vec2-BERT model on a 40-hour Sinhala speech dataset, achieving remarkable improvements with a Word Error Rate (WER) of 1.79% and a Character Error Rate (CER) of 0.33%, surpassing existing Sinhala ASR systems. The chatbot component utilizes a Retrieval-Augmented Generation (RAG) approach, combining the strengths of Large Language Models (LLMs) with dynamic knowledge retrieval to deliver context-aware and accurate responses in Sinhala. The TTS module, powered by the Variational Inference TTS (VITS) model, generates natural-sounding Sinhala speech, achieving a Mean Opinion Score (MOS) of 4.62 for intelligibility and 4.18 for naturalness in male voices, and 4.24 for intelligibility and 4.07 for naturalness in female voices.

The proposed system addresses a significant gap in voice-based human-computer interaction for Sinhala speakers, with applications spanning education, accessibility, and digital services. By combining cutting-edge ASR, RAG-powered chatbot intelligence, and high-quality TTS, this research not only advances the field of NLP for low-resource languages but also sets a benchmark for future developments in multilingual speech technologies. The modular architecture and methodologies developed in this study provide a foundation for extending similar solutions to other underrepresented languages, fostering greater inclusivity in the digital age.

Acknowledgment

Foremost, we would like to express our sincere gratitude to our supervisor, Dr. Randil Pushpananda, for his invaluable guidance, support, and encouragement throughout the course of this project. Despite his demanding schedule, he consistently took the time to listen, advise, and ensure we stayed on the right path, while also arranging the necessary resources to facilitate our work. We are truly grateful for his commitment and mentorship.

Our gratitude also goes to the members of the UCSC LTRL Research group, especially Dr. A.R. Weerasinghe, for sharing their expertise and providing valuable insights into the research domain, which significantly contributed to our understanding and development.

We would also like to extend our heartfelt appreciation to Dr. Randil Pushpananda and Ms. Amali Perera for their roles as coordinators of the SCS4223 Final Year Project in Software Engineering at the University of Colombo School of Computing. Their continuous support, feedback, and coordination played a crucial role in the successful progression of this project.

Finally, we wish to thank our parents and colleagues for their unwavering support, encouragement, and motivation throughout this journey. Their presence and belief in us have been instrumental in reaching this milestone.

Table of Contents

List of Figures

List of Tables

List of Acronyms

NLP Natural Language Processing

NLU Natural Language Understanding

TTS Text to Speech

DLB Deep Learning-Based

E2E End-to-End

ASR Automatic Speech Recognition

LF-MMI Lattice-free Maximum Mutual Information

HMM Hidden Markov Model

DNN Deep Neural Network

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LLM Large Language Model

RAG Retrieval-Augmented Generation

IR Information Retrieval

CA Conversational Agents

LRL Low-Resource Languages

WER Word Error Rate

GMM Gaussian Mixture Model

CER Character Error Rate

MOS Mean Opinion Score

NAR Non-autoregressive

LSTM Long Short-Term Memory

Wandb Weights & Biases

DSR Design Science Research

MOS Mean Opinion Score

SUS Semantically Unpredictable Sentences

SPSS statistical parametric speech synthesis

VITS Variational Inference TTS

Chapter 1

Introduction

1.1 Background

Speech-to-speech chatbots are transformative technologies that are reshaping how people communicate across linguistic and geographic boundaries in an increasingly connected world. As communication continues to evolve beyond traditional borders, the field of Natural Language Processing (NLP) has seen remarkable advancements, particularly in voice-enabled interactive systems. These systems can now interpret and respond to user queries through spoken language, offering a more intuitive and human-like interaction experience compared to text-based interfaces.

Around the world, both industry and academia are investing heavily in the development and deployment of speech-to-speech chatbots across various sectors. Popular voice assistants such as (Microsoft 2024), Amazon Alexa (Amazon.com 2023), Apple Siri (Inc. 2024) and Google Assistant (Google 2024) demonstrate the growing adoption of these technologies. These systems offer users enhanced accessibility and convenience by enabling seamless voice communication with machines.

However, despite these advancements, mainstream chatbot technologies often fail to adequately support low-resource languages—those with limited linguistic datasets and minimal technological infrastructure. As a result, speakers of such languages face a significant barrier in accessing and benefiting from speech-based AI systems, since most existing solutions are predominantly designed for English and a few other high-resource languages.

This research project seeks to address this gap by developing a speech-to-speech chatbot tailored for Sinhala speakers. By enabling users to interact with technology in their native language, the project aims to create a more inclusive, accessible, and user-friendly digital experience for Sinhala-speaking communities.

1.2 Problem Definition

Sinhala is the main language spoken by the majority of people in Sri Lanka, but most conversational AI systems do not fully support speech-based interaction in Sinhala. Existing chatbots are often limited to text input and output or are designed primarily for English and other widely spoken languages (D. Rajapakshe et al. 2020). This creates a significant barrier for Sinhala-speaking users, especially those who are illiterate, visually impaired, or simply more comfortable speaking than typing. While components like Sinhala speech recognition and text-to-speech have seen progress, there is still a lack of integrated systems that can support a complete speech-to-speech interaction. The absence of such systems limits accessibility and usability for a large portion of the population. Therefore, there is a clear need for a Sinhala speech-to-speech chatbot that can accept spoken Sinhala as input, understand the user's intent, and generate appropriate spoken responses in Sinhala. This would enable more natural and inclusive communication, improve access to information and services, and empower Sinhala-speaking individuals to interact with technology in their native language.

1.3 Key Terms and Concepts

This project is centered on the development of a Sinhala speech-to-speech chatbot, a type of conversational agent that facilitates natural spoken communication between humans and machines. Unlike traditional text-based systems, speech-to-speech chatbots process spoken input and generate spoken output, mimicking human conversation more effectively. These systems rely heavily on advancements in NLP, a subfield of artificial intelligence that enables computers to understand, interpret, and generate human language.

The chatbot is powered by three main components, including ASR, chatbot, and TTS. ASR converts spoken Sinhala language into text by analyzing the acoustic signal and predicting the corresponding word sequences. TTS performs the inverse operation, transforming the chatbot's textual response into natural and intelligible Sinhala speech. The intelligence of the chatbot lies in the Natural Language Understanding (NLU) component, which interprets the intent and context of the user to provide relevant answers.

Together, these components create a seamless, end-to-end conversational experience for Sinhala speakers, bridging the language gap in modern digital communication tools and contributing to the broader field of speech technology for underrepresented languages.

1.4 Scope and Delimitation

1.4.1 Scope

1. The system will allow users to interact with a virtual assistant which will only support Sinhala language interaction.

2. The system will convert Sinhala speech to text, understand the intent of the user's query, and respond in natural-sounding synthesized Sinhala speech.

3. The system will allow users to change the chatbot's domain by uploading relevant documentation.

1.4.2 Delimitation

1. The system will not support languages other than Sinhala in the initial phase.

2. This system will not be able to process or understand blended Sinhala and English words and phrases. The system will only recognize and respond to formal, Sinhala language.

1.5 Significance of the Research

This research holds significant value in promoting digital inclusivity and accessibility for Sinhala-speaking communities in Sri Lanka. By developing a Sinhala speech-to-speech chatbot, the study addresses a critical gap in current conversational AI systems, which often overlook native language support and voice-based interaction. The proposed solution has the potential to benefit individuals who face barriers when using text-based or non-native language interfaces, including those who are visually impaired, illiterate, elderly, or less familiar with digital technologies.

Furthermore, this research contributes to the advancement of NLP and speech technologies for low-resourced languages like Sinhala. It integrates speech recognition, natural language understanding, and speech synthesis into a unified system, demonstrating how deep learning and language-specific modeling can enhance user experience in real-world applications.By enabling seamless voice-based communication in Sinhala, the chatbot can help bridge the digital divide and empower users to access services and information more effectively using their own voice and language.

1.6 Overview of the Dissertation

The dissertation is structured as follows: Chapter 2 discusses some of the existing approaches for ASR systems, chatbots, and TTS systems, including the identified research gaps, aims, objectives, and research questions. Chapter 3 details the process of dataset collection and the experiments conducted for the ASR, chatbot, and TTS systems. Chapter 4 presents the methods and technologies used to implement the application. Chapter 5 discusses the evaluations conducted for the three main components, along with the corresponding results. Chapter 6 outlines the key research and individual contributions, and the challenges encountered during the study. Finally, Chapter 7 provides the conclusions based on the research conducted, along with suggestions for future work.

Chapter 2

Literature Review

2.1 Chapter Overview

This chapter provides a detailed overview of the research domain, including a comprehensive literature review. It introduces essential background information and key terminology to support a deeper understanding of the topic. The discussion covers foundational concepts related to ASR, chatbots, TTS systems, and speech-to-speech chatbot frameworks. It also reviews prior work conducted in each of these areas, with particular emphasis on integrative approaches that combine ASR, chatbot, and TTS technologies to develop a complete speech-to-speech chatbot system. Special attention is given to research efforts focused on Sinhala-language ASR, TTS, and speech-to-speech chatbot development. In addition, this chapter outlines the identified research gaps, clearly states the research aim and objectives, and presents the key research questions that guide this study.

2.2 Speech to Speech Chatbot

Speech-to-speech chatbots are revolutionizing human-machine interaction by enabling natural conversations. This technology has seen significant research, particularly for high-resource languages like English. The early 2000s marked a significant shift in the development of speech-to-speech chatbots with the introduction of machine learning techniques, particularly deep learning. The application of Recurrent Neural Network (RNN)s and Long Short-Term Memory (LSTM) networks to the tasks of speech recognition and speech generation resulted in substantial improvements in the accuracy and naturalness of these chatbots. These advancements enabled chatbots to better understand and generate human-like speech, thereby enhancing the overall user experience and paving the way for more sophisticated conversational agents Hinton et al.

(2012). Google's release of the WaveNet model in 2016 further revolutionized speech generation. WaveNet, a deep generative model, produced highly realistic and human-like speech by modelling raw audio waveforms (Oord et al. 2016). This innovation set a new standard for speech synthesis in chatbots.

Recent years have seen the rise of end-to-end speech-to-speech systems, which streamline the process by directly converting speech inputs to speech outputs without intermediate text representation. According to Cho et al. (2014) and Bahdanau et al. (2014), models like Sequence-to-Sequence (Seq2Seq) with attention mechanisms have been instrumental in this advancement. As an example, a medical chatbot has been developed in English (Dharwadkar and Deshpande 2018). This chatbot allows users to ask health-related questions using their voice and receive spoken responses. It leverages Google APIs for speech-to-text and text-to-speech conversion, facilitating a smooth user experience.Pathirage et al. (2023) developed voicebot specifically for scheduling appointments in the English language. This chatbot utilizes Flask API to transmit voice data and relies on external APIs for both speech-to-text and text-to-speech conversion. Additionally, it leverages RASA NLU, a natural language processing framework, along with a relevant database to understand user queries and generate appropriate spoken responses.

While research in speech-to-speech chatbots has primarily focused on high-resource languages like English, advancements are being made in low-resource languages as well. One such example is a Sinhala language chatbot that facilitates appointment scheduling and offers medical advice through voice or text interaction (Rajapakshe et al. 2020). This chatbot leverages Flutter, Rasa framework, and Firebase for its functionality. Another project developed a Tamil language learning chatbot named Tilly, allowing users to learn through voice commands and responses (Goonatilleke et al. 2020). This was built using Dialogflow and Google Cloud platforms. M et al. (2021) has explored creating a multi-lingual voice assistant for farmers, utilizing Google Translator, PySTTSX3, and Google Search Engines to answer their queries in various languages. These examples showcase the growing potential of speech-to-speech chatbots in bridging language barriers and reaching diverse populations.

These Speech to speech chatbots consist of three basic components: ASR translates user voice to text, the chatbot brain processes the text and formulates a response, and TTS converts that response back into spoken language for the user to hear, as illustrated in the Figure 2.1. This innovative technology enhances accessibility and user experience by allowing natural interaction with machines.

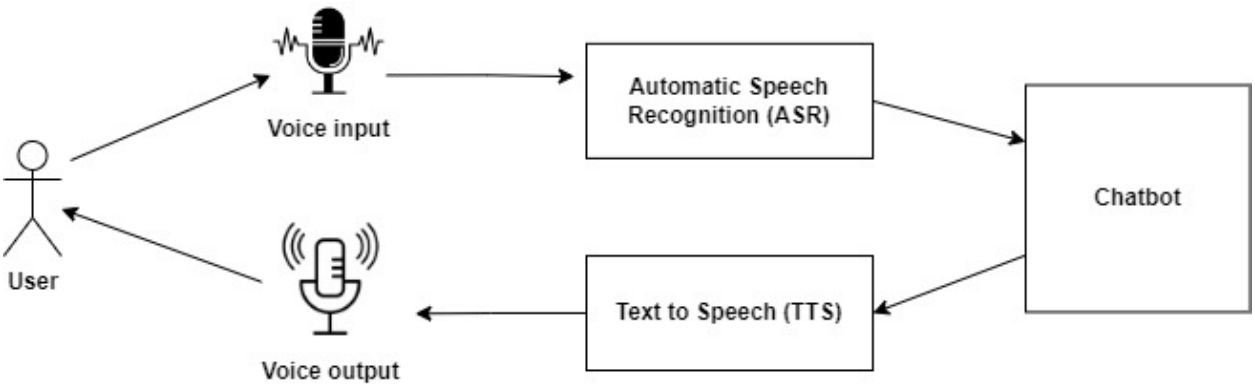

Figure 2.1: Main Components of Speech to Speech Chatbot

2.3 Automatic Speech Recognition

ASR systems have significant advancements in the last few decades. They are designed to convert spoken language into text, providing an interface for various applications such as virtual assistants, transcription services, and accessibility tools.

2.3.1 History of ASR

A few decades ago, researchers used template-based methods to match speech signals, mapping them to templates based on frequency representations. This approach varied with individual speech rates. Subsequently, HMM-GMM models became popular for speaker-independent continuous speech recognition with large vocabularies, and they remain in use today (Li et al. 2013). Over time, machine learning techniques emerged, starting with decision trees and moving to statistical methods that made probabilistic decisions about words or sentences. However, these statistical models often had accuracy issues.

Deep learning has recently advanced numerous fields. Deep Neural Network (DNN) have been introduced, outperforming statistical approaches in terms of accuracy (Mohamed 2014). Consequently, current research focuses on developing speech systems using DNNs. In 2020, a new deep learning method called End-to-End (E2E) speech recognition was introduced. This method consists of encoder, aligner and decoder (Nanayakkara and Weerasinghe 2023).

With the evolution of ASR, most research studies have focused on this field, resulting in the release of more open-source ASR toolkits. DeepSpeech is an Open-source Speech-To-Text engine, which was introduced in 2014, employs a bidirectional RNN model with LSTM (Hannun 2014). Wav2vec 2.0, an encoder model by Facebook, was trained on 60,000 hours of LibriVox audiobooks using a self-supervised objective (Baevski et al.

2020). Whisper is an Encoder/decoder ASR model trained on a large corpus with 2D Convolutional Neural Network (CNN)s architecture. Unlike other ASR models, Whisper generates punctuated and capitalized text with segment-level timestamps, improving readability and usability in long-form audio (Radford et al. 2023).

2.3.2 Transfer Learning

The most recent advancement in ASR for low-resourced languages is Transfer learning. Transfer learning involves working with two distinct datasets: a large, data-rich dataset, which is used to train the system for specific tasks, and a second, much smaller dataset that pertains to the real-world problem we want to solve. Despite the smaller size of the second dataset, the system remains focused on the same task, leveraging the knowledge acquired from the first dataset. This process allows us to transfer insights from the initial system to the second one, ultimately enhancing its performance. Similarly, in speech recognition, transfer learning enables the use of knowledge derived from a well-trained, data-rich model to improve predictions in a low-resource model, leading to more accurate output generation.

As Sinhala is a low-resourced language, it's hard to find such a large dataset. In bottleneck feature transfer learning, a pre-trained model trained on a well-known, large dataset is utilized by removing its original output layer and replacing it with a customized layer tailored to the target low-resource speech dataset. The weights of the original output layer are discarded, while all preceding layers remain unchanged. During retraining with the target speech data, only the newly introduced output layer undergoes weight adjustments, ensuring the model refines its predictions based on the specific characteristics of the low-resource dataset.

There are multiple multi lingual pre-trained models are available for ASR including, Wav2Vec2-XLSR Baevski et al. (2020), Whisper Radford et al. (2023), and Wav2Vec2-BERT (Barrault et al. 2023).

Wav2Vec2-XLSR

Wav2Vec2-XLSR (Cross-Lingual Speech Representations) is a multilingual extension of the original Wav2Vec 2.0 model developed by Facebook AI. It is trained in a self-supervised manner on unlabelled speech data across 128 languages. The model utilizes a convolutional encoder to extract latent speech representations, followed by a Transformer-based context network that captures long-term dependencies in speech sequences. During pretraining, it learns to identify masked latent representations from a quantized latent space, allowing it to generalize effectively across various languages. Wav2Vec2-XLSR achieves impressive results in low-resource and cross-lingual ASR tasks by leveraging the diversity of multilingual datasets (Baevski et al. 2020). Its ability to fine-tune on a limited amount of labelled data makes it particularly useful for underrepresented languages in ASR research.

Whisper

Whisper, developed by OpenAI, is a large-scale multilingual and multitask ASR model trained on 680,000 hours of diverse audio data collected from the web. Unlike conventional ASR models, Whisper is trained in a supervised fashion using paired audio-text data and supports multiple languages, translation, and language identification tasks. Its architecture is based on an encoder-decoder Transformer model, where the encoder processes raw audio features (mel-spectrograms), and the decoder generates transcriptions token by token. One of Whisper's strengths lies in its robustness to accents, background noise, and poor-quality recordings due to its diverse training data. Furthermore, Whisper includes capabilities for language detection and speech translation, setting it apart as a general-purpose speech model that can be zero-shot applied to many tasks (Radford et al. 2023)

Wav2Vec2-BERT

Wav2Vec2-BERT is a hybrid model that integrates Wav2Vec 2.0 for acoustic modeling and BERT for contextual language modeling. This combination aims to improve recognition accuracy by incorporating richer linguistic context into the decoding process. The architecture first processes raw speech with the Wav2Vec2 encoder to extract frame-level acoustic representations. These are then fed into a pre-trained BERT model that enhances the contextual understanding of speech segments, particularly beneficial in noisy environments and for languages with complex grammatical structures. This approach bridges the gap between acoustic and language modeling, leveraging the bidirectional contextual strengths of BERT to refine decoding outputs in ASR systems (Barrault et al. 2023). It is particularly effective in scenarios involving ambiguous phonetic patterns or homophones.

2.3.3 The evolution of Sinhala ASR technologies

While high-resourced languages like English and Mandarin have shown substantial progress in ASR, there were very few studies that have focused on developing Sinhala ASR systems. Sinhala, the primary language spoken in Sri Lanka, poses unique challenges for ASR due to its phonetic and morphological complexity. The table 2.1 shows the research studies done in the Sinhala ASR domain.

Year	Title	Model	Dataset	Accuracy	Reference
2011	Continuous Sinhala Speech Recognizer	HMM	UCSC 10M Sinhala Corpus	WER 3.86% SER 24.26%	Nadungodage and Weerasinghe (2011)

2012	Speaker Independent Sinhala Speech Recognition for voice dialling	HMM	Sinhala speech samples from various people	CRD Noisy-82.19% Quiet-87.37%	Amarasingha and Gamini (2012)
2012	Dynamic Time Warping Based Speech Recognition for isolated Sinhala words	DTW	N/A	WRR 93.92%	Priyadarshani et al. (2012)
2013	Efficient use of training data for Sinhala speech recognition using active learning	Active Learning	N/A	WRR 95.17%	Nadungodage and Weerasinghe (2013)
2015	Real-time Translation of Discrete Sinhala Speech to Unicode text	HMM	Noise free recordings with a frequency of 22050Hz	WRR 94.80%	Gunasekara and Meegama (2015)
2018	Sinhala speech recognition for interactive voice response systems accessed through mobile phone	HMM	A speech corpus of 2.63 hours	WER 11.2%	Manamperi et al. (2018)
2018	Sinhala speech to sinhala unicode text conversion for disaster relief facilitation in Sri Lanka	HMM	N/A	Noiseless 65% Noisy 54%	Prasangini and Nagahamulla (2018)
2019	Transfer learning based free-form speech command classification for low-resource languages	DeepSpeech model	N/A	93.16%	Karunanayake et al. (2019)

2020	Usage of Combinational Acoustic Models and Identifying the Impact of Language Models in Sinhala Speech Recognition	SGMM and DNN-HMM	UCSC Novel Corpus	WER 31.72%	Gamage et al. (2020)
2021	Improve sinhala speech recognition through e2e lf-mmi model	LF-MMI	N/A	WER 28.55%	Gamage et al. (2021)
2023	Exploring Model-Level Transfer Learning to Improve the Recognition of Sinhala Speech	Transfer learning	Speech dataset at LTRL at UCSC	WER 17.19% CER 5.9%	Nanayakkara and Weerasinghe (2023)

Table 2.1: Sinhala ASR Models and Systems

The evolution of Sinhala ASR technologies reflects a dynamic journey from traditional methodologies to deep neural approaches. Research done in the early stages like Amarasingha and Gamini (2012), experimented on speaker-independent voice dialling, and the Gunasekara and Meegama (2015) introduced Hidden Markov Model (HMM) to the Sinhala ASR landscape. When proceeding further, Karunanayake et al. (2019) incorporated CNN for transcribing free from Sinhala and Tamil speech. Gamage et al. (2020) explored hybrid acoustic models by integrating the DNN-HMM architecture. Gamage et al. (2021) uses end-to-end Lattice-free Maximum Mutual Information (LF-MMI) models as recent innovation and achieved a Word Error Rate (WER) of 28.55%. This method overcomes most of the challenges in low-resourced language identification. The most recent research done in Sinhala Speech recognition systems, have used a transfer learning-based approach to build the ASR system and gained WER of 17.19% and 5.9% in CER (Nanayakkara and Weerasinghe 2023).

2.4 Chatbot

The concept of chatbots originates from early discussions on artificial intelligence, including Alan Turing's seminal question, "Can machines think?" and the Turing Test introduced in the 1950s (TURING 1950). Over the years, various research has explored developing chatbots across different languages and domains.

For instance, a chatbot designed as a virtual assistant for tourists in Bengkulu was created using extreme

programming, achieving an accuracy rate of 87% in responding to user inputs (Rohman and Subarkah 2024).

A chatbot system for material sciences was developed using the LLaMA-2 language model, pre-trained on the S2ORC dataset containing domain-specific articles. This research produced four open-sourced language models to support chatbot applications in this field (X. Yang and Petzold 2024). Similarly, chatbots employing Large Language Model (LLM) have been developed in areas like radiology to enhance their capabilities. However, these studies also identified limitations in LLM-based chatbots and proposed solutions to address them (Bhayana 2024).

This study focuses on the development of an interactive communication application designed to simplify information exchange in a user-friendly manner. The application enables users to submit queries in plain language and receive clear, actionable responses, making it particularly useful for individuals seeking precise and detailed information across various domains. Designed with accessibility in mind, the system allows users, including students, to easily access the knowledge they need. To enhance usability, a built-in guide assists users in navigating the application independently, reducing reliance on external support. Beyond personal and educational use, the application also strengthens Information Retrieval (IR) processes by streamlining data retrieval and delivery, ensuring that users receive relevant information efficiently. With a global reach, the system is particularly beneficial in regions like Pakistan, where access to educational resources may be limited. By facilitating knowledge dissemination across different countries and cultures, the platform helps bridge information gaps and enhance learning opportunities. Furthermore, the application supports educational growth and enables users to apply their acquired knowledge in real-world scenarios, ultimately contributing to personal and socio-cultural development. Its design ensures accessibility for a diverse audience, improving learning experiences and optimizing IR capabilities on an international scale (Yousaf 2025).

This study explores the role of AI-driven Conversational Agents (CA) tailored for Low-Resource Languages (LRL), with a particular focus on Mooré, a widely spoken language in Burkina Faso. By facilitating natural language interactions in local languages, these AI-powered systems can empower informal traders to manage financial transactions more effectively, promoting business autonomy and security while fostering gradual integration into formal economic structures. In many African nations, the informal business sector serves as a crucial pillar of the economy, offering essential livelihood opportunities in environments where formal employment is scarce. Despite the increasing adoption of digital technologies, entrepreneurs operating in this sector frequently encounter challenges related to literacy limitations and language barriers. These obstacles not only hinder accessibility but also expose individuals to financial insecurity and potential fraud. The research also highlights key challenges in developing AI models for African languages, including data scarcity and linguistic diversity, and discusses potential solutions such as cross-lingual transfer learning and data augmentation techniques (Ouattara et al. 2025).

In Sinhala-language chatbot research, initial efforts include a prototype chatbot which capable of answering simple queries and performing basic tasks like displaying the current date and time (Hettige and Karunananda 2006). Later studies developed domain-specific chatbots, such as one providing details about degree programs at the University of Ruhuna's Faculty of Technology (Kumanayake 2015) and another offering information about Sri Lanka Railway services, including schedules, ticket prices, and reservations (Harshani 2019). Efforts to enhance chatbot accuracy using Sinhala word embeddings, based on the RASA framework, have also been reported (Bimsara Gamage et al. 2020). Furthermore, a comparative study found that the RASA framework outperformed Microsoft LUIS in terms of accuracy, adaptability, and cost-effectiveness for Sinhala chatbot development (Avishka et al. 2021).

Retrieval-Augmented Generation

Fine-tuning Large Language Model (LLM)s helps address key limitations in domain-specific applications by supplementing pre-trained models with additional training data. However, fine-tuned models become static, with their knowledge fixed at the time of training. While this approach is beneficial for ensuring a specific response style, it is impractical for organizations requiring real-time updates to domain knowledge. Continuously retraining an LLM with new data is both time-consuming and costly. Retrieval-Augmented Generation (RAG) offers a solution to this challenge by allowing general-purpose LLMs to dynamically access external knowledge sources. Instead of directly interpreting a user query, RAG first retrieves relevant information from a designated database before processing the response with an LLM. For example, a traditional LLM-based chatbot may fail to answer queries about a restaurant's current menu if such data was not present in its training set. However, by incorporating RAG, the chatbot can retrieve real-time menu information and generate an accurate, contextually relevant response. This fusion of generative AI with external knowledge retrieval enhances both accuracy and adaptability. The adoption of RAG extends beyond commercial applications. Organizations can leverage this technique to create chatbots capable of querying internal resources such as employee handbooks, technical manuals, or legal databases. For instance, legal professionals can use RAG-powered systems to retrieve relevant case law or administrative guidance, eliminating the need for direct model retraining. Instead of embedding all domain-specific knowledge within the LLM, RAG efficiently integrates stored knowledge with the model's natural language processing capabilities. This innovation is particularly promising for government agencies, which manage vast repositories of data but often struggle with accessibility. By utilizing RAG, agencies can enhance public services by enabling seamless, conversational access to complex information. The integration of RAG-based AI in public-sector applications has the potential to revolutionize how governments deliver information, transforming citizen interactions into a more intuitive and efficient experience (Mazur and Thimmesch 2024).

Retrieval-Augmented Generation (RAG) combines LLMs with retrieval techniques to improve response

quality by incorporating relevant external information. For example, a voting advice application integrated RAG to ensure fairness, impartiality, and transparency, utilizing post-retrieval processing methods for enhanced accuracy (Gittmann et al. 2024). Similarly, advanced multi-agent systems using RAG and LLMs have been developed, showcasing strategies to create sophisticated, collaborative applications (Lehto 2024).

Nvidia's framework offers enterprise-grade RAG-based chatbots addressing critical challenges within RAG pipelines. It identifies optimization points and evaluates accuracy-latency trade-offs between large and small LLMs, enabling balanced performance (Akkiraju et al. 2024).

Furthermore, intelligent tutoring chatbots combining RAG and custom LLMs have been proposed to provide tailored, contextually relevant educational experiences (Modran et al. 2024). A similar RAG-optimized LLM was assessed in Finnish housing law applications, demonstrating its efficacy in domain-specific legal guidance (Rafat 2024).

The development of a PDF-based chatbot utilizing Large Language Model (LLM) in generative AI serves as a practical application of these advanced models. This system employs prompt-based querying, tokenization, and a vector-based storage mechanism for processing PDF documents. By leveraging LLMs, the chatbot extracts relevant information directly from uploaded PDFs and generates contextually accurate responses, even for complex queries. Furthermore, the chatbot incorporates an advanced feature known as Language Chains, enhancing its ability to maintain coherence and contextual relevance in multi-turn interactions (Deekshita et al. 2024).

Mansurova, Nugumanova, and Makhambetov focused on developing a chatbot designed to answer open-domain questions related to blockchain technology. Their research explores various techniques and approaches for creating an effective chatbot capable of handling complex queries within this domain. By leveraging advanced natural language processing methods, the study aims to bridge the gap between users and blockchain knowledge, enabling seamless communication and efficient information exchange (Mansurova et al. 2023).

This study introduces an alternative approach to chatbot development using Retrieval-Augmented Generation (RAG) with a focus on Frequently Asked Questions (FAQs). The research demonstrates that domain-specific retrieval embedding models trained with the infoNCE loss outperform general-purpose embeddings. Additionally, it is the first to implement a Reinforcement Learning (RL)-based optimization strategy within the RAG pipeline, aiming to minimize costs and maximize efficiency, thereby showcasing its broader applicability beyond FAQ-based chatbots (Kulkarni et al. 2024).

This study explores the significant impact of LLMs on AI advancements, particularly in developing chatbots tailored for specialized domains such as therapy. It also investigates innovative methods like Reinforcement Learning from Human Feedback to enhance chatbot performance. Furthermore, the research highlights the profound influence of LLMs on the future of conversational AI, emphasizing their potential to

refine and improve human-computer interactions (Bill and Eriksson 2023).

This survey examines the evolution of chatbot technology, particularly the integration of Artificial Intelligence (AI) and Natural Language Processing (NLP) in modern applications. It highlights how businesses increasingly rely on chatbots for virtual customer support while overlooking key limitations and challenges in their implementation. The study's findings identify critical areas for future research, emphasizing the need for innovative approaches to enhance chatbot efficiency and effectiveness (Caldarini et al. 2022).

This study presents a chatbot designed to generate precise and contextually relevant responses within the framework of a Brazilian consortium company. The system is built on a custom codebase that integrates highly contextualized information through unique data generation techniques. By LLMs and employing the parent retriever document approach, the model enhances document retrieval efficiency, ensuring seamless access to company-specific data. A key contribution of this work is the integration of RAG with a customized chunk-splitting technique, which optimizes the chatbot's performance. Additionally, the proposed methodology for database generation and data analysis fills a gap in the literature, addressing the need for a structured approach to chatbot data evaluation. The chatbot demonstrated a high level of accuracy in answering user queries, with minimal errors primarily caused by gaps in the company's documentation rather than model limitations. The chunk-splitting technique proved to be a critical factor in maintaining performance, particularly when handling large datasets. The model underwent expert evaluation and was approved for its adherence to business rules and accuracy. From a practical perspective, the chatbot serves as an interactive communication tool for both customers and employees, providing valuable insights that can inform marketing strategies, highlight key customer concerns, and uncover underlying demands (Melo et al. 2025).

LLMs exhibit remarkable capabilities but face limitations such as hallucinations, outdated knowledge, and a lack of transparency in reasoning. RAG has emerged as an effective solution by integrating external knowledge sources to improve accuracy, credibility, and domain-specific adaptability. This approach enables continuous updates and enhances knowledge-intensive tasks by combining LLMs' inherent language understanding with dynamic external databases. This study provides a comprehensive analysis of the evolution of RAG methodologies, including Naïve RAG, Advanced RAG, and Modular RAG. It explores the core components of RAG systems—retrieval, generation, and augmentation—examining the latest advancements in each area. Additionally, the paper presents an updated evaluation framework and benchmarking standards. Lastly, it discusses existing challenges and potential future directions for enhancing RAG-based systems (Gao et al. 2024).

Large pre-trained language models have demonstrated the ability to store factual knowledge within their parameters and achieve state-of-the-art results when fine-tuned for downstream NLP tasks. However, their effectiveness in precisely retrieving and manipulating knowledge remains limited, particularly in knowledge-

intensive applications, where they often underperform compared to task-specific architectures. Furthermore, challenges such as providing decision provenance and updating world knowledge persist. To address these limitations, models incorporating a differentiable access mechanism to explicit non-parametric memory have been explored, though primarily for extractive tasks. This study investigates a general-purpose fine-tuning approach for RAG, integrating both parametric and non-parametric memory for language generation. Specifically, the proposed RAG models utilize a pre-trained sequence-to-sequence model as parametric memory and a dense vector index of Wikipedia, accessed via a neural retriever, as non-parametric memory. Two RAG variations are compared: one where the same retrieved passages are used throughout the generated sequence, and another that allows different passages to be retrieved per token. The models are fine-tuned and evaluated on multiple knowledge-intensive NLP tasks, achieving state-of-the-art results on three open-domain QA benchmarks. Additionally, the study finds that RAG models generate responses that are more specific, diverse, and factually accurate compared to parametric-only sequence-to-sequence baselines (Lewis et al. 2020).

RAG techniques have demonstrated their effectiveness in incorporating real-time information, reducing hallucinations, and improving response accuracy, particularly in domain-specific applications. While various RAG methods have been developed to enhance large language models by retrieving contextually relevant data, challenges such as complex implementation and increased response latency persist. A typical RAG pipeline consists of multiple sequential processing steps, each with different implementation possibilities. This study systematically examines existing RAG methodologies and explores their optimal configurations to improve efficiency and performance. Through extensive experimentation, the research identifies strategies that balance response quality and computational efficiency. Additionally, the study highlights the advantages of multimodal retrieval in enhancing question-answering tasks involving visual data and accelerating content generation through a "retrieval as generation" approach (Wang et al. 2024).

2.5 Text To Speech

TTS has become a prominent technology over the past years. Conversion of text to speech helps to keep human-computer interaction smoother and more natural because text is one of the basic forms which computers use to interact with humans. Computer systems can convert text in natural language to a speech format equivalent to a native speaker pronouncing the same given text using TTS (Tan et al. 2021).

2.5.1 History of TTS

Since the 12th century, several experiments have been conducted to create speech synthesisers. Tan et al. (2021) explained that Wolfgang von Kempelen, a Hungarian scientist, created a "speaking machine" in the

second decade of the 18th century that produced short words and basic phrases using a variety of "bellows", "resonance boxes", "bagpipes", and "springs".

Speech synthesisers have become most commonly used in many computer operating systems after the introduction of the initial speech synthesis system based on computers in 1960. There were initial methods such as Articulatory, formant, and concatenative synthesis used in computer-based speech synthesis. Subsequently, the statistical parametric speech synthesis (SPSS) method was produced with the development of statistical machine learning. It forecasts characteristics including fundamental frequency, duration, and spectrum. From the 2010s, the most widely used technique was Deep Learning based speech synthesis, which produced noticeably better voice quality (Arachchige and Weerasinghe 2023).

2.5.2 Deep Learning Approaches in TTS

Deep learning has revolutionized TTS systems, enabling the generation of highly natural and expressive synthetic speech. Traditional statistical and concatenative methods have been increasingly replaced by deep learning architectures that model the entire speech generation pipeline end-to-end. Early advancements like Tacotron and WaveNet introduced the concept of learning alignments between text and audio representations, significantly improving the naturalness and flexibility of TTS outputs. These models were followed by further refinements such as Tacotron 2 and Transformer TTS, which enhanced robustness and prosody modelling by leveraging attention mechanisms and sequence modelling techniques.

A notable evolution in TTS came with the introduction of models such as FastSpeech and VITS, which addressed issues of speed and quality by introducing non-autoregressive generation and variational inference, respectively. These models not only improved training and inference efficiency but also facilitated higher quality and more expressive speech generation.

Building upon these deep learning-based architectures, the latest trend in TTS is the integration of LLMs into speech synthesis systems. One significant breakthrough is the HALL-E model by Nishimura et al. (2024), which combines an LLM with a hierarchical neural codec. This approach enables high-quality, zero-shot speech synthesis without requiring speaker-specific training data. HALL-E is capable of generating natural, minute-long speech segments that are adaptable across speakers, languages, and accents. This advancement continues the trajectory set by earlier models such as VALL-E, VALL-E X, and RALL-E, all of which leverage LLMs to reduce dependency on large training datasets. These models exemplify a shift from data-intensive architectures to more dynamic and context-aware systems, capable of producing personalized and multilingual speech outputs with minimal customization.

Overall, Deep Learning-Based (DLB) TTS approaches have progressed from autoregressive models to non-autoregressive, variational autoencoder-based architectures, and now LLM-integrated frameworks. These innovations have not only improved speech quality and training efficiency but also significantly broadened

the adaptability and scalability of modern TTS systems.

Deep Learning Approaches for Low-Resource TTS

While DLB approaches in TTS synthesis have demonstrated remarkable success in English, largely due to its relatively simple phonetic structure and the availability of extensive, high-quality training datasets. However, numerous other languages continue to encounter significant obstacles in developing effective TTS systems. In particular, languages spoken across regions such as India, Japan, China, Nepal, Turkey, Sri Lanka, and Arabic-speaking countries require TTS systems that are specifically tailored to their unique linguistic characteristics.

Historically, TTS systems for low-resource languages have relied on traditional methods such as concatenative synthesis and SPSS. However, these techniques have often produced speech that lacks naturalness and intelligibility. The recent emergence of deep learning techniques has sparked a shift towards more sophisticated models, with growing research efforts dedicated to adapting these techniques to low-resource settings. Approaches such as semi-supervised and self-supervised learning have shown considerable promise in addressing the limitations imposed by insufficient annotated data, enabling the generation of more natural and expressive speech outputs. Moreover, a variety of advanced strategies such as dataset mining, cross-speaker transfer, cross-lingual transfer and dynamic prosody extraction have been proposed and successfully implemented in recent studies to enhance performance in low-resource TTS systems, as highlighted by Tan et al. (2021).

Year	**Language**	**System**	**Model**	**Dataset**	**Accuracy Rates (MOS)**	**References**
2017	Turkish	Turkish TTS	Concatenation Synthesis basedRC8660	N/A	3.29	Oyucu (2023)
2020	Hindi, Bengali, Malayalam	IndicSpeech	Deep Voice 3	IndicSpeech Corpus	Hindi: 4.31 Malayalam: 3.87 Bengali: 3.96	Srivastava et al. (2020)
2020	Kannada, Malayalam, Tamil, Telugu, Bengali, Gujarati, Hindi, Odia, Rajasthani	Generic TTS	Tacotron2 + WaveGlow	IndicSpeech Corpus	3.98	Prakash and Murthy (2020)
2022	Mongolian	MnTTS	FastSpeech2+ HiFi-GAN	MnTTS dataset	4.46 ± 0.06	Hu et al. (2022)
2023	Telugu	prosody-TTS	Encoder + Decoder + Neural Vocoder	IndicSpeech Corpus	3.98	Pamisetty and Sri Rama Murty (2023)
2023	Turkish	Turkish TTS	Tacotron2 + WaveGlow	New Turkish Corpus	4.49	Oyucu (2023)

Year	Language	System	Model	Dataset	Accuracy Rates (MOS)	References
2023	Dravidian, Indo-Aryan, Sino-Tibetan based 13 Indian languages	IndicTTS	FastPitch + HiFiGAN V1	IndicSpeech Corpus	3.8	Kumar et al. (2023)
2023	Nepali	Nepali TTS	Tacotron2 + HiFiGAN	OpenSLR dataset, self-recorded data	4.03	Khadka et al. (2023)
2024	Mundari, Hindi	MUNTTS	Transformer based XTTS, VITS models	Mundari speech dataset	VITS-44K: 3.69 ± 1.18	Gumma et al. (2024)

Table 2.2: Recent Low Resource TTS Synthesis Systems

Recent developments in TTS synthesis for low-resource languages, including Turkish, Hindi, Bengali, Malayalam, Kannada, Tamil, Mongolian, Telugu, Nepali, and Mundari have increasingly adopted DLB models such as Tacotron2, FastSpeech2, VITS, and XTTS as shown in Table 2.2. These models have demonstrated notable improvements in naturalness and intelligibility, often achieving MOS ranging from approximately 3.8 to 4.5 depending on the language, dataset quality, and model configuration. These advancements indicate a clear and ongoing transition from traditional synthesis methods to DLB approaches, even in contexts where linguistic resources are limited.

Variational Inference TTS (VITS) Model

Recent advancements in deep learning have revolutionized the field of TTS, with models such as Tacotron 2, FastSpeech, and Glow-TTS demonstrating impressive results in generating human-like speech. As a continuation of this evolution, VITS has emerged as a powerful end-to-end neural TTS framework that unifies the advantages of variational autoencoders (VAEs), adversarial training, and normalizing flows (Kim et al. 2021).

VITS is a DLB model that eliminates the need for external aligners by integrating a stochastic duration predictor directly into the model architecture. This non-autoregressive approach allows for parallel training and fast inference while maintaining high naturalness and speaker similarity. It comprises three main components:

- **Conditional Variational Autoencoder (VAE)** – learns a latent representation of the input text and audio features.
- **Normalizing Flow-based Decoder** – enables flexible and expressive waveform generation.
- **Stochastic Duration Predictor** – models the alignment between text and audio without external aligners.

The VITS model has proven effective not only in standard TTS but also in a wide range of advanced applications such as multilingual synthesis, zero-shot speaker adaptation, emotion-aware speech generation, and robust speech synthesis in noisy environments. The following Table 2.3 summarizes several notable VITS-based TTS systems developed in recent years, showcasing the versatility and growing popularity of VITS in global TTS research.

Year	**Language**	**TTS System**	**Reference**
2021	English	VITS Model	Kim et al. (2021)
2022	English, French, Portuguese	YourTTS	Casanova et al. (2022)
2022	English	TriniTTS	Ju et al. (2022)
2023	Chinese	An Emotion Speech Synthesis Method Based on VITS	Zhao and Z. Yang (2023)
2023	Mongolian	Comparative Study for Multi-Speaker Mongolian TTS with a New Corpus	Liang et al. (2023)
2024	English, French	Llama-VITS	Feng and Yoshimoto (2024)
2024	Hindi	STORiCo	Tankala et al. (2024)

Table 2.3: Recent VITS Model-Based TTS Systems

These recent contributions illustrate the flexibility and extensibility of the VITS model across various languages, speaker profiles, and speech characteristics. Its ability to integrate emotion, handle multiple speakers, and generate speech under challenging conditions (e.g., noise, low resources) makes it a robust candidate for next-generation TTS systems.

The subsequent sections present experiments conducted using the VITS model for the Sinhala language. These sections describe the adaptation and training process of the model to accommodate the linguistic characteristics of Sinhala and evaluate its performance in generating natural and intelligible speech. The results demonstrate the effectiveness of the VITS model in handling under-resourced languages and highlight its potential for the development of robust Sinhala TTS systems.

2.5.3 The evolution of Sinhala TTS Systems

Sinhala, the primary language spoken by over 16 million people in Sri Lanka, belongs to the Indo-Aryan language family. Despite its richness and adaptability, Sinhala poses challenges for TTS development due to

its linguistic complexity, limited resources, and distinct phonetic and grammatical features. The language consists of 61 letters, including 41 consonants, 18 vowels, and 2 semi-consonants. Unique traits, such as differences between spoken and written styles, further complicate synthesis, but can also be leveraged to improve expressiveness in speech generation.

Year	**System**	**Technique**	**Model**	**Dataset**	**Accuracy**	**References**
2005	Letter-to-Sound Algorithm based model	Diphone concatenation	N/A	UCSC Sinhala Corpus	98.21%	Wasala and K. Gamage (2005)
2007	Festival-si	Diphone concatenation	Festival	1413 Sinhala diphone units database	71.5%	Weerasinghe et al. (2007)
2009	AMORA	Diphone concatenation	Festival	N/A	N/A	Dias and Jayasena (2009)
2018	Human Quality TTS	Unit Selection based HMM model	MaryTTS	UCSC Sinhala Corpus	70%	Nanayakkara, Liyanage, et al. (2018)
2018	Clustergen algorithm-based Festival Framework	Unit Selection	Festival, Merlin	UCSC Sinhala Corpus	MOS - 3.285 $\pm$ 0.161	Sodimana et al. (2018)
2019	English and Sinhala TTS System	Deep Learning	Merlin, E2E model	"Delowak Atarin Eha" Short story	1000+ utterances within 96 hours	Jayawardhana et al. (2019)

2022	MARY TTS Platform	Unit Selection	MaryTTS	N/A	91.7% MOS - 4.972	Senarathna et al. (2022)
2023	TacoSi	Deep Learning	Tacotron, WaveNet	"PathNirvana"	84.0% MOS - 4.39	Arachchige and Weerasinghe (2023)
2023	Mobile Base Sinhala Book Reader	Deep Learning	SpeechT5, HiFIGAN	"PathNirvana"	N/A	Madhusha et al. (2023)

Table 2.4: Sinhala TTS synthesis Systems

The 2.4 table shows currently available Sinhala TTS systems and their used techniques, models and accuracy rates. Unfortunately, very few studies have been conducted on Sinhala TTS synthesis beginning in the 21st century, and even fewer of them have been effective. The first well-documented Sinhala TTS system is *Festival-si*, which was introduced by (Weerasinghe et al. 2007) using the diphone concatenation. Following that, Nanayakkara, Liyanage, et al. (2018) implemented human quality TTS using MaryTTS framework based on a Unit selection HMM model. Recent DLB Sinhala TTS systems, such as *Tacosi*, have demonstrated significant improvements in intelligibility and naturalness over traditional systems like *Festival-si*, by leveraging deep learning models such as Tacotron and WaveNet (Arachchige and Weerasinghe 2023). This highlights the potential of DLB approaches for Sinhala TTS systems. However, compared to other languages, Sinhala TTS research has seen less exploration of novel DLB TTS models and techniques.

Given the success of DLB approaches in other low-resource languages, applying techniques like transformer-based models, diffusion models, Non-autoregressive (NAR) models, and fully E2E models to Sinhala TTS holds immense promise. By leveraging these advancements, Sinhala TTS can achieve significant improvements in accuracy, naturalness, and efficiency, opening doors for wider adoption in various applications.

2.6 Research Gap

After studying the previous research studies, which are mentioned in the literature review, the following are the identified gaps.

- **Low ASR Accuracy:** Existing systems continue to face challenges to achieve high accuracy, indicating that there is still a significant gap in the development of more effective techniques. This highlights the need for further research and innovation within the Sinhala ASR domain.

- **Lack of LLM-Powered Sinhala chatbots:** The chatbot systems' replies can be enhanced to provide more logical and human-like dialogues by using LLM models. Furthermore, more customised answers can be given. As a result, user interaction can be enhanced. Compared to other chatbot systems, those using LLM models will be more successful when using high-quality training data since they have a deeper comprehension of languages and their contexts.

- **Inadequate TTS Quality:** Most existing models are trained on high-resource languages and do not generalize well to Sinhala, which has distinct phonetic and syntactic features. As a result, synthesized speech often sounds unnatural or unclear. In particular, fully E2E models and advanced architectures like non-autoregressive, transformer-based, and diffusion models have not yet been explored for Sinhala. This creates a clear research gap and an opportunity to investigate how these approaches can be adapted and optimized to improve TTS quality for the Sinhala language.

- **Lack of End-to-End Integration:** While some progress has been made in individual components, no existing solution fully integrates ASR, an intelligent dialogue engine, and natural TTS into a complete speech-to-speech system for Sinhala. The lack of such a system limits accessibility and usability, particularly for non-literate users or those with visual impairments.

2.7 Aim, Objectives and Research Questions

2.7.1 Research Aim

This research aims to explore and develop deep learning-based methods for building a Sinhala speech-to-speech chatbot that can accurately understand spoken Sinhala, process the input, and generate natural spoken Sinhala responses.

2.7.2 Objectives

The following are the objectives of the project.

- To enhance digital inclusivity by enabling natural and accessible human-computer interaction for Sinhala speakers.

- To address the lack of speech technology support for low-resource languages, specifically Sinhala.

- To enable real-time, end-to-end voice-based communication in Sinhala through the development of a speech-to-speech chatbot.

- To improve the accuracy and effectiveness of Sinhala voice-based interactions by leveraging modern ASR, TTS, and language modelling technologies.
- To demonstrate the feasibility and usability of a Sinhala speech-to-speech chatbot across everyday conversational scenarios.

2.7.3 Research Questions

1. **How can pre-trained models and transfer learning techniques be leveraged to enhance the accuracy of ASR models for the Sinhala language?**

   Using transfer learning methods to improve the accuracy of ASR systems is a popular approach for enhancing models in high-resource languages. According to the literature review, the latest Sinhala ASR research by Nanayakkara and Weerasinghe (2023) achieved a WER of 17.19%, an improvement over the LF-MMI model used in previous research Gamage et al. (2021) . In this research,a method is expected to be implemented for the model developed by Nanayakkara and Weerasinghe (2023) to further improve the accuracy of the Sinhala ASR model.

2. **What methods can be employed to develop an effective Sinhala language chatbot using LLMs with a limited dataset?**

   Employing RAG can greatly improve the functionality of a Sinhala-language chatbot, or any chatbot developed in a low-resource language setting. RAG's combination of information retrieval with generative language modelling helps mitigate the limitations of smaller datasets by allowing the chatbot to draw on external knowledge sources. This method equips Sinhala chatbots to produce accurate, context-sensitive responses, offering a robust solution for languages with limited computational resources or training data. This study will focus on exploring the application of RAG in the development of a Sinhala-language chatbot, examining how it leverages external knowledge bases to optimize response quality.

3. **How can deep learning-based models be effectively adapted and fine-tuned to synthesize natural and intelligible Sinhala speech?**

   While deep learning has revolutionized TTS technology, creating highly natural and intelligible speech in low-resource languages like Sinhala continues to face significant challenges. These challenges arise from the unique phonetic and syntactic characteristics of Sinhala, which are not well-captured by models primarily trained in more common languages. Adapting and fine-tuning DLB models for Sinhala requires specialized approaches to address these linguistic nuances. This research seeks to explore the effective adaptation and fine-tuning of DLB TTS models to enhance the naturalness and

intelligibility of synthesized Sinhala speech, ultimately advancing the applications of TTS technology for the Sinhala language.

4. **How can the integration of ASR, TTS and chatbot functionalities be streamlined for real-time communication?**

   Integration of ASR, TTS and the chatbot system is critical for providing a smooth and responsive user experience. This explores the technical and architectural considerations necessary to ensure real-time interaction without latency.

Chapter 3

Analysis and Design

3.1 Chapter Overview

This chapter provides an overview of the research method and approach used to analyze and design the system. The process of data collection and validation is further discussed in the chapter. The methodology section provides a detailed analysis of the methodologies experimented and used for the Sinhala language ASR, chatbot and TTS modules. The architectural design of the system, highlighting its main components and functions is also presented in this chapter.

3.2 Research Approach

The research approach employed in this project is based on the Design Science Research (DSR) methodology. DSR is a problem-solving approach that seeks to design and develop innovative solutions to complex problems across various domains, including engineering, computer science, and business. It involves a cyclical process of creating, evaluating, and refining artifacts such as software systems, models, and processes to address practical issues and enhance understanding within a given domain (see Figure 3.1). The DSR methodology is distinguished by its emphasis on practical relevance, rigorous evaluation, and the integration of both theoretical and practical knowledge. The primary objective of DSR is to generate knowledge through the creation of artifacts that offer practical value in real-world applications.

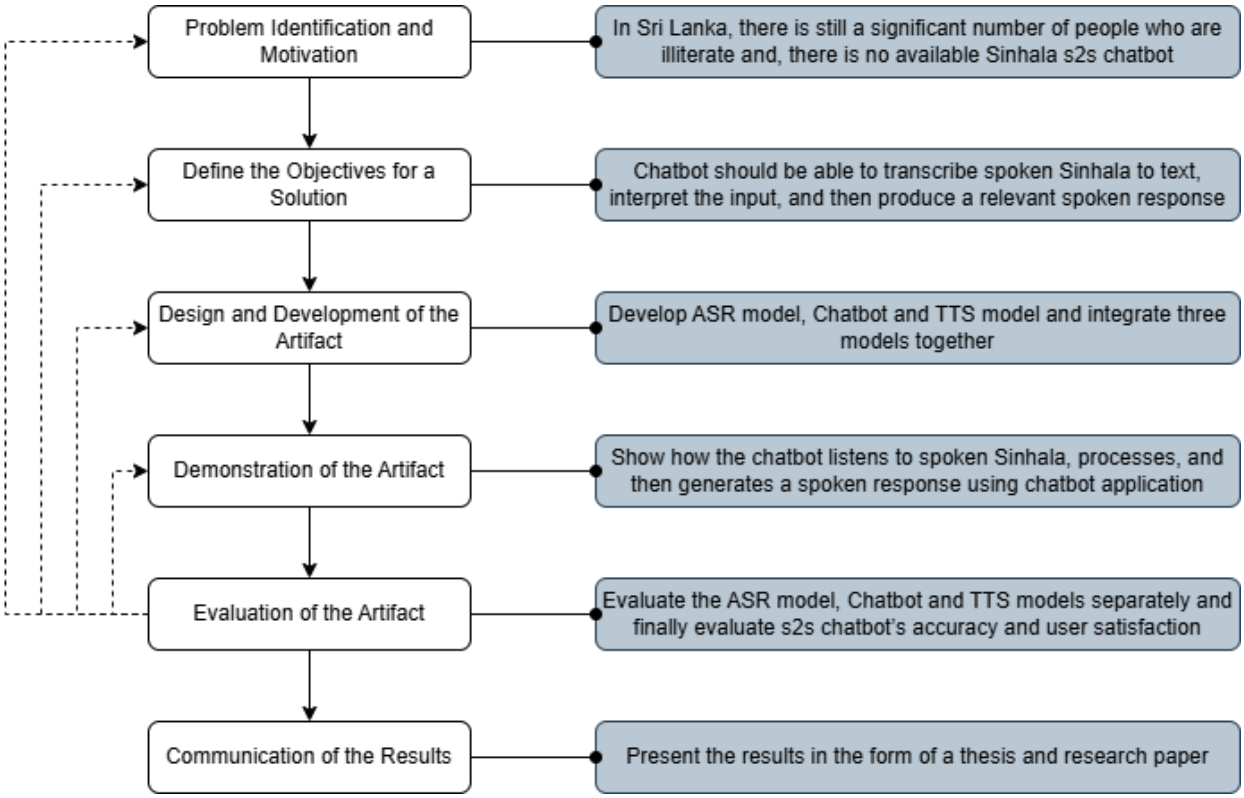

Figure 3.1: Steps of Design Science Research Methodology

3.3 Software Development Approach

The software product was developed using an iterative and incremental software development approach, as it is a research-based product where the requirements and timeline are not clearly defined at the outset. Iterative and incremental software development is a methodology that emphasizes a cyclical process of development and refinement, focusing on delivering functional software in small increments.

In iterative development, the development team addresses a subset of the software requirements at a time, creating a working prototype or minimum viable product (MVP) that can be tested and evaluated. Feedback from users and stakeholders is incorporated to refine and improve the product, with the process repeating for each new subset of requirements until the entire product is completed. In incremental development, the product is built in stages, with each stage introducing a new set of features or functionalities. Each stage builds upon the previous one, with continuous feedback and testing. This allows the development team to adjust the project scope based on feedback and evolving requirements, enabling a more flexible and adaptable development process.

3.4 Methodology

The following subsections outline the methodology adopted to develop the components of the proposed Sinhala speech-to-speech chatbot system, including ASR, TTS, and the chatbot , along with the experimental setups and configurations employed at each stage.

3.4.1 Automatic Speech Recognition

Data Collection and Preparation

Pre-collected speech datasets from the Language Technology Research Lab (LTRL) at UCSC were used in this study. Before training the models, the dataset was pre-processed. The transcriptions, originally in CSV format, contained 40 hours of Sinhala speech data, with corresponding audio files in WAV format. Special characters were removed from the transcriptions for both models, and the CSV file was then modified to meet the specific formatting requirements of each model.

According to Nanayakkara and Weerasinghe (2023), the dataset was created using two different software programs. Below table 3.1 shows the content of the UCSC LTRL dataset.

Software	**Male Speakers**	**Female Speakers**	**Total Speakers**
Praat	14	31	45
Redstart	23	55	78

Table 3.1: ASR dataset content

The UCSC LTRL dataset was selected for this study due to its suitability for dataset augmentation and its high level of speaker variability, both of which are crucial for improving the robustness and generalization of ASR models. In terms of dataset augmentation, the dataset's relatively clean and structured recordings allow for the application of various augmentation techniques such as time-stretching, pitch shifting, and noise injection without introducing excessive artifacts. Additionally, since the dataset was collected using two different software tools, Praat and Redstart, the variations in recording conditions naturally introduce acoustic diversity, which simulates real-world environments and enhances model robustness. Moreover, the dataset's 40-hour duration provides a sufficiently large corpus for augmentation strategies, enabling the creation of expanded training sets to improve ASR performance.

Another key factor in selecting this dataset is its extensive speaker variability. With contributions from 123 speakers, including 37 male and 86 female speakers, the dataset ensures a diverse range of vocal characteristics, which is essential for training an ASR model that generalizes well across different voices. The inclusion of recordings from multiple speakers helps address variations in accent, pitch, and speaking rate,

all of which are critical challenges in Sinhala ASR. Furthermore, since the recordings were collected using different software tools, they inherently capture variations in microphone types, audio quality, and recording environments, further enhancing the dataset's diversity.

The dataset was partitioned into training, validation, and testing subsets to ensure effective model training and evaluation. Specifically, 70% of the data was allocated for training, 20% for validation, and 10% for testing. This split was chosen to maintain a balance between providing sufficient data for training while reserving adequate samples for validation and testing to assess model performance. The training set was used to optimize model parameters, while the validation set helped fine-tune hyperparameters and prevent overfitting. The test set, which remained unseen during training, was used to evaluate the final model's generalization ability. This distribution ensured a robust assessment of the ASR model's accuracy and effectiveness across different subsets of the dataset.

As a pre-processing step, the audio files were resampled to 16 kHz to match with expected input. Transcriptions were cleaned by removing special characters and formatted appropriately. The dataset format should be as follows as in the figure 3.2.

- **path** - The path of the audio file
- **sentence** - The transcription of the audio file

path	sentence
/kaggle/input/sinhala-asr-v3/final_dataset/final_dataset/LSD_data/train/F071/wav16000/F071_122.wav	සඳට පුලුවන්ද සාවුන් අල්ලන්ට
/kaggle/input/sinhala-asr-v3/final_dataset/final_dataset/LSD_data/train/F071/wav16000/F071_123.wav	මාම්පිටියේ බණ්ඩාර
/kaggle/input/sinhala-asr-v3/final_dataset/final_dataset/LSD_data/train/F071/wav16000/F071_124.wav	ලසන්නාගේ අකල් වියෝව
/kaggle/input/sinhala-asr-v3/final_dataset/final_dataset/LSD_data/train/F071/wav16000/F071_125.wav	මුගුරු පාලිය
/kaggle/input/sinhala-asr-v3/final_dataset/final_dataset/LSD_data/train/F071/wav16000/F071_126.wav	එමෙන් ම හංසයා බෙදුම්වාදයට දක්ෂයකු වන්නට පුළුවන

Figure 3.2: Format of the dataset

Experiments with ASR Models

Whisper

OpenAI's Whisper, an open-source multilingual ASR model, was fine-tuned to develop an ASR system. The Whisper model is a Transformer-based encoder-decoder architecture designed for multilingual speech recognition tasks. It comprises an encoder that processes audio inputs into hidden state representations and a decoder that generates text transcriptions from these representations. According to Radford et al. (2023), the model was pre-trained on 680,000 hours of labeled data, including 117,000 hours of multilingual ASR data, enabling it to generalize across various languages and domains. However, the multilingual dataset used

by Whisper contains only 5 hours of Sinhala speech data, which is insufficient for accurately recognizing Sinhala.

The Whisper feature extractor and tokenizer were configured to process the audio inputs and their corresponding transcriptions. The feature extractor converted audio waveforms into log-Mel spectrograms, while the tokenizer encoded the transcriptions into token sequences. These components ensured that the inputs were in a format suitable for the Whisper model.

A data collator was implemented to batch the processed data effectively. This component handled padding and ensured that batches were uniform in size, which is crucial for efficient training and optimal GPU utilization.

The training arguments have been defined as follows. The model trained with a per-device batch size of 8 and gradient accumulation steps set to 2, which allows for effective training despite memory constraints. A learning rate of 0.00005 is used, and the warmup steps are set to 500 to gradually increase the learning rate at the start. The training is limited to 9000 steps which is 10 epochs, with gradient checkpointing and mixed-precision enabled to reduce memory usage and accelerate training. The model is evaluated every 1000 steps, using WER as the metric, with the best model loaded at the end based on the lowest WER. The model is set to be pushed to the Huggingface Hub after training (Link to huggingface).

According to the figure 3.3, the Wandb matrices shows an increase in loss over time, suggesting potential overfitting or instability in the model's learning. The samples per second and steps per second both increase, indicating that the model's processing speed improves as training progresses. The runtime decreases, which can suggest a more efficient evaluation or a reduction in processing load. Finally, the WER initially decreases sharply, reflecting improvements in model accuracy.

During the transcription process, some text outputs included special characters, indicating issues with character encoding. These characters represent cases where the model was unable to properly decode certain segments of the audio.

Wav2Vec2-XLSR

In the second experiment of this study, we fine-tuned Facebook's open source Wav2Vec 2.0 XLSR (Cross-Lingual Speech Representation) model to develop a Sinhala ASR system, using transfer learning. According to Baevski et al. (2020), the Wav2Vec 2.0 XLSR model is a Transformer-based architecture designed for multilingual speech recognition tasks. It comprises a convolutional feature encoder that processes audio inputs into latent representations and a Transformer context network that captures contextual information from these representations. The model was pre-trained on 56,000 hours of unlabeled multilingual speech data, enabling it to learn cross-lingual speech representations applicable to various languages.

Additionally, a column named "audio" should be added to the dataset. The audio data should be decoded

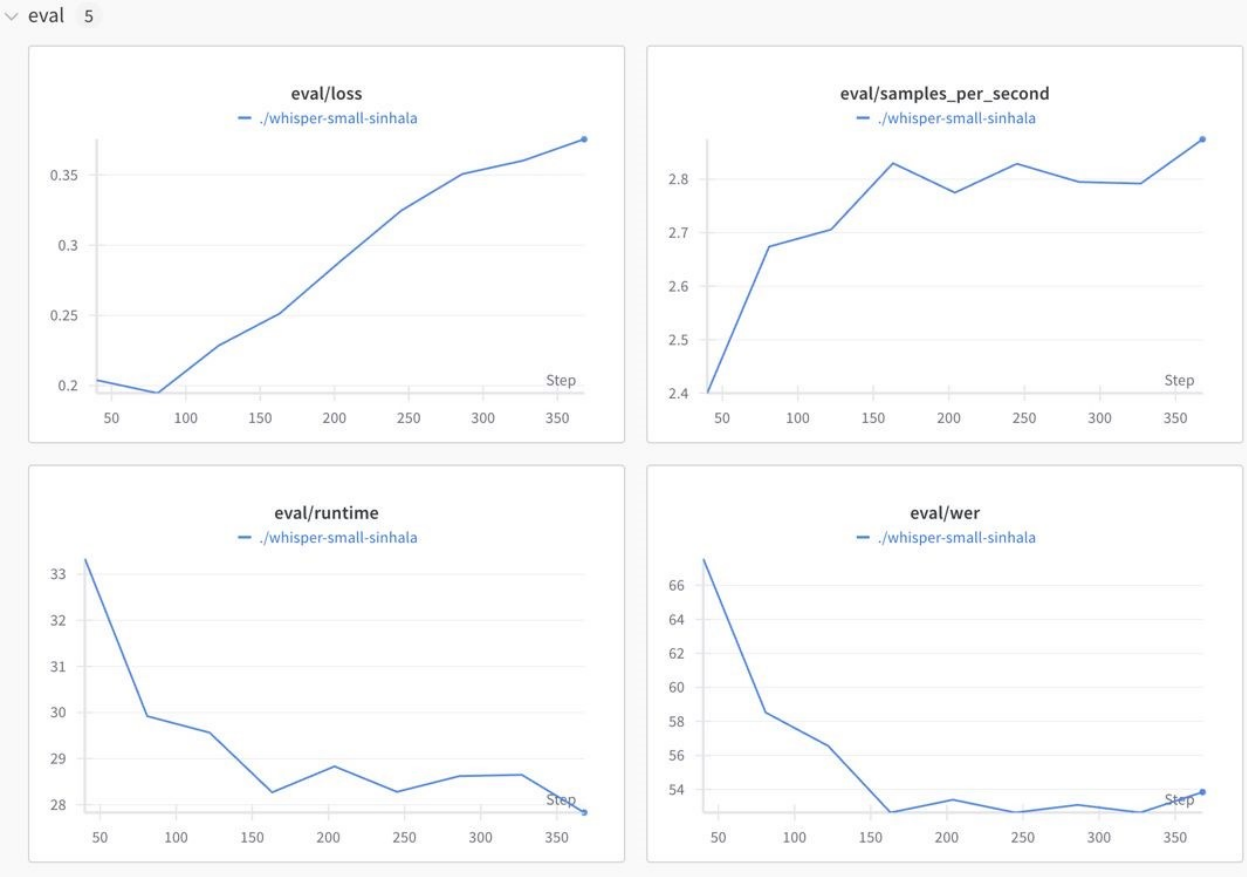

Figure 3.3: Wandb evaluation metrics for the Whisper model

and converted into a one-dimensional array with a sampling rate of 16,000 Hz, and then stored in the "audio" column.

We employed the Wav2Vec 2.0 feature extractor and a tokenizer tailored for the Sinhala language to process the audio inputs and transcriptions. The feature extractor converted audio waveforms into latent speech representations, while the tokenizer encoded the transcriptions into token sequences. These components ensured that the inputs were in a format suitable for the Wav2Vec 2.0 XLSR model.

A data collator was implemented to batch the processed data effectively. This component handled padding and ensured uniform batch sizes, which is crucial for efficient training and optimal GPU utilization.

Finally, the training arguments have been defined. We set the learning rate to 0.00005 and used a batch size of 8 per device, with gradient accumulation steps set to 2. The model is trained for 10 epochs, with gradient checkpointing enabled to optimize memory usage. We save and evaluate the model every 1000 steps, logging progress at the same interval. Additionally, the warmup steps are set to 500, and the total

number of saved checkpoints is limited to 2. For efficient checkpoint management, a checkpoint was pushed and saved to the Hugging Face Hub every 1000 steps completed (Link to huggingface).

The model training process was visualized using Wandb. According to the figure 3.4, the evaluation loss, which decreases rapidly initially and then stabilizes, indicating convergence. The evaluation WER also decreases, showing improvement in transcription accuracy. The samples per second and steps per second measure the model's performance speed, which fluctuates slightly but remains consistent overall. The runtime graph shows the time taken for each evaluation step.

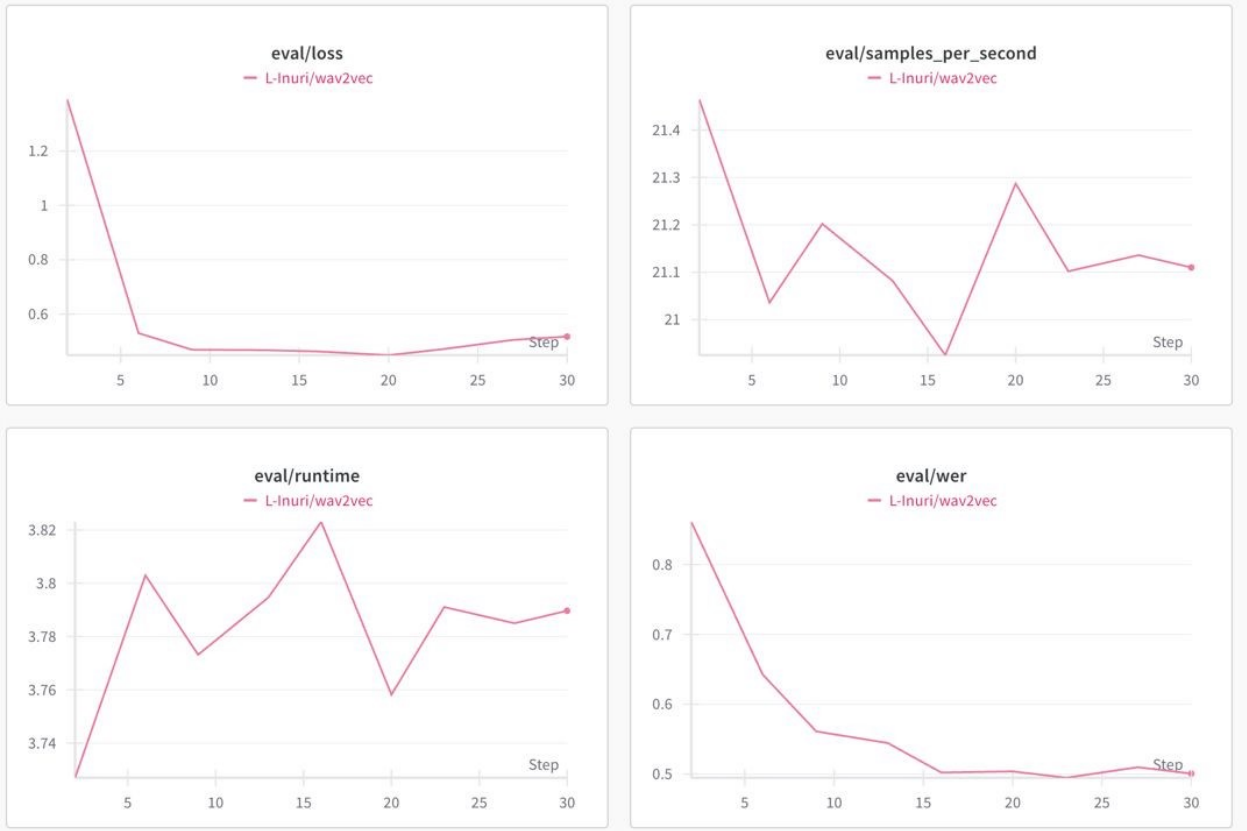

Figure 3.4: Wandb evaluation metrics for the Wav2Vec2.0-XLSR model

Wav2Vec2-BERT

In the third experiment, we fine-tuned the Wav2Vec2-BERT model to develop a Sinhala ASR system. According to Barrault et al. (2023), Wav2Vec2-BERT is a self-supervised speech representation model that combines the methodologies of Wav2Vec 2.0 and BERT, integrating contrastive learning and masked language modeling to learn contextualized speech representations.

Additionally, a column named "audio" should be added to the dataset. The audio data should be decoded and converted into a one-dimensional array with a sampling rate of 16,000 Hz, and then stored in the "audio" column.

We employed the SeamlessM4TFeatureExtractor and a Wav2Vec2CTCTokenizer tailored for the Sinhala language to process the audio inputs and transcriptions. The feature extractor converted audio waveforms into latent speech representations, while the tokenizer encoded the transcriptions into token sequences. These components ensured that the inputs were in a format suitable for the Wav2Vec2-BERT model.

A data collator was implemented to batch the processed data effectively. This component handled padding and ensured uniform batch sizes, which is crucial for efficient training and optimal GPU utilization.

During the fine-tuning of the Wav2Vec2-BERT model for Sinhala ASR, we employed the Trainer class from the Transformers library on the Kaggle platform. The training process was configured with a learning rate of 5e-5 to ensure stable convergence. We set the per-device training batch size to 8 and utilized gradient accumulation steps of 2, effectively achieving a total batch size of 16, which balances memory constraints and training efficiency. A warmup phase of 500 steps was incorporated to gradually increase the learning rate, aiding in model stabilization during initial training phases. The training was scheduled for a maximum of 5000 steps, with evaluations conducted every 1000 steps to monitor performance metrics, particularly the WER. Logging intervals were set at every 25 steps to capture detailed training progress, and model checkpoints were saved every 1000 steps to safeguard against potential data loss in huggingface (Link to huggingface). To optimize memory usage and training speed, gradient checkpointing and mixed precision training (fp16) were enabled. Post-training, the best-performing model, determined by the lowest WER, was automatically loaded, ensuring optimal performance for subsequent evaluations.

The given Figure 3.5, the WER graph demonstrates a steady decline over the training process, indicating improved transcription accuracy. As training progresses, the WER decreases significantly, showing that the model effectively learns from the dataset. Similarly, the CER follows the same trend, reinforcing that the model is making fewer character-level mistakes over time. The evaluation loss graph also shows a consistent downward trajectory, with a sharp decline in the initial phases followed by gradual improvement, which is typical of deep learning models as they converge.

The steps per second and samples per second graphs remain relatively stable, with minor fluctuations. These graphs suggest that training speed was maintained without major slowdowns, although slight variations could be attributed to differences in dataset complexity or computational loads. The runtime graph indicates a mostly stable evaluation duration, with occasional fluctuations, potentially due to system resource allocation. Overall, the steady decrease in error rates, coupled with the stable processing speed, confirms that the Wav2Vec2-BERT model effectively learned from the dataset, optimizing its transcription performance over the course of training.

3.4.2 Chatbot

Proposed Method

This section presents the proposed approach for employing Retrieval-Augmented Generation (RAG) to tackle the challenges in Sinhala Natural Language Processing (NLP). The method combines advanced retrieval mechanisms with generative models to produce accurate and contextually relevant responses to user queries. The system is designed to effectively manage various types of queries by integrating its components into a well-structured and efficient architecture.

The system architecture, depicted in Figure 3.6, is structured to handle user queries in Sinhala and provide precise, contextually appropriate responses. Central to this architecture is the Interface, which acts as the main interaction point, allowing users to upload Sinhala documents, submit their questions, and receive corresponding responses.

The proposed method follows a step-by-step process to handle user interactions efficiently. Initially, the user uploads a Sinhala document through the interface, which is then stored in the backend (RAG-C1) for further processing. The document is subsequently loaded (RAG-C2) and divided into smaller chunks of approximately 500–1000 characters (RAG-C6), a size chosen to balance semantic coherence and embedding model limitations. This chunking ensures meaningful context is preserved while keeping input lengths within the acceptable token limits of the embedding model. Each chunk is then processed by the embedding model (RAG-C4), which transforms the textual data into high-dimensional vector representations. These vectors capture the semantic meaning of each chunk and are stored in the vector database (RAG-C7). The purpose of this entire process is to enable efficient semantic search and retrieval during user queries, ensuring that relevant information from the uploaded document can be accurately and quickly retrieved in response to user inputs.

Once the document processing is complete, the chat interface becomes available for user queries related to the uploaded document. Queries can be submitted via text or voice. If a voice query is received, it is converted into text using the ASR component. The text query is then passed through the same embedding model used for document processing to generate its vector representation (RAG-C8).

A semantic search is conducted in the vector database to identify content from the uploaded document that closely matches the query. The retrieved content is provided as context to the large language model, which generates a contextually appropriate response (RAG-C9). The text response is then sent to the front-end. If the user requests a voice response, the TTS component is activated to convert the text into speech before delivering it to the user.

This sequence outlines the internal workflow of the proposed method, with specific control points in the RAG pipeline and techniques implemented to optimize chatbot performance at each stage(Akkiraju et al.

2024).

Experimentation

Experiments were conducted at each of the control points mentioned in the proposed method to optimize the system's performance. To ensure the relevance of responses, the RAG model requires the uploaded document to contain related information. Various types of materials, including PDF and TXT documents, were used for testing. The pdfLoader and TXTLoader from LangChain were utilized to load these documents into the RAG model.

For document chunking, both chunk size and overlap size were defined to balance efficiency and contextual coherence between chunks. Chunk size determines the number of words into which a document is segmented before being processed by the embedding model, while overlap size defines the number of shared words between consecutive chunks. Selecting appropriate values for these parameters is crucial for maintaining efficiency and preserving contextual links. To determine optimal values, experiments were conducted with different configurations, considering the limitations of the embedding model, and responses were evaluated accordingly.

Regarding the embedding model, multiple models were tested to identify one best suited for Sinhala language processing and question-answering tasks. LangChain provides a range of embedding models (LangChain 2025), and Hugging Face maintains a leaderboard ranking various models (HugginFace 2025). Experiments were performed with different embedding models, and responses were analyzed to assess their effectiveness.

The next consideration was selecting a vector database suitable for semantic search operations, which is essential for retrieving relevant information from stored document embeddings. Experiments were conducted with various open-source vector databases, and their performance in retrieving relevant content was evaluated.

Finally, the LLM plays a critical role in generating responses. Various open-source LLMs were tested, focusing on their capability to process Sinhala language inputs and perform question-answering tasks effectively. Experiments were conducted using models such as LLAMA, and their responses were assessed to determine their suitability for the system.

3.4.3 Text to Speech

Data Collection and Preparation

For this study, the publicly available Pathnirvana Sinhala TTS dataset was utilized, with several modifications implemented to suit the requirements of the experiments (PathNirvana 2023). The Pathnirvana dataset

is available in two versions: an earlier version comprising recordings from a female speaker only, and a more recent multi-speaker version that includes recordings from both male and female speakers. To enhance the quality and diversity of the training data, both versions were combined to construct three distinct datasets: a male-only dataset, a female-only dataset, and a combined multi-speaker dataset. These datasets were subsequently used for training the TTS model. A summary of the modified dataset is presented in Table 3.2.

Speaker Gender	**Speaker Name**	**Total Audio Length**	**Number of Audio Clips**
Male	Ven. Mettananda	11.8 hours	5400
Female	Mrs. Oshadi	9 hours	4285

Table 3.2: Modified Pathnirvana Dataset Summary

The audio files are stored in WAV format, while the accompanying transcription CSV file contains Romanized text and Sinhala text corresponding to each audio file. This dual-script format enables the model to be trained using either script as input. The dataset's CSV file contained four columns as shown in Figure 3.7:

- **Path**: The path to the audio file
- **Romanized text**: The transcription of the audio file in Roman letters
- **Sinhala text**: The transcription of the audio file in Sinhala letters
- **Speaker**: The name of the speaker associated with the audio file

	A	B	C	D	E	F	G	H	I	J	K
1	sinh_0001|keṇḍaya dakvā vasālana vahan no dæriya yutu ya.|-ෙප- කෙණ්ඩය දක්වා වසාලන වහන් නො දැරිය යුතු ය.|mettananda										
2	sinh_0002|keṇḍaya dakvā vasālana vahan no dæriya yutu ya.|-ෙප- කෙණ්ඩය දක්වා වසාලන වහන් නො දැරිය යුතු ය.|mettananda										
3	sinh_0003|theyyasaṁvāsakayā upāddhyāya koṭa gena upasapan kereti.|-ෙප- ථෙය්‍යසංවාසකයා උපාද්ධ්‍යාය කොට ගෙණ උපසපන් කෙරෙති.|mettananda										
4	sinh_0004|maha anullapanādhipṛ අනුල්ලපනාධිප්‍රාය ඇතියහුට ඇවැත් නො වේ.|mettananda										
5	sinh_0005|eyin kiyanu læbeyi 'evametassa kēvalassa dukkhakkhandhassa samudayō hoti' yi.|එයින් කියනු ලැබෙයි 'එවමෙතස්ස කේවලස්ස දුක්ඛක්ඛන්ධස්ස සමුදයෝ හොනි' යි.|mettananda										

Figure 3.7: CSV File Format of the Pathnirvana Dataset

The average duration of the audio recordings is approximately 7.78 seconds, with a maximum length of 15 seconds and a minimum of 2 seconds. To ensure consistency and improve model performance, silences were removed from both the beginning and the end of each recording. All audio files are sampled at 22,050 Hz and encoded in 16-bit PCM format, adhering to the same specifications as the ljspeech dataset.

To support efficient training and thorough evaluation of the TTS model, the dataset was divided into three distinct subsets: training, validation, and testing. A proportion of 80% was allocated to the training set, 10% to the validation set, and the remaining 10% to the test set. This division was designed to ensure that the model had access to a substantial amount of data for learning, while also preserving sufficient samples for validation and testing to enable accurate performance monitoring.

The training set was employed to learn and adjust model parameters, whereas the validation set played a critical role in tuning hyperparameters and preventing overfitting. The test set, which was not used during any stage of model development, served as an independent benchmark to assess the model's generalization capabilities. This systematic data partitioning approach contributed to a comprehensive and reliable evaluation of the TTS model's effectiveness.

Experiments with VITS Model

Building upon the limitations of conventional Sinhala TTS systems discussed in Section 2.5.3, we conducted a series of experiments using the VITS model—a fully E2E variational autoencoder-based architecture known for its effectiveness in low-resource scenarios. Five configurations were evaluated by varying dataset composition (see Section 3.4.3 *Data Collection and Preparation*) and input modality (Romanized text vs. Sinhala text).

Training was performed using a single GPU on the following hardware: 4 × GeForce RTX 2080 Ti (11GB GDDR6 memory), 2 × Intel(R) Xeon(R) E5-2620 v4 @ 2.10GHz CPUs (32 cores), 128GB RAM, and 3 × 3.7TB storage, with CUDA version 10 or higher. The complete training process typically took around 24 hours; however, this duration varied depending on the dataset composition and the specific training configuration used in each model.

Romanized Text – Single-Speaker Male Dataset

As an initial experiment, the VITS model was trained using Romanized transcriptions of Sinhala sentences from the Pathnirvana dataset. The dataset's CSV file, as described in Section 3.4.3 *Data Collection and Preparation*, contained four columns. For this experiment, only the **Path**, **Romanized Text**, and **Speaker** columns were utilized, as they provided the necessary information for generating Sinhala speech from Romanized text inputs.

The Romanized transcription in the second column was selected as the reference text for training. To isolate male voice data, a speaker selection function was implemented to filter the dataset accordingly, ensuring that only male speaker recordings were used. The character set was defined using Roman letters to align with the Romanized representation of Sinhala speech.

Training was conducted with a batch size of 16 and an evaluation batch size of 32. Mixed-precision

training was enabled to improve computational efficiency. Training progress was logged every 50 steps, and model checkpoints were saved every 600 steps, with a maximum of 10 checkpoints retained to balance storage and recovery. The model was trained for 300 epochs, demonstrating effective learning and convergence in significantly fewer epochs compared to the Sinhala script-based model discussed in Section 3.4.3 *Sinhala Text – Single-Speaker Male Dataset*, which required 1,000 epochs. This indicates that the Romanized text input model achieved faster convergence and efficiency.

Figure 3.8: Validation vs. Training Loss for the Romanized Male Voice Model

Training and validation loss trends as shown in Figure 3.9 confirmed effective learning behavior with minimal overfitting. This experiment successfully demonstrated the adaptability of the VITS model for Sinhala TTS using Romanized text, producing coherent and natural-sounding Sinhala speech.

Romanized Text – Single-Speaker Female Dataset

To generate Sinhala female voice outputs from Romanized text input, the VITS model was trained using a configuration similar to that employed for the male voice model, as outlined in Section 3.4.3 *Romanized Text – Single-Speaker Male Dataset*. A speaker selection function was used to isolate female speaker data from the Pathnirvana dataset.

The same Romanized transcription column was used as input, and the character set consisted of Roman characters. The model was trained over 4,000 epochs, with other configurations retained as in the previous experiment.

Although the training loss followed a smooth decreasing trend as shown in Figure 3.9, the validation loss fluctuated significantly, indicating the possibility of overfitting or sensitivity to speaker-specific characteristics. Additional fine-tuning and regularization may be needed to improve generalization.

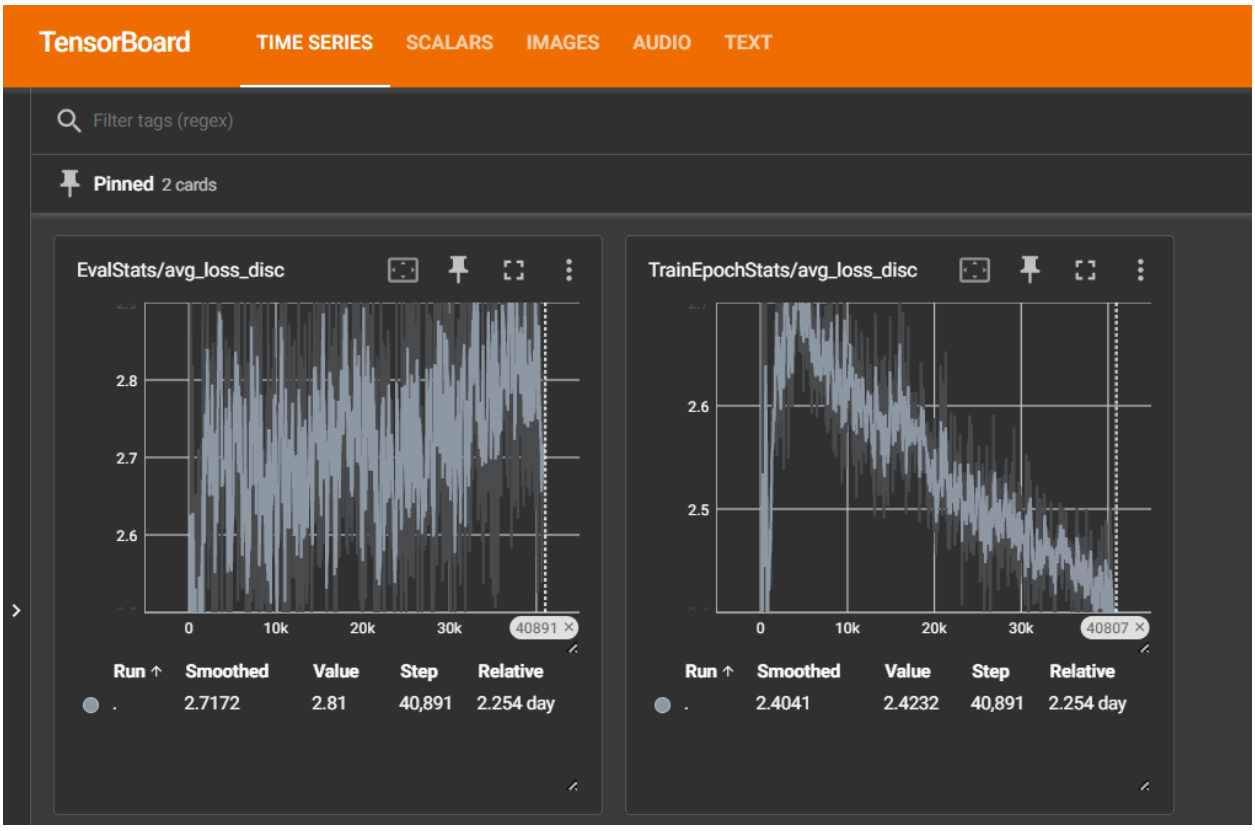

Figure 3.9: Validation vs. Training Loss for the Romanized Female Voice Model

Sinhala Text – Single-Speaker Male Dataset

Following the initial experiments using Romanized Sinhala, the next phase focused on Sinhala script input. This shift enabled direct Sinhala character input and generated synthesized speech in the native script.

The VITS model was trained using a modified version of the Pathnirvana dataset filtered for a single male speaker. The dataset columns used were **Path**, **Sinhala text**, and **Speaker**, as noted in Section 3.4.3 *Data Collection and Preparation*. The Sinhala sentence (third column) served as the model input, and a filtering function ensured only male speaker recordings were included.

The character set excluded Roman characters and included only Sinhala letters and punctuation. The training configuration was identical to the Romanized Sinhala setup (see Section 3.4.3 *Romanized Text – Single-Speaker Male Dataset*), except that the model was trained for **1,000 epochs** instead of 300.

Figure 3.10: Validation vs. Training Loss for Single-Speaker Male Sinhala Model

TensorBoard metrics indicated consistent improvement throughout training. Both validation and training loss curves, as shown in Figure 3.10, exhibited a clear downward trend, despite some fluctuations in validation

loss. This suggests stable convergence and effective adversarial training.

Sinhala Text – Single-Speaker Female Dataset

This model configuration largely mirrored the male speaker setup as Section 3.4.3 *Sinhala Text – Single-Speaker Male Dataset*, but was adapted for a female voice using filtered Pathnirvana data. The same data columns were used, and speaker filtering ensured that only female recordings were used.

Key differences included an extended training duration of 4,000 epochs and increased data loader workers (8 for training, 4 for evaluation) to handle processing more efficiently. Mixed-precision training and Sinhala character set configurations remained unchanged.

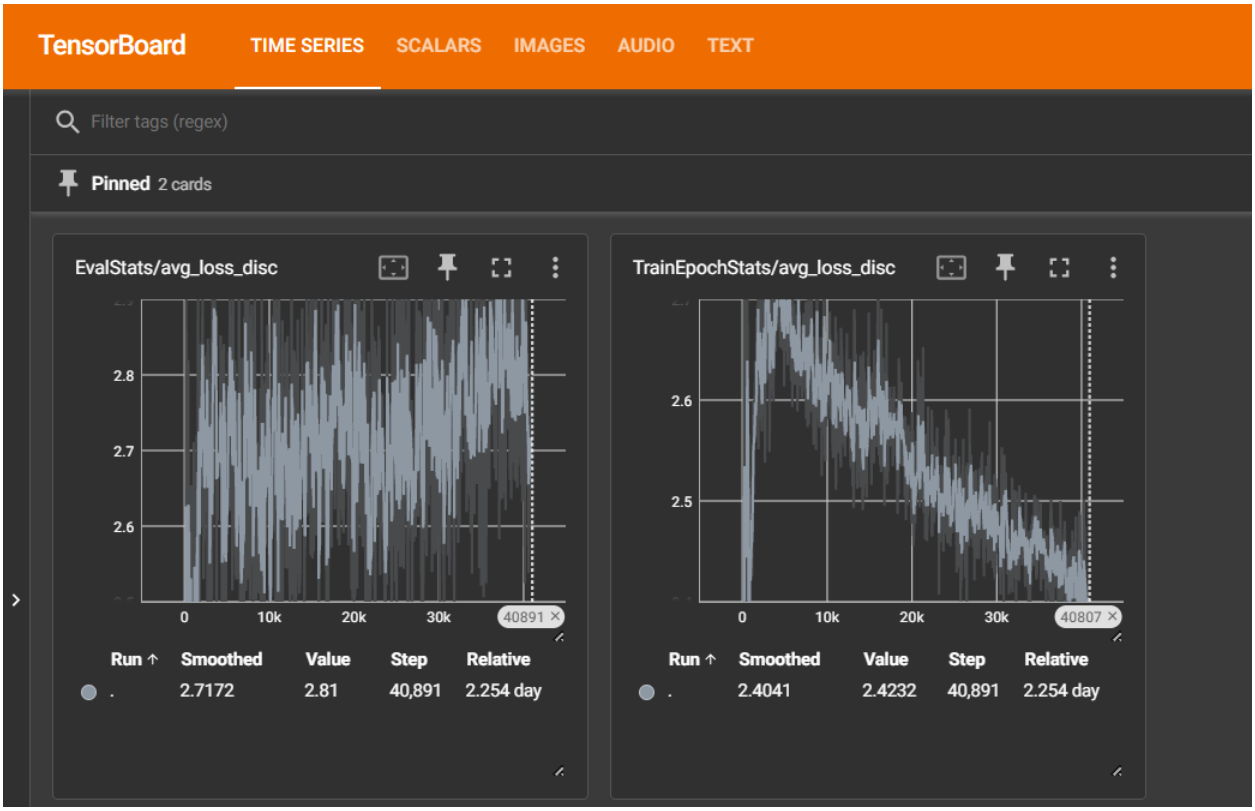

Figure 3.11: Validation vs. Training Loss for Single-Speaker Female Sinhala Model

Training metrics showed learning progress as shown in Figure 3.11; however, the validation loss fluctuated significantly, suggesting overfitting and weaker generalization. The alignment plots lacked the clarity seen in the male model, indicating the need for additional fine-tuning, possibly due to a smaller dataset size.

Sinhala Text – Multi-Speaker Dataset

This configuration extended the previous models by incorporating both male and female speaker data to train a multi-speaker VITS model. The same columns—**Path**, **Sinhala sentence**, and **Speaker**—were used. No speaker filtering was applied.

The model used a customized Sinhala character set and accepted both text and speaker ID at inference time to synthesize speech in the corresponding voice. Training parameters remained consistent: batch size of 16, evaluation batch size of 32, 1,000 epochs, and logging/checkpoint intervals as before.

Figure 3.12: Validation vs. Training Loss for Multi-Speaker Sinhala Model

TensorBoard plots (Figure 3.12) showed a clear downward trend in loss metrics. Alignment plots were consistently diagonal, confirming effective attention learning across speakers. The duration prediction improved steadily, and output waveforms resembled realistic speech, demonstrating the model's flexibility and effectiveness across speaker variations.

Overall, these experiments highlighted key insights across speaker configurations and input types. Sinhala

script inputs yielded better alignment, while multi-speaker models demonstrated superior generalization and speaker diversity. However, Romanized training remained faster in convergence.

Preprocessing Methods

Text preprocessing is a crucial component in any TTS system, especially for low-resource languages such as Sinhala. In this research, a set of language-specific normalization techniques were developed and implemented to improve the quality and intelligibility of synthesized speech. As Sinhala lacks comprehensive natural language processing libraries and toolkits, all preprocessing steps were manually designed using rule-based methods, relying primarily on regular expressions with Python's `re` library.

All these preprocessing steps shown in Figure 3.13 were implemented as a part of the number normalization module and were critical to ensuring natural and accurate pronunciation in Sinhala TTS. The system uses a custom converter function to transform digit strings and Sinhala abbreviations into their full spoken forms in Sinhala. This preprocessing pipeline played a vital role in bridging the gap between written and spoken Sinhala in the TTS model.

Category	Description
Abbreviations	Certain Sinhala abbreviations, such as "ෙප.ව.", are frequently used in written text but are pronounced differently. For instance, "ෙප.ව." is pronounced as "ෙපරවරු". A rule-based approach was developed to expand such abbreviations to their full spoken forms, ensuring accurate pronunciation in generated speech.
Currency Handling	In Sinhala, rupees are denoted by the abbreviation "රු." but pronounced as "රුපියල්", while dollars and pounds are represented by the symbols "$" and "£", respectively. Recognizing these symbols and converting them into spoken equivalents is crucial for natural speech synthesis. A rule-based mechanism utilizing regular expressions was implemented to detect and expand currency symbols in the text. For example, the value "රු.150.50" was normalized to "රුපියල් එකසිය පනහයි සත පනහ", reflecting the appropriate spoken format.
Decimal Points	Although decimal points are written using a period ("."), they are pronounced using the Sinhala term "දශම". For instance, the number "12.5" is expanded to "ෙදාළහයි දශම පහ". This transformation was handled through a custom rule-based strategy that distinguishes decimal numbers from other numeric formats and expands them accordingly.
Number Expansion	Sinhala follows a rule-based grouping strategy similar to English, where large numbers are segmented into three-digit units starting from the right. These segments are then expanded into full spoken forms based on predefined patterns and keywords. For example, the number "123456" is grouped as "123,456" and pronounced as "එකසිය විසි තුන් දහස් හාරසිය පනස් හය" (one hundred twenty-three thousand four hundred fifty-six). Similarly, "12345" becomes "දොළොස් දහස් තුන්සිය හතලිස් පහ" (twelve thousand three hundred forty-five). This rule-based approach supports number expansion up to the scale of trillions.

Figure 3.13: TTS Number Normalization Steps

3.5 System Architecture

3.5.1 Use Case Diagram

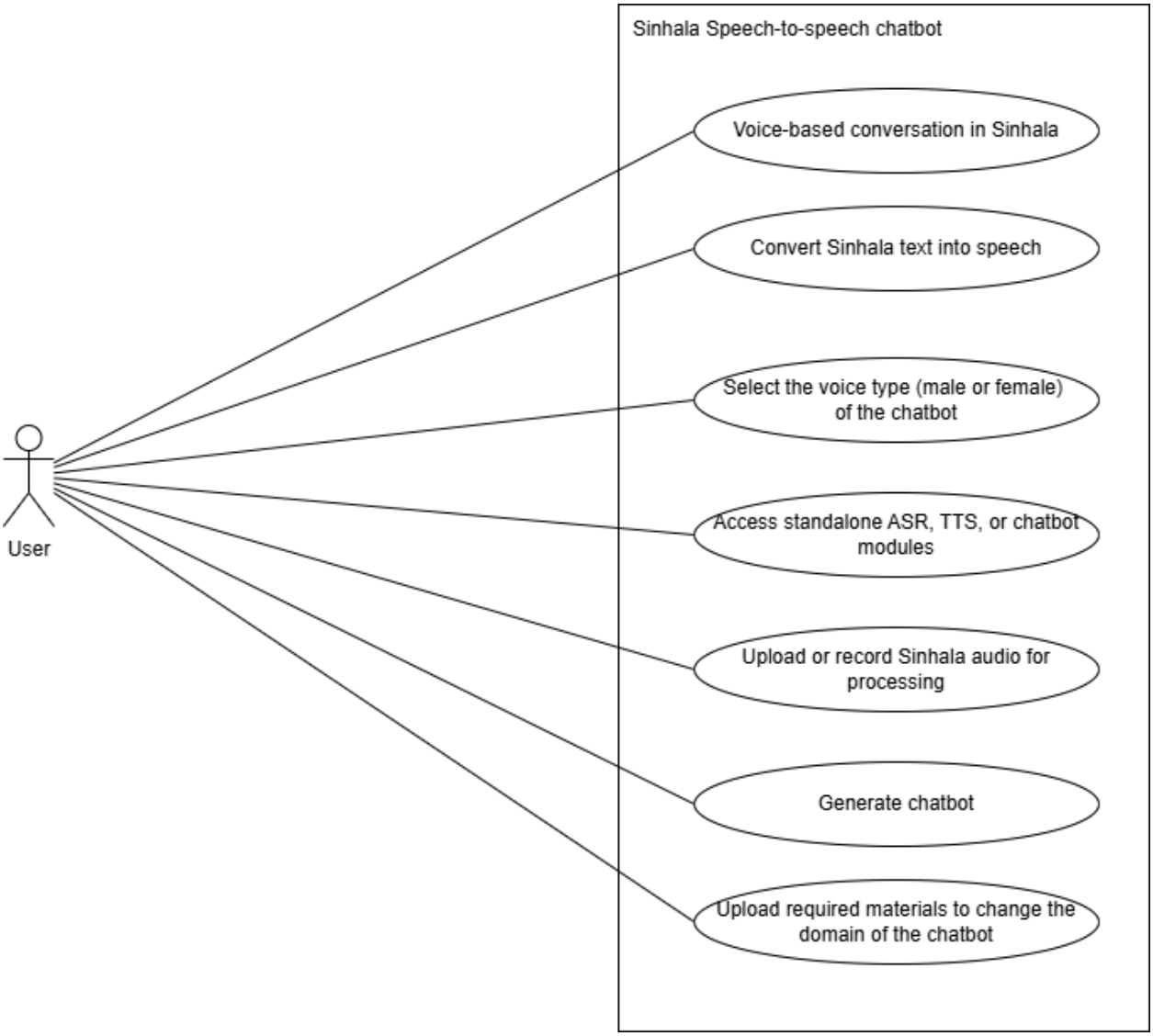

Figure 3.14: Use case diagram

Figure 3.14 presents the use case diagram of the Sinhala speech-to-speech chatbot, showing the various interactions a user can perform with the system. These include engaging in Sinhala voice conversations, selecting voice type, uploading audio or domain-specific data, and accessing individual ASR, TTS, or chatbot

modules. The diagram highlights the system's flexibility and user control in customizing and using the chatbot.

3.5.2 Activity Diagram

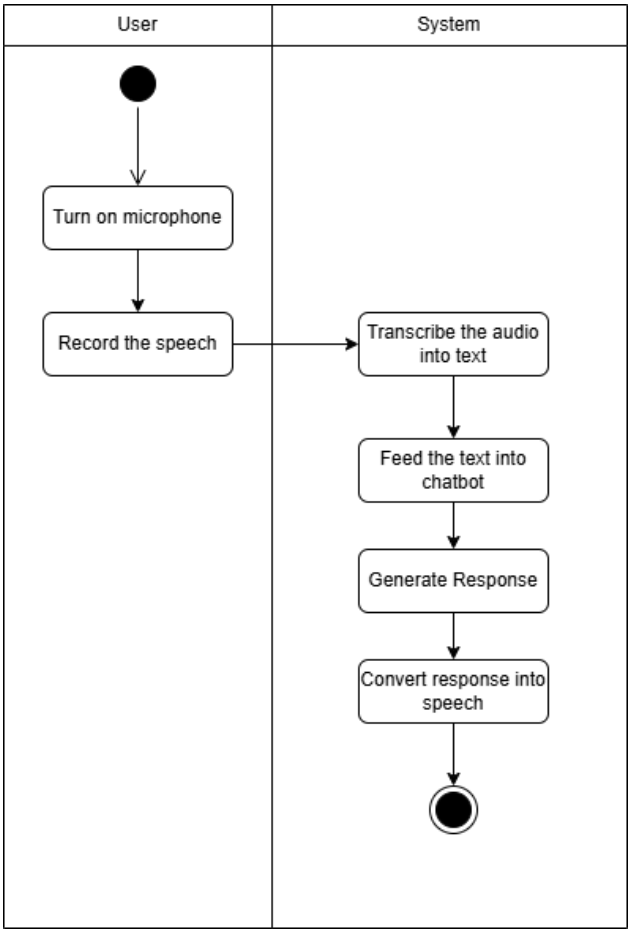

Figure 3.15: Activity diagram

Figure 3.15 illustrates the activity flow of the Sinhala speech-to-speech chatbot system. The user initiates the process by recording speech, which the system transcribes into text, processes through the chatbot, and converts the generated response back into speech. This sequence enables seamless spoken interaction in Sinhala.

3.6 Product Workflow Diagram

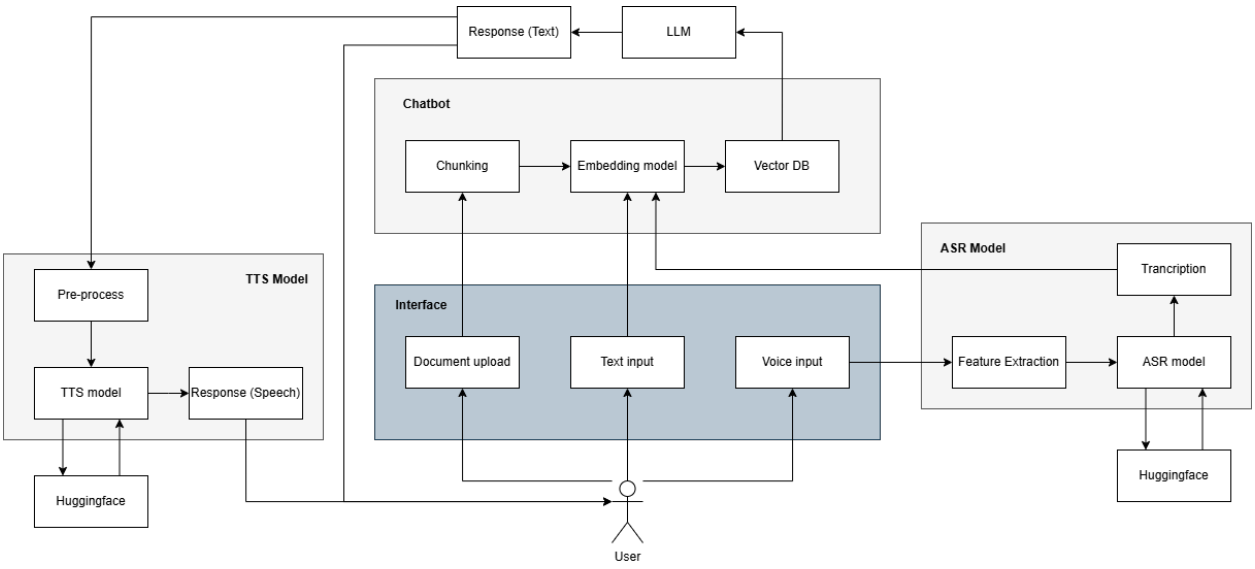

Figure 3.16: Product workflow diagram

Figure 3.16 illustrates the product workflow of the Sinhala speech-to-speech chatbot system. It shows how user inputs—whether voice, text, or documents—are processed through ASR, chatbot, and TTS components, with integration of models from Hugging Face and a large language model (LLM) for generating accurate responses. The diagram highlights the complete data flow from input to speech output.

3.7 Functional Requirements

- The system shall enable users to engage in voice-based conversations in Sinhala.
- The system shall allow users to upload or record Sinhala audio for processing.
- The system shall transcribe spoken Sinhala audio into text using the ASR module.
- The system shall provide a graphical interface that includes options to record audio and send input.
- The system shall generate chatbot responses based on transcribed user input.
- The system shall convert Sinhala text into speech using the TTS module.
- The system shall enable users to select between different speaker types and voice preferences.

- The system shall provide functionality to play back the generated audio output to the user.
- The system shall support access to standalone ASR, TTS, and chatbot modules independently.
- The system shall allow users to upload domain-specific materials to customize or retrain the chatbot for specific domains.
- The system shall support Sinhala language processing for both input (speech/text) and output (text/speech).
- The system shall maintain real-time conversational flow with minimal latency.

3.8 Quality Attributes

- **Usability**

  The system provides a minimal and user-friendly interface to ensure a smooth interaction for users. The intuitive design reduces the cognitive workload and improves the overall experience when engaging with the Sinhala speech-to-speech chatbot.

- **Performance**

  The system leverages trained ASR and TTS models hosted on Hugging Face and uses FastAPI to make real-time API calls. This cloud-based approach eliminates the need for local model storage and enhances response time, making the process seamless and fast.

- **Maintainability**

  By decoupling model hosting and application logic, the system allows easy updates and replacements of ASR and TTS models through Hugging Face without modifying the application code. This supports easier maintenance and version control.

- **Scalability**

  The use of cloud-based APIs allows the system to efficiently scale with the number of users. As user demand increases, backend infrastructure can be expanded without affecting the frontend interface or user experience.

- **Modularity**

  The architecture separates ASR, TTS, and chatbot components into independent modules. This modular design facilitates isolated development, testing, and reuse across different use cases or domains.

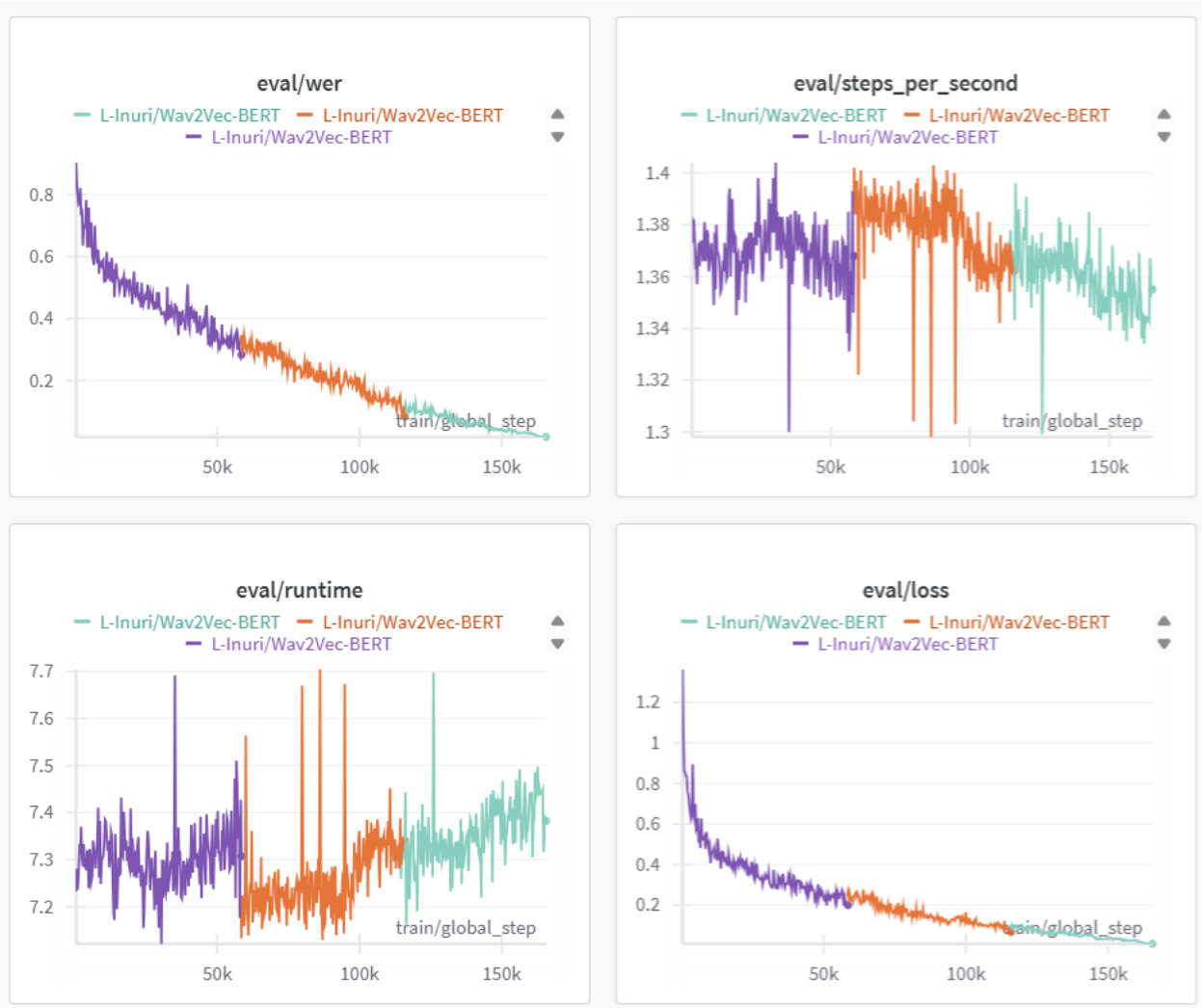

Figure 3.5: Wandb evaluation metrics for the Wav2Vec 2.0-BERT model

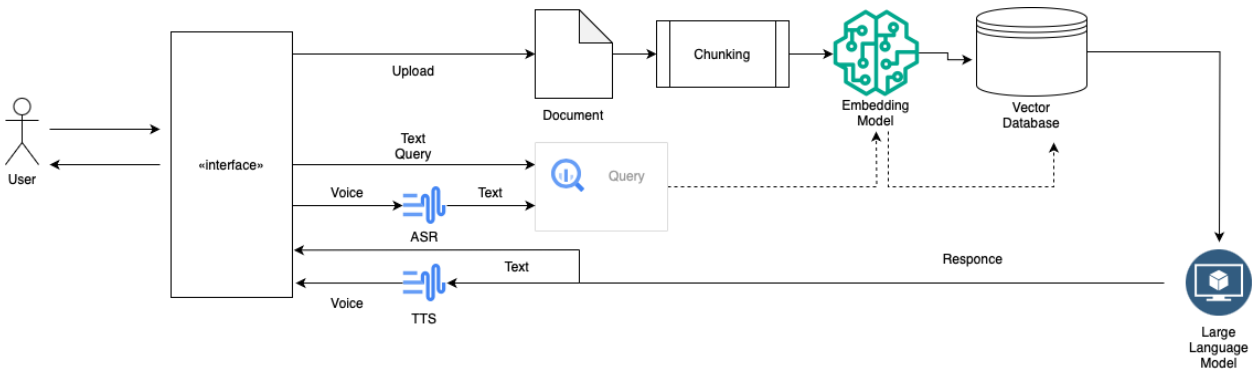

Figure 3.6: The system architecture demonstrates the interaction between the Interface, the vector database, and the RAG system, ensuring smooth query processing and response generation.

Chapter 4

Implementation

4.1 Chapter Overview

This chapter presents the implementation of the ASR, Chatbot, and TTS components. It begins by detailing the individual stages involved in developing each module—ASR, Chatbot, and TTS, highlighting the data processing pipelines, model selection, and deployment strategies. The chapter then describes the integration of these components to form a complete speech-to-speech chatbot system. Furthermore, the design, functionality, and user-friendliness of the system's user interfaces are discussed, emphasizing seamless interaction and accessibility for Sinhala-speaking users.

4.2 Automatic Speech Recognition

The implementation of the ASR model involved multiple stages, including data preprocessing, model selection, and deployment. Initially, three different models Wav2Vec2-XLSR, Whisper, and Wav2Vec2-BERT were trained and evaluated to determine the most suitable model for Sinhala ASR.

4.2.1 Data Reading

To support real-time transcription, the ASR system was designed to accept Sinhala speech recorded directly from the device's microphone. As mentioned in the chapter 3, under Wav2Vec2-BERT model, the audio files should be in WAV format. Therefore, the recorded audio is captured on the client side and transmitted to the server in WAV format. Upon receiving the file, it is temporarily saved on the server and processed to ensure compatibility with the selected ASR models. This includes converting all audio inputs to a standard sampling rate of 16kHz using librosa, and ensuring mono-channel audio by averaging stereo channels when

necessary. These preprocessing steps guarantee consistent input across models and contribute to improved transcription accuracy. The entire backend pipeline was developed using FastAPI, which enables seamless interaction with the frontend interface and provides efficient audio handling for real-time transcription.

4.2.2 Model Selection

Three pre-trained models were evaluated: Wav2Vec2-XLSR, Whisper, and Wav2Vec2-BERT. Each model was fine-tuned using transfer learning on a 40 hour Sinhala speech dataset. The experiments were conducted on the Kaggle platform, with training metrics visualized using Wandb. Performance evaluation was based on WER, CER, and validation loss trends. As mentioned in chapter 5 section 5.2.3, among the three models, the Wav2Vec2-BERT model showed the best results in terms of transcription accuracy and overall performance, as evidenced by the lowest WER and CER curves. Based on this comparative analysis, the Wav2Vec2-BERT model was selected for final deployment.

4.2.3 Generating the Transcription

After the audio input is received and pre-processed, it is passed to the selected Wav2Vec2-BERT model, which generates a transcription of the spoken Sinhala input. This raw transcription is then refined through a custom post-processing stage developed to enhance its linguistic correctness and user readability. Specifically, a rule-based approach was employed to determine the type of sentence whether it is a question or a statement. For this, a list of Sinhala question-indicating words was manually extracted from the UCSC Sinhala corpus. A function was implemented to detect the presence of these question words within the transcribed sentence. If such indicators were found, the system would automatically append a question mark to the sentence; otherwise, a full stop would be added. This process not only improves punctuation consistency but also makes the transcription more natural and grammatically appropriate for downstream use in conversational interfaces or language-based applications.

4.3 Chatbot

After evaluation, the system was developed to support the upload of Sinhala text (TXT) and PDF documents. The intfloat/multilingual-e5-large-instruct model was selected as the embedding model. The chunk size was set to 514, as it represents the maximum token limit that the embedding model can process. An overlap size of 20 was chosen to minimize redundant content between adjacent chunks while preserving semantic continuity. FAISS was selected as the vector database because of its superior efficiency in performing a semantic search compared to other open-source alternatives. To overcome computational constraints, Groq API(GroqCloud 2025) was used to make calls to the large language model, and llama-3.3-70b-versatile was

identified as the most suitable model based on the evaluation results. With these configurations, the system is capable of processing Sinhala PDF and TXT documents and generating relevant, human-like responses to queries based on the provided content.

4.4 Text-to-Speech

The implementation of the TTS system focuses on generating natural Sinhala speech based on user-defined input type and voice preferences. This section describes the complete pipeline, including user input handling, dynamic model selection, and speech synthesis. The backend was developed using `FastAPI` to ensure efficient and responsive communication with the frontend, supporting real-time synthesis.

4.4.1 User Input Handling

The system first prompts users to choose the input text type—either **Sinhala text** or **Romanized text**. As discussed in Chapter 3 Section 3.4.3, the VITS model was trained using both representations, offering flexibility based on user preference.

After selecting the input text type, users are prompted to choose a voice—**male** or **female**. A dedicated text input field is provided on the interface for entering the desired sentence in the selected format. Once the user clicks the Generate button, the request is sent to the backend for processing. The backend, powered by `FastAPI`, ensures fast and reliable handling of these requests, enabling real-time performance.

4.4.2 Model Selection

Based on the selected input type and voice, the system dynamically identifies and loads the corresponding trained VITS model. These models were selected based on a comprehensive evaluation, as detailed in Chapter 5 Section 5.4, to ensure high-quality and natural speech synthesis. This dynamic mapping guarantees that each input is handled by the most suitable model, resulting in accurate and natural-sounding speech.

4.4.3 Synthesizing the Speech

Upon receiving the request, the server performs preprocessing on the input text, as outlined in Chapter 3 Section 3.4.3 *Preprocessing Methods*. This step ensures the input is appropriately formatted for the chosen VITS model. The preprocessed text is then passed to the model, which generates the synthesized speech in the form of a waveform.

The generated audio is saved in `WAV` format on the server. It is then sent back to the frontend, where an embedded audio player presents the output. Users can listen to the synthesized speech and replay it as

needed, allowing for clear assessment of the output's naturalness and intelligibility.

4.5 Integration

Following the successful implementation of the three core components, ASR, Chatbot, and TTS—they were integrated to function as a unified voice-based system. When a user inputs a query through speech, the ASR module is triggered to convert the spoken input into textual form. This transcribed text is then passed to the chatbot module, which processes the input and generates an appropriate textual response. Subsequently, the TTS module converts this response into speech format, allowing the system to deliver the final output back to the user in both text and audio forms.

4.6 User Interfaces

This research project delivers a user-friendly Sinhala speech-to-speech chatbot system enriched with several key features. It includes an ASR module that transcribes Sinhala speech into text, both speech-based and text-based chatbot interactions, and a TTS module that converts Sinhala text back into speech. Additionally, the system empowers users to create their own domain-specific chatbots by uploading relevant documents.

4.6.1 Chatbot

The chatbot interface is designed to support both speech-based and text-based interactions in Sinhala. Users can either speak or type their queries, and the system responds appropriately using natural language understanding. As shown in Figure 4.1, the interface is intuitive, allowing smooth and responsive communication.

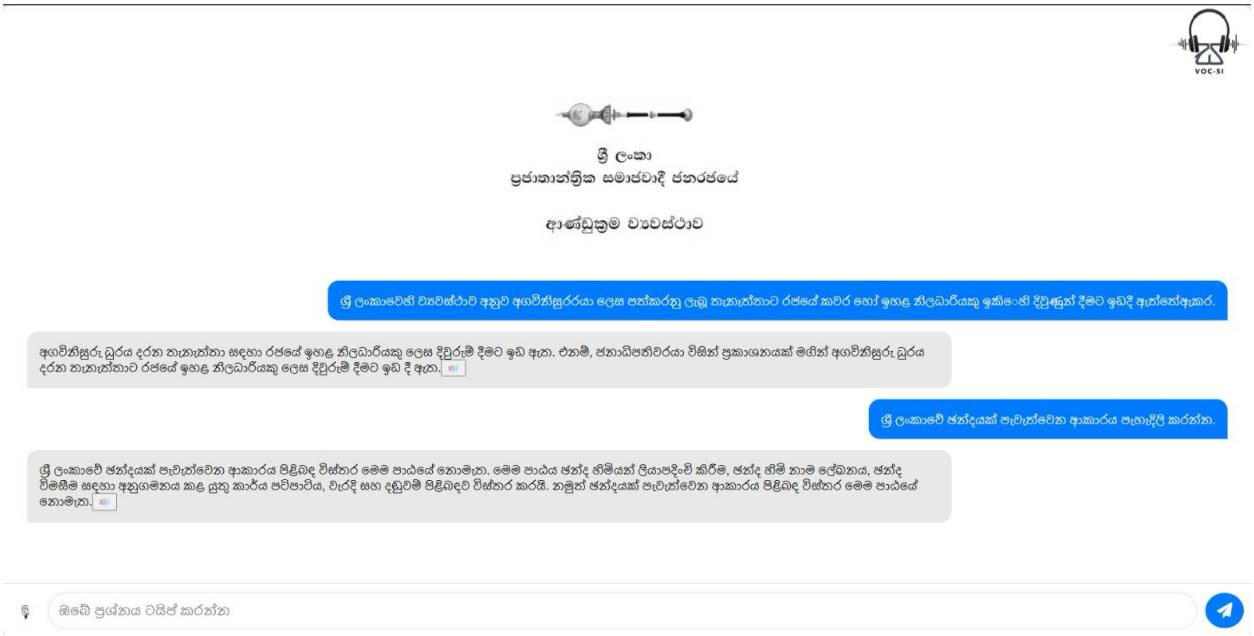

Figure 4.1: Chatbot interface

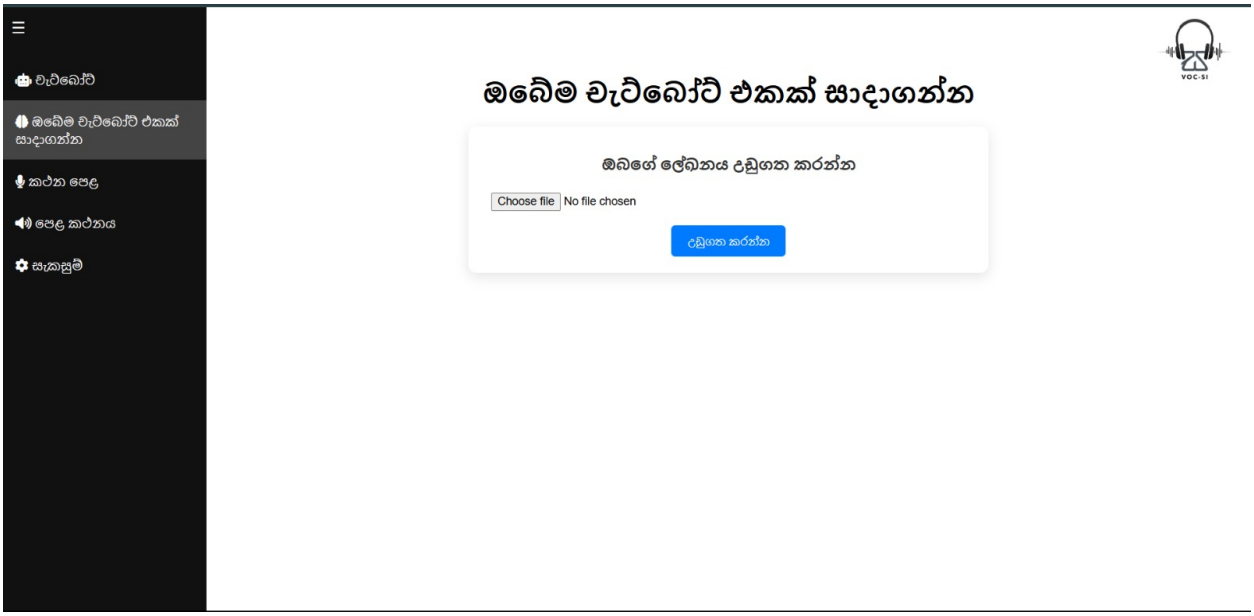

Figure 4.2: Generate chatbot

An additional feature is the ability to generate custom chatbots tailored to specific domains. As illustrated

in Figure 4.2, users can upload relevant documents, and the system will create a chatbot capable of answering queries based on the content provided. This makes the platform adaptable for various fields such as tourism, finance, or healthcare.

4.6.2 Automatic Speech Recognition

ASR module plays a vital role in enabling seamless human-computer interaction through speech. As shown in Figure 4.3, the interface allows users to input their speech in Sinhala, which is then transcribed into written text. This functionality is particularly beneficial for users who find it difficult to type in Sinhala or prefer speaking over writing, such as the elderly, visually impaired, or individuals with lower literacy levels.

The ASR system captures the user's voice through the device's microphone and processes the audio signal using a deep learning-based model trained on a large corpus of Sinhala speech data. The model converts the waveform into phonetic units, which are then mapped to corresponding words using a language model to ensure syntactic and grammatical correctness.Once the transcription is complete, the recognized text is automatically displayed in the interface.

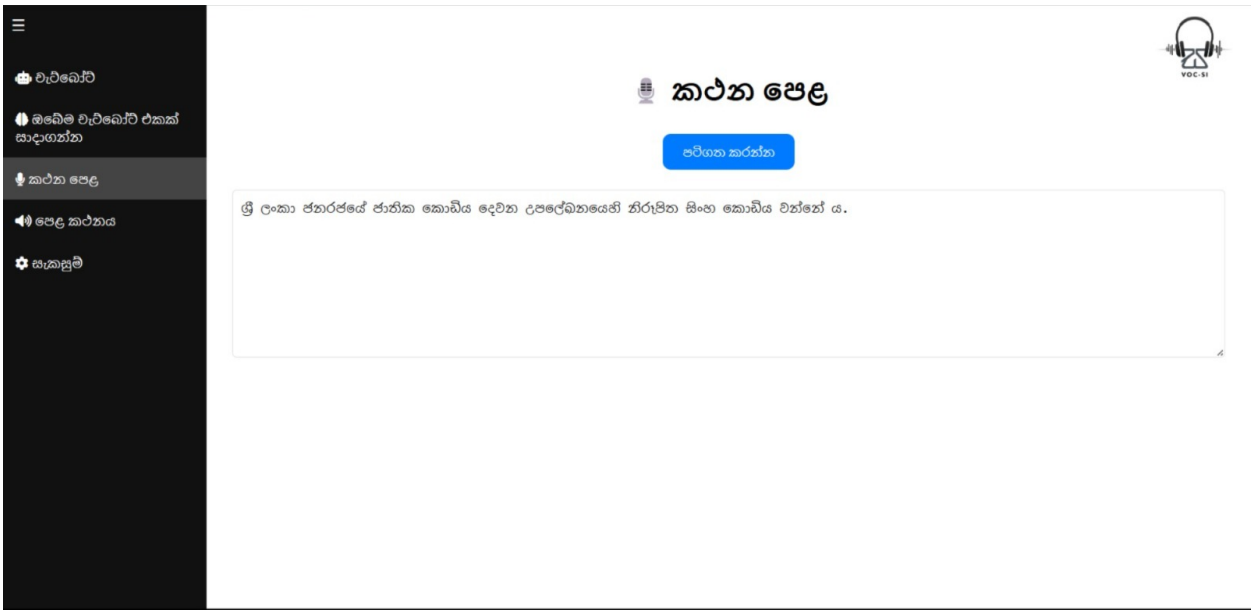

Figure 4.3: ASR interface

4.6.3 Text to Speech

As depicted in Figure 4.4, TTS interface enables the conversion of Sinhala text into natural-sounding spoken Sinhala. This functionality enhances accessibility for users who prefer auditory content, as well as for those with visual impairments, reading difficulties, or limited literacy skills. The output speech aims to be both natural and intelligible, preserving the prosody and rhythm of native Sinhala.

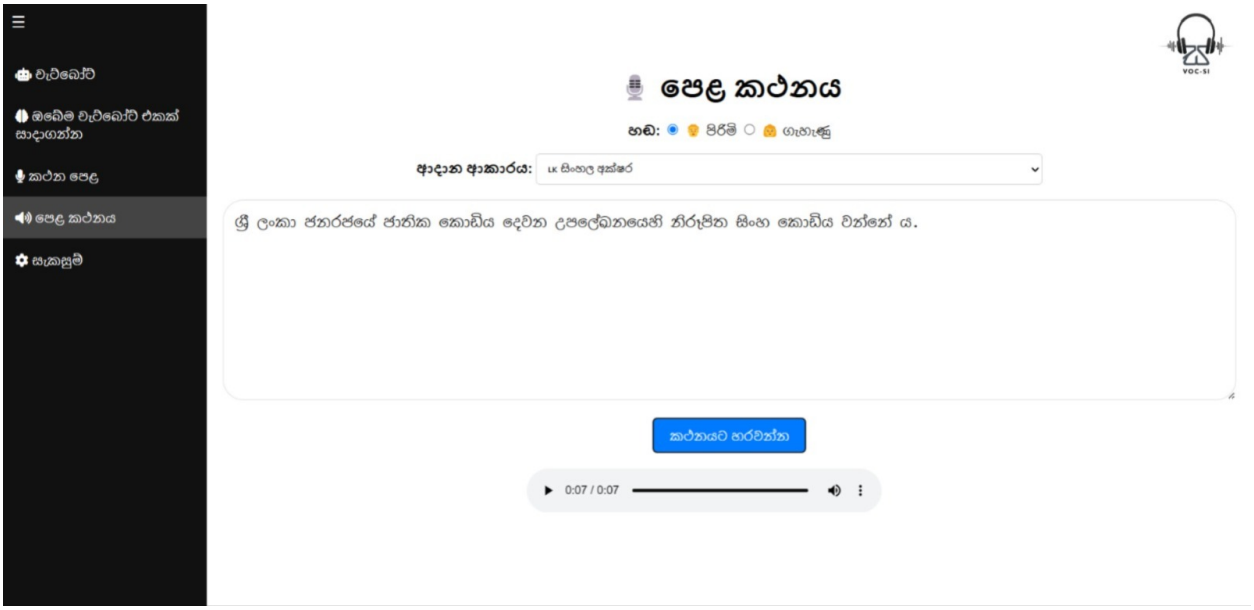

Figure 4.4: TTS interface

Chapter 5

Evaluation and Results

5.1 Chapter Overview

This chapter outlines the evaluations conducted for the ASR, Chatbot, and TTS models, as well as for the final product, the Sinhala Speech-to-Speech Chatbot. The evaluations are carried out through a combination of qualitative and quantitative methodologies. It provides an overview of the evaluation process, including the evaluation questions, the methods applied throughout the research, and the results derived from the evaluation.

5.2 Evaluation of the ASR Models

To ensure a fair and reliable assessment of the proposed ASR models, a separate test dataset was used for evaluation purposes. This dataset was strictly isolated from the training and validation sets to prevent any data leakage and to ensure that the models' generalization ability could be properly measured. Each model was evaluated only after the completion of the training process using this unseen test dataset.

To quantify the transcription accuracy of the models, two widely accepted evaluation metrics were used: WER and CER. These metrics are particularly useful in speech recognition tasks as they measure how closely the predicted text matches the reference transcription at both word and character levels, respectively. Both metrics are expressed as percentages, with lower values indicating better performance.

WER evaluates the number of word-level errors in the predicted transcription. It takes into account the number of substitutions (S), insertions (I), and deletions (D) required to convert the predicted sentence into the reference sentence, normalized by the total number of words (N) in the reference sentence. The formula is given by:

$$\text{WER} = \frac{S + D + I}{N} \tag{5.1}$$

CER is a similar metric to WER, but operates at the character level instead of the word level. It is especially useful for evaluating ASR systems working in languages like Sinhala, where word boundaries can be complex. CER is calculated using the same components (S, D, I), but with the normalization factor being the total number of characters (C) in the reference transcription:

$$\text{CER} = \frac{S + D + I}{C} \tag{5.2}$$

To ensure a fair comparison between the models, all ASR models were trained using the same set of hyperparameters. Specifically, a learning rate of 0.00005, batch size of 16, and gradient accumulation steps set to 2 were used consistently across the Whisper, Wav2Vec2-XLSR, and Wav2Vec2-BERT models. By keeping these parameters constant, the effect of the model architecture and feature extraction strategies on recognition performance could be more accurately assessed. The evaluation results were then compared with the existing state-of-the-art Sinhala ASR systems to validate the effectiveness of the proposed models.

5.2.1 Whisper Model

The Whisper model achieved a WER of 55.8% on the test dataset without the use of any external language model or post-processing techniques. This result highlights the model's baseline capability to transcribe Sinhala speech accurately, even in the absence of additional linguistic support.

Figure 5.1 illustrates several example transcriptions generated by the trained Whisper model. While the model was able to capture the overall structure and content of the audio in many cases, certain limitations were observed. Notably, the outputs of the third and fourth predictions in Figure 5.1 contain replacement symbols, indicating potential character encoding issues or incomplete decoding by the model. These anomalies typically occur when the model fails to correctly interpret or generate certain segments of the audio input, possibly due to noise, unclear pronunciation, or underrepresented phonetic patterns in the training data.

Such occurrences underscore the need for integrating post-processing mechanisms and language modeling in future iterations to refine transcription quality, handle edge cases more effectively, and produce syntactically and semantically cleaner outputs.

Given these limitations, especially the instability of the generated text in the absence of a language model, a decision was made to proceed with training a second model, Wav2Vec2 XLSR. This model architecture is known for its robustness in handling multilingual and low-resource scenarios through transfer learning and

Ground Truth	**අලාබ නම් වෙළඳාම අත්හැර දමන එකයි කරන්න තියෙන්නේ**
Prediction	අලාභ නම් වෙළඳාම අත්හැර දමනිකයි කරන්න තියෙන්නේ
Ground Truth	**එම කාමරය තුළ හුස්ම ගැනීමේ අපහසුවක් ඔහුට දැනුණි**
Prediction	එම කාමරය තුළ හුස්ම ගැනීමේ අපහසුවක් ඔහුට දැනුනි
Ground Truth	**ඒ බයානක සතාගෙං බේරිලා දුවනව වෙනුවට මං උඟව පැගුව**
Prediction	ඒ භයානේ සතාගෙන බේරිලා දුවනව�වෙනුවට මංඋව පැගුව��
Ground Truth	**ඔවුන්ගේ අරමුණ ගැන කියනව නම් මා ඇත්තම කියන්නම්**
Prediction	ඔවුන්නේ අරමුණ ගැන කියන �නම් මා ඇත්තම කියන්නම්
Ground Truth	**මේ ලියුම් පත් බොහෝම කාලයක සිට පාවිච්චි කරනවා ද**
Prediction	මේ ලියුම්පත් බොහොම කාලයක සිට පාවිච්චි කරනවාද

Figure 5.1: Transcriptions generated with transfer-learning based Whisper model

has shown promising results in prior studies. The shift aimed to improve transcription quality and address the decoding issues observed with Whisper, especially in terms of character-level and word-level accuracy.

5.2.2 Wav2Vec2-XLSR Model

The Wav2Vec2-XLSR model achieved a WER of 50.2% on the Sinhala test dataset, even without utilizing any external language model or applying post-processing techniques. This result indicates a noticeable improvement in transcription accuracy compared to the previously evaluated Whisper model in figure 5.1. As shown in Figure 5.2, the trained model was able to produce more coherent and accurate transcriptions across a variety of speech samples. When comparing Figure 5.1 and Figure 5.2, it is noticeable that the character encoding issue was resolved in the results of the Wav2Vec2-XLSR model.

The Wav2Vec2-XLSR architecture is particularly well-suited for low-resource languages like Sinhala due to its multilingual pre-training on a diverse set of languages. This allows the model to generalize better and handle linguistic variations more effectively. In the absence of a language model, the outputs retained a relatively high degree of fluency and correctness, with significantly fewer encoding or character-level issues than

Figure 5.2: Transcriptions generated with transfer-learning based Wav2Vec 2.0 model

those observed with the Whisper model. However, there are still some notable issues with the transcriptions, such as spacing errors in the first and fifth sentences in figure 5.2.

These encouraging results motivated the exploration of further improvements using a more advanced variation of the Wav2Vec2 architecture, Wav2Vec2-BERT, which incorporates contextual embeddings to enhance language understanding during the transcription process.

5.2.3 Wav2Vec2-BERT Model

The Wav2Vec2-BERT model produced the most accurate results among all the evaluated models. It achieved an average WER of 1.79% and an average CER of 0.33%, which demonstrates a substantial improvement over the previously tested models. These metrics highlight the model's strong ability to transcribe Sinhala speech with minimal errors, even without relying on an external language model.

Figure 5.3 presents a set of qualitative results, comparing the model's predictions against the ground truth sentences. The predictions closely match the reference transcriptions, capturing even subtle nuances in Sinhala speech. Unlike earlier models such as Whisper and Wav2Vec2-XLSR, this model shows robust

performance across diverse sentence structures, significantly reducing omissions, substitutions, and encoding issues. When comparing the results of the previous Whisper model (Figure 5.1) and the Wav2Vec2-XLSR model (Figure 5.2), the Wav2Vec2-BERT model provides the best results. All the spelling errors, spacing issues, and decoding errors identified in the previous model outputs have been resolved in the results of the Wav2Vec2-BERT model.

Figure 5.3: Transcriptions generated with transfer-learning based Wav2Vec 2.0 - BERT model

The impressive performance of this model can be attributed to the integration of contextualized embeddings through the BERT architecture, which helps the system better understand the structure and semantics of Sinhala sentences. These results validate the decision to adopt the Wav2Vec2-BERT model as the final architecture for deployment.

5.2.4 Results Comparison

Model	WER	CER
Whisper (Trained)	55.8%	11.8%
Wav2Vec2-XLSR (Trained)	50.2%	9.8%
Wav2Vec2-BERT (Trained)	**1.79%**	**0.33%**
Nanayakkara and Weerasinghe (2023)	17.19%	5.9%
Buddhi Gamage, Pushpananda, Nadungodage, et al. (2021)	28.55%	-

Table 5.1: WER and CER Comparison of Sinhala ASR Models

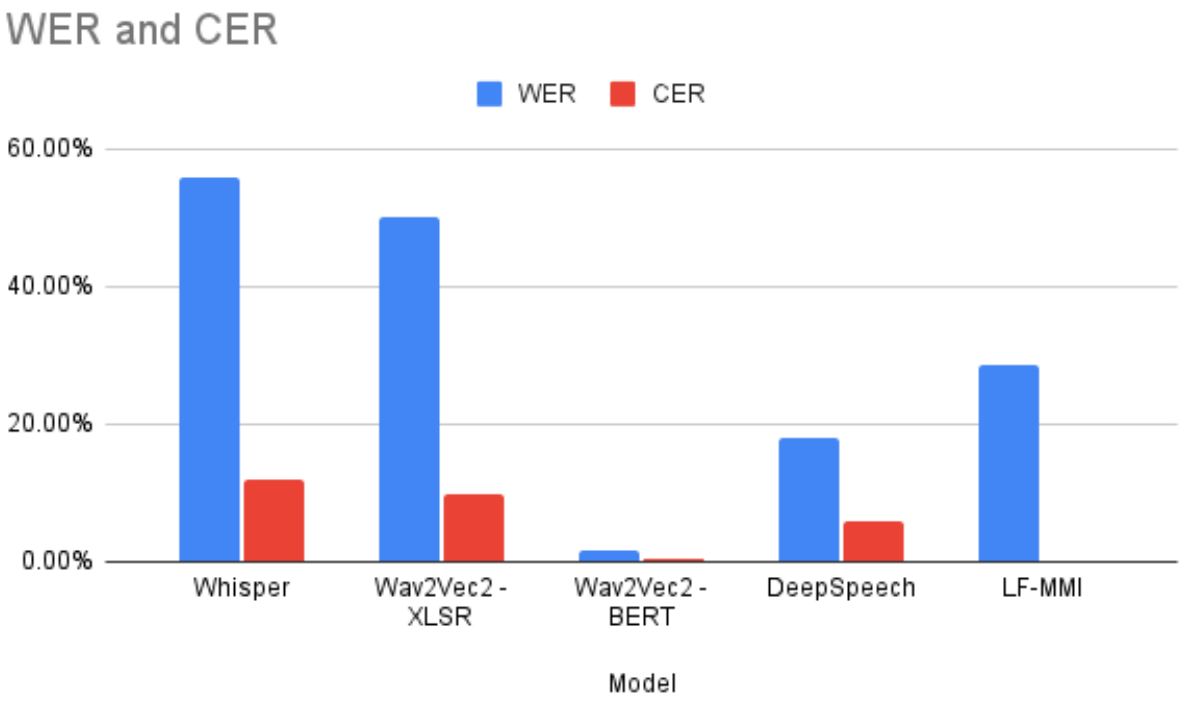

Figure 5.4: Results comparison with state of the art ASR models

The results demonstrate a clear progression in performance across the ASR models we experimented with. The Whisper model, although known for its multilingual capabilities, yielded the highest WER at 55.8% when trained on our Sinhala dataset. This indicates that more than half of the predicted words were incorrect, which we suspect is due to character encoding issues and the model's difficulty in handling certain phonetic

and linguistic structures specific to Sinhala. Additionally, Whisper did not benefit from any post-processing or external language model, further contributing to its lower accuracy.

The Wav2Vec2-XLSR model showed some improvement, reducing the WER to 50.2%. This model is specifically pre-trained on multiple languages and is designed for low-resource scenarios. However, even though it performed better than Whisper, the improvement was not substantial, and there was still a noticeable gap between its predictions and the ground truth. Like Whisper, this model was also evaluated without the use of a language model or punctuation correction, which may have limited its full potential.

The most notable advancement came from the Wav2Vec2-BERT model. This model achieved a significantly lower WER of 1.79% and an impressively low CER of 0.33%, making it the most accurate model among those tested. The integration of a BERT-based language model allowed for better linguistic context understanding, while the addition of a custom post-processing function further refined the output by correcting punctuation and sentence endings. This hybrid approach proved highly effective in minimizing both word- and character-level errors.

When compared to previously published Sinhala ASR models, our Wav2Vec2-BERT model outperformed the model by Nanayakkara and Weerasinghe (2023), which reported a WER of 17.19% and CER of 5.9%, and the model by Gamage et al. (2021), which achieved a WER of 28.55%. These results highlight the strength of our model architecture and training methodology, and position Wav2Vec2-BERT as a new state-of-the-art baseline for Sinhala ASR.

5.3 Evaluation of the Chatbot

The evaluation process involved testing the system with a set of 20 questions with answers derived from the Constitution of Sri Lanka text document. Questions and relevant answers were used to assess the system's relevance, accuracy, and consistency. These questions, along with the corresponding answer options, are provided in Appendix B. The generated responses were manually validated by comparing them with the correct answers. Figure 5.5 displays sample questions and relevant answers in Sinhala.

ප්‍රශ්නය	පිළිතුරු
වෘත්තීය සමිති පිහිටුවීමේ සහ වෘත්තීසමිතිවලට බැඳීමේ නිදහසක් ව්‍යවස්ථාවෙන් ලබා දී ඇත්තේ වරප්‍රසාද ලත් පිරිසකට පමණක් ද?	නැත. සෑම පුරවැසියෙකුටම ලබා දී තිබේ
ශ්‍රී ලංකා ජනරජයේ පරමාධිපත්‍යය ඇත්තේ විධායක ජනාධිපතිවරයාට ද?	නැත. එය ඇත්තේ ජනතාවටයි
ශ්‍රී ලංකා පුරවැසියෙකු අල්ලස් චෝදනාවකට වරදකරු වී පස් වසරක් ගත වී ඇත. ඔහුට පාර්ලිමේන්තු මැතිවරණය සඳහා ඡන්දය ලබා දිය හැකි ද? එසේ නොවන්නේ නම් ඊට ඇති නීතිමය බාධාව කුමක් ද?	නොහැක. ඒ සඳහා වසර හතක් ගත විය යුතු ය.
ශ්‍රී ලංකා ව්‍යවස්ථාව අනුව අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ අගමැතිවරයා ද?	නැත. අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ ජනාධිපතිවරයාය.
ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාරවරයෙකු පත් කිරීමේ යම් දෝෂයක් ඇති විට ශ්‍රේෂ්ඨාධිකරණයේ පවතින නඩු කටයුත්තක් නිර්බල වන්නේ ද?	නැත. ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාර ධුරයක් හිස්ව තිබුණ ද, යම් විනිශ්චයකාරවරයෙකු පත් කිරීමේ දෝෂ තිබුණ ද එහි පවතින නඩු කටයුතු නිර්බල වන්නේ නැත.
වරක් තේරී පත් වී සිටින ජනාධිපතිවරයෙකුට නැවත ජනවරමක් ලබා ගැනීම සඳහා තේරී පත් වීමෙන් පසු අවම වශයෙන් වසර කීයක් ඉක්ම විය යුතු ද?	අවම වශයෙන් පත්වීමෙන් වසර හතරක් ඉක්ම විය යුතු ය.
ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාව සාමාජිකයින් කී දෙනෙකුගෙන් සමන්විත විය යුතු ද?	සාමාජිකයින් 10කි
අමාත්‍ය මණ්ඩලයට අයත් උපරිම ඇමතිවරුන් ගණන සහ අමාත්‍ය මණ්ඩල සාමාජිකයින් නොවන ඇමතිවරුන් සහ නියෝජ්‍ය ඇමතිවරුන්ගේ උපරිම ගණන කොපමණද?	අමාත්‍ය මණ්ඩලයට අයත් උපරිම සංඛ්‍යාව 30කි අමාත්‍ය මණ්ඩල සාමාජිකයින් නොවන ඇමතිවරුන් සහ නියෝජ්‍ය ඇමතිවරුන් උපරිම ගණන 40කි
පර්ලිමේන්තුව අවම වශයෙන් වසරකට කීවරක් කැඳවිය යුතු ද?	අවම වශයෙන් එක් වරක්වත් කැඳවිය යුතුය.
අධිකරණ සේවා කොමිෂන් සභාවෙහි යම් රැස්වීමක ගණපූරණය සඳහා සාමාජිකයින් කී දෙනෙකු සහභාගී විය යුතුද?	ගණපූරණය සඳහා සාමාජිකයින් දෙදෙනෙකුගෙන් සමන්විත විය යුතුය.

Figure 5.5: Sample question set in Sinhala Language.

Table 5.2 presents the evaluation results for different chunk and overlap sizes, while Table 5.3 illustrates the evaluation results for various vector databases.

Chunk Size and Overlap Size	**Correct Percentage (%)**
chunk size=514, chunk overlap=20	80
chunk size=1000, chunk overlap=100	70
chunk size=1000, chunk overlap=50	65
chunk size=500, chunk overlap=100	70
chunk size=500, chunk overlap=50	75

Table 5.2: Evaluation Results of Chunk Sizes & Overlap Sizes

Vector Database	**Correct Percentage (%)**
Chroma	75
FAISS	80
Qdrant	70
Weaviate	70

Table 5.3: Evaluation Results of Vector Databases

Table 5.4 summarizes the performance metrics for the various embedding models evaluated, whereas Table 5.5 outlines the results obtained from different LLMs. Figure 5.7 depicts the comparison between predicted and actual responses generated by the LLaMA-3.3-70B-Versatile model. Similarly, Figure 5.6 illustrates the evaluation outcomes for the intfloat/multilingual-e5-large-instruct embedding model, highlighting its predicted versus actual answers.

Embedding Model	**Correct Percentage (%)**
intfloat/multilingual-e5-large-instruct	80
Ransaka/sinhala-sentence-transformer	20
Ollama	20

Table 5.4: Evaluation Results of Embedding Models

Questions	Answer	intfloat/multilingual-e5-large-instruct
වෘත්තීය සම්මේ පිහිටුවීමේ සහ වෘත්තීසම්මේනිවලට	නැත. සෑම පුරවැසියෙකුටම ලබා දී තිබේ	වෘත්තීය සම්මේ පිහිටුවීමේ සහ වෘත්තීසම්මේනිවලට බැඳීමේ නිදහසක් වාවස්ථාවෙන් ලබා දී ඇත්තේ සාමාන්‍ය ජනතාවට ද අනුමත වසමකි.
ශ්‍රී ලංකා ජනරජයේ පරමාධිපත්‍යය ඇත්තේ විධාය	නැත. එය ඇත්තේ ජනතාවටයි	නැත. ශ්‍රී ලංකා ජනරජයේ පරමාධිපත්‍යය ජනතාව කෙරෙහි පිහිටා ඇත්තේ ය.
ශ්‍රී ලංකා පුරවැසියෙකු අල්ලස් චෝදනාවකට වරද	**නොහැක. ඒ සඳහා වසර හතක් ගත විය යුතු ය.**	**ශ්‍රී ලංකා පුරවැසියෙකු අල්ලස් චෝදනාවකට වරදකරු වී සත් වසරක් ඉකුත් වී නැත්නම්, ඔහු පාර්ලිමේන්තු මැතිවරණය සඳහා ඡන්දය ලබා දිය නොහ**
ශ්‍රී ලංකා වාවස්ථාව අනුව අමාත්‍ය මණ්ඩල ප්‍රධාන	නැත. අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ ජනාධිපතිවරයාය.	නැත, අමාත්‍ය මණ්ඩලයේ ප්‍රධානයා ජනාධිපතිවරයා බව සඳහන් වා ඇත.
ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාරවරයෙ	නැත. ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාර ධුරයක් හිස්ව තිබුණ ද, යම් විනිශ්චයකාරවරයෙ	නිර්බල නොවන්නේ ය; නිර්බල වන්නාක් සේ නොසැලෙන්න යුත්තේය.
වරක් භේරී පත් වී සිටින ජනාධිපතිවරයෙකුට නැ	අවම වශයෙන් පස්වීමෙන් වසර හතරක් ඉක්ම විය යුතු ය.	වරක් භේරී පත් වී සිටින ජනාධිපතිවරයෙකුට නැවත ජනවරමක් ලබා ගැනීම සඳහා භේරී පත් වීමෙන් පසු අවම වශයෙන් වසර 4ක් ඉක්ම විය යුතුය.
ආණ්ඩුකුම වාවස්ථා සභාව සාමාජිකයින් කී දෙ	සාමාජිකයින් 10කි	ආණ්ඩුකුම වාවස්ථා සභාව පිළිබඳ සාමාන්‍ය සංඛ්‍යාව නිර්දේශය නොසරන ලද නිසා ආණ්ඩුකුම වාවස්ථා සභාව සාමාජිකයින් පිළිබඳ පිට දුන්න සමාග එකඟ නැත
අමාත්‍ය මණ්ඩලයට අයත් උපරිම ඇමතිවරුන් ගණ	**අමාත්‍ය මණ්ඩලයට අයත් උපරිම සංඛ්‍යාව 30කි අමාත්‍ය මණ්ඩල සාමාජිකයින් නො**	අමාත්‍ය මණ්ඩලයට අයත් උපරිම ඇමතිවරුන් ගණන 30. අමාත්‍ය මණ්ඩල සාමාජිකයින් නොවන ඇමතිවරුන් සහ නියෝජ්‍ය ඇමතිවරුන්ගේ උපරිම ගණන 40.
පර්ලිමේන්තුව අවම වශයෙන් වසරකට කීවා	**අවම වශයෙන් එක් වරක්වත් කැඳවිය යුතුය.**	ඔව්, පාර්ලිමේන්තුව සෑම වසරකට වරක්වත් කැඳවිය යුත්තේය.
අධිකරණ සේවා කොමිෂන් සභාවෙහි යම් රැස්වීම	**ගණසුරණය සඳහා සාමාජිකයින් දෙදෙනෙකුගෙන් සමන්විත විය යුතුය.**	අධිකරණ සේවා කොමිෂන් සභාවෙහි යම් රැස්වීමක ගණසුරණය සඳහා සාමාජිකයින් දෙදෙනකු සහභාගී විය යුතුය
පාර්ලිමේන්තුවෙහි ඡන්ද විමසීමක දී මුලසුන දරන	**ඡන්ද සංඛ්‍යාව සම වූ අවස්ථාවන්හි දී මූලසුන දරන තැනැත්තාට තීරණ ඡන්දයක් හි**	මූලසුන දරන මන්ත්‍රීවරයා ඡන්දය දිය යුත්තේ පක්ෂ ඡන්ද සංඛ්‍යාව සම වූවහොත්, එවිට මූලසුන දරන මන්ත්‍රීවරයා තීරණ ඡන්දය දිය යුතුය.
කිසියම් ආණ්ඩුකාරවරයෙකු සිය බලය අහිසි	ආණ්ඩුකාරවරයා ඉවත් කිරීමට සභාවට යෝජනාවක් ඉදිරිපත් කොට සියලු මන්ත්‍රීවරුන්ගෙ	ආණ්ඩුකාරවරයා ධුරයෙන් ඉවත් කිරීම සඳහා ජනාධිපතිවරයා විසින් ආණ්ඩුකාරවරයා වෙත යවන ලිපියක් මගින් මාසු සිය ධුරයෙන් ඉල්ලා අස්විය හැකි බව සඳහන් කවි.
ජනාධිපතිවරණයක හෝ ජනමත විචාරණයක	අවම වශයෙන් ශ්‍රේෂ්ඨාධිකරණ විනිශ්චයකාරවරුන් පස් දෙනෙකු විසින් විභාග කොට තීරණ	ශ්‍රේෂ්ඨාධිකරණයට
ආණ්ඩුකුම වාවස්ථා සභාවේ සංයුතිය කුමක්	සාමාජිකයින් 10කි. පහත පරිදි වේ. අගමැති කථානායක විපක්ෂ නායක ජනාධිපති පත් කළ මන්ත්‍රී ආණ්ඩු පක්ෂයේ එක් මන්ත්‍රීවරයෙක් විපක්ෂයේ එකඟත්වයෙන් එක් මන්ත්‍රීවරයෙක් අගමැති සහ විපක්ෂ නායක අලුතැමියෙන් කථානායක පත් කරන 3කි ආණ්ඩු පක්ෂයේ හෝ විපක්ෂ නායකයාගේ පක්ෂයේ නොවන එක් මන්ත්‍රීවරයෙක්	**ආණ්ඩුකුම වාවස්ථා සභාව නව දෙනෙකුගෙන් සමන්විත විය යුතු බව සඳහන් වේ. එහි සාමාජිකයන් තිදෙනකුට නොඅඩු සංඛ්‍යාවක් රජයේ නිලධාර**
වාවස්ථානුකූලව විගණකාධිපති ධුරය හිස් වී	මිය යෑම ඉල්ලා අස් වීම වයස 60 සම්පූර්ණ වීම රෝගාතුර වීම හෝ මානසික දුබලතාවක් නිසා ජනපතිවරයා විසින් ඉවත් කිරීම පාර්ලිමේන්තු යෝජනාවක් සම්මත වීමෙන් ජනපතිවරයා විසින් ඉවත් කිරීම	විගණකාධිපති ධුරය හිස් විය හැකි අවස්ථා පහත දක්වා ඇත: 1. විගණකාධිපතිවරයා මිය ගිය විට 2. විගණකාධිපතිවරයා ජනාධිපතිවරයා වෙත යවන ලිපියක් මගින් ඉල්ලා අස් වී
ශ්‍රී ලංකාවෙහි වාවස්ථාව අනුව අගවිනිසුරුවරයා	නැත. අගවිනිසුරුවරයා ලෙස පත් කරනු ලබූ තැනැත්තා පළමුව ජනාධිපතිවරයා ඉදිරියේ දිවු	ශ්‍රී ලංකාවෙහි වාවස්ථාව අනුව අගවිනිසුරුවරයා ලෙස පත් කරනු ලබූ තැනැත්තා ශ්‍රී ලංකාවෙහි හතරවන උපලේඛනයේ දැක්වෙන ප්‍රතිඥාව දී රට අත්සන් තැබීමෙන් හෝ විධුරණ දී
ශ්‍රී ලංකාවෙහි වාවස්ථාව අනුව ජනාධිපතිවරයාට	අසත්‍යයකි. ජනාධිපතිවරයාට අගමැතිවරයාගේ ද අදහස් විමසා කිසියම් මන්ත්‍රීවරයෙකු අමා	**නැත, මෙය සත්‍යයක් නොවේ. ශ්‍රී ලංකාවෙහි වාවස්ථාව අනුව, ජනාධිපතිවරයා විසින් අග්‍රාමාත්‍යවරයාගේ අදහස් විමසා, පාර්ලිමේන්තුවේ මන්ත්‍රීවරය**
ශ්‍රී ලංකාවෙහි වාවස්ථාව අනුව අමාත්‍ය මණ්	වැරදියි. **අමාත්‍ය මණ්ඩලයට රජයේ නිලධාරීන් පත් කිරීමට, මාරු කිරීමට හෝ සේවයෙන් පහ කිරීමට විධිවිධාන සැපයීමට බලය ඇත.**	නැත. අමාත්‍ය මණ්ඩලයට රජයේ නිලධාරීන් පත් කිරීමට, මාරු කිරීමට හෝ සේවයෙන් පහ කිරීමට විධිවිධාන සැපයීමට බලය ඇත.
පළාත් පාලන ආයතන සහ වෙනත් අධිකාරීන්ට බ	අසත්‍යයකි. පාර්ලිමේන්තුව විසින් සම්මත කරනු ලබූ නීතියක හේ පවත්නා නීතියක අධිකාරය	නැත. ආණ්ඩුකුම වාවස්ථාව අනුව, පළාත් පාලන ආයතන සහ වෙනත් අධිකාරීන්ට බදු සහ විරිපනම් කිරීමේ බලය නැත. ආණ්ඩුකුම වාවස්ථාව අනුව, මූලික පිළිබඳ සම්පූර්ණ
ශ්‍රී ලංකා ආණ්ඩුකුම වාවස්ථාව අනුව රාජ්‍ය සේවා	අසත්‍යයකි. රාජ්‍ය සේවා කොමිෂන් සභාව සාමාජිකයින් නව දෙනෙකුගෙන් සමන්විත වන අ	මෙම ප්‍රශ්නයට අනුව, රාජ්‍ය සේවා කොමිෂන් සභාවට සාමාජිකයින් පත් කරන බව ආණ්ඩුකුම වාවස්ථාව අනුව ජනාධිපතිවරයා විසින් ආණ්ඩුකුම වාවස්ථා සභාවේ නිර්දේශය මත
Correct Percentage (%)		80

Figure 5.6: Comparison between predicted and actual responses generated by the intfloat/multilingual-e5-large-instruct model

Large Language Model	**Correct Percentage (%)**
llama-3.1-8b-instant	40
llama-3.1-70b-versatile	80
llama-3.3-70b-versatile	80
timpal0l/mdeberta-v3-base-squad2	45

Table 5.5: Evaluation Results of Large Language Models

Questions	Answer	llama-3.3-70b-versatile
වෘත්තීය සමිති පිහිටුවීමේ සහ වෘත්තීයසමිතිවලට බැඳීමේ නිදහසක	නැත. සෑම පුරවැසියෙකුටම ලබා දී තිබේ	නැත, වෘත්තීය සමිති පිහිටුවීමේ සහ වෘත්තවලට බැඳීමේ නිදහස වරප්‍රසාද ලත් පිරිසට පමණක් නොවේ. ශ්‍රී ලංකා ප්‍රජාතන
ශ්‍රී ලංකා ජනරජයේ පරමාධිපත්‍යය ඇත්තේ විධායක ජනාධිපතිවර	නැත. එය ඇත්තේ ජනතාවටයි	නැත, ශ්‍රී ලංකා ජනරජයේ පරමාධිපත්‍යය ජනතාව කෙරෙහි පිහිටා ඇත. ජනාධිපතිවරයා ජනතාව විසින් තෝරා පත්කර ගැ
ශ්‍රී ලංකා පුරවැසියෙකු අල්ලස් චෝදනාවකට වරදකරු වී පස් වසරක	**නොහැක. ඒ සඳහා වසර හතක් ගත විය යුතු ය.**	1946 ලංකා (පාර්ලිමේන්තු මැතිවරණ) රාජසභා ආඥාව යටතේ දූෂණ ක්‍රියාවකට හෝ එකී දූෂණ ක්‍රියාවකට අනුරූප වන්නා
ශ්‍රී ලංකා ව්‍යවස්ථාව අනුව අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ අගමැ	නැත. අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ ජනාධිපතිවරයාය.	අමාත්‍ය මණ්ඩලයේ ප්‍රධානයා ජනාධිපතිවරයා ම වේ. අගමැතිවරයා අමාත්‍ය මණ්ඩලයේ ප්‍රධානයා නොවේ.
ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාරවරයෙකු පත් කිරීමේ ය	නැත. ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාර ධුරයක් හිස්ව තිබුණ ද, යම් විනිශ්චයකාර	නිර්බල නොවන්නේ ය, නිර්බල වන්නාක් සේ නොසැලකිය යුත්තේ ය.
වරක් තේරී පත් වී සිටින ජනාධිපතිවරයෙකුට නැවත ජනවරමක් ල	අවම වශයෙන් පත්වීමෙන් වසර හතරක් ඉක්ම විය යුතු ය.	4
ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාව සාමාජිකයින් කී දෙනෙකුගෙන් සමන්වි	සාමාජිකයින් 10කි	ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාව ගැන මා ලබන තොරතුරු අනුව එය සාමාජිකයන් 10 දෙනෙකුගෙන් සමන්විත විය යුතු බව පෙ
අමාත්‍ය මණ්ඩලයට අයත් උපරිම ඇමතිවරුන් ගණන සහ අමාත්‍ය	**අමාත්‍ය මණ්ඩලයට අයත් උපරිම සංඛ්‍යාව 30කි අමාත්‍ය මණ්ඩල සාමාජිකයිද**	අමාත්‍ය මණ්ඩලයට අයත් උපරිම ඇමතිවරුන් ගණන 30 කි. අමාත්‍ය මණ්ඩල සාමාජිකයන් නොවන ඇමතිවරුන් සහ නි
පාර්ලිමේන්තුව අවම වශයෙන් වසරකට කීවරක් කැඳවිය යුතු	**අවම වශයෙන් එක් වරක්වත් කැඳවිය යුතුය.**	පාර්ලිමේන්තුව සෑම වසරකට වරක්වත් කැඳවිය යුත්තේය.
අධිකරණ සේවා කොමිෂන් සභාවෙහි යම් රැස්වීමක ගණපූරණය ස	**ගණපූරණය සඳහා සාමාජිකයින් දෙදෙනෙකුගෙන් සමන්විත විය යුතුය.**	අධිකරණ සේවා කොමිෂන් සභාවෙහි යම් රැස්වීමක ගණපූරණය සඳහා සාමාජිකයින් 5 දෙනෙකු සහභාගී විය යුතුය.
පාර්ලිමේන්තුවෙහි ඡන්ද විමසීමක දී මුලසුන දරන මන්ත්‍රීවරයා ඡන්	**ඡන්ද සංඛ්‍යාව සම වූ අවස්ථාවන්හි දී මුලසුන දරන තැනැත්තාට තීරණ ඡන්දය**	පාර්ලිමේන්තුවෙහි ඡන්ද විමසීමක දී මුලසුන දරන මන්ත්‍රීවරයා ඡන්දය දිය යුත්තේ පක්ෂ ඡන්ද සංඛ්‍යාව හා විපක්ෂ ඡන්ද ස
කිසියම් ආණ්ඩුකාරවරයෙකු සිය බලය අනිසි ලෙස භාවිත ක	ආණ්ඩුකාරවරයා ඉවත් කිරීමට සභාවට යෝජනාවක් ඉදිරිපත් කොට සියලු මන්ත්‍රීවරු	ආණ්ඩුකාරවරයා ආණ්ඩුක්‍රම ව්‍යවස්ථාවේ විධිවිධාන චේතනාත්මකව උල්ලංඝනය කර තිබීම; ආණ්ඩුකාරවරයා සිය බලය
ජනාධිපතිවරණයක හෝ ජනමත විචාරණයක වලංගුභාවය	අවම වශයෙන් ශ්‍රේෂ්ඨාධිකරණ විනිශ්චයකාරවරුන් පස් දෙනෙකු විසින් විභාග කොට	ශ්‍රේෂ්ඨාධිකරණයට බලය ඇත.
ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාවේ සංයුතිය කුමක් ද?	සාමාජිකයින් 10කි. පහත පරිදි වේ. අගමැති කථානායක විපක්ෂ නායක ජනාධිපති පත් කළ මන්ත්‍රී ආණ්ඩු පක්ෂයේ එක් මන්ත්‍රීවරයෙක් විපක්ෂයේ එකඟත්වයෙන් එක් මන්ත්‍රීවරයෙක් අගමැති සහ විපක්ෂ නායක අනුමැතියෙන් කථානායක පත් කරන 3ක් ආණ්ඩු පක්ෂයේ හෝ විපක්ෂ නායකගේ පක්ෂයේ නොවන එක් මන්ත්‍රීවරයෙක්	පිළිතුර: ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාවේ නිර්දේශය මත ජනාධිපතිවරයා විසින් පත් කරනු ලබන සාමාජිකයන් නව දෙනෙකු
ව්‍යවස්ථානුකූලව විගණකාධිපති ධුරය හිස් විය හැකි අවස්ථ	මිය යෑම ඉල්ලා අස් වීම වයස 60 සම්පූර්ණ වීම රෝගාතුර වීම හෝ මානසික දුබලතාවක් නිසා ජනපතිවරයා විසින් ඉවත් කිරීම පාර්ලිමේන්තු යෝජනාවක් සම්මත වීමෙන් ජනපතිවරයා විසින් ඉවත් කිරීම	විගණකාධිපතිවරයා මිය ගිය හොත්, ජනාධිපතිවරයා වෙත ලිපියක් මඟින් ඉල්ලා අස්වුවහොත්, වයස අවුරුදු හැට සම්පූර්ණ
ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව අගවිනිසුරුවරයා ලෙස පත් කරනු ල	නැත. අගවිනිසුරුවරයා ලෙස පත් කරනු ලැබූ තැනැත්තා පළමුව ජනාධිපතිවරයා ඉදිරි	ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව අගවිනිසුරුවරයා ලෙස පත් කරනු ලැබූ තැනැත්තාට ජනාධිපති ඉදිරියෙහි දිවුරුම් දීමට ඉඩ
ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව ජනාධිපතිවරයාට කිසියම් මන්ත්‍රීව	අසත්‍යයකි. ජනාධිපතිවරයාට අගමැතිවරයාගේ ද අදහස් විමසා කිසියම් මන්ත්‍රීවරයෙ	අමාත්‍ය මණ්ඩලයේ සාමාජිකයන් නොවන අමාත්‍යවරයන් පත් කරනු ලැබිය හැක. ජනාධිපතිවරයා විසින් අග්‍රාමාත්‍යවරයා
ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව අමාත්‍ය මණ්ඩලයට රජයේ ඒ	**වැරදියි. අමාත්‍ය මණ්ඩලයට රජයේ නිලධාරීන් පත් කිරීමට, මාරු කිරීමට හෝ**	අමාත්‍ය මණ්ඩලයට රජයේ නිලධාරීන් පත් කිරීමට, මාරු කිරීමට හෝ සේවයෙන් පහ කිරීමට විධිවිධාන සැපයීමට බලය න
පළාත් පාලන ආයතන සහ වෙනත් අධිකාරීන්ට බදු සහ වරිපනම් නි	අසත්‍යයි. පාර්ලිමේන්තුව විසින් සම්මත කරනු ලැබූ නීතියක හේ පවත්නා නීතියක අධි	පළාත් පාලන ආයතන සහ වෙනත් අධිකාරීන්ට බදු සහ වරිපනම් නියම කිරීමේ බලය ශ්‍රී ලංකා ආණ්ඩුක්‍රම ව්‍යවස්ථාවෙන්
ශ්‍රී ලංකා ආණ්ඩුක්‍රම ව්‍යවස්ථාව අනුව රාජ්‍ය සේවා කොමිෂන් සභාව	අසත්‍යයකි. රාජ්‍ය සේවා කොමිෂන් සභාව සාමාජිකයින් නව දෙනෙකුගෙන් සමන්විත ද	ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාවේ නිර්දේශය මත ජනාධිපතිවරයා විසින් රාජ්‍ය සේවා කොමිෂන් සභාවේ සාමාජිකයන් පත් කරු
Correct Percentage (%)		80

Figure 5.7: Comparison between predicted and actual responses generated by the llama-3.3-70b-versatile model

5.4 Evaluation of the TTS Model

To ensure a comprehensive and reliable assessment of the developed TTS systems, both subjective and objective evaluation methods were employed. This dual evaluation strategy allowed us to measure not only the technical correctness of the synthesized speech but also how natural and understandable it was to human listeners. The goal was to evaluate two key aspects of TTS performance: **intelligibility** and **naturalness**. **Intelligibility** refers to how accurately and easily a listener can understand the spoken output. It measures how well the TTS system communicates the intended message through clear pronunciation and phrasing. **Naturalness**, on the other hand, assesses how closely the synthetic voice resembles real human speech. This includes evaluating prosodic features such as rhythm, pitch, tone, and overall expressiveness, which contribute to the human-likeness of the audio.

It is important to note that only the models trained with Sinhala text were used for the final evaluation, as the overall system is designed to process Sinhala script inputs. The models trained with Romanized text were developed solely for preliminary experimentation and were excluded from the final evaluation phase.

5.4.1 Subjective Evaluation

While objective methods offer quantifiable measurements, subjective evaluations are particularly critical in TTS research, as the final output is designed for human listeners. Therefore, **Mean Opinion Score (MOS)** and **Semantically Unpredictable Sentences (SUS)** tests were used for subjective analysis, with a focus on intelligibility and naturalness.

Mean Opinion Score (MOS)

MOS is the most widely adopted subjective evaluation method used in TTS research to assess both the intelligibility and naturalness of synthesized speech. It involves gathering human listener ratings based on a standardized five-point scale as shown in Table 5.6:

Rating	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 5.6: MOS Rating Scale

A total of 12 native Sinhala speakers participated in the MOS evaluation. The group was carefully balanced to avoid demographic bias, consisting of 6 male and 6 female participants, equally distributed across two age groups: 15–30 and 30–60 years. All participants were well-educated and fluent in Sinhala.

For the test, 15 Sinhala sentences (Figure 5.8) were selected to represent a diverse range of linguistic features and sentence structures. The dataset included:

- 6 short sentences (5–10 words)
- 6 medium-length sentences (10–20 words)
- 3 long sentences (more than 20 words)

The selected sentences also included abbreviations, numerical values, dates, and currency expressions to test the robustness of the system in handling common real-world variations.

Participants are typically asked to listen to a set of synthesised audio samples and rate each one according to how natural or intelligible the speech sounds.

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N} \tag{5.3}$$

Gender: Male	Age: 25	
Synthesize Sentences	VITS_sinhala_male_single	
	Intelligibility (1 to 5)	Naturalness (1 to 5)
ෆිනෑන්ස් ආයතනය අඛණ්ඩව වර්ධනය වෙමින් ඉදිරියට පැමිණියේය. ළෙන්ගතු පාරිභෝගික පදනම මත එල්.බී. ෆිනෑන්ස් ඉදිරියට.	3	4
සිරිකොත ඉදිරිපිට ගැටුමක් - කිහිප දෙනෙකුට තුවාලයි	5	3
ආණ්ඩුක්‍රම ව්‍යවස්ථාවේ විධිවිධාන යටතේ අමාත්‍ය මණ්ඩලය විසිර ගිය ද ජනාධිපතිවරයා නවදුරටත් තම ධුරය දරන්නේය.	4	5
සැලසුම් සහගතව නිවෙසට ගෙන්වා ගෙන ඖෂධ අලෙවි නිලධාරියාට හොඳටම ගහලා.	4	5
වෙළඳසැලේ නිබු මාලය නැති වුණ හැටි - සීසීටීවී දර්ශන සහිතයි.	5	5
රටේ ආරක්ෂාව ඇතුළුව ජනතාවගේ විධායක බලය ජනතාව විසින් තෝරා පත්කර ගනු ලබන ජනරජයේ ජනාධිපතිවරයා විසින් ක්‍රියාත්මක කළ යුත්තේ ය.	4	5
ලෝක සෞඛ්‍ය සංවිධානය කියන්නේ නවත් වසර දෙකක් යන තුරු කොරෝනා වසංගතය පැවැතීමට ඉඩ තිබෙන බවයි.	2	5
රැස්වීම් තුනකට නැත්නම් කමිටු සාමාජිකත්වය නෑ	4	4
$1 ක වටිනාකම රු.365 ක් දක්වා 2022 වර්ශයේදී ඉහල නැඟ තිබේ.	5	5
ජනතාව විසින් ජනාධිපති ධුරයට දෙවරක් නෝරා පත්කර ගනු ලැබූ නැනැත්තකු ජනතාව විසින් නැවත එකී ධුරය සඳහා නෝරා පත්කර ගනු ලැබීමට සුදුස්සකු නොවන්නේය.	5	5
දකුණු නායිලන්තයේ ස්ථාන 17ක අද දිනයේ පිපිරීම් සහ ගිනිනැබීම් වාර්තා විය.	5	5
දැනටමත් අයිස්ලන්තය, ග්‍රීසිය ඒ මග ගොස් ඇත.මෙය අනිවාර්යයෙන්ම ඒ රටවල ජාතිකවාදී කැලඹිලි ඇති කරනු ඇත.	4	4
ශ්‍රී ලංකාවේ රාජ්‍ය භාෂාව සිංහල භාෂාව වන්නේය.	5	4
ලන්දේසින් විසින් ක්‍රි.ව.1640 දී ගාල්ලේ බලය තහවුරු කර ගත්ත ද ඊට පෙර සිටම දෙපාර්ශවයේ බලය තහවුරු කරගැනීමමේ කටයුතු සිදුවූ බව පෙනෙයි.	3	4
පාර්ලිමේන්තුවේ මංගල රැස්වීම්වල මූලසුන දැරීමට බලය ඇත්තේය.	5	4
Average MOS of Participant	**4.2**	**4.466666667**

Figure 5.8: Test sentences used to calculate the MOS value

The final MOS is calculated as the mean of all individual scores using Equation 5.3, providing an aggregate measure of perceived speech quality. Higher scores indicate better performance.

Model	Intelligibility (MOS)	Naturalness (MOS)
Male_Single	**4.62**	**4.18**
Male_Multi	4.07	3.98
Female_Single	3.59	3.73
Female_Multi	**4.24**	**4.07**

Table 5.7: MOS results for each VITS model trained with Sinhala text

Table 5.7 presents the MOS results for each VITS model trained with Sinhala text, evaluating both intelligibility and naturalness.

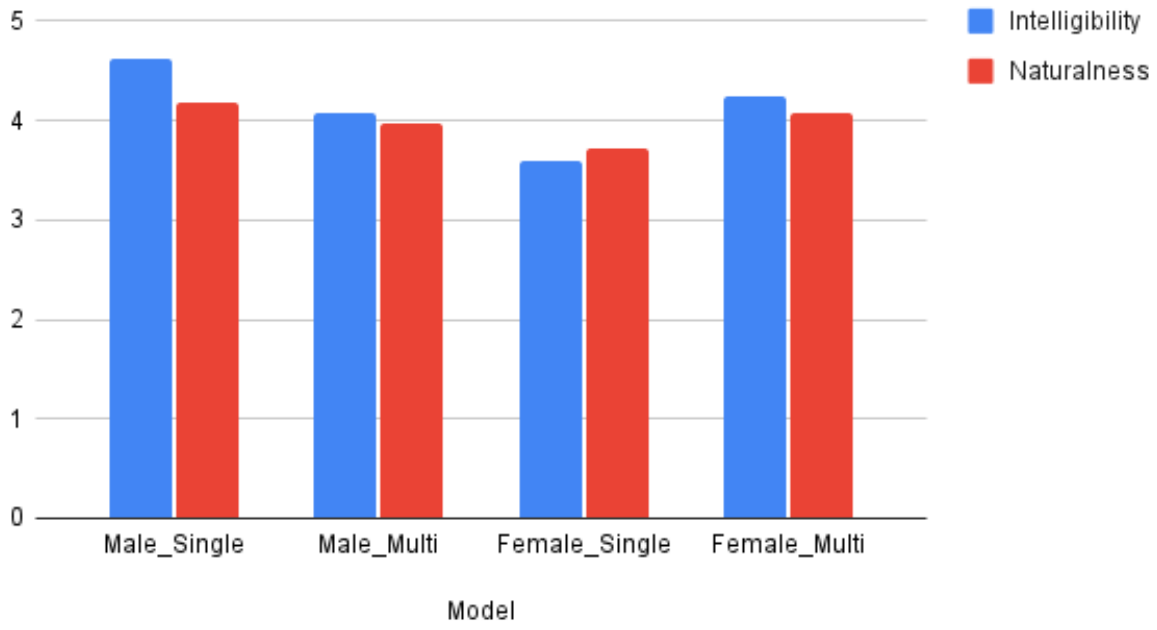

Figure 5.9: Visualization of MOS Results for Intelligibility and Naturalness

Based on the results (Figure 5.9), the most optimal model for the male voice is the Sinhala Text – Single-Speaker Male model (Section 3.4.3 *Sinhala Text – Single-Speaker Male Dataset*). For the female voice, the most effective model is the Sinhala Text – Multi-Speaker model (Section 3.4.3 *Sinhala Text – Multi-Speaker Dataset*).

Semantically Unpredictable Sentences (SUS)

To further evaluate intelligibility in a controlled way, Semantically Unpredictable Sentences (SUS) were used. These sentences as shown in Figure 5.10, are grammatically correct but lack meaningful semantic context, making it harder for listeners to guess missing words. This test provides a stricter and more objective measure of intelligibility.

කළු නයා කෝපි බීලා පන්සලට ගියා.
තණකොළ වළඳා කවුරුත් නැති ගඟක් ගිලිහුණා.
මැටි ගල මල්වල උඩ නැඟුණ ගායකයා නටන්න ගියා.
අඳුරු අහස යටින් නැටියක් ගෙනා මියෝ නැටුම් දැක්වුවා.
බිම වැටුණු පොත හරින රථය කුඹුක් වනයට ගියා.

Figure 5.10: SUS sentences used to calculate the SUS value

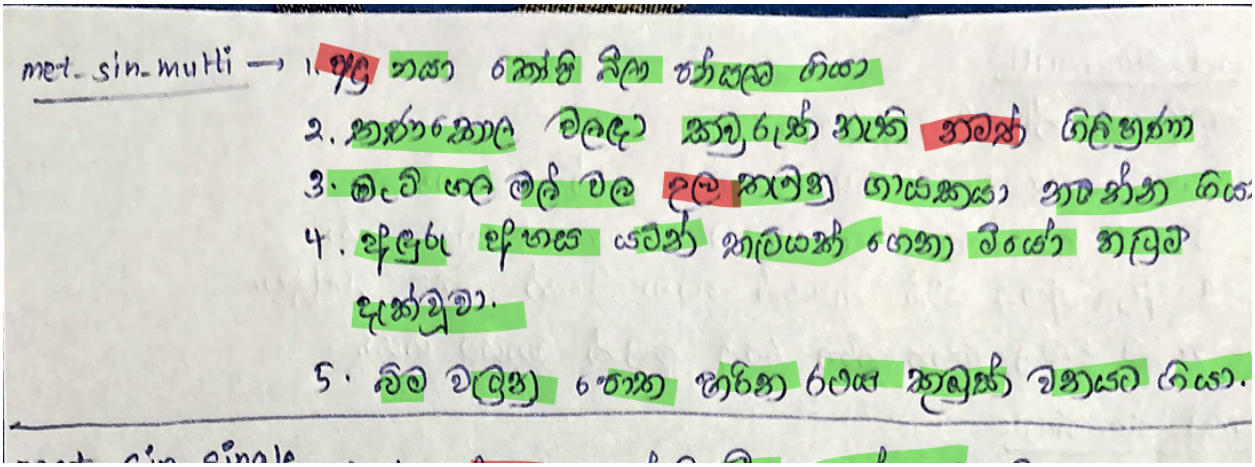

Figure 5.11: Transcription of SUS Sentences by Participants

Speeches were generated using the trained Sinhala TTS models for the given sentences. These generated speeches were shuffled and distributed among 10 participants, who were instructed to transcribe what they heard (Figure 5.11). While they were permitted to replay the audio, they were advised to do so sparingly. The handwritten responses were meticulously compared to the original text, with spelling errors being forgiven.

$$\text{Intelligibility (SUS)} = \frac{\sum_{\text{respondent}=1}^{10} \text{correct words}}{\left(\sum_{\text{sentence}=1}^{10} \text{words}\right) \times \text{respondents}} \tag{5.4}$$

The intelligibility score was then calculated using the 5.4 equation. The participants were specifically asked to listen to the synthesized SUS sentences and transcribe them without relying on context or guessing. The accuracy of their transcriptions was analyzed to assess the intelligibility of the synthetic speech.

Model	Intelligibility
Male_Single	**85.83%**
Male_Multi	82.50%
Female_Single	77.50%
Female_Multi	**81.39%**

Table 5.8: SUS Intelligibility Scores for Different Models

Figure 5.12: Visualization of SUS Results for Intelligibility

The SUS values were converted into percentages, and the results (Figure 5.12) obtained from the SUS values are consistent with those of the MOS values. The most optimal model for the male voice is the Sinhala Text – Single-Speaker Male model (Section 3.4.3 *Sinhala Text – Single-Speaker Male Dataset*). For the female voice, the most effective model is the Sinhala Text – Multi-Speaker model (Section 3.4.3 *Sinhala Text – Multi-Speaker Dataset*).

5.4.2 Objective Evaluation

To further evaluate the performance of the TTS system, an objective evaluation was conducted. This type of evaluation is based on quantitative methods, where values are computed using algorithms, equations, and automated tools. In this study, WER and CER were used as objective evaluation metrics.

WER and CER

As part of the objective evaluation, the WER and CER were calculated using the trained Sinhala Wav2Vec2-BERT ASR model, which is described in Chapter 3 Section 3.4.1 *Wav2Vec2-BERT*. Although this evaluation depends on the performance of the ASR system, it provides a reasonable estimation of the intelligibility of the TTS system from an objective perspective.

For this evaluation, a set of 15 test sentences—previously used for the MOS evaluation—was selected. These sentences were synthesized using each of the TTS models. The generated audio was then fed into the ASR model, and the resulting transcriptions were compared with the original reference text. The WER and CER were calculated using Equations 5.1 and 5.2, respectively.

Model	**WER**	**CER**
Male_Single	**0.495**	**0.109**
Male_Multi	0.574	0.147
Female_Single	0.574	0.158
Female_Multi	**0.560**	**0.142**

Table 5.9: WER and CER scores for each TTS model using the trained Sinhala Wav2Vec2-BERT ASR

The results of this evaluation are presented below, further confirming that the Sinhala Text – Single-Speaker Male model and the Sinhala Text – Multi-Speaker Female model perform best in terms of intelligibility.

5.4.3 Results Comparison

When comparing all the results obtained from both subjective and objective evaluation metrics—namely MOS, SUS, WER and CER, the Sinhala Text – Single-Speaker Male model (Section 3.4.3 *Sinhala Text – Single-Speaker Male Dataset*) is identified as the best-performing model for male voice synthesis. For the female voice, the Sinhala Text – Multi-Speaker model (Section 3.4.3 *Sinhala Text – Multi-Speaker Dataset*) demonstrates the highest performance.

Metric	[1]	[2]	VITS (Trained)
MOS - Intelligibility	–	–	**92.30%**
MOS - Naturalness	70%	78.2%	**82.34%**
SUS - Intelligibility	70%	84.00%	**85.83%**

Table 5.10: Comparison of evaluation metrics across Sinhala TTS systems. [1]: Nanayakkara, Liyanage, et al. 2018, [2]: Arachchige and Weerasinghe 2023.

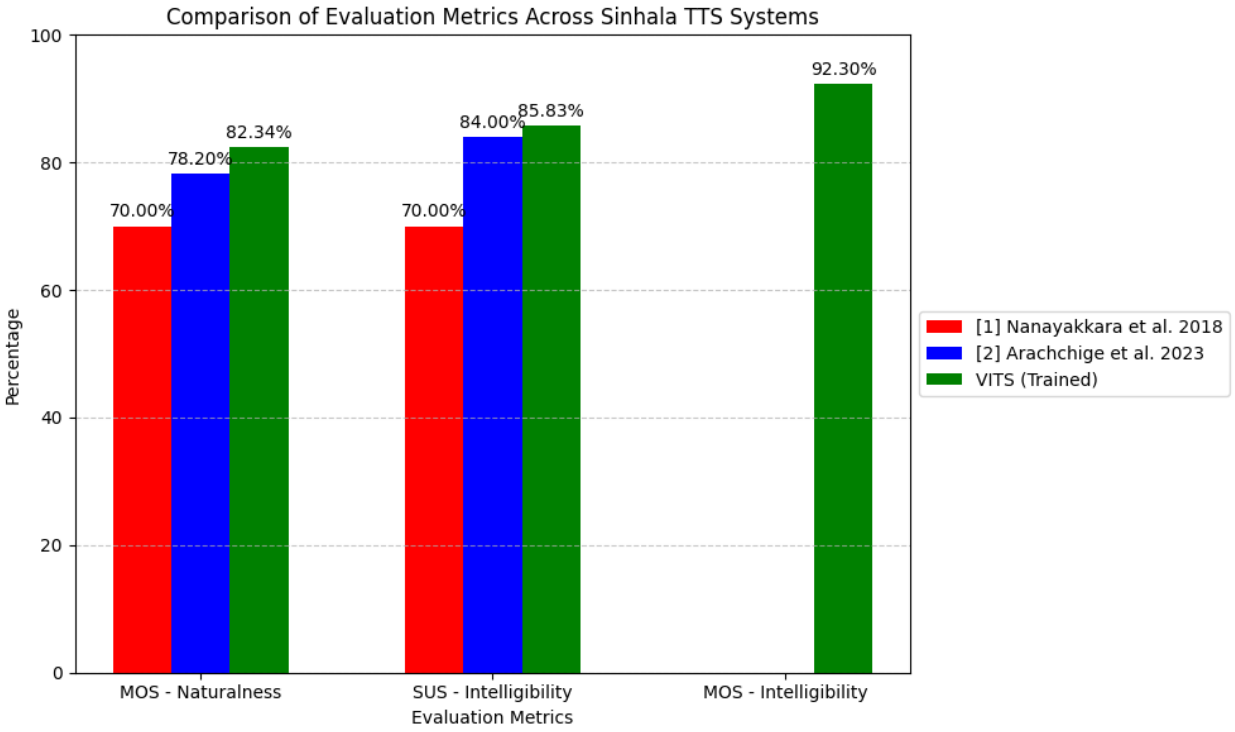

Figure 5.13: Comparison of Evaluation Metrics Across Sinhala TTS Systems

Furthermore, Table 5.10 and Figure 5.13 present a comparison between the best-performing trained Sinhala VITS model (Sinhala Text – Single-Speaker Male model) and the values reported in recent Sinhala TTS systems, such as those proposed by Nanayakkara, Liyanage, et al. (2018) and the Arachchige and Weerasinghe (2023) model.The best-performing trained Sinhala VITS model (Section 3.4.3 *Sinhala Text – Single-Speaker Male Dataset*), which is the Sinhala Text - Single-Speaker Male model, demonstrates a slight improvement over previous systems.

5.5 Evaluation of Speech to Speech Chatbot

After integrating the three main components—Automatic Speech Recognition (ASR), Chatbot, and Text-to-Speech (TTS)—and developing the front-end interface, a comprehensive user-based evaluation was conducted to assess the system's functionality, accuracy, and user experience. A group of 20 participants, diverse in terms of age, language proficiency, and technical background, were invited to interact with the complete speech-to-speech chatbot system in a controlled environment.

Participants were assigned a set of predefined tasks designed to test different aspects of the system, such as open-ended question answering, information retrieval. During the evaluation, users engaged in multiple speech-based interactions where their spoken queries were processed by the ASR, interpreted by the chatbot, and responded to via the TTS engine. Each session was monitored to record interaction flow, response time, and the relevance of the chatbot's answers.

Upon completing the interaction session, participants provided feedback through a structured Google Form, which collected both quantitative and qualitative responses. The form included Likert-scale items (1–5) evaluating criteria such as ASR accuracy, chatbot relevance, TTS clarity, response coherence, ease of use, and overall satisfaction. Open-ended questions were also included to gather user suggestions and highlight any observed limitations.

Evaluation Metric	**Average Score (Out of 5)**
Task Success Rate	4.2
Utterance and Response Matching Score	3.8
Response Time	4.0
Confusion Rate	2.3
User Satisfaction	4.3
Clarity and Concision	3.95
Conversation Relevancy	4.05
Conversation Completeness	4.1
Overall experience with ASR Component	4.1
Overall experience with TTS Component	4.45

Table 5.11: Evaluation Results of Speech-to-Speech Chatbot

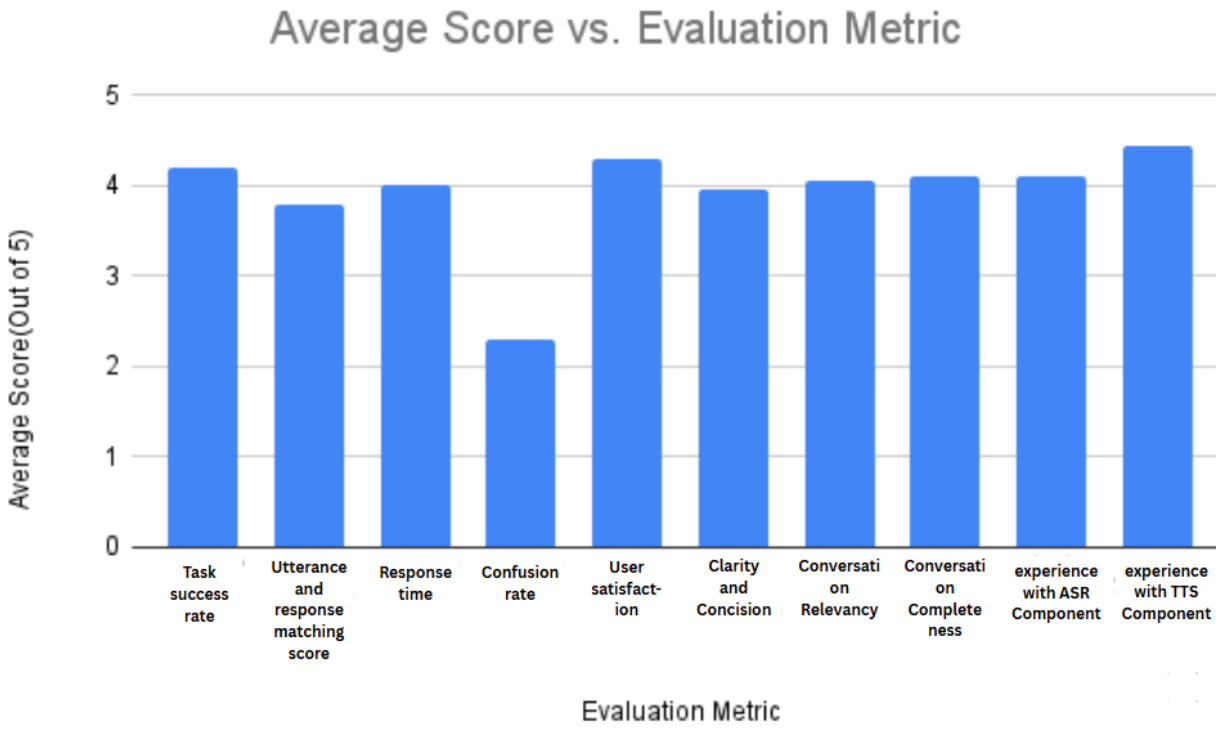

Figure 5.14: Evaluation Results of Speech to Speech Chatbot

The Google Form used for this evaluation is included in Appendix A, and a summarized overview of participant feedback is presented in 5.11. The results demonstrate that the developed Sinhala speech-to-speech chatbot system is both functional and user-friendly, achieving high average scores across multiple performance metrics. Specifically, the ASR and TTS components received average ratings of 4.1 and 4.45, respectively, reflecting their strong performance in speech recognition and output synthesis. While the utterance-response matching and contextual understanding showed moderate variability—highlighting areas for future improvement—the system still achieved a high task success rate, indicating its practical usability in real-world scenarios.

These evaluation outcomes not only confirm the technical feasibility of integrating ASR, chatbot, and TTS technologies for Sinhala language interactions but also provide actionable insights for further refinements. The study supports the potential of such systems in enhancing speech-based human-computer interaction, especially in underrepresented languages like Sinhala.

Chapter 6

Discussion

6.1 Chapter Overview

This chapter covers the discussion of the significant discoveries, implications, and obstacles encountered in the research and development stages of developing the Vocsi product.

6.2 Contributions

6.2.1 Research Contribution

The main contributions of this research can be highlighted as follows:

- **Developing a complete Sinhala speech-to-speech chatbot system integrating ASR, Chatbot, and TTS modules.**
  This research proposes and implements a complete end-to-end Sinhala speech-to-speech chatbot pipeline. Unlike existing solutions that address only individual components, this system integrates Sinhala ASR, a Sinhala language capable chatbot using RAG, and a Sinhala TTS system to enable natural, spoken dialogue in Sinhala. This holistic integration bridges a significant technological gap for Sinhala speakers.

- **Enhancing Sinhala ASR using transfer learning techniques.**
  Building upon the most recent work by Nanayakkara and Weerasinghe (2023), this research explores the use of alternative pre-trained models through transfer learning to further improve the performance of Sinhala ASR. The primary objective is to enhance the accuracy of speech recognition beyond the results achieved in the mentioned study, specifically for conversational Sinhala.

- **Identifying and adapting suitable LLMs for Sinhala chatbots.**

  The research evaluates and selects appropriate LLMs capable of generating high-quality, context-aware responses in Sinhala. This contributes significantly to the underexplored area of Sinhala NLP using LLMs.

- **Developing suitable RAG pipeline for Sinhala chatbots.**

  The research evaluates and selects appropriate chunk sizes, overlap sizes, embedding models and vector databases capable of generating up to date, high-quality, context-aware responses in Sinhala. These models are used to fine-tuned the RAG pipeline to improve their contextual understanding of Sinhala inputs, enabling more natural, human-like, and personalized responses. This contributes significantly to the underexplored area of Sinhala NLP using RAG architecture.

- **Training end-to-end VITS TTS model specifically for the Sinhala language.**

  This research successfully trains and evaluates a VITS model as the first fully end-to-end DLB TTS solution for Sinhala. The model achieves natural and intelligible speech synthesis in Sinhala by addressing the language's unique phonetic and syntactic characteristics. Before this work, no fully end-to-end DLB TTS system had been developed specifically for the Sinhala language, marking this as a significant advancement in Sinhala TTS research.

- **Laying the groundwork for future Sinhala NLP research.**

  This project provides a reusable, modular architecture for ASR, Chatbot, and TTS in Sinhala, creating a foundation for future developments. The trained models and associated code can be extended to different domains or used as baselines for other low-resource language research.

6.2.2 Individual Contribution

The following members contributed to the successful implementation of the Sinhala Speech-to-Speech Chatbot system, each focusing on a key research component. The team comprised three dedicated individuals:

Inuri – *L.I.L. Harischandra (Index No: 20000715)*

Sasangi –*K.K.S. Nayanathara (Index No: 20001207)*

Thamira – *T.V.R. Weerakoon (Index No: 20002009)*

ASR (Automatic Speech Recognition)	
• Dataset Selection • Preprocessing for ASR • ASR model training using transfer learning • Evaluation of ASR models	Inuri
TTS (Text-to-Speech)	
• Dataset Selection and Preparation • VITS TTS Model Configuration and Training • Implementation of Preprocessing Steps • Evaluation of TTS models	Sasangi
Chatbot System	
• Exploration and evaluation of Suitable chunk sizes and overlap sizes • Exploration and evaluation of Suitable embedding models • Exploration and evaluation of Suitable vector databases • Exploration and evaluation of Suitable LLMs • Development of Sinhala Chatbot using RAG pipeline	Thamira
Speech-to-Speech Chatbot System	
Integration of ASR and TTS Models into Chatbot	Inuri, Sasangi, Thamira
Speech-to-Speech Chatbot System UI Design	Inuri, Sasangi, Thamira
Frontend Development	Inuri, Sasangi, Thamira
System Evaluation and Testing	Inuri, Sasangi, Thamira

Table 6.1: Individual Contributions

6.3 Challenges Faced

6.3.1 Automatic Speech Recognition

- **Fine-tuning large models with limited resources**

  Adapting large pre-trained models like Whisper and Wav2Vec2-BERT required significant computational power and memory, which were limited during experimentation. Model training was conducted on Kaggle, which provides 30 hours of GPU usage per week. However, each model required more than 30 hours to train, making the process time-consuming and requiring careful scheduling and resource management.

6.3.2 Chatbot

- **Limited Availability of Sinhala Language Resources**

  High-quality embeddings and large-scale language models for Sinhala are scarce compared to high-resource languages like English. This limits the accuracy and fluency of responses generated by the system in Sinhala.

- **Chunking and Context Loss**

  Although the system uses an optimized chunking strategy, breaking long documents into smaller segments may cause slight context loss between chunks. This can affect the chatbot's ability to provide fully coherent and context-aware responses.

- **Limited Computational Resources**

  The system's performance is affected by the availability of processing power. Limited hardware capabilities can lead to slower processing, especially during model inference and retrieval steps. Scalability is also restricted, making it challenging to serve a large number of users simultaneously.

- **Dependency on External APIs**

  The system relies on third-party services such as Groq for LLM inference. This introduces latency in response times and creates a dependency on external infrastructure, which may affect reliability.

6.3.3 Text-to-Speech

- **Hardware Setup and GPU Requirements**

  Training the VITS TTS model required a high-performance GPU-enabled environment. Initially, platforms like Google Colab were considered; however, they lacked sufficient storage and computational capabilities to efficiently handle the complete training process. The university provided a GPU machine

(Ant PC) that was used as a solution. However, this machine initially had an outdated CUDA version, which was incompatible with the TTS training requirements. The CUDA toolkit had to be upgraded to version 10 or higher, along with necessary driver updates, before training could proceed.

- **Library Installation and Dependency Conflicts**

  Another significant issue involved installing the required TTS library. Attempting to install it through conventional package managers resulted in dependency conflicts with other pre-installed libraries, which caused the installation process to hang or fail. To overcome this, the TTS library was manually cloned and installed from its official GitHub repository, resolving the compatibility issues.

6.3.4 Speech-to-Speech Chatbot

- **Low-Resource Language Constraints**

  Lack of large, annotated Sinhala speech and text datasets. Limited availability of pre-trained Sinhala models for ASR, TTS, and NLP.

- **Handling Diverse Speech Inputs in ASR**

  While the ASR system demonstrated solid performance overall, recognizing the wide range of Sinhala dialects, accents, and informal speech patterns remained a challenge. Variability in speaker pronunciation, background noise, and frequent Sinhala-English code-switching affected transcription accuracy in some cases.

- **Improving the Expressiveness of TTS**

  The TTS system successfully generated intelligible Sinhala speech. However, producing speech with natural prosody and emotion, especially in dynamic conversational contexts, required further fine-tuning. Minor issues such as unnatural intonation or occasional mispronunciations were observed, which are common in low-resource language TTS systems.

- **Integration Complexity**

  Combining ASR, RAG-based chatbot, and TTS into one seamless pipeline requires synchronization and low-latency processing. Error propagation from one module to another affects overall performance.

- **LLM Hallucination**

  The chatbot may generate confident but incorrect or entirely fabricated information, especially when relevant data is missing from the retrieval step or training corpus. Users may interpret hallucinated answers as factual, leading to misinformation—this is especially critical in knowledge-sensitive domains like healthcare, education, or finance. Detecting and correcting hallucinations in Sinhala (a low-resource language) is more challenging due to the limited tools for fact-checking and verifying outputs.

Chapter 7

Conclusion

7.1 Chapter Overview

This chapter serves as a summary and reflection on the research study, providing insight into the limitations of the study and future research directions.

7.2 Conclusion about the research questions

This research successfully presents the design and implementation of a comprehensive Sinhala speech-to-speech chatbot system by integrating ASR model, a Sinhala-capable chatbot, and TTS synthesis into a unified pipeline. The system addresses the critical technological gap for Sinhala speakers by enabling natural, spoken human-computer interaction in their native language.

The ASR component demonstrated significant improvements in transcription accuracy by leveraging transfer learning approaches. Among the evaluated models, Whisper, Wav2Vec2-XLSR, and Wav2Vec2-BERT, the Wav2Vec2-BERT model achieved the best performance with WER and CER. These results emphasize the effectiveness of contextual embeddings in enhancing recognition of conversational Sinhala speech.

The chatbot module utilized a RAG approach, offering both factual accuracy and contextually relevant responses to user queries. This ensured meaningful and intelligent dialogue generation tailored to Sinhala linguistic and cultural nuances. The integration of LLMs further enhanced the depth and diversity of responses, even in low-resource settings.

The TTS component was developed using the VITS model, marking the first attempt to train a fully end-to-end deep learning-based TTS system for the Sinhala language. The results from both objective (WER,

CER) and subjective (MOS, SUS) evaluations confirmed that the system could generate highly intelligible and natural-sounding Sinhala speech across single and multi-speaker datasets.

Despite the successful implementation, the project faced several challenges, including limited availability of high-quality Sinhala datasets, computational resource constraints, and phonetic complexity of the language. Nonetheless, through rigorous experimentation and model fine-tuning, the research achieved promising results.

This project not only contributes to the advancement of Sinhala NLP and speech technologies but also lays a solid foundation for future research. The reusable modular architecture, trained models, and documented methodology can serve as valuable resources for extending similar applications to other domains or low-resource languages. Future work could focus on improving real-time performance, expanding domain-specific capabilities, and incorporating multilingual or code-mixed speech handling.

Overall, this research demonstrates the feasibility and impact of using deep learning to build inclusive, accessible, and intelligent speech-driven systems for underrepresented languages like Sinhala.

7.3 Future Works

7.3.1 Automatic Speech Recognition

- Extend the ASR system to support Sinhala-English code-mixed speech, which is commonly used in informal communication, especially on digital platforms.
- Add modules for identifying and segmenting different speakers in a conversation and detecting actual speech segments, improving usability in multi-speaker scenarios.
- Optimize the model for real-time transcription with minimal delay, allowing seamless integration into live speech applications such as virtual assistants or live translation tools.
- Enhance readability and usability of ASR output by automatically generating grammatically correct text with proper punctuation and casing.

7.3.2 Chatbot

- Refine Semantic Search Techniques: Enhance document retrieval accuracy by improving the relevance of retrieved passages through better similarity metrics and retriever models.
- Improve Sinhala Language Understanding: Develop or integrate more robust Sinhala embeddings and language models to improve understanding and generation in this morphologically rich, low-resource language.

- Optimize Chunking Strategies: Experiment with adaptive or context-aware chunking methods to minimize information loss between segments and maintain coherence.
- Enhance Real-Time Performance: Optimize the system for faster response times by reducing latency in retrieval and inference, particularly important for speech-based interfaces.
- Reduce Dependency on External APIs: Transition towards using on-premise or open-source LLMs and infrastructure to reduce latency and increase system autonomy.
- Handle Hallucinations More Effectively: Incorporate post-processing or verification layers to detect and correct hallucinated outputs, ensuring more reliable responses.

7.3.3 Text-to-Speech

- Fine-tune a pretrained multilingual VITS model to improve Sinhala TTS performance by leveraging multilingual knowledge transfer.
- Expand and enhance the Sinhala TTS dataset with more diverse, high-quality data to improve the naturalness and intelligibility of synthesized speech.
- Explore voice cloning techniques using the VITS model to enable personalized and speaker-adaptive Sinhala speech synthesis.

7.3.4 Speech-to-Speech Chatbot

- Improve Sinhala ASR Accuracy: Enhance ASR for Sinhala by training with diverse and dialect-rich speech datasets.
- Refine Sinhala TTS Output: Increase naturalness and intelligibility of TTS synthesis by fine-tuning models on native voice data with varied expressions.
- Optimize Real-Time Interaction: Reduce latency throughout the speech-to-speech pipeline to support smooth, real-time communication.
- Improve Low-Resource Language Modeling Build higher quality Sinhala embeddings and fine-tune multilingual LLMs to better serve low-resource language scenarios.
- Multimodal Expansion: Enable support for multimodal queries by integrating visual data such as scanned documents or images alongside text and speech.

References

Akkiraju, Rama et al. (2024). *FACTS About Building Retrieval Augmented Generation-based Chatbots*. arXiv: 2407.07858 [cs.LG]. URL: https://arxiv.org/abs/2407.07858.

Amarasingha, WGTN and DDA Gamini (2012). "Speaker independent sinhala speech recognition for voice dialling". In: *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*. IEEE, pp. 3–6.

Amazon.com (2023). *Amazon Alexa*. URL: https://developer.amazon.com/en-US/alexa.

Arachchige, Tharuka Kasthuri and Ruvan Weerasinghe (2023). "TacoSi: A Sinhala Text to Speech System with Neural Networks". In: DOI: 10.1109/ICARC57651.2023.10145749.

Avishka, Pasan et al. (2021). "Towards a More Intuitive Sinhala Chatbot: Leveraging NLU for Enhanced Intent Identification and Entity Extraction". In: *Sabaragamuwa University Journal* 19.2, pp. 46–63.

Baevski, Alexei et al. (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in neural information processing systems* 33, pp. 12449–12460.

Bahdanau, Dzmitry et al. (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473*.

Barrault, Loıc et al. (2023). "Seamless: Multilingual Expressive and Streaming Speech Translation". In: *arXiv preprint arXiv:2312.05187*.

Bhayana, Rajesh (2024). "Chatbots and Large Language Models in Radiology:A Practical Primer for Clinical and Research Applications". In: *Radiology* 310.

Bill, Desire;e and Theodor Eriksson (2023). *Fine-tuning a LLM using Reinforcement Learning from Human Feedback for a Therapy Chatbot Application*.

Caldarini, Guendalina et al. (Jan. 2022). "A Literature Survey of Recent Advances in Chatbots". In: *Information* 13.1, p. 41. ISSN: 2078-2489. DOI: 10.3390/info13010041. URL: http://dx.doi.org/10.3390/info13010041.

Casanova, Edresson et al. (2022). "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone". In: *International conference on machine learning*. PMLR, pp. 2709–2720.

Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

Deekshita, P et al. (2024). "PDF Chatbot Using Generative AI (LLMs & RAG)". In: *Journal of Nonlinear Analysis and Optimization* 15.

Dharwadkar, Rashmi and Dr.Mrs. Neeta A. Deshpande (2018). "A Medical ChatBot". In: *International Journal of Computer Trends and Technology (IJCTT)* 60.1.

Dias, Gihan and Sanath Jayasena (2009). "Sinhala Text to Speech System". In.

Feng, Xincan and Akifumi Yoshimoto (2024). "Llama-vits: Enhancing tts synthesis with semantic awareness". In: *arXiv preprint arXiv:2404.06714*.

Gamage, Bimsara et al. (2020). "The impact of using pre-trained word embeddings in Sinhala chatbots". In: *20th International Conference on Advances in ICT for Emerging Regions*.

Gamage, Buddhi, Randil Pushpananda, Thilini Nadungodage, et al. (2021). "Improve sinhala speech recognition through e2e lf-mmi model". In: *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pp. 213–219.

Gamage, Buddhi, Randil Pushpananda, Ruvan Weerasinghe, et al. (2020). "Usage of Combinational Acoustic Models (DNN-HMM and SGMM) and Identifying the Impact of Language Models in Sinhala Speech Recognition". In: *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 17–22. DOI: 10.1109/ICTer51097.2020.9325439.

Gao, Yunfan et al. (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv: 2312.10997 [cs.CL]. URL: https://arxiv.org/abs/2312.10997.

Gittmann, Yannick et al. (2024). "Voting Advice Applications: Implementation of RAG-supported LLMs". MA thesis.

Google (2024). *Google Assistant*. URL: https://assistant.google.com/intl/en_in/.

Goonatilleke, M.A.S.T et al. (2020). "Tilly – A Tamil Learning Chatbot for Non-Native Tamil Speakers". In: *International Conference on Artificial Intelligence*.

GroqCloud (2025). *Documentation*. URL: https://console.groq.com/docs/models.

Gumma, Varun et al. (2024). "MunTTS: A Text-to-Speech System for Mundari". In: *arXiv preprint arXiv:2401.15579*.

Gunasekara, M.K.H. and R.G.N. Meegama (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. DOI: 10.1109/ICTER.2015.7377680.

Hannun, A (2014). "Deep Speech: Scaling up end-to-end speech recognition". In: *arXiv preprint arXiv:1412.5567*.

Harshani, S.A.D.U. (2019). "Sinhala Chatbot for Train Information". MA thesis.

Hettige, B. and Asoka. S. Karunananda (2006). "First Sinhala Chatbot in action". In: *Sri Lanka Association for Artificial Intelligence (SLAAI) Proceedings of the third Annual Sessions*.

Hinton, Geoffrey et al. (2012). "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal processing magazine* 29.6, pp. 82–97.

Hu, Yifan et al. (2022). "MnTTS: an open-source mongolian text-to-speech synthesis dataset and accompanied baseline". In: *2022 International Conference on Asian Language Processing (IALP)*. IEEE, pp. 184–189.

HugginFace (2025). *MMTEB: Massive Multilingual Text Embedding Benchmark*. URL: `https://huggingface.co/spaces/mteb/leaderboard`.

Inc., Apple (2024). *Siri*. URL: `https://www.apple.com/siri/`.

Jayawardhana, Pabasara et al. (2019). "An intelligent approach of text-to-speech synthesizers for english and sinhala languages". In: *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*. IEEE, pp. 229–234.

Ju, Yooncheol et al. (2022). "TriniTTS: Pitch-controllable End-to-end TTS without External Aligner." In: *Interspeech*, pp. 16–20.

Karunanayake, Yohan et al. (2019). "Transfer learning based free-form speech command classification for low-resource languages". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 288–294.

Khadka, Supriya et al. (2023). "Nepali Text-to-Speech Synthesis using Tacotron2 for Melspectrogram Generation". In: *Proc. 2nd Annual Meeting of the ELRA/ISCA SIG on Under-resourced Languages (SIGUL 2023)*, pp. 73–77.

Kim, Jaehyeon et al. (2021). "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech". In: *International Conference on Machine Learning*. PMLR, pp. 5530–5540.

Kulkarni, Mandar et al. (2024). *Reinforcement Learning for Optimizing RAG for Domain Chatbots*. arXiv: `2401.06800 [cs.CL]`. URL: `https://arxiv.org/abs/2401.06800`.

Kumanayake, U.E. (2015). "A Sinhala chatbot for user inquiries regarding Degree Programs at University of Ruhuna". MA thesis.

Kumar, Gokul Karthik et al. (2023). "Towards Building Text-to-Speech Systems for the Next Billion Users". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1–5.

LangChain (2025). *Embedding models*. URL: `https://python.langchain.com/docs/integrations/text_embedding/`.

Lehto, Timo (2024). "Developing LLM-powered Applications Using Modern Frameworks". MA thesis.

Lewis, Patrick et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Asso-

ciates, Inc., pp. 9459–9474. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Li, Longfei et al. (2013). "Hybrid Deep Neural Network–Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition". In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 312–317. DOI: 10.1109/ACII.2013.58.

Liang, Kailin et al. (2023). "Comparative Study for Multi-Speaker Mongolian TTS with a New Corpus". In: *Applied Sciences* 13.7, p. 4237.

M, Kiruthiga Devi et al. (2021). "FARMER'S ASSISTANT using AI Voice Bot". In: *3rd International Conference on Signal Processing and Communication*.

Madhusha, Thirosh et al. (2023). "Mobile Base Sinhala Book Reader for Visually Impaired Students". In: *International Research Journal of Innovations in Engineering and Technology* 7.11, p. 127.

Manamperi, Wageesha et al. (2018). "Sinhala speech recognition for interactive voice response systems accessed through mobile phones". In: *2018 Moratuwa Engineering Research Conference (MERCon)*. IEEE, pp. 241–246.

Mansurova, Aigerim et al. (Sept. 2023). "DEVELOPMENT OF A QUESTION ANSWERING CHATBOT FOR BLOCKCHAIN DOMAIN". In: *Scientific Journal of Astana IT University*, pp. 27–40. DOI: 10.37943/15XNDZ6667.

Mazur, Orly and Adam B. Thimmesch (2024). "Beyond ChatGPT: Transforming Government with Augmented LLMs". In: *Tennessee Law Review*. Forthcoming, SMU Dedman School of Law Legal Studies Research Paper No. 651. URL: https://ssrn.com/abstract=4846472.

Melo, Maisa Kely de et al. (2025). "Improving Customer Journeys: Data-Driven LLM Chatbot Customization". In.

Microsoft (2024). *Cortana*. URL: https://www.microsoft.com/en-us/cortana.

Modran, Horia et al. (July 2024). "LLM Intelligent Agent Tutoring in Higher Education Courses using a RAG Approach". In: DOI: 10.20944/preprints202407.0519.v1.

Mohamed, Abdel-rahman (2014). "Deep Neural Network Acoustic Models for ASR." PhD thesis. University of Toronto Toronto, Canada.

Nadungodage, Thilini and Ruvan Weerasinghe (2011). "Continuous sinhala speech recognizer". In: *Conference on Human Language Technology for Development, Alexandria, Egypt*. Citeseer, pp. 2–5.

— (2013). "Efficient use of training data for Sinhala speech recognition using active learning". In: *2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 149–153.

Nanayakkara, Lakshika, Chamila Liyanage, et al. (2018). "A Human Quality Text to Speech System for Sinhala." In: *SLTU*, pp. 157–161.

Nanayakkara, Lakshika and Ruvan Weerasinghe (2023). "Exploring Model-Level Transfer Learning to Improve the Recognition of Sinhala Speech". In: *International Conference on Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication*. Springer, pp. 17–28.

Nishimura, Yuto et al. (2024). "HALL-E: Hierarchical Neural Codec Language Model for Minute-Long Zero-Shot Text-to-Speech Synthesis". In: *arXiv preprint arXiv:2410.04380*.

Oord, Aaron van den et al. (2016). "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499*.

Ouattara, Maimouna et al. (Jan. 2025). "Bridging Literacy Gaps in African Informal Business Management with Low-Resource Conversational Agents". In: *Proceedings of the First Workshop on Language Models for Low-Resource Languages*. Ed. by Hansi Hettiarachchi et al. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 193–203. URL: https://aclanthology.org/2025.loreslm-1.15/.

Oyucu, Saadin (2023). "A Novel End-to-End Turkish Text-to-Speech (TTS) System via Deep Learning". In: *Electronics* 12.8, p. 1900.

Pamisetty, Giridhar and K Sri Rama Murty (2023). "Prosody-TTS: An end-to-end speech synthesis system with prosody control". In: *Circuits, Systems, and Signal Processing* 42.1, pp. 361–384.

Pathirage, N.M. et al. (2023). "Voice-bot for Book An Appointment". In: *The 4th International Conference on Innovations in Info-business Technology*.

PathNirvana (2023). *Path Nirvana Sinhala TTS Dataset*. https://github.com/pnfo/sinhala-tts-dataset. GitHub repository. URL: https://github.com/pnfo/sinhala-tts-dataset.

Prakash, Anusha and Hema A Murthy (2020). "Generic indic text-to-speech synthesisers with rapid adaptation in an end-to-end framework". In: *arXiv preprint arXiv:2006.06971*.

Prasangini, Nishadi and Harshani Nagahamulla (2018). "Sinhala speech to sinhala unicode text conversion for disaster relief facilitation in sri lanka". In: *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*. IEEE, pp. 1–6.

Priyadarshani, PGN et al. (2012). "Dynamic time warping based speech recognition for isolated Sinhala words". In: *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, pp. 892–895.

Radford, Alec et al. (2023). "Robust speech recognition via large-scale weak supervision". In: *International conference on machine learning*. PMLR, pp. 28492–28518.

Rafat, Md Irfan (2024). "AI-powered Legal Virtual Assistant: Utilizing RAG optimized LLM for Housing Dispute Resolution in Finland". MA thesis.

Rajapakshe, DDS et al. (2020). "Sinhala conversational interface for appointment management and medical advice". In: *2020 2nd International Conference on Advancements in Computing (ICAC)*. Vol. 1. IEEE, pp. 85–90.

Rajapakshe et al. (2020). "Sinhala Conversational Interface for Appointment Management and Medical Advice". In: *2020 2 nd International Conference on Advancements in Computing*.

Rohman, M. Abdul and Pungkas Subarkah (2024). "Design and Build Chatbot Application for Tourism Object Information in Bengkulu City". In: *Journal of Information Technology and Strategic Innovation Management* 1.1, pp. 28–34.

Senarathna, Manuri et al. (2022). "Step-by-Step Process of Building Voices for Under Resourced Languages using MARY TTS Platform". In: *2022 4th International Conference on Advancements in Computing (ICAC)*. IEEE, pp. 18–23.

Sodimana, Keshan et al. (2018). "A step-by-step process for building tts voices using open source data and framework for bangla, javanese, khmer, nepali, sinhala, and sundanese". In.

Srivastava, Nimisha et al. (2020). "Indicspeech: text-to-speech corpus for indian languages". In: *Proceedings of the 12th language resources and evaluation conference*, pp. 6417–6422.

Tan, Xu et al. (June 2021). "A Survey on Neural Speech Synthesis". In: URL: `http://arxiv.org/abs/2106.15561`.

Tankala, Pavan et al. (2024). "STORiCo: Storytelling TTS for Hindi with Character Voice Modulation". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 426–431.

TURING, A.M. (1950). "COMPUTING MACHINERY AND INTELLIGENCE". In: *Mind* LIX.236, pp. 433–460.

Wang, Xiaohua et al. (Nov. 2024). "Searching for Best Practices in Retrieval-Augmented Generation". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan et al. Miami, Florida, USA: Association for Computational Linguistics, pp. 17716–17736. DOI: `10.18653/v1/2024.emnlp-main.981`. URL: `https://aclanthology.org/2024.emnlp-main.981/`.

Wasala, Asanka and Kumudu Gamage (2005). "Research report on phonetics and phonology of Sinhala". In: *Language Technology Research Laboratory, University of Colombo School of Computing* 35, p. 11.

Weerasinghe, Ruvan et al. (2007). "Festival-si: A sinhala text-to-speech system". In: *Text, Speech and Dialogue: 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, 2007. Proceedings 10*. Springer, pp. 472–479.

Yang, Xianjun and Stephen D. Wilson andLinda Petzold (2024). *Quokka: An Open-source Large Language Model ChatBot for Material Science*.

Yousaf, Maya Bint (Jan. 2025). “Artificial Intelligence Based Chatbot on RasaNLU System Empowered with Deep Learning”. In: *SES* 3.1, pp. 320–338.

Zhao, Wei and Zheng Yang (2023). “An emotion speech synthesis method based on vits”. In: *Applied Sciences* 13.4, p. 2225.

Appendix A: Evaluation Form

Click here to visit the evaluation form

User Evaluation of VocSi

In an increasingly interconnected world, speech-to-speech chatbots are revolutionizing the way people communicate, breaking down language barriers and enabling more natural interactions with technology. Recent advancements in Natural Language Processing (NLP) have significantly enhanced chatbot capabilities, particularly in voice-based communication. While widely used voice assistants like Microsoft Cortana, Amazon Alexa, Apple Siri, and Google Assistant demonstrate the potential of these technologies, they often fall short in supporting low-resource languages. This project addresses that gap by developing a speech-to-speech chatbot tailored for Sinhala speakers, promoting digital inclusivity and enhancing access to technology in native languages.

This project focuses on developing a conversational AI system for Sinhala, a low-resource language that faces challenges due to limited training data. To overcome these limitations, a Retrieval-Augmented Generation (RAG) approach was implemented, significantly improving the relevance and quality of responses compared to standalone language models. The system integrates Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) components, enabling users to interact through spoken Sinhala. Among various models tested, the LLaMA-3.3-70B combined with RAG delivered the best performance. While RAG may slightly increase response time, it greatly enhances the fluency and contextual accuracy of responses. This research highlights the potential of RAG-based systems to support conversational AI in Sinhala and other underrepresented languages.

Thank you for taking the time to participate in this evaluation. Your feedback is valuable and will help improve the quality and effectiveness of Sinhala conversational AI systems.

Sample Questions : Sample Question Set

* Indicates required question

Email *

Record **2020cs071@stu.ucsc.cmb.ac.lk** as the email to be included with my response

Email *

Your answer

Name *

Your answer

Task Success Rate: *

Measures how often the chatbot successfully fulfills user requests or completes tasks.

1 2 3 4 5

☆ ☆ ☆ ☆ ☆

Utterance and Response Matching Score: *

Assesses how well the chatbot's responses align with user inputs, ensuring coherence and relevance.

1 2 3 4 5

☆ ☆ ☆ ☆ ☆

Response Time: *

Measures the speed at which the chatbot responds to user queries.

1 2 3 4 5

Confusion Rate: *

Tracks the frequency of user confusion or frustration during the interaction, indicating areas for improvement.

1 2 3 4 5

User Satisfaction: *

Gathers feedback from users to assess their overall experience with the chatbot.

1 2 3 4 5

Clarity and Concision: *

Evaluates the chatbot's ability to communicate clearly and concisely, ensuring easy understanding.

1 2 3 4 5

Conversation Relevancy: *

Measures how relevant the chatbot's responses are to the user's queries.

1 2 3 4 5

☆ ☆ ☆ ☆ ☆

Conversation Completeness: *

Assesses whether the chatbot provides all the necessary information to fulfill the user's request.

1 2 3 4 5

☆ ☆ ☆ ☆ ☆

Overall experience with Automatic Speech Recognition Component *

1 2 3 4 5

☆ ☆ ☆ ☆ ☆

Overall experience with Text to Speech Component *

1 2 3 4 5

☆ ☆ ☆ ☆ ☆

Suggestions for improvement

Your answer

Send me a copy of my responses.

Submit

Clear form

Never submit passwords through Google Forms.

Appendix B: Set of Sample Questions

Level 1 - Yes & No Questions

1. වෘත්තීය සමිති පිහිටුවීමේ සහ වෘත්තීසමිතිවලට බැඳීමේ නිදහසක් ව්‍යවස්ථාවෙන් ලබා දී ඇත්තේ වරප්‍රසාද ලත් පිරිසකට පමණක් ද?

    නැත. සෑම පුරවැසියෙකුටම ලබා දී තිබේ

2. ශ්‍රී ලංකා ජනරජයේ පරමාධිපත්‍යය ඇත්තේ විධායක ජනාධිපතිවරයාට ද?

    නැත. එය ඇත්තේ ජනතාවටයි.

3. ශ්‍රී ලංකා පුරවැසියෙකු අල්ලස් චෝදනාවකට වරදකරු වී පස් වසරක් ගත වී ඇත. ඔහුට පාර්ලිමේන්තු මැතිවරණය සඳහා ඡන්දය ලබා දිය හැකි ද? එසේ නොවන්නේ නම් ඊට ඇති නීතිමය බාධාව කුමක් ද?

    නොහැක. ඒ සඳහා වසර හතක් ගත විය යුතු ය.

4. ශ්‍රී ලංකා ව්‍යවස්ථාව අනුව අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ අගමැතිවරයා ද?

    නැත. අමාත්‍ය මණ්ඩල ප්‍රධානියා වන්නේ ජනාධිපතිවරයාය.

5. ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාරවරයෙකු පත් කිරීමේ යම් දෝෂයක් ඇති විට ශ්‍රේෂ්ඨාධිකරණයේ පවතින නඩු කටයුත්තක් නිර්බල වන්නේ ද?

    නැත. ශ්‍රේෂ්ඨාධිකරණයේ යම් විනිශ්චයකාර ධුරයක් හිස්ව තිබුණ ද, යම් විනිශ්චයකාරවරයෙකු පත් කිරීමේ දෝෂ තිබුණ ද එහි පවතින නඩු කටයුතු නිර්බල වන්නේ නැත.

Level 2 - What Questions

6. වරක් තේරී පත් වී සිටින ජනාධිපතිවරයෙකුට නැවත ජනවරමක් ලබා ගැනීම සඳහා තේරී පත් වීමෙන් පසු අවම වශයෙන් වසර කීයක් ඉක්ම විය යුතු ද?

අවම වශයෙන් පත්වීමෙන් වසර හතරක් ඉක්ම විය යුතු ය.

7. ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාව සාමාජිකයින් කී දෙනෙකුගෙන් සමන්විත විය යුතු ද?

සාමාජිකයින් 10කි.

8. අමාත්‍ය මණ්ඩලයට අයත් උපරිම ඇමතිවරුන් ගණන සහ අමාත්‍ය මණ්ඩල සාමාජිකයින් නොවන ඇමතිවරුන් සහ නියෝජ්‍ය ඇමතිවරුන්ගේ උපරිම ගණන කොපමණද?

අමාත්‍ය මණ්ඩලයට අයත් උපරිම සංඛ්‍යාව 30කි
අමාත්‍ය මණ්ඩල සාමාජිකයින් නොවන ඇමතිවරුන් සහ නියෝජ්‍ය ඇමතිවරුන් උපරිම ගණන 40කි

9. පර්ලිමේන්තුව අවම වශයෙන් වසරකට කීවරක් කැඳවිය යුතු ද?

අවම වශයෙන් එක් වරක්වත් කැඳවිය යුතුය.

10. අධිකරණ සේවා කොමිෂන් සභාවෙහි යම් රැස්වීමක ගණපූරණය සඳහා සාමාජිකයින් කී දෙනෙකු සහභාගී විය යුතුද?

ගණපූරණය සඳහා සාමාජිකයින් දෙදෙනෙකුගෙන් සමන්විත විය යුතුය.

Level 3 - How Questions

11. පාර්ලිමේන්තුවෙහි ඡන්ද විමසීමක දී මුලසුන දරන මන්ත්‍රීවරයා ඡන්දය දිය යුත්තේ කිනම් අවස්ථාවන්හි දී ද?

ඡන්ද සංඛ්‍යාව සම වූ අවස්ථාවන්හි දී මුලසුන දරන තැනැත්තාට තීරණ ඡන්දයක් හිමිවෙයි.

12. කිසියම් ආණ්ඩුකාරවරයෙකු සිය බලය අනිසි ලෙස භාවිත කරන්නේ නම් ඔහු ධුරයෙන් ඉවත් කළ හැක්කේ කෙසේ ද?

ආණ්ඩුකාරවරයා ඉවත් කිරීමට සභාවට යෝජනාවක් ඉදිරිපත් කොට සියලු මන්ත්‍රීවරුන්ගෙන් තුනෙන් දෙකකට වැඩි සංඛ්‍යාවක් විසින් සම්මත කොට ජනාධිපතිවරයාට ඉදිරිපත් කිරීමෙන් ආණ්ඩුකාරවරයා ධුරයෙන් ඉවත් කළ හැකි ය.

13. ජනාධිපතිවරණයක හෝ ජනමත විචාරණයක වලංගුභාවය සම්බන්ධයෙන් වන නඩු කටයුත්තක තීරණය ලබා දීමට බලය ඇත්තේ කාට ද?

අවම වශයෙන් ශ්‍රේෂ්ඨාධිකරණ විනිශ්චයකාරවරුන් පස් දෙනෙකු විසින් විභාග කොට තීරණය කළ යුතු වෙයි. අග විනිසුරුවරයා අන්‍යාකාර විධානයක් කළහොත් මිස ඔහු ද ඒ පස්දෙනාගෙන් කෙනෙකු විය යුතු ය.

14. ආණ්ඩුක්‍රම ව්‍යවස්ථා සභාවේ සංයුතිය කුමක් ද?

සාමාජිකයින් 10කි. පහත පරිදි වේ.

අගමැති
කථානායක
විපක්ෂ නායක
ජනාධිපති පත් කළ මන්ත්‍රී
ආණ්ඩු පක්ෂයේ එක් මන්ත්‍රීවරයෙක්
විපක්ෂයේ එකඟත්වයෙන් එක් මන්ත්‍රීවරයෙක්
අගමැති සහ විපක්ෂ නායක අනුමැතියෙන් කථානායක පත් කරන 3ක්
ආණ්ඩු පක්ෂයේ හෝ විපක්ෂ නායකගේ පක්ෂයේ නොවන එක් මන්ත්‍රීවරයෙක්

15. ව්‍යවස්ථානුකූලව විගණකාධිපති ධුරය හිස් විය හැකි අවස්ථා කවරේද?

මිය යෑම
ඉල්ලා අස් වීම
වයස 60 සම්පූර්ණ වීම

Level 4 - False Premise Questions

16. ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව අගවිනිසුරුවරයා ලෙස පත් කරනු ලැබූ තැනැත්තාට රාජ්‍යයේ කවරහෝ ඉහළ නිලධාරියෙකු ඉදිරියෙහි දිවුරුම් දීමට ඉඩ දී ඇත්තේ ඇයි? පැහැදිලි කරන්න.

නැත. අගවිනිසුරුවරයා ලෙස පත් කරනු ලැබූ තැනැත්තා පළමුව ජනාධිපතිවරයා ඉදිරියේ දිවුරුම් දිය යුත්තේය.

17. ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව ජනාධිපතිවරයාට කිසියම් මන්ත්‍රීවරයෙකු අමාත්‍ය මණ්ඩල සාමාජිකයෙක් නොවන අමාත්‍යවරයෙක් ලෙස පත් කිරීමට නොහැකිය. මෙය සත්‍යයක් ද?

අසත්‍යයකි. ජනාධිපතිවරයාට අගමැතිවරයාගේ ද අදහස් විමසා කිසියම් මන්ත්‍රීවරයෙකු අමාත්‍ය මණ්ඩල සාමාජිකයෙක් නොවන අමාත්‍යවරයෙක් ලෙස පත් කළ හැකිය.

18. ශ්‍රී ලංකාවෙහි ව්‍යවස්ථාව අනුව අමාත්‍ය මණ්ඩලයට රජයේ නිලධාරීන් පත් කිරීමට, මාරු කිරීමට හෝ සේවයෙන් පහ කිරීමට විධිවිධාන සැපයීමට බලය නැත. මෙය සාධනීය තත්ත්වයක් ද?

වැරදියි. අමාත්‍ය මණ්ඩලයට රජයේ නිලධාරීන් පත් කිරීමට, මාරු කිරීමට හෝ සේවයෙන් පහ කිරීමට විධිවිධාන සැපයීමට අදාළ බලය ව්‍යවස්ථාවෙන් ලබා දී තිබේ.

19. පළාත් පාලන ආයතන සහ වෙනත් අධිකාරීන්ට බදු සහ වරිපනම් නියම කිරීමේ බලය ශ්‍රී ලංකා ආණ්ඩුක්‍රම ව්‍යවස්ථාවෙන් සලසා තිබේ. මෙය සාධනීය තත්ත්වයක් ද?

අසත්‍යයි. පාර්ලිමේන්තුව විසින් සම්මත කරනු ලැබූ නීතියක හේ පවත්නා නීතියක අධිකාරය යටතේ මිස කිසිම පළාත් පාලන ආයතනයක් සඳහා බදු සහ වරිපනම් නියම කිරීමේ බලයක් ආණ්ඩුක්‍රම ව්‍යවස්ථාවෙන් සලසා නැත.

20. ශ්‍රී ලංකා ආණ්ඩුක්‍රම ව්‍යවස්ථාව අනුව රාජ්‍ය සේවා කොමිෂන් සභාවට රාජ්‍ය පරිපාලන ඇමතිවරයා විසින් සාමාජිකයින් පත් කරනු ලබයි. අගමැතිවරයා ඊට විරුද්ධ නම් ඔහු එම සාමාජිකයින් පත් කරන්නේ කෙසේ ද?

අසත්‍යයකි. රාජ්‍ය සේවා කොමිෂන් සභාව සාමාජිකයින් නව දෙනෙකුගෙන් සමන්විත වන අතර ඔවුන් පත් කරනු ලබන්නේ ජනාධිපතිවරයා විසිනි.