Developing a domain-dependent Automatic Speech Recognition system with minimal data

A Thesis submitted for the Degree of Master of Computer Science

P. D. I. V. Sugathadasa

University of Colombo School of Computing

2025

DECLARATION

Name of the student: P.D.I.V. Sugathadasa
Registration number: 2020/MCS/089
Name of the Degree Programme: Master of Computer Science
Project/Thesis title: Developing a domain-dependent Automatic Speech Recognition system with minimal data

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

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

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

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

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

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

Signature of the Student	**Date (DD/MM/YYYY)**
	24/06/2025

Certified by Supervisor

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

	Supervisor
Name	**Dr. B.H.R. Pushpananda**
Signature	
Date	24 - 06 -2025

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

ABSTRACT

The development of Automatic Speech Recognition (ASR) systems has substantially improved through deep learning models and extensive datasets during recent years. Accurate deployment of Automatic Speech Recognition systems within the banking domain proves difficult for the low-resource language Sinhala because of present obstacles in building efficient systems. The primary research objective of this research was to develop a domain-dependent ASR system that functions effectively with minimal data.

The research investigates limitations of present-day ASR systems for low-resource languages, highlighting how few Sinhala speech datasets are accessible to the public and showing no domain-specific corpora exist. A domain-specific private dataset was developed to meet this requirement by focusing on banking-related terminology and situations. The research achieved the objectives through OpenAI's Whisper model fine-tuning process with limited data for banking applications. The Whisper model achieved performance evaluation through measurements of Word Error Rate (WER) along with loss values after receiving fine-tuning through the specified dataset. The fine-tuning process of the model produced 22.39% WER which surpassed the primary error rates of commercially available Whisper models that directly interacted with the Sinhala banking dataset.

Transfer learning and fine-tuning methods play a critical role according to the research for improving ASR accuracy by using small data sets. The performance of automatic speech recognition depends on four key elements which involve data quality as well as domain suitability and model structures and appropriate tuning parameters. The research succeeded in its goals yet recognizes two major obstacles involving excessive computation requirements and insufficient database diversity.

This research demonstrates two essential outcomes: first it proves the development of domain-specific Sinhala ASR systems using limited data and second it offers a framework which can apply to low-resource languages and special applications for inclusive speech recognition technology accessibility. Future research directions include expanding the dataset, optimizing computational efficiency, and exploring advanced machine learning techniques to further enhance ASR accuracy and applicability.

ACKNOWLEDGEMENTS

I would like to express special thanks & gratitude to my research supervisor, Dr. B.H.R. Pushpananda Senior lecturer of the University of Colombo School of Computing who gave me a lot of ideas and help to work on this project on the topic of “Developing a domain-dependent Automatic Speech Recognition system with minimal data”, which led me into doing a lot of Research which diversified my knowledge to a huge extent for which We are thankful.

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

P.D.I.V. Sugathadasa

2020/MCS/089

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ASR – Automatic Speech Recognition

NLP – Natural Language Processing

GPU - Graphics Processing Unit

DNN - Deep Neural Networks

CNN - Convolutional Neural Networks

WER – Word Error Rate

LoRA - Low-Rank Adaptation

GMM - Gaussian Mixture Models

HMMs - Hidden Markov Models

TDNNs - Time Delay Neural Networks

LPC - Linear Predictive Coding

MFCC - Mel-Frequency Cepstral Coefficients

PT – Pre-Trained model

FT - fine-tuning

MLS - Multilingual LibriSpeech

FLEURS - Few-shot Learning Evaluation of Universal Representations of Speech

RNNs - Recurrent Neural Networks

LSTM - Long Short Term Memory

GRU - Gated Recurrent Units

BERT - Bidirectional Encoder Representations from Transformers

GPT - Generative Pre-trained Transformer

BOS - Beginning-Of-Sequence

CHAPTER 1 - INTRODUCTION

1.1. Overview

The ability of machines to interpret human language through speech recognition brought significant interest in the field in recent decades because analyzing language constitutes an essential part of human-computer interaction. Significant advancements in speech recognition have been made possible by the fourth generation of IT's broad adoption of big data, cloud computing, and machine learning. Large-scale data processing, the use of distributed computing, and the creation of sophisticated algorithms that can learn from data, make predictions, and perform activities that humans would perform have all been made possible by these technologies.

Automatic Speech Recognition (ASR) has undergone a transformational development, evolving from a simple template-based recognition system in the mid-20th century to a complex architecture capable of processing large vocabulary and continuous language. The first ASR system, developed in the early 1950s, could recognize only one speaker's isolated number (Levis and Suvorov, 2012). Since then, ASR technology has progressed to the point where a common distributed speech recognition service enables researchers to incorporate speech into their applications with minimal development efforts or special knowledge. This progress has been driven by statistical methods, particularly hidden Markov models (HMMs) and, recently, by the integration of deep learning technologies. The integration of deep learning significantly improved ASR performance, thereby enhancing the system's ability to model complex linguistic structures. Despite these advances, however, the development of an efficient, specific area-specific ASR system is still difficult, especially with limited data resources.

Speech is the most efficient and convenient method of communication, and many commercial applications such as Google Assistant, Apple Siri, Bing, and Amazon Alexa use ASR technology for a variety of purposes (Twiefel *et al.*, 2014). The main objective of the ASR system is to accurately convert the spoken language into written text or machine-readable format. This process involves the conversion of speech's acoustic waveform into a series of sounds, words, or sentences that computers can understand and process (Karunathilaka, 2020). Although ASR systems are widely used in various fields, their practical application in specific industries can be difficult. An important factor is that ASR systems are highly dependent on

training data and are typically trained using fixed vocabulary, grammar or statistical language models. This dependency can reduce performance when applied to external content (Morbini *et al.*, 2013). Therefore, the development of domain-dependent ASR systems developed and optimized for specific industries and environments has gained importance. These systems are tailored to handle unique vocabularies, grammar, and statistical language models relevant to their target fields.

In a country such as Sri Lanka, Sinhala is both the national language and the state language spoken by about 75 percent of the population (Wijesekera and Alford, 2019), and the domain-dependent ASR system of Sinhala holds important promise for local industry. A customized ASR system can benefit a user base of over 16 million people and improve organizational processes and user interactions. However, the development of such a system poses a particular challenge, especially in obtaining a sufficiently large and diverse corpus of speech to ensure the model is able to accurately recognize a wide range of patterns and styles of speech. Other difficulties include the treatment of variations in pronunciation, intonation and phoneme levels, as well as the treatment of Sinhala's morphological wealth, which can generate many forms of words from a single root.

Deep learning technologies help overcome these challenges by allowing systems to learn domain-specific language and terminology along with context-specific keywords which leads to better accuracy and effectiveness in domain-specific queries. The Karunathilaka, (2020) research demonstrates how deep neural network models enable competitive low-resource Sinhala ASR performance despite limited data availability. The study demonstrated that time delay neural networks (TDNNs) represent the best choice of architecture over traditional Gaussian mixture models and Hidden Markov models (GMM-HMMs). The research demonstrates that successful neuronal models still depend on both phonologically balanced corpora and expanded vocabularies for optimal performance.

Feature extraction serves as a critical component for effective operation of Automatic Speech Recognition systems. The feature extraction methods Linear Predictive Coding (LPC), Relative Spectral Filtering (RASTA), and Mel-Frequency Cepstral Coefficients (MFCC) are widely utilized despite possessing both strengths and weaknesses. LPC demonstrates reliable vocal tract characteristics estimation but encounters difficulties when identifying words with similar vowel sounds. The RASTA feature extraction technique provides noise resistance but usually requires additional methods to achieve optimal performance. MFCC stands as an industry

standard because it corresponds with human auditory perception but its effectiveness decreases in noisy environments. There is a clear need for advanced feature extraction systems which merge the advantages of various techniques to become robust.

Levis and Suvorov, (2012), research demonstrated a major advancement in ASR capabilities through the combination of HMMs with neural network models. These innovative hybrid models integrated HMMs' sequential modeling capabilities with the deep learning pattern recognition abilities to create a powerful framework for handling continuous speech with greater efficiency. The significant advancements in these systems require substantial training sets and computational power which limits their applicability in domain-specific or resource-constrained environments unless they are modified.

1.2. Motivation

This study aims to create a domain-specific Automatic Speech Recognition (ASR) system for the Sinhala language using minimal data. Developing ASR systems for low-resource languages such as Sinhala poses significant challenges, prompting research communities to explore modern deep learning techniques to enhance accuracy. By evaluating the methodologies of existing ASR systems, this research seeks to develop a more efficient ASR system for Sinhala, which would be especially valuable in areas where precise speech-to-text conversion is essential. Although there are sophisticated ASR systems available for English and other widely used languages, Sinhala currently lacks a strong, domain-targeted ASR solution, underscoring the importance of this research.

The primary research question addressed in this study is:
"How can we develop an efficient and accurate domain-dependent Sinhala ASR system with minimal data?"

To answer this, the following sub-questions will be explored:

1. What are the current limitations of ASR systems for low-resource languages, and how can they be improved for domain-specific applications?
2. How can machine learning and deep learning techniques be adapted to enhance ASR performance with limited training data?
3. How can we fine-tune the transfer learning approach to improve the accuracy of the domain-dependent Sinhala ASR?

By exploring these research questions, this study intends to create an ASR solution that successfully adjusts to specific domain needs while addressing the issues related to limited data availability.

1.3. Statement of Problem

The project work targets developing an ASR system able to achieve efficient performance using limited domain-specific data. System deployment requires solving three research gaps including domain-specific language model development with data augmentation methods and powerful feature extraction mechanisms. This research aims to create a reliable speech recognition solution for specialized applications that requires different methods because data-intensive methods prove ineffective based on the findings of this study.

1.4. Research Aims and Objectives

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

1.4.1. Aim

So, the primary aim of this research was to develop a domain-dependent ASR system that functions effectively with minimal data.

1.4.2. Objectives

- Developing a domain-dependent Sinhala Speech Recognition (ASR) system that can operate on a limited amount of training data.
- Investigating the effects of different hyperparameters, and training strategies on the performance of the proposed system.
- Performance comparison of the proposed system with the other ASR systems developed for other low-resource languages

1.5. Scope of the Study

1.5.1. In Scope

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

- Development of a domain-dependent ASR system for the Sinhala language.
- Focus on one specific domain with a small but domain-specific speech dataset.
- Addressing data scarcity through transfer learning modeling.
- Optimizing the ASR system for domain-specific language models to enhance

performance and applicability.

1.5.2. Out of Scope

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

- Development of a general-purpose ASR system for Sinhala or other languages.
- Consideration of multiple domains within the ASR system.
- Large-scale ASR development with extensive datasets.
- Multi-language or cross-lingual ASR capabilities.

1.6. Structure of the Thesis

The first chapter of this dissertation presents an overview of the research area, outlining the research background, key questions, objectives, methodologies, and scope. Chapter 2 provides a critical review of existing literature, highlighting the research gap and emphasizing the significance of the problem. It also explores potential solutions and discusses the methods and tools utilized in developing the proof-of-concept prototype. Chapter 3 details the research design, including its evolution and the rationale behind design decisions. Chapter 4 delves into the implementation phase, addressing key challenges encountered. Chapter 5 focuses on the evaluation of the proposed model, analyzing its results. Lastly, Chapter 6 presents a discussion on the findings, both positive and negative, along with research limitations and future directions.

CHAPTER 2 - LITERATURE REVIEW

People now communicate with devices through Automatic Speech Recognition technology which has transformed the communication experience making all interactions voice-activated on different platforms. Recent developments in neural networks combined with deep learning techniques have improved ASR system accuracy to a critical level required by healthcare fields and finance operations as well as telecommunication systems. The development of ASR systems that work with limited domain data proves vital for discovering next-generation voice-based applications. This study works to develop a domain-focused Sinhala ASR system that needs only a small amount of training data.

2.1. Evolution and Advancements in ASR

The ASR technology sector emerged in the early 1950s by focusing on identifying simple vocal expressions and spoken digits. From Bell Telephone Laboratories came the first ASR system in 1952 which detected single digits between 0 and 9 yet worked only with one voice. A phonetic typewriter controlled by Olson and Belar appeared in 1956 but needed thorough speaker training for recognizing ten distinct syllables. The first ASR models operated using pattern matching methods that analyzed speech-by-comparing it against pre-made acoustic templates, (Levis and Suvorov, 2012). When implemented this method delivered good results for identifying specific words with minimal vocabulary yet it failed to operate with extended wordlists or different speaking styles. Early ASR systems faced two severe practical problems because they depended heavily on manually created patterns which made them both resource-heavy and very sensitive to user-specific characteristics and background noises.

The years from 2010 to 2020 represented a breakthrough period for ASR because deep learning technology experienced swift advancements. Progress in automated speech recognition systems during that decade delivered momentous reductions in word error rates which made speech recognition technology accessible to new practical applications (Figure 2.1). Deep learning achieved success in ASR through three essential factors. Major speech datasets obtained from transcription allowed models to learn generalization capabilities that work across different languages and speaker ranges. Deep neural network training received substantial acceleration from the rapid technological development of Graphics Processing Units (GPUs) which made complex models usable for real-world implementations. The understanding capabilities of ASR systems grew more accurate and robust because of progressive learning algorithm

breakthroughs along with adopted deep neural networks (DNNs) and convolutional neural networks (CNNs) and finally transformer-based models. The progress enabled the implementation of ASR technology into different mainstream applications including virtual assistants and real-time transcription services as well as multilingual communication tools.(Hannun, 2021)

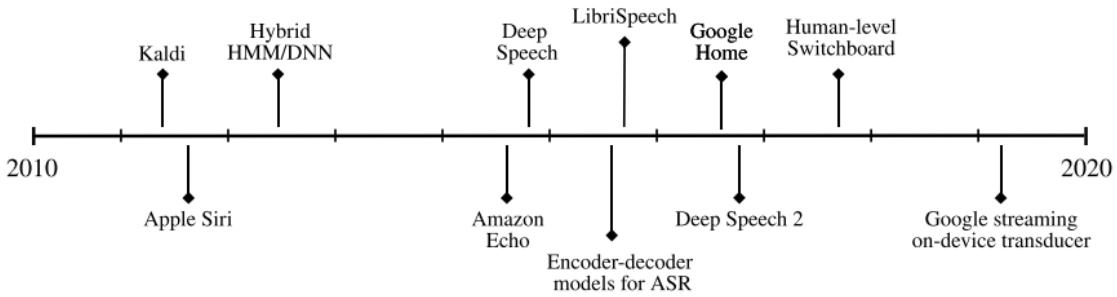

Figure 2.1: A timeline of some of the major developments in ASR

2.2. ASR System Architecture

ASR involves the conversion of spoken language into text through system analysis of audio input which produces textual outputs (Figure 2.2). Multiple approaches have been created since the beginning to improve both accuracy and efficiency in ASR systems. According to researchers' speech recognition methods rely on five different techniques including Acoustic-Phonetic modeling based on language understanding and Pattern Recognition methods using statistical pattern matching and expert-based Knowledge-Based strategies as well as Connectionist processing with neural networks and Support Vector Machine classification of speech features.

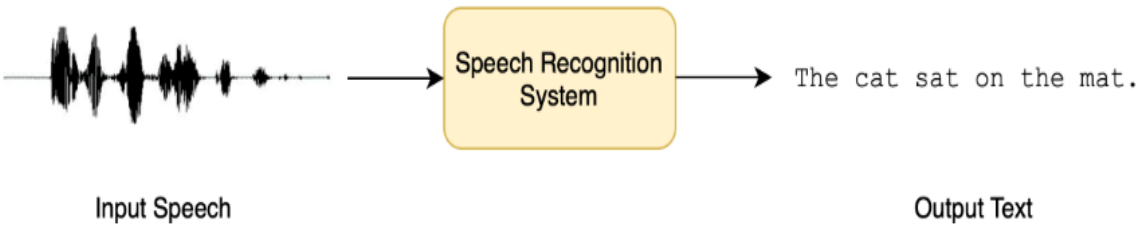

Figure 2.2: Automatic speech recognition (ASR) or speech-to-text (STT)

Two main stages mark the operation of an ASR system beginning with training followed by recognition. Training the system requires pre-defined speech samples along with their transcription documents which enables model development for spoken word recognition. The

analyzed speech enters during the recognition phase where the trained system compares it against the previously learned patterns. ASR technology detects recorded words during their training process but it recognizes unknown words through matched sub-words which exist in the system dictionary. The ability to understand linguistic patterns outside trained samples boosts contemporary ASR system adaptability for various speech patterns.

The three main components of ASR architecture follow signal processing through decoding with feature extraction in between. System performance depends heavily on the decoding phase that requires acoustic and language modeling together with pronunciation modeling and proves especially challenging (Figure 2.3) (Ghai and Singh, 2012).

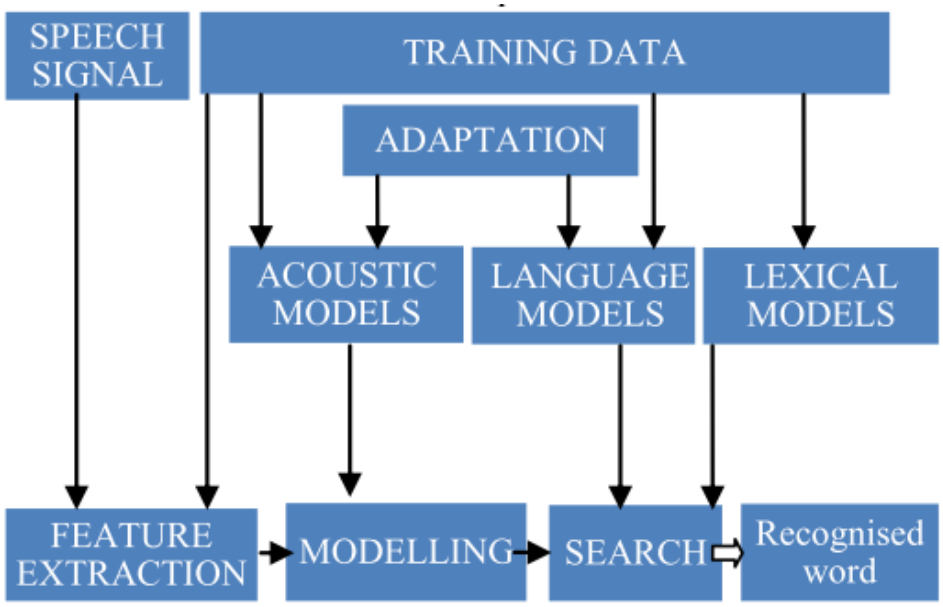

Figure 2.3: ASR block diagram

2.3. Transfer Learning for ASR

Transfer learning is a particularly important method that allows ASR systems to use the knowledge learnt for modelling in different domains to improve learning for new information-poor domains. Instead of training a new model from scratch for a low resource language, it is more effective to fine-tune an acoustic model which is trained on a different language with abundance of data. It enhances the general performance of the model to a great extend and cuts down time that would have taken in developing the model through the conventional machine learning. When it comes to ASR, name transfer learning is highly effective particularly for low-resource languages due to the shortage of speech material with annotations. For instance, Sinhalese, which do not have large labeled datasets, can be benefited from the perform pre-training on larger dataset in high resource languages. By incorporating such small language-

specific data in occurrence, transfer learning improves recognition accuracy and generalization and is therefore a feasible solution for ASR systems of underrepresented languages. (Nanayakkara, 2022)

Transfer learning has also been investigated in the context of end-to-end ASR to cover the low resource scenario in Persian language where our experiment presents an enhanced performance at a considerably shorter training time as compared to models trained without transfer learning. In one of the study which is (Kermanshahi, Akbari and Nasersharif, 2021), the authors adopted the said model by first pre-training it with 960 hours of the English LibriSpeech before fine-tuning with only 3.5 hours of Persian speech data from the FarsDat corpus as indicated in figure 2.4 below.

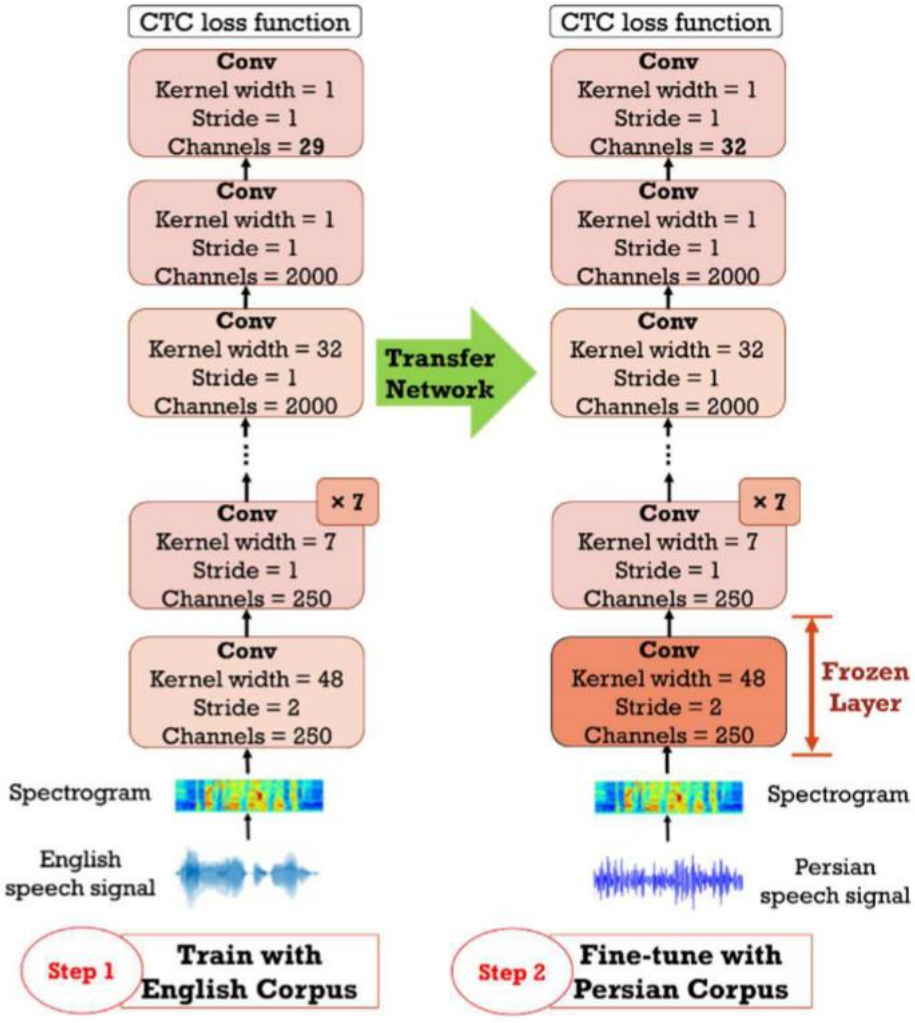

Figure 2.4: The illustration of transfer learning for the ASR model with English

As part of the regularization, some of the layers of the neural network were made untrainable in order to control the adaptation to the new language for efficient learning. This was evident from the outcome of which the TL was able to outperform models trained with the scratch (as

seen in Figure 2.5), making it even more effective to fine tune pre-trained ASR models for low resource languages but with more efficiency in training.

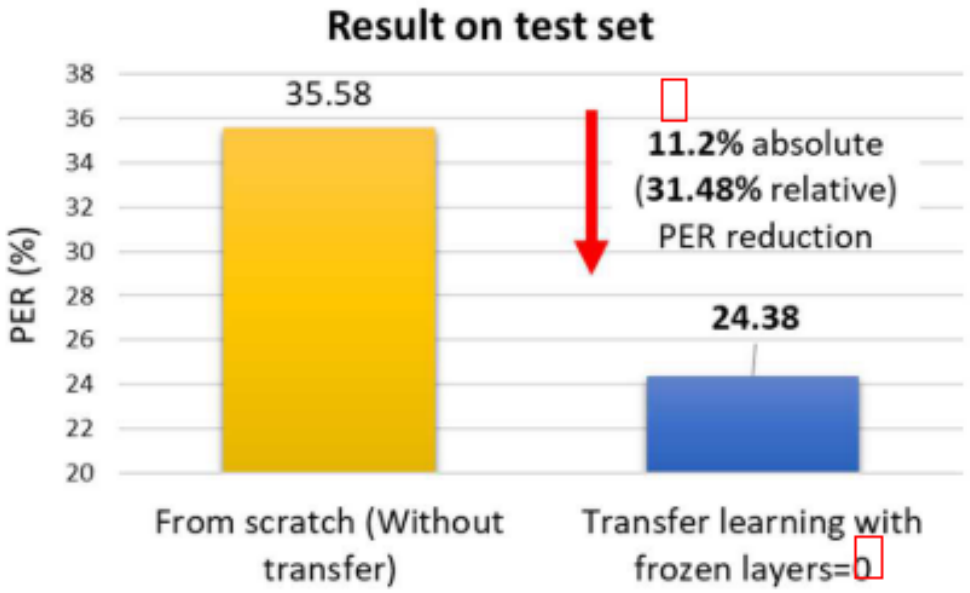

Figure 2.5: Comparing the results of transfer learning and learning from scratch

Recent research by Polat et al., (2024), also has also discussed the use of transfer learning in ASR for the low resource language by utilizing the Whisper architecture for Turkish speech recognition. Turkish is one of the underrepresented languages in the field of speech recognition and, like many other languages, it introduces certain difficulties in its emancipation from the ASR errors. The study showed that fine-tuning of pre-trained Whisper models can be effectively used for ASR of Turkish, which in return facilitates a reliable end-to-end ASR system. The first assessments revealed a word error rate between 4.3% and 14.2%, which are the outcomes of the work done with such language. The Low-Rank Adaptation (LoRA) technique was also used to apply additional improvements in fine-tuning since it decreases the number of trainable parameters but increases model performance. This has boosted the ASR accuracy with a WER decrease of up to 52.38%. The Figure 2.6 shows the ASR transfer learning for German (CV-ge), Spanish (CV-sp) and Russian (CV-ru) parts of Common Voice dataset. Training from scratch vs fine-tuning, greedy WER (%). The results suggest that the approach of transfer learning and the use of adaptation models can help to enhance ASR in low resource languages and make viable for daily use.

To sum up, there is a significant body of literature on cross-language transfer learning, continuous learning, and domain adaptation of end-to-end ASR confirming the effectiveness of the transfer learning method in addressing various linguistic and domain issues (Huang *et al.*, 2020). A study will also show that well-trained base model can be effectively fine-tuned for

different accents of English, different languages including German, Spanish, and Russian and for application-specific domains separately. The experiments used a QuartzNet model trained with the CTC loss (Figure 2.7); they proved that transfer learning helps the ASR models learn multiple languages and domains without much fine-tuning. This method greatly improves the flexibility of the ASR systems to work with the changes in speech and also improve specific models for niche applications.

Language	Model	Train from scratch	Fine-tune	WER,%
German	5x3	CV-ge	-	30.42
		WSJ	CV-ge	28.61
	15x5	CV-ge	-	23.35
		5D	CV-ge	**18.65**
Spanish	5x3	CV-sp	-	23.29
		WSJ	CV-sp	22.93
	15x5	CV-sp	-	19.82
		5D	CV-sp	**14.96**
Russian	5x3	CV-ru	-	42.18
		WSJ	CV-ru	40.24
	15x5	CV-ru	-	59.50
		5D	CV-ru	**32.20**

Figure 2.6: ASR transfer learning of Common Voice dataset

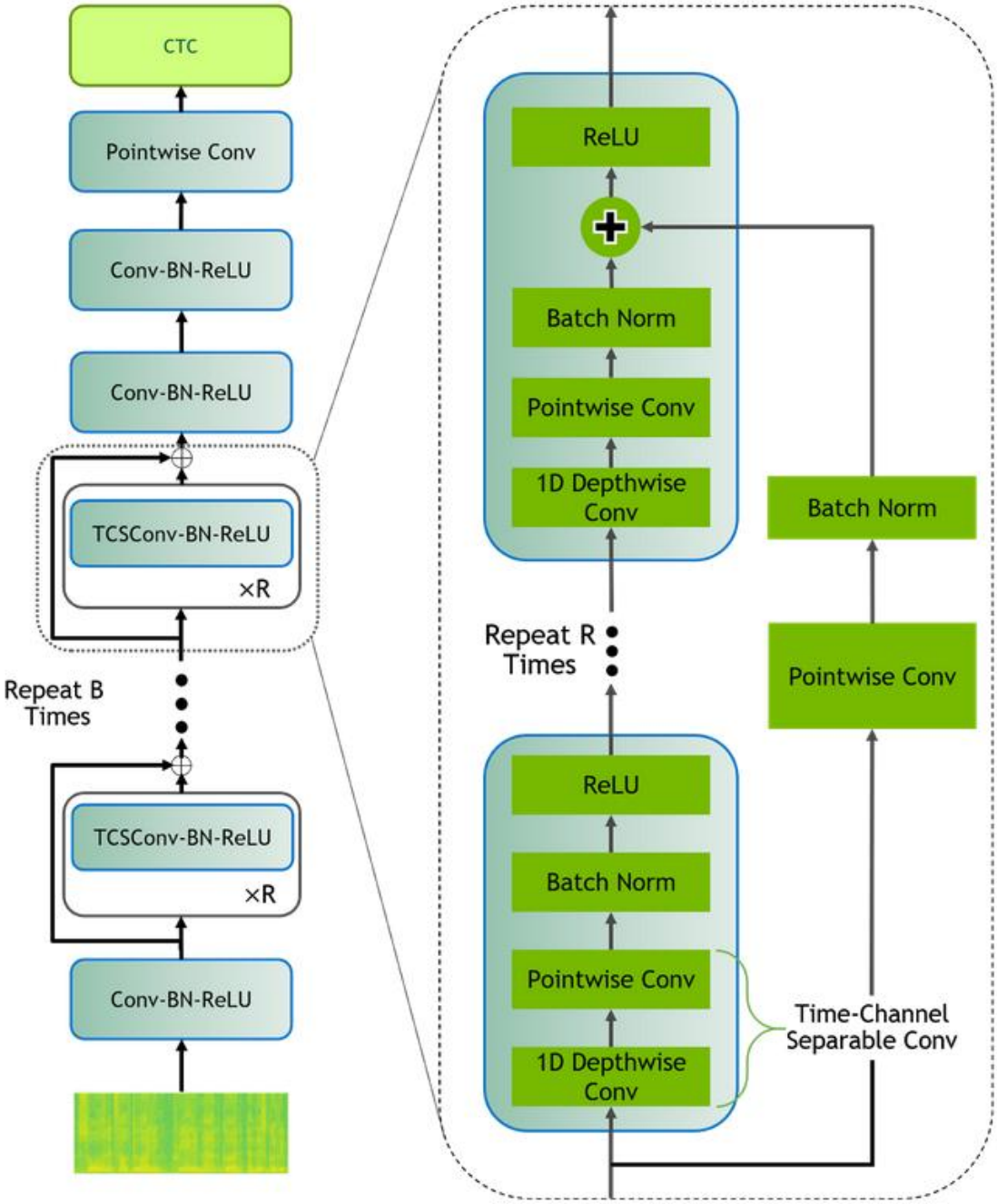

Figure 2.7: QuartzNet BxR architecture

Liu, Yang and Qu, (2024) performed a quantitative and qualitative assessment of applying transfer learning for training Whisper models on seven low-resource ASR languages. The work has explained five strategies of fine tuning and this empirical was an attempt to show that through fine tuning gains could be realized using restricted level of supervised information in enhancing the performance of ASR systems. The fine-tuning also simply means adjusting the parameters of the pre-trained model to match the characteristics of the target language when only a relatively small sample size is available, which is different from the sort of dataset used for pretraining. In this work, Large-v2 variant of Whisper was used as the baseline which will be referred to as pre-trained model (PT). The performance measures obtained from WER

showed preliminary ASR across several languages with Whisper (Figure 2.8). The number after each language is the duration of training data in Whisper pretraining, and h means hour. These 7 languages have less than 20 h of data, which satisfies the low-resource feature

Model	Af(4.1 h)	Be(2.4 h)	Is(16 h)	Kk(12 h)	Mr(0.6 h)	Ne(0.6 h)	Sw(5.4 h)	Average
Tiny	91.2	94.0	113.3	165.2	100.3	101.8	100.9	109.53
Base	81.5	91.3	95.5	109.2	100.3	102.4	92.5	96.10
Small	61.1	75.1	72.6	70.3	60.2	69.5	73.7	68.93
Medium	44.9	60.4	49.9	48.8	63.2	54.4	52.8	53.49
Large	42.6	56.6	43.0	43.8	43.7	52.2	47.9	47.11
Large-v2	36.7	45.4	38.2	37.7	38.3	47.1	39.3	40.39
Large-v3	32.4	42.5	30.4	32.4	34.1	40.2	34.1	35.18

Figure 2.8: WERs (%) of 7 languages tested by OpenAI

As shown in Figure 2.9, there has been an improvement in test results from before fine-tuning (PT) and after fine-tuning (FT). The outcomes indicate that the end-to-end multilingual and multi-task pretraining not only naturally equips Whisper with a diverse base of ASR but also fine-tunes the parameters into the précised high-dimensional space, which accelerates the adaptation to the target task.

Model	Af	Be	Is	Kk	Mr	Ne	Sw	Average
PT	36.7	45.4	38.2	37.7	38.3	47.1	39.3	40.39
FT	19.33	12.98	25.56	13.53	15.17	19.57	15.62	17.39

Figure 2.9: WERs (%) of 7 languages tested before (PT) and after (FT) fine-tuning

Further, recommendations were made on the features of ASR for new languages to be improved by integrating them with the use of LoRA and language similarities in enhancing the accuracy of the machine while developing new languages (Song *et al.*, 2024). It is essential to note that the work involved the Multilingual LibriSpeech (MLS) and the Few-shot Learning Evaluation of Universal Representations of Speech (FLEURS) datasets; the languages utilized are Polish, Portuguese, and Italian from the MLS, and Chinese, Danish, Greek, Welsh, and Japanese from FLEURS. For the multilingual ASR test, four languages were used in training while the other four were used as new languages in the language expansion scenario as shown (Figure 2.10).

Experiment	Language	Train	Test
Multilingual ASR	Polish(PL)	103.0	2.0
	Portuguese(PT)	160.0	5.0
	Italian(IT)	247.0	5.0
	Chinese(ZH)	9.7	3.1
Language expansion	Danish(DA)	7.5	2.9
	Greek(EL)	10.0	1.9
	Welsh(CY)	12.2	4.3
	Japanese(JA)	7.4	2.4

Figure 2.10: Statistics of training and testing data (in hours)

This means for the ability of the LoRA-Whisper model to improve performance on the new languages without causing a loss of performance on the previously learned languages, the study was successful. This way the model appears to be flexible for new languages and at the same time retaining the high level of accuracy for the base languages that it is originally created in. Thus, the utilization of warm start LoRA and MoE LoRA also resulted in improvement with relative improvement of 23% from full fine-tuning and 5% from the standard LoRA fine-tuning. In the current study, the above findings also amplify the prospects of using the LoRA-based techniques in broadening the ASR systems adaptability for various linguistic environments (Figure 2.11). E1 denotes the results of original Whisper model and E2 denotes the results on base languages before language expansion. Full means multilingual full fine-tune with new language data and full+ means using both new language and base language data. Seed model serves as an initialization for continual training

ID	Model	Finetune	Seed model	#Train param	New languages					Base languages				
					DA	EL	CY	JA	Avg	PL	PT	IT	ZH	Avg
E1	Whisper-small	No	E1	-	33.97	31.81	58.62	12.04	34.11	10.90	13.82	20.43	9.25	13.60
E2		No	E2	-	-	-	-	-	-	8.30	13.34	10.69	14.37	11.68
E5		Full	E2	240M	38.88	26.08	32.90	15.46	28.33	17.25	25.56	17.41	68.93	32.29
E6		Full+	E2	240M	41.35	28.89	33.65	16.61	30.13	9.77	15.41	12.16	16.07	13.35
E7	LoRA-Whisper	LoRA	E4	13M*4	28.28	22.84	30.63	10.11	22.97	**7.94**	**10.81**	**10.49**	**8.82**	**9.52**
E8		Warm start	E4	13M*4	**27.45**	21.77	**28.05**	**10.03**	21.83	**7.94**	**10.81**	**10.49**	**8.82**	**9.52**
E9		LoRA MoE	E4	13M*4	27.56	**21.56**	28.07	**10.03**	**21.81**	**7.94**	**10.81**	**10.49**	**8.82**	**9.52**

Figure 2.11: Experimental results of language expansion

2.4. Summary

Specifically, this literature review discusses ASR's growth and progress in the low-resource language adaptation by applying transfer learning. Early generation of ASR used the template-based recognition and normally covered a small set of words which normally demanded a large

amount of speaker training. More particularly, from 2010 to 2020, large dataset, newly supported GPUs, and better neural network models all together boosted deep learning technology to enhance ASR accuracy tremendously. The contemporary approaches of ASR are the acoustic-phonetic models, pattern recognition and machine learning, rule-based system, connectionist models, and support vector machines. These include systems that work through training and recognition and a model learns from data and applies it to new inputs.

Transfer learning has been seen as a potential solution in low resource languages which helps in quickly adapting efficiently in cases of low resources with sufficient target language that is used as a starting point for training. Given the past work on Whisper-based ASR systems, transfer learning has been depicted to be efficient. There are studies on using the Turkish ASR with Low-Rank Adaptation technique (LoRA) that demonstrated lower WER. Therefore, by conducting the multilingual ASR experiments using the MLS and FLEURS datasets, it was confirmed that the LoRA-Whisper models have the capability to extend to new languages without surrendering the primary performance. To sum up, the methods of deep learning and transfer learning have transformed as much as improved the ASR, extending it even to minority languages that had been unseen before.

CHAPTER 3 - METHODOLOGY

3.1. Introduction

This chapter shows how it is possible to respond to the research questions. It also explains the course of action engaged in the study and the establishment of the model and an evaluation of every design strategy. Furthermore, in the chapter presented is the concept of proposed solution and the definition and assumptions that were made throughout the study.

3.2. Design Evolution

3.2.1. Research High level Design

When it comes to the conceptual level of the ASR system based on a particular domain, there are several crucial phases and main processes that need to be taken into account in order to create and implement the necessary system. The following are the elements of this design framework exemplified in Figure 3.1.

Figure 3.1: Research High-Level Design

In the design of any SRS, the first stage is referred to as the data acquisition stage, where there is the gathering of speech data specific to the domain in question. In the case of this specific ASR system, the application of interest is for the banking sector, thus, it means that; The audio samples retrieved should be related to the banking environment by including the kind of language and terms used in banking. To maintain the integrity of the collected data, there is a pre-processing stage that is carried out on the data that has been collected. This step is necessary to make the medium and audio data more qualitative in order to improve the quality of the required data for model training. Some pre-processing operations may be reducing level of noise, equalization of volumes as well as splitting the recordings into smaller chunks that can easily be processes.

The next step comes when one has a clean and well-structured dataset and that now the best transfer learning model should be chosen. It lets one use old model that was trained with lots of

data and it is used to train a new model with little data. The training process is involved in the process of the refinement of the pre-trained model on the collected Sinhala banking dataset. In this step, there is an optimization of the model's hyperparameters so that the model can perform as expected. The training stage is in turns which means that the process of training needs to be monitored and modified in order to improve its ability to learn from the data provided.

The last activity in high-level design is to assess and compare the trained model. This includes fitting the model on a sample data-set different from the training data-set in order to check on the over-fitting problem. New variables like word error rate (WER), accuracy percentage and response time are recorded so as to check the performance of the ASR system.

3.2.2. Selecting a Proper Sinhala Dataset

Typically, the performance of the ASR system must be based on the availability of the speech data and its corresponding text in the required domain and the desired language.
It is therefore important to consider the above when selecting a dataset while developing a domain-dependent ASR system especially when working with less data. In this particular study, the subject area of interest is banking, while the language to be adopted is Sinhala.

This is under the understanding that globally, there was no primary dataset found concerning the banking domain in the Sinhala language. This lacuna in the existing literature called for the development of a new private dataset that would suit the need of this study.

In the creation of this dataset, the use of the recording equipment called Audacity was deemed efficient as it is a basic audio capturing software that guarantees the quality of the recorded data. Given that audio quality is highly crucial in training an efficient ASR system, further steps were taken to improve the recordings' quality. Specifically for the purpose of reducing background noise, Adobe Audition and NVIDIA Broadcast were used. This was important particularly regarding the audio sera that would enable the recorded speech data portray the spoken language in a banking environment to enable efficient creation of ASR system.

Dataset - https://huggingface.co/datasets/IshanSuga/sinhala-bank-speech

Stats

The dataset consists of a total of 100 audio recordings, all of which were recorded by a single speaker. To ensure consistency and linguistic diversity, the speaker was provided with a predefined script comprising 100 sentences. This methodical data collection process aimed to create a clean and focused dataset suitable for fine-tuning the Whisper model on the Sinhala language.

- Number of Recordings: 100
- Total Duration: 700.283 seconds
- Maximum Length: 15 seconds
- Minimum Length: 2 seconds
- Sample Rate: 16kHz
- Language(s) (NLP): Sinhala
- License:MIT


```
{
  "1": {
    "newfn": "bank_sin_01_00001.wav",
    "sinhala": "සුභ උදෑසනක්! අද මම කෙෂේද ඔබට සහය වන්නේ?",
    "duration": 5.356
  },
  "2": {
    "newfn": "bank_sin_01_00002.wav",
    "sinhala": "මට නව බැංකු ගිණුමක් විවෘත කිරීමට අවශ්‍යයි, කරුණාකර මට උපදෙස් දෙන්න.",
    "duration": 7.097
  },
  "3": {
    "newfn": "bank_sin_01_00003.wav",
    "sinhala": "බැංකු ගිණුමක් විවෘත කිරීම සඳහා අවශ්‍ය ලියකියවිලි මොනවාද?",
    "duration": 5.907
  },
  "4": {
    "newfn": "bank_sin_01_00004.wav",
    "sinhala": "මට මගේ බැංකු ගිණුමේ ශේෂය පරීක්ෂා කිරීමට අවශ්‍යයි, කරුණාකර මට උදව් කරන්න.",
    "duration": 8.447
  },
  "5": {
    "newfn": "bank_sin_01_00005.wav",
    "sinhala": "මට මගේ ගිණුමට මුදල් තැන්පත් කිරීමට අවශ්‍යයි, එයට අවශ්‍ය ක්‍රියාවලිය කුමක්ද?",
    "duration": 6.97
  },
  "6": {
    "newfn": "bank_sin_01_00006.wav",
    "sinhala": "මට මගේ ගිණුමේ අවසාන ගනුදෙනු පිළිබඳව විස්තර ලබාගැනීමට අවශ්‍යයි.",
    "duration": 5.82
  },
  "7": {
    "newfn": "bank_sin_01_00007.wav",
    "sinhala": "මගේ ගිණුම අක්‍රිය වී ඇත, කරුණාකර එය නැවත සක්‍රිය කිරීමට උපදෙස් දෙන්න.",
    "duration": 6.943
  },
  "8": {
    "newfn": "bank_sin_01_00008.wav",
    "sinhala": "මගේ බැංකු ගිණුමේ නම වෙනස් කිරීමට අවශ්‍යයි, කරුණාකර ක්‍රියාවලිය පැහැදිලි කරන්න.",
    "duration": 6.571
  },
```


Figure 3.2: Audio-text mapping in dataset

To sum up, the lack of collections in the Sinhala language for the banking domain meant the generation of the new dataset recommended during ideal acoustic environment for the purpose of achieving high sound quality. This dataset is going to be used in training and testing of the ASR system, which would help in enhancing the ability of the system to understand the Sinhala spoken words and phrases employed in the context of banking and financial operations.

3.2.3. Selecting Transformers

The advancement of deep learning models in sequence-based problems has been made with recurrent neural networks (RNNs) and some of the variations of the same like LSTM (Long Short Term Memory) and GRU (Gated Recurrent Units) that have proved to be quite effective in machine translation, in language modeling, in and in sequential data analysis. However, these architectures fail when it comes to processing long input sequence since all the information is packed into a low dimensional vector (Polat *et al.*, 2024). That is true because this constraint often leads to loss of extremely important contextual information thus compromising performance. To address it, the suggestiveness of the model was enhanced with the introduction of the attention mechanism, in which the model can focus on various segments of the input during its operation at each level of the output. Although this improvement helped to overcome some issues, there are still some limitations Like long-term dependency, high computational complexity and can not be paralleled (Bahdanau, Cho and Bengio, 2014). Realizing these constraints, Vaswani et al., (2017) came up with the Transformer architecture which disrupted the sequence modeling by doing away with recurrence and replaced it with self attention. This one enable parallel processing and improves scalability and speed with the important information retention over many iterations. Therefore, transformers have turned into the core of the large-scale language models which include bidirectional encoder representations from transformers (BERT) as well as generative pre-trained transformer (GPT), and are widely used in modern natural language processing and speech recognition systems.

The original transformer model was designed to translate written text from one language into another which consists of an encoder–decoder structure (Figure 3.3).

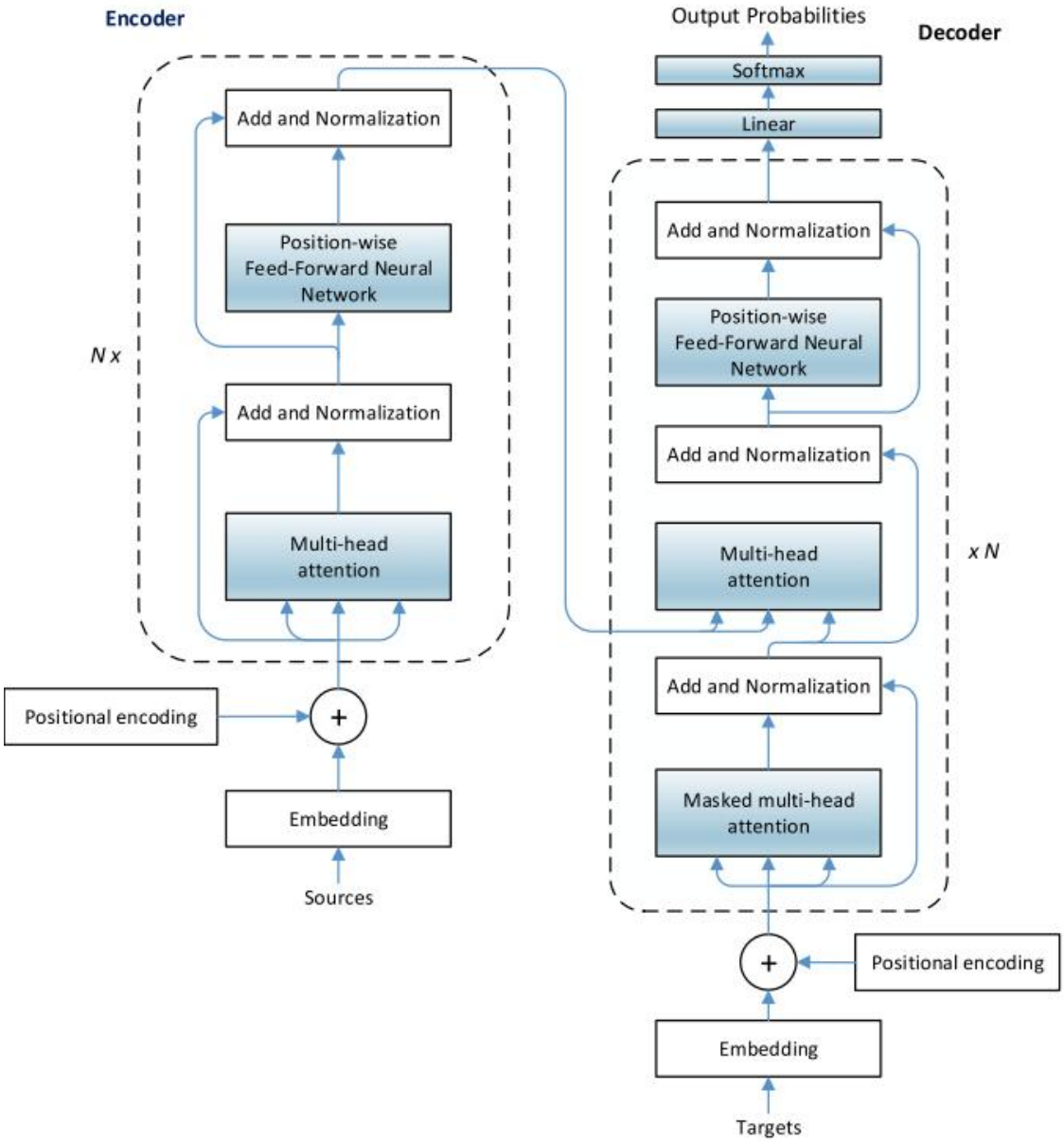

Figure 3.3: Basic transformer architecture

3.2.4. Whisper Model Architecture

In this study, Whisper is the latest ASR model by OpenAI (Radford *et al.*, 2022), that relies on large-scale supervised pre-training for boosting its performance in speech recognition. Hinging on encoder-decoder transformer (Figure 3.4), Whisper is capable of performing several speech tasks that include speech recognition in several languages, speech-to-speech translation, language identification, and Voicing Activity Detection (Polat *et al*., 2024; Song *et al*., 2024).

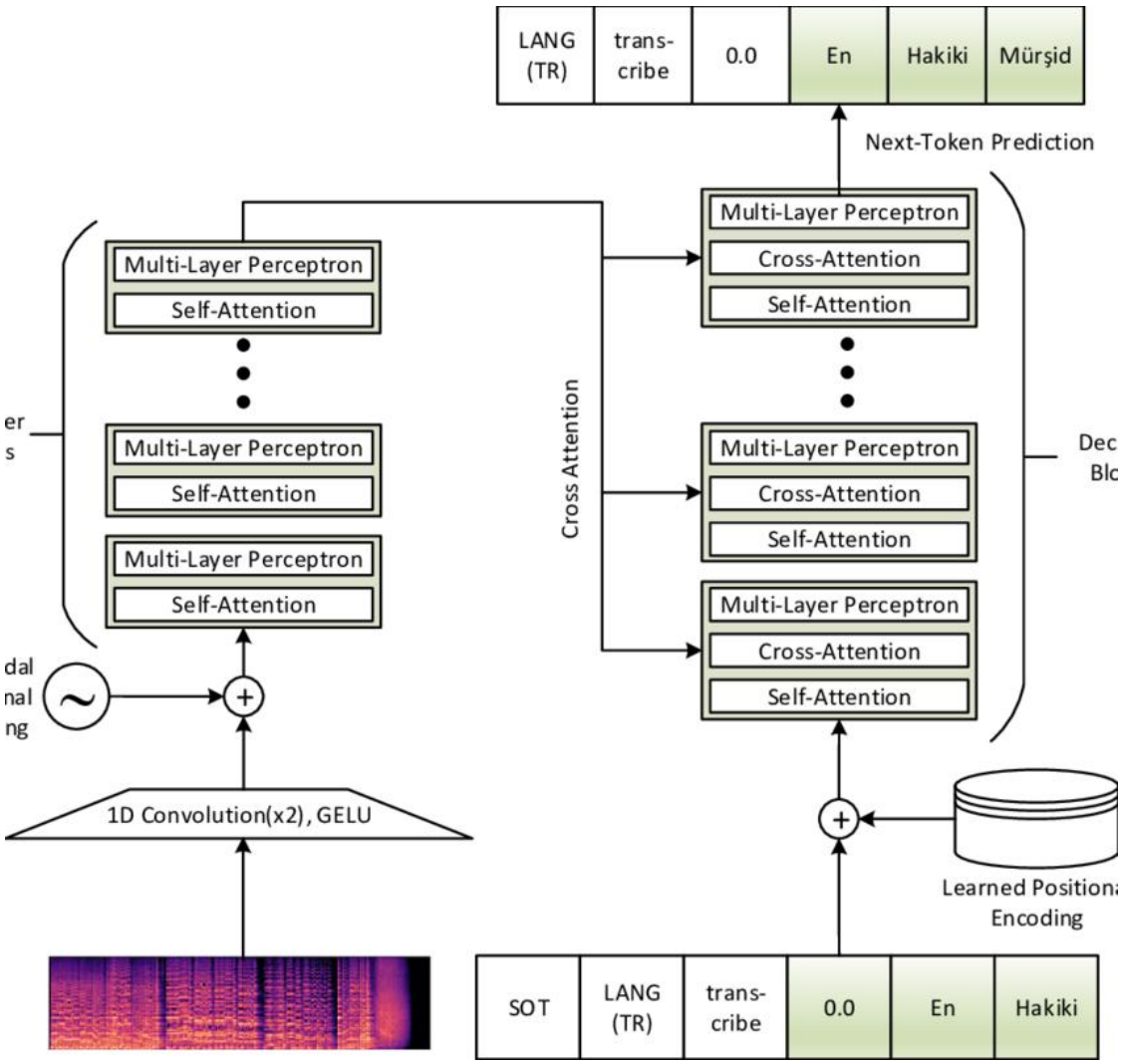

Figure 3.4: Whisper model architecture

A fully end-to-end modeling framework replaces the classic ASR system architecture of combining deep learning techniques with HMM and GMM components that needed trained independently. By merging acoustic and language model components into one system the traditional training boundaries are eliminated thus improving operational effectiveness (Polat *et al.*, 2024). Through parallel processing along with the transformer-based design Whisper achieves fast processing rates and powerful computational efficiency in addition to handling long-range speech dependencies by using the attention mechanism.

The distinctive attribute of Whisper emerges from its support of approximately 100 languages (Polat *et al.*, 2024), while utilizing a vast multilingual dataset (approximately 680,000 h with 117,000 h of multilingual content (Polat *et al.*, 2024)), to obtain solid results across different speech recognition benchmarks. The system recommends more precise results than both available open-source models and existing commercial ASR services while surpassing NVIDIA's STT Conformer-CTC Large model from NeMo Toolkit especially when processing

audio datasets dominated by infrequent complex words to obtain solid results across different speech recognition benchmarks. The system recommends more precise results than both available open-source models and existing commercial ASR services while surpassing NVIDIA's STT Conformer-CTC Large model from NeMo Toolkit especially when processing audio datasets dominated by infrequent complex words (Radford *et al.*, 2022).

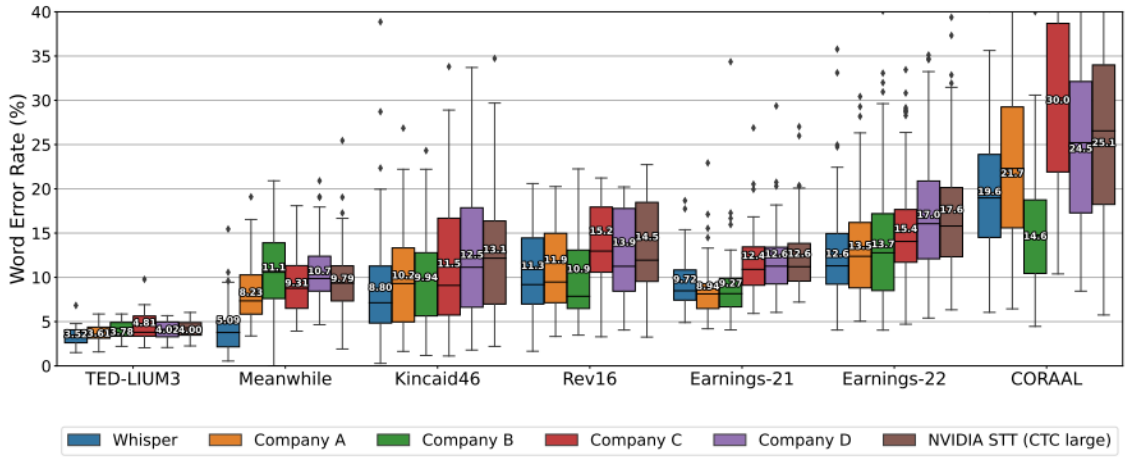

Figure 3.5:Ddistribution of WER from Whisper and the 4 commercial ASR services

Whisper is available in multiple configurations, each varying in complexity and computational requirements based on the number of layers, the width of feature Whisper comes in versatile versions which differ in complexity by featuring distinct numbers of layers and width of feature representations along with attention head capabilities. The product line features Tiny, Base, Small, Medium and Large editions which enable users to pick the most suitable model for their resource limits (Figure 3.6) (Radford *et al.*, 2022). Whisper offers two categories of models which comprise Tiny and Base for cellular devices with efficiency needs and Whisper Large designed for demanding speech recognition workloads across various languages and difficult acoustic situations. Whisper achieves scalability which allows users to deploy the system for activities ranging from fast transcription to detailed Automatic Speech Recognition operations with robust processing resources.

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Figure 3.6: Architecture details of the Whisper model family

3.3. Evaluation metrics for ASR

The assessment of ASR system performance requires examination of transcription matches between machine-generated output and reference text alongside detection of all variations between them. ASR system errors break down into three types including substitutions where incorrect words take the place of correct words such as transcribing “learn” when the author intended “earn” and insertions of extra words in transcriptions and deletions which occur when reference words vanished from transcriptions. Every evaluation metric for ASR uses these fundamental types of errors which remain constant. The measurement scale of word substitution errors differs across evaluation metrics which track them either through individual words or single characters. The targeted application determines the evaluation level selection where overall transcription metrics employ word analysis while character-based metrics excel for evaluating speech processing of complex languages and noisy inputs.

3.3.1. The Word Error Rate (WER)

The Word Error Rate serves as a common evaluation metric which measures ASR system accuracy. Absent Levenshtein distance as its foundation this metric operates under the name edit distance to determine how many substitutions (S), insertions (I) and deletions (D) require to modify the ASR output into the reference text. WER enables engineers to measure the system performance by showing the mathematical similarity between automatic transcriptions and human-generated text (du Simplon *et al.*, 2005). It is mathematically defined as:

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

N stands for the total word count in reference transcription while S represents incorrect substitution errors and D represents omitted words and I stand for extra added words from the Automatic Speech Recognition system. The WER metric determines the transcription precision and functions as an essential assessment standard to help evaluate the progress of speech recognition systems through model benchmarking.

```
!pip install evaluate jiwer

from evaluate import load

def cal_wer(reference, prediction)

  wer_metric = load("wer")

  wer = wer_metric.compute(references=[reference], predictions=[prediction])

  print(wer)

  return wer
```

Figure 3.7: Compute WER

3.4. Training Phase

The training together with evaluation of the ASR model took place on Google Colab primarily because this platform enables direct access to GPU resources essential for speeding up deep learning operations. OpenAI's Whisper model 'small' version underwent fine-tuning with a Sinhala banking domain-specific dataset by making use of the 16GB T4 GPU gained from Google Colab's free tier. The decision to select the smaller model for fine-tuning purposes enabled more efficient processing because it worked with both limited computational power and minimal disk storage requirements. The system offered a perfect ratio of performance potential alongside resource availability which permitted quick model optimization and testing processes.

Model checkpoints sent to Hugging Face Hub during training benefited from its strong management features during the process. During training operations model checkpoints reside in the Hugging Face Hub environment so users ensure complete checkpoint preservation throughout the execution phase. TensorBoard logs accessible through Hugging Face Hub enable users to supervise essential training metrics such as loss and accuracy in real time during the procedure. The platform enables users to create model cards that provide necessary documentation about model functions along with intended applications and restrictions. Users can easily share their models through the Hub platform since it creates a community environment that facilitates team work with other researchers and developers. The process to integrate Colab notebooks with the Hugging Face Hub demands no more than entering a Hub authentication token when asked since this step simplifies the training workflow (Figure 3.8).

```
from huggingface_hub import notebook_login

notebook_login()
```

Figure 3.8: Hugging face authentication

3.4.1. Feature Extractor, Tokenizer, and Processor

The sequence of operations in an ASR pipeline consists of three main processes including feature extraction and model inference and tokenization. Raw audio inputs are transformed through the feature extractor which produces log-Mel spectrograms for the model before processing as a vital pre-processing phase of sound wave conversion. The model conducts sequence-to-sequence mapping after which it produces predicted tokens from audio features. The text transcription ends with the tokenizer converting produced tokens into human-readable textual content.

The Hugging Face Transformers ecosystem includes two components for feature extraction and tokenization in its Whisper model which are known as **WhisperFeatureExtractor** and **WhisperTokenizer**. The **WhisperProcessor** class contains both components as an encapsulated system to process audio beforehand and text afterward. Proper initialization of the processor requires users to specify correct "language" and "task" parameter settings when operating the model for multilingual speech recognition or translation tasks. The source audio language should be defined as the "language" parameter and the "task" parameter needs to be set either as "transcribe" or "translate" according to the requirement (Figure 3.9). The specific configurations control how the tokenizer functions which results in better target label encoding and improves prediction accuracy from the model.

```
from transformers import WhisperProcessor

processor = WhisperProcessor.from_pretrained(
    base_model, language="sinhalese", task="transcribe"
)
```

Figure 3.9: Calling WhisperProcessor

CHAPTER 4 - IMPLEMENTATION

4.1. Introduction

The ASR system development procedures for the Sinhala banking domain are thoroughly presented in this chapter. The method explains the entire project progression starting with data preparation and preprocessing before reaching model training after evaluation and before deployment begins. This chapter discusses the fundamental tools including the Whisper model configuration along with Hugging Face integration and the deployment of Google Colab for the complete process. This chapter presents an organized approach that both documents the process well and enables effortless reproduction which provides important knowledge about creating domain-specific ASR models using restricted data.

4.2. Dataset alignment and preparing techniques

The ASR model refinement process employed the datasets library for managing and preparing the Sinhala banking dataset. The data preparation process included importing raw information which underwent structure transformation before applying it for training and evaluating the model.

A Python implementation of the json module allowed the dataset to be loaded from its JSON file format. A JSON file contained dual information regarding audio files and their match Sinhala transcription. A list of records is built by the program while it traverses through the database containing audio file paths alongside Sinhala sentence transcripts and respective audio duration measurements. A specified folder maintained the audio files within which the processing code generated dynamic paths to properly synchronize audio data alongside their corresponding transcripts.

Through the *Dataset.from_list()* method the structured data transformed into a Hugging Face Dataset. The Audio feature type accepted the audio column of the dataset after casting it while setting the sampling rate to 16000 Hz since this is the standard used for ASR purposes. Using this step the automatic loading combined with processing of audio files in the dataset became possible.

The data split operation with *train_test_split()* method generated 80% training data and 20% testing data. Effective evaluation and training of the model become possible through this

stratification process. The split data was organized into a DatasetDict structure to provide convenient access to the datasets during finetuning operations. The systematic preprocessing methods created a firm basis for training an advanced ASR model.

```
!pip install datasets

import json
from datasets import Dataset, Audio, DatasetDict

# Step 1: Load your dataset
with open("/content/drive/MyDrive/Colab Notebooks/"+
          "Sinhala_TTS_data/bank_sinhala_dataset.json", "r", encoding="utf-8") as f:
    raw_data = json.load(f)

# Step 2: Transform the dataset into a list of records
data = []

audio_dir = "/content/drive/MyDrive/Colab Notebooks/Sinhala_TTS_data/bank_audio/"
for _, entry in raw_data.items():
    data.append({
        "audio": f"{audio_dir}{entry['newfn']}",  # Path to audio file
        "sentence": entry["sinhala"],  # Sinhala transcription
        "duration": entry["duration"]  # Duration (optional)
    })

# Step 3: Create a Hugging Face Dataset
dataset = Dataset.from_list(data)

# Step 4: Cast the 'audio' column to the Audio feature
dataset = dataset.cast_column("audio", Audio(sampling_rate=16000))

# Step 5: Split the dataset into train (80%) and test (20%)
split_dataset = dataset.train_test_split(test_size=0.2)

# Step 6: Organize splits into a DatasetDict
dataset_dict = DatasetDict({
    "train": split_dataset["train"],
    "test": split_dataset["test"]
})
print(dataset_dict)
```

```
DatasetDict({
    train: Dataset({
        features: ['audio', 'sentence', 'duration'],
        num_rows: 80
    })
    test: Dataset({
        features: ['audio', 'sentence', 'duration'],
        num_rows: 20
    })
})
```

Figure 4.1: Preparing and organizing the Sinhala banking dataset

4.3. Feature Extractor, Tokenizer and Processor

Through *TO_LANGUAGE_CODE* mapping the Whisper model enables the processing of multiple tongue languages with Sinhalese featured as one of its pre-trained language choices (Figure 4.2). The model maintains excellent capability for processing Sinhalese audio data thanks to its language configuration. The implementation of the WhisperProcessor required the installation of a pre-trained checkpoint while specifying "sinhalese" as the language parameter and "transcribe" as the task parameter (Figure 3.9). The combination of processor settings allows the model to correctly execute audio pre-processing and text post-processing tasks according to Sinhala banking domain requirements.

```
from transformers.models.whisper.tokenization_whisper import TO_LANGUAGE_CODE

TO_LANGUAGE_CODE
```

```
{'english': 'en',     'khmer': 'km',           'javanese': 'jw',
 'chinese': 'zh',     'shona': 'sn',           'sundanese': 'su',
 'german': 'de',      'yoruba': 'yo',          'cantonese': 'yue',
 'spanish': 'es',     'somali': 'so',          'burmese': 'my',
 'russian': 'ru',     'afrikaans': 'af',       'valencian': 'ca',
 'korean': 'ko',      'occitan': 'oc',         'flemish': 'nl',
 'french': 'fr',      'georgian': 'ka',        'haitian': 'ht',
 'japanese': 'ja',    'belarusian': 'be',      'letzeburgesch': 'lb',
 'portuguese': 'pt',  'tajik': 'tg',           'pushto': 'ps',
 'turkish': 'tr',     'sindhi': 'sd',          'panjabi': 'pa',
 'polish': 'pl',      'gujarati': 'gu',        'moldavian': 'ro',
 'catalan': 'ca',     'amharic': 'am',         'moldovan': 'ro',
 'dutch': 'nl',       'yiddish': 'yi',         'sinhalese': 'si',
 'arabic': 'ar',      'lao': 'lo',             'castilian': 'es',
 'swedish': 'sv',     'uzbek': 'uz',           'mandarin': 'zh'}
 'italian': 'it',     'faroese': 'fo',
 'indonesian': 'id',  'haitian creole': 'ht',
 'hindi': 'hi',       'pashto': 'ps',
 'finnish': 'fi',     'turkmen': 'tk',
 'vietnamese': 'vi',  'nynorsk': 'nn',
 'hebrew': 'he',      'maltese': 'mt',
 'ukrainian': 'uk',   'sanskrit': 'sa',
 'greek': 'el',       'luxembourgish': 'lb',
 'malay': 'ms',       'myanmar': 'my',
 'czech': 'cs',       'tibetan': 'bo',
 'romanian': 'ro',    'tagalog': 'tl',
 'danish': 'da',      'malagasy': 'mg',
 'hungarian': 'hu',   'assamese': 'as',
 'tamil': 'ta',       'tatar': 'tt',
 'norwegian': 'no',   'hawaiian': 'haw',
 'thai': 'th',        'lingala': 'ln',
 'urdu': 'ur',        'hausa': 'ha',
 'croatian': 'hr',    'bashkir': 'ba',
```

Figure 4.2: Whisper pre-trained languages

4.4. Pre-Process the Data

The processing of audio data for fine-tuning the Whisper model must ensure alignment with the model requirements for its specified input format. The audio column of the dataset contains specific information about samples which includes sampling rate data. The model requires 16kHz sampling rate (Figure 4.3) so no further down sampling is necessary to use the data directly.

```
common_voice["train"].features
```

```
{'audio': Audio(sampling_rate=16000, mono=True, decode=True, id=None),
 'sentence': Value(dtype='string', id=None)}
```

Figure 4.3: Dataset features

The *prepare_dataset()* function serves as the data preprocessing mechanism to prepare audio data alongside accompanying transcribed text for the model (Figure 4.4). The function accepts one dataset entry and extracts the audio material from it. After sampling the audio array the Whisper model component known as the processor converts both the audio array data and sampling rate together with its transcription text output. The processor performs two operations during data preparation by turning audio signals into log-Mel spectrograms and converting text into tokens for model processing. The function computes audio sample length in seconds through dividing the audio array length by sampling rate. Audio samples can be filtered or batched through the information available from this step during training. The function delivers the processed example which is prepared for successful model training purposes.

```
def prepare_dataset(sample):
    audio = sample["audio"]

    sample = processor(
        audio=audio["array"],
        sampling_rate=audio["sampling_rate"],
        text=sample["sentence"],
    )

    # compute input length of audio sample in seconds
    sample["input_length"] = len(audio["array"]) / audio["sampling_rate"]

    return sample
```

Figure 4.4: Function- prepare_dataset

Once the *prepare_dataset()* function is defined, it is applied to the entire training dataset using

the *.map* method from the Hugging Face Datasets library (Figure 4.5). During this mapping process, unnecessary columns from the original dataset are removed, leaving only the processed features generated by the *prepare_dataset()* function.

```
common_voice = common_voice.map(
    prepare_dataset, remove_columns=common_voice.column_names["train"], num_proc=1
)
```

Figure 4.5: Apply prepare_dataset function

Before the processing begins the system eliminates any audio records exceeding 30-second durations. The Whisper feature extractor cuts down extended audio files which can make training operations less stable. A filter function has been defined to check which samples have duration shorter than 30 seconds while returning True values for these shorter samples and False for the longer ones (Figure 4.6). The *.filter* method operates across all training dataset samples through the defined function to keep only samples with appropriate sizes (Figure 4.7).

```
max_input_length = 30.0


def is_audio_in_length_range(length):
    return length < max_input_length
```

Figure 4.6: Function to identify longer audio clips

```
common_voice["train"] = common_voice["train"].filter(
    is_audio_in_length_range,
    input_columns=["input_length"],
)
```

```
common_voice["train"]
```

```
Dataset({
    features: ['input_features', 'labels', 'input_length'],
    num_rows: 80
})
```

Figure 4.7: Filter-out the longer audio clips

The dataset contained no samples beyond the set 30-second limit therefore the dataset remained unchanged. The preprocessed dataset stands ready to serve as training material for the ASR model through its optimized condition from the preprocessing phase.

4.5. Model Training and Evaluation

Hugging Face Trainer serves as a production tool for efficient model development and evaluation of the Sinhala banking domain ASR system. The methodology cuts down the requirement for human interaction while maintaining efficient and standardized operations.

The training setup requires defining a data collator as its first step. The data collator transforms processed data into proper PyTorch tensor formats needed by the model for use. The correct preparation of data batches occurs for both training and evaluation phases thanks to this step. Evaluation metrics need to be selected first for precise performance assessment of the model. WER serves as the accuracy metric because it reliably scores transcription accuracy through reference transcription comparison with predicted text. Users can perform WER calculations during evaluation through the custom *compute_metrics()* function which the system provides. The model begins from its pre-trained checkpoint before loading it into operation. A pre-trained checkpoint optimizes resource usage because it applies previous model learning to adapt to the Sinhala banking data during fine-tuning. The checkpoint configuration needs appropriate setup to establish correct parameter values for the new training assignment. The Trainer utilizes training arguments that consist of elements which control characteristics such as batch size along with learning rate and training epochs. The specified parameters serve as essential components for developing optimized training plans together with optimized model learning processes.

The completed model receives its evaluation through testing on the test dataset. The verification of model learning to transcribe speech within the Sinhala banking domain represents this essential last evaluation phase which reveals the model's accuracy level in practical applications.

4.5.1. Define a Data Collator

The data collator for a sequence-to-sequence speech recognition model is designed to handle both audio features and tokenized labels independently, given their different length requirements and padding needs. The provided code (Figure 4.8) implements a custom data collator class, *`DataCollatorSpeechSeq2SeqWithPadding`*, which efficiently prepares batches of data for training and evaluation.

```python
import torch

from dataclasses import dataclass
from typing import Any, Dict, List, Union


@dataclass
class DataCollatorSpeechSeq2SeqWithPadding:
    processor: Any

    def __call__(
        self, features: List[Dict[str, Union[List[int], torch.Tensor]]]
    ) -> Dict[str, torch.Tensor]:
        ''' split inputs and labels since they have to be of
         different lengths and need different padding methods'''
        # first treat the audio inputs by simply returning torch tensors
        input_features = [
            {"input_features": feature["input_features"][0]} for feature in features
        ]
        batch = self.processor.feature_extractor.pad(input_features, return_tensors="pt")

        # get the tokenized label sequences
        label_features = [{"input_ids": feature["labels"]} for feature in features]
        # pad the labels to max length
        labels_batch = self.processor.tokenizer.pad(label_features, return_tensors="pt")

        # replace padding with -100 to ignore loss correctly
        labels = labels_batch["input_ids"].masked_fill(
            labels_batch.attention_mask.ne(1), -100
        )

        # if bos token is appended in previous tokenization step,
        # cut bos token here as it's append later anyways
        if (labels[:, 0] == self.processor.tokenizer.bos_token_id).all().cpu().item():
            labels = labels[:, 1:]

        batch["labels"] = labels

        return batch
```

Figure 4.8: Custom data collator class

The class uses the processor that integrates both the feature extractor and tokenizer. During the `__call__` method, the input features and labels are separated for distinct processing. The audio inputs are first converted into PyTorch tensors by padding the input features using the feature extractor. This ensures consistent input shapes across the batch, which is essential for efficient batch processing in the model. For the labels, the collator retrieves the tokenized label sequences and applies padding using the tokenizer. The padding process aligns the length of label sequences, which allows the model to process variable-length transcriptions uniformly. To properly compute the loss during training, the padding tokens are replaced with -100. This specific value is used by PyTorch to indicate that these positions should be ignored in the loss calculation, preventing the padding from affecting model training.

Additionally, the code handles the case where the beginning-of-sequence (BOS) token might be redundantly included during the tokenization process. If the BOS token is present at the start of all labels, it is removed since it will be automatically appended later. Finally, the processed labels are added to the batch dictionary, which is then returned for use in the training loop. This custom data collator ensures that both input features and labels are correctly formatted, padded, and ready for the model, contributing to robust and efficient training of the ASR system.

4.5.2. Evaluation Metrics

The ASR model evaluation critically depends on the WER metric. A quantitative model accuracy score can be found in WER as it evaluates predicted text against reference text. The provided function *compute_metrics()* (Figure 4.9) calculates orthographic WER along with a normalized WER metric to provide complete assessment of model transcription performance.

```
import evaluate

metric = evaluate.load("wer")
```

```
from transformers.models.whisper.english_normalizer import BasicTextNormalizer

normalizer = BasicTextNormalizer()


def compute_metrics(pred):
    pred_ids = pred.predictions
    label_ids = pred.label_ids

    # replace -100 with the pad_token_id
    label_ids[label_ids == -100] = processor.tokenizer.pad_token_id

    # we do not want to group tokens when computing the metrics
    pred_str = processor.batch_decode(pred_ids, skip_special_tokens=True)
    label_str = processor.batch_decode(label_ids, skip_special_tokens=True)

    # compute orthographic wer
    wer_ortho = 100 * metric.compute(predictions=pred_str, references=label_str)

    # compute normalised WER
    pred_str_norm = [normalizer(pred) for pred in pred_str]
    label_str_norm = [normalizer(label) for label in label_str]
    # filtering step to only evaluate the samples that correspond to non-zero references:
    pred_str_norm = [
        pred_str_norm[i] for i in range(len(pred_str_norm)) if len(label_str_norm[i]) > 0
    ]
    label_str_norm = [
        label_str_norm[i]
        for i in range(len(label_str_norm))
        if len(label_str_norm[i]) > 0
    ]

    wer = 100 * metric.compute(predictions=pred_str_norm, references=label_str_norm)

    return {"wer_ortho": wer_ortho, "wer": wer}
```

Figure 4.9: Custom function for model evaluation

The function Retrieval starts by extracting predicted token IDs along with label IDs present in the model output. The function applies the tokenizer padding token ID to replace all -100 placeholder entries in the labels because they serve as padding tokens in input data processing. The processor employs batch decoding to translate both predictions and labels into readable text after their conversion from tokenized sequences to string format by skipping special tokens. The orthographic WER calculation depends on the original transcriptions to provide a starting performance measurement. The function implements both a WER calculation along with normalized WER for a comprehensive evaluation. The normalization procedure requires applying the `*BasicTextNormalizer*` to both predicted and reference strings in order to normalize inconsistencies including punctuation and capitalization and extra whitespace. Speech recognition systems need normalization techniques because it maintains performance

accuracy despite minor orthographic differences in practical deployments.

A filtering process within the function enables selection of reference samples that contain content leading to useful evaluation results. Such precautions ensure proper calculation of metrics because empty or invalid entries would otherwise introduce bias to the evaluation process. The function combines its output with both orthographic and normalized WER scores which provides detailed understanding of an ASR model's performance level. The given metrics serve as fundamental tools for improving and validating model efficiency when assessing its suitability to serve the Sinhala banking market.

4.5.3. Load the Pre-Trained Checkpoint

The training starts with the Whisper "small" model checkpoint for loading into the system to serve as a groundwork before fine-tuning on the Sinhala banking dataset. The training requires a `use_*cache*` setting of `False` because gradient checkpointing requires memory optimization yet cannot work with caching. When generating text the model requires users to set two parameters which specify "*sinhalese*" as language and "transcribe" as task type through the `*language*` and `*task*` fields respectively. The mechanism for caching remains disabled during training while enabling it during inference to boost the speed of produced transcriptions. (Figure 4.10).

```
from transformers import WhisperForConditionalGeneration

model = WhisperForConditionalGeneration.from_pretrained(base_model)
```

```
from functools import partial

# disable cache during training since it's incompatible with gradient checkpointing
model.config.use_cache = False

# set language and task for generation and re-enable cache
model.generate = partial(
    model.generate, language="sinhalese", task="transcribe", use_cache=True
)
```

Figure 4.10: Load the whisper-small pre-trained checkpoint

4.5.4. Define the Training Configuration

During training configuration all crucial parameters should be defined to perform fine-tuning operations on the Whisper model (Figure 4.11). The training intensity is evaluated by using

training steps set to 50, 100, 200 and 250 in order to assess performance variations. The Hugging Face Trainer receives training arguments together with the model and prepared dataset and custom data collator as well as the `*compute_metrics*` function. (Figure 4.12).

```
from transformers import Seq2SeqTrainingArguments

training_args = Seq2SeqTrainingArguments(
    output_dir=f"./{hugging_face_push}",  # name on the HF Hub
    per_device_train_batch_size=16,
    gradient_accumulation_steps=2,  # increase by 2x for every 2x
    learning_rate=1e-5,
    lr_scheduler_type="constant_with_warmup",
    warmup_steps=50,
    max_steps=250,  #2 increase to 4000 on own GPU or a Colab paid plan
    gradient_checkpointing=True,
    fp16=True,
    fp16_full_eval=True,
    evaluation_strategy="steps",
    per_device_eval_batch_size=16,
    predict_with_generate=True,
    generation_max_length=225,
    save_steps=500,
    eval_steps=2, #500
    logging_steps=25,
    report_to=["tensorboard"],
    load_best_model_at_end=True,
    metric_for_best_model="wer",
    greater_is_better=False,
    push_to_hub=True,
)
```

Figure 4.11: Define training configurations

```
from transformers import Seq2SeqTrainer

trainer = Seq2SeqTrainer(
    args=training_args,
    model=model,
    train_dataset=common_voice["train"],
    eval_dataset=common_voice["test"],
    data_collator=data_collator,
    compute_metrics=compute_metrics#,
    #tokenizer=processor,
)
```

Figure 4.12: Define hugging face trainer

4.6. Summary of the chapter

The chapter presented the complete workflow for fine-tuning the Whisper model for ASR operation in Sinhala financial text. The initial procedure started with a dataset preparation phase

utilizing Hugging Face Datasets library for alignment and preprocessing operations. Both audio features went into log-mel spectrogram formation while the *WhisperProcessor* performed tokenization after it extracted features along with tokenizing textual inputs.

True and effective training required custom functions to validate data format and afterward the filtering of audio samples that surpassed 30 seconds. The Hugging Face Trainer served as the training configuration platform to ease work with data alongside model training along with performance evaluation. Both input features and labels got padding assistance from the *DataCollatorSpeechSeq2SeqWithPadding* class before processing and the custom *compute_metrics()* function assessed model effectiveness using WER.

A pre-trained Whisper small checkpoint received designated modifications which disabled the cache during training then allowed it to activate during inference to deliver better performance results. Multiple training step values from 50 to 100 to 200 to 250 were combined during the optimization process of the model's learning parameters. The chapter provided an organized method to transform the Whisper model while developing efficient training techniques alongside solid evaluation practices for ASR applications.

CHAPTER 5 - EVALUATION AND FINDINGS

5.1. Introduction

A thorough assessment exists within this section which details the performance evaluation of the fine-tuned Whisper model when applied to ASR in the Sinhala banking domain. The evaluation methodology section presents information about metrics along with benchmarks which measure how precise and effective the model functions. Testing is conducted on new data to obtain quantitative results which receive a critical interpretation in this section. The section focuses on the analysis of significant findings along with the evaluation of aids and weaknesses within the model framework. The research aims to show model performance relative to its research targets and create foundations that support both improvement and practical worldwide applications.

The evaluation of the implemented ASR system through transfer learning sought to adapt successful methods already used for different languages while making necessary adjustments to suit Sinhala speech patterns specifically in banking domains. This evaluation assessed different Whisper model types including small, medium and other options to identify the best configuration suitable for this application. The evaluation of the Sinhala-specific model included performance testing based on results obtained from other low-resource languages which employed the same Whisper framework. The comparison analysis helped show how well the system performs under various linguistic conditions and revealed its capability to fulfill banking industry specifications in Sinhala.

5.2. Result of Implemented Sinhala ASR for Banking Domain

Performance results from fine-tuning the Whisper model with Sinhala banking data under varying training conditions appear in the following table (Table 5.1). The researchers performed experiments to determine the best configurations of training steps along with gradient accumulation for achieving maximum accuracy in automatic speech recognition.

Table 5.1: Evaluation training results of Sinhala ASR

Training Steps	Gradient accumulation steps	Model Card	Outcome
100	None	whisper-small-si-bank-v4	Loss: 0.8527 Wer Ortho: 81.5217 Wer: 68.4337
200	None	whisper-small-si-bank-v1	Loss: 0.6361 Wer Ortho: 69.5652 Wer: 46.9880
250	None	whisper-small-si-bank-v3	Loss: 0.7054 Wer Ortho: 64.1791 Wer: 47.4041
250	2	whisper-small-si-bank-v5	Loss: 0.6785 Wer Ortho: 65.2174 Wer: 54.6988

For the initial experiment with 100 training steps and no gradient accumulation, the model (v4) achieved a loss of 0.8527, with a WER of 68.43% and an orthographic WER of 81.52%. Increasing the training steps to 200 (v1) significantly improved performance, reducing the loss to 0.6361 and achieving a lower WER of 46.99% and an orthographic WER of 69.57%. When the training steps were further increased to 250 without gradient accumulation (v3), the loss slightly rose to 0.7054, with the WER and orthographic WER improving to 47.40% and 64.18%, respectively.

The last set of experiments applied 250 training steps with the gradient accumulation set to 2 (v5) to achieve a competitive loss of 0.6785 and orthographic WER rate of 65.22% with WER at 54.70%. The experimental findings indicate that longer training steps always improved results but gradient accumulation failed to yield substantial performance benefits thus indicating that better hyperparameter adjustments or different strategies should be considered for enhanced performance.

5.2.1. Qualitative evaluation of the model's transcription performance

The evaluation of model transcription performance involved qualitative assessment of test samples sourced from the Sinhala banking domain along with quantitative measurements of loss and WER. The assessment process involved comparing automatic transcriptions from the model with the established ground truth to measure performance in actual situations.

For instance, when using the model version "*IshanSuga/whisper-small-si-bank-v3*", which achieved a WER of 22.39%, the model accurately transcribed the audio file "bank_sin_01_00081.wav." The transcription perfectly matched the ground truth, "**ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න**", The model proved its capability in handling complex financial terminology while demonstrating excellent accuracy regarding auditory input.

However, not all transcriptions were flawless. The audio file "bank_sin_01_00082.wav" presented a challenging case where the ground truth was "**විදේශ මුදල් තැන්පතු කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු දැනගන්න**". The model's output, "**විද්ද්දුම්දර තැන්පත් කිරීමේ දිවිදිදිදිදිදිදෙන් චින්නාවිල ගාසත දෙන්න**", The expected transcription levels showed a substantial difference from what was predicted. The system generates transcripts that differ significantly from the original content especially when processing complex and nuanced sentences thus pointing out improvement opportunities for developers.

The qualitative results in Appendix 1 demonstrate why researchers should extend their analysis past numerical scores because they reveal critical performance strengths and limitations of the model. Data from both successful and erroneous transcriptions yield essential knowledge to enhance performance and future developments of the training model.

5.3. Comparison of Whisper Small vs Medium on Sinhala banking dataset

This evaluation of Whisper Small and Whisper Medium models performing text transcription on the Sinhala banking dataset demonstrates essential findings about their transcription accuracy when processing specialized language speech. The Sinhala language benefited from training through the Whisper architecture because it is among its supported languages thus creating excellent conditions for low-resource language evaluation.

5.3.1. Whisper-Small Model

The assessment of the Whisper small model on Sinhala banking data showed major inconsistencies in transcription outcomes due to its lack of direct applicability to this particular domain.

For example, the Whisper small model ("openai/whisper-small") produced a WER score of 361.39%, a stark contrast to the 22.39% WER achieved by the fine-tuned model ("IshanSuga/whisper-small-si-bank-v3"). The transcription output for audio file "bank_sin_01_00091.wav" was particularly problematic, where the ground truth "මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න" was distorted into a repetitive and nonsensical sequence. Similarly, the transcription for "bank_sin_01_00092.wav" showed an equally incoherent output, failing to capture the intended message of "ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න." (Figure 5.1)


```
{
    "model_name": "openai/whisper-small",
    "wer_score": 3.613861386138614,
    "transcriptions": [
        {
            "audio_file": "bank_sin_01_00091.wav",
            "ground_truth": "මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.",
            "transcription": " අපි
දැතියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියි

යියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියි
ය",
            "log": "[Audio File]: bank_sin_01_00091.wav\n[Reference]: මට ගෙවීම් ඉතිහාසය
බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.\n[Hypothesis]:   අපි
දැතියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියි
යියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියියිය"
        },
        {
            "audio_file": "bank_sin_01_00092.wav",
            "ground_truth": "ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය'
පිටුව කියවා බලන්න.",
            "transcription": " අපින් අතින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින්
කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින්
කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින්",
            "log": "[Audio File]: bank_sin_01_00092.wav\n[Reference]: ඔබගේ අන්තර්ජාල
ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.\n[Hypothesis]:   අපින් අතින්
කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින්
කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින්
කින් කින් කින් කින් කින් කින් කින් කින් කින් කින් කින්"
        },
```


Figure 5.1: openai/whisper-small transcription

5.3.2. Whisper-Medium Model

A detailed evaluation of the Whisper medium model on Sinhala banking dataset echoed the same translation issues identified within the Whisper small model assessment although the

model was more advanced. Many of the transcriptions contained repetitive or nonsensical text, such as "අපි දැකියියියි..." and "අපින් කින් කින්...," which do not align with the expected outputs.

The Whisper medium model ("*openai/whisper-medium*") recorded a WER score of 320.90%, significantly higher than the fine-tuned model's 22.39% WER ("*IshanSuga/whisper-small-si-bank-v3*"). A detailed examination of the transcriptions shows the model's limitations. For instance, the transcription of "bank_sin_01_00081.wav," where the ground truth was "ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න," resulted in an output filled with repetitive, meaningless phrases. Similarly, for "bank_sin_01_00082.wav," the model produced a transcription in different language text, which not only deviated from the original meaning but also introduced inaccuracies in the language format (Figure 5.2).


```
{
    "model_name": "openai/whisper-medium",
    "wer_score": 3.208955223880597,
    "transcriptions": [
        {
            "audio_file": "bank_sin_01_00081.wav",
            "ground_truth": "ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති
පිළිපදින්න.",
            "transcription": " අපි කිරින්න් කිරියික් කිරියික් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින්
කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින්
කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිර",
            "log": "[Audio File]: bank_sin_01_00081.wav\n[Reference]: ඔබගේ විදෙස් ගිණුම
විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න.\n[Hypothesis]:  අපි කිරින්න් කිරියික් කිරියික් කිරින්
කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින්
කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිරින් කිර"
        },
        {
            "audio_file": "bank_sin_01_00082.wav",
            "ground_truth": "විදේශ මුදල් තැන්පතු කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු
දැනගන්න.",
            "transcription": " विदेश मुदाल तम्पक्किरीमे दी अधाल विदम् सहा अवस्तर गास्तू दिनने गान्ना",
            "log": "[Audio File]: bank_sin_01_00082.wav\n[Reference]: විදේශ මුදල් තැන්පතු
කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු දැනගන්න.\n[Hypothesis]:  विदेश मुदाल तम्पक्किरीमे दी अधाल विदम्
सहा अवस्तर गास्तू दिनने गान्ना"
        },
```


Figure 5.2: openai/whisper-medium transcription

These results (Appendix 1) demonstrates that domain-specific data training is indispensable for developing accurate ASR systems. The Whisper small and medium model illustrates the essential nature of fine-tuning for transcription accuracy enhancement in specialty applications due to its inability to accommodate the intricate banking terminology and context of Sinhala.

5.4. Comparison of other ASR systems developed for other low-resource languages

Table 5.1 shows the Whisper small model achieves better accuracy while processing Sinhala banking data due to fine-tuning procedures. The Whisper small model leads to distinct performance outcomes across ASR systems that use it to process Cantonese and Spanish as well as Hindi and Japanese and Korean and Turkish based on dataset sizes training steps and domain characteristics.

Table 5.2: Comparison of other ASR systems developed for other low-resource languages

Base Model	Language	steps	wer	loss	dataset	link
whisper-small	Cantonese	15000	7.93 CER (without punctuations) 9.72 CER (with punctuations)	0.6007	Common Voice 16.0	Ref:
whisper-small	Spanish	25000	20.6842	0.4485	common voice dataset v11	Ref:
whisper-small	Hindi	5000	32.0113	0.4519	Common Voice 11.0	Ref:
whisper-small	Japanese	None	23.1 CER	None	Common Voice, JVS and JSUT	Ref:
whisper-small	Korean	None	9.48	0.1943	AI hub dataset	Ref:
whisper-small	Turkish	4000	20.3954	0.2221	Common Voice 16.1	Ref:

According to Table 5.2, the Cantonese model based on Whisper small reached a Character Error Rate (CER) of 7.93% without punctuation when trained using Common Voice 16.0 to 15,000 steps at a loss of 0.6007. Spanish ASR systems solved 20.68% WER and reached a loss rate of 0.4485 after spending 25,000 training steps on Common Voice v11 due to their exposure to large datasets. Training a Hindi model on Common Voice 11.0 with 5,000 steps produced a WER of 32.01% along with a loss of 0.4519. The Turkish model that underwent training on the Common Voice 16.1 dataset during 4,000 steps achieved a performance metric of 20.39% WER using a low loss value of 0.2221.

The ASR systems in Japanese and Korean demonstrate different results in their performance. The Japanese model operated with three datasets consisting of Common Voice and Japanese versatile speech (JVS), and Japanese speech corpus of Saruwatari-lab., University of Tokyo (JUST) resulting in a CER of 23.1% although no specialized training techniques or loss metrics

were mentioned. The Korean model processed data from the AI Hub dataset to produce a WER assessment of 9.48% while maintaining a low metric loss of 0.1943 thus demonstrating reliable system stability. ASR model performance receives substantial improvement from a customized method which implements adequate training levels and suitable dataset selection.

The base model of Whisper forms a useful starting point but firms require domain-specific training with purpose-built datasets to fulfill transcription needs of particular lower-resource language domains.

5.5. Summary of the chapter

A full performance assessment of the fine-tuned ASR model with the Whisper framework occurred inside the evaluation and findings chapter for the Sinhala banking domain. A systematic performance comparison took place between the fine-tuned Whisper-small model together with its untrained counterparts Whisper-small and Whisper-medium as well as the evaluation of its effectiveness against alternative low-resource language ASR systems.

Using pre-trained Whisper models with Sinhala as one of the training languages produced substantial WER score and transcription faults according to the experimental results. The analysis confirmed the requirement to optimize the model structure to understand the specific linguistic elements and specialized words which appear in Sinhala banking language. The fine-tuned Whisper-small model attained a WER measure of 22.39% whereas it surpassed the untrained models which demonstrated the merits of specialized training methods.

The research on fine-tuned model performance revealed that specialized domain training together with suitable data selection and suitable training steps supports low-resource language accuracy improvement. Specialized ASR applications gained ground when models for Cantonese, Spanish, Hindi, Japanese, Korean and Turkish were examined through systematic fine-tuning protocols.

The chapter demonstrated that model fine-tuning enhances transcription quality while making it ready to address particular language and domain requirements for bank deployment.

CHAPTER 6 - CONCLUSION

6.1. Introduction

Final findings emerge after completing the research and investigation activities. The research objective from Section 1.4 achieved success through the technical methods and approaches contained in the 4th chapter along with the methodologies from the 3rd Chapter.

6.2. Conclusions about Research Questions

6.2.1. Reflections of Main Research Question

The core research question explored in this study is: " *How can we develop an efficient and accurate domain-dependent Sinhala ASR system with minimal data*?" The evaluation of the Sinhala ASR system in banking produced meaningful findings. This approach's adaptation process showed its effectiveness through combining theoretical knowledge with specific training modifications that used optimized training procedures.

Research on Sinhala ASR enhanced performance while using transfer learning solutions for low-resource language systems. The base model from Whisper brings solid foundation yet research shows that domain-specialized training data becomes essential to reach the preferred transcription results within low-resource contexts.

Simply relying on pre-training a model for the target language fails to deliver satisfactory results during direct application before fine-tuning. The banking dataset confirms this conclusion through its evaluation of Whisper-small and Whisper-medium models. Fine-tuning is essential because it makes the model suitable for distinct linguistic patterns and sector-specific needs to reach better results and measurement accuracy.

Whisper-small achieved a Word Error Rate (WER) of 22.39% on the test set after conducting fine-tuning through maximum 250 steps. The fine-tuned model showed its ability to work with financial terminology complexity along with strong accuracy in processing audible data points.

Not every transcription managed to maintain absolute accuracy in its translation. The system generated results which diverged considerably from the source content mainly during transcription processes of complex sentences. The current model needs refinement by

developers to achieve better results since additional training data with more training steps should be applied to its optimization process.

A project was initiated to optimize the Whisper-medium model through training with the Sinhala banking domain data. The project encountered major obstacles because of problems with computing capacity. Memory errors appeared as the system limitations prevented advancement when using smaller step sizes during the fine-tuning process.

Certain constraints restricted exploratory studies of accuracy and model performance so the smaller model provided better results compared to the larger model and limited experiments with sophisticated training methods for better transcription quality.

6.2.2. Reflections of Sub Research Question 1

Addressing the sub-research question, "*What are the current limitations of ASR systems for low-resource languages, and how can they be improved for domain-specific applications*?" this study has identified several critical challenges and proposed effective solutions.

Present ASR systems operate poorly with low-resource languages because they face three challenges: limited training data quantity and language differences and insufficient pre-trained model support. The accuracy level decreases significantly when specialized domains require processing because the systems struggle with a narrow vocabulary and context which is highly domain-specific. An assessment of the banking domain confirmed that the Whisper ASR model which included Sinhala as a training language fell behind in performance standards because it generated high Word Error Rates (WER) errors and inaccurate transcriptions.

These research outcomes show that dedicated domain training serves as the necessary solution for these performance challenges. The study improved the model performance through applying optimal parameter tuning technique after preparing and aligning the model with a dataset suitable for its specific domain context. The ASR system developed from this process reached outstanding WER results while proving capable of effectively processing the complex banking sector terminology.

The development of domain-specific applications for low-resource languages requires fine-tuned general-purpose ASR models using targeted datasets. The research delivers important

findings about improving ASR system performance and provides a method that works for enhancing precision in specialized areas using minimal data requirements.

6.2.3. Reflection of Sub Research Question 2

Reflecting on the sub-research question, "*How can machine learning and deep learning techniques be adapted to enhance ASR performance with limited training data?*" this study has demonstrated effective strategies for maximizing model performance in low-resource settings.

Transfer learning exists as a robust technique to defeat data deficits mainly through pre-trained models like Whisper. The study built upon existing model knowledge to accomplish successful ASR system fine-tuning for the Sinhala banking domain with limited available data. The training process included accumulation steps together with optimized hyperparameter tuning which equated to effective training along with better accuracy.

Proper data preparation required careful normalization techniques to minimize transcription errors according to this research. The model results improved when researchers selected Whisper models of different sizes and examined performance efficiency for small and medium models. Domain-specific adaptation proves necessary for achieving robust performance since pre-trained models make an excellent initial framework according to research findings.

The study results show that deep learning approaches paired with targeted fine-tuning produce major gains in automated speech recognition capability when dealing with small dataset sizes. This investigation presents workable methods which can guide ASR system creation for domains with limited resources and specialized use conditions.

6.2.4. Reflection of Sub Research Question 3

Reflecting on the sub-research question, "*How can we fine-tune the transfer learning approach to improve the accuracy of the domain-dependent Sinhala ASR?*" The research findings from this study enabled better understanding of ASR optimization through strategic model refinement. This research proves that pre-trained Whisper models provide solid foundational capabilities yet specialized tuning delivers substantial accuracy advances over these models.

Factors Impacting ASR Accuracy:

Several factors influenced the ASR system's accuracy, including:

- Dataset Quality and Size: The growth of both science and practice depends on high-quality and well-annotated audio examples specific to banking institutions.
- Language Characteristics: The particular phonetic along with syntactic properties found in the Sinhala language needed proper preprocessing normalization techniques to minimize transcription errors.
- Model Configuration: Using Whisper models of different sizes required a performance efficiency trade-off because banking terminology required specialized fine-tuning.

Optimization Strategies:

To enhance accuracy, the study implemented several optimization strategies:

- Gradient Checkpointing & Accumulation Steps: The technical methods-controlled memory usage while enabling bigger batch sizes during training and improved both model stability and performance results.
- Hyperparameter Tuning: Trial and error testing with training step numbers between 100 and 250, in addition to examining configurations like using accumulation steps revealed the most suitable fine-tuning parameters.
- Normalization Techniques: The implementation of basic text normalization minimized prediction accuracy differences compared to ground truth results especially when processing punctuation elements and text variations.
- Domain Adaptation: The model developed enhanced capabilities to understand bank-related terminology through the integration of banking-specific language in its training process.

6.4. Conclusions about Research Problem

The study focused on fixing essential performance barriers that prevent an efficient and precise automatic speech recognition (ASR) system development for Sinhala banking operations. The research effectively addressed original problems by exploring transfer learning approaches combined with fine-tuning methods and evaluation procedures.

The main barrier to ASR systems operates in low-resource languages emerged as a critical issue. The pre-trained Whisper models included Sinhala but failed to achieve acceptable results for banking transcription despite their language training capabilities. The analysis emphasized that

the Sinhala language with banking domain content demands its own specialized fine-tuning procedures.

The research established that machine learning and deep learning approaches need modification to maximize ASR performance utilization with restricted training material availability. Transfer learning strategies proved effective because traditional training methods were not applicable when dealing with limited data availability. The models required additional parameter adjustment to achieve better accuracy levels in domain-specific implementations. The model's performance gained substantial improvement from the utilization of gradient checkpointing together with hyperparameter tuning and normalization procedures.

The research solved its main issues effectively through the finding that customized pre-trained model optimization which prioritizes domain-specific content along with language features produces significant ASR performance enhancements. Research reveals that purposeful system modifications hold key importance for addressing the current challenges of low-resource language ASR systems thus enabling their widespread business application in specialized domains including banking.

6.5. Limitations

The research successfully developed an ASR system for banking-related Sinhala speech but various constraints restricted the study's effectiveness. The research encountered several restrictions particularly because of limited data sources and complex models and insufficient processing capabilities.

6.5.1. Data Availability and Diversity

The main obstacle researchers faced throughout this project was the lack of suitable Sinhala speech datasets available to the public in the banking domain. Sinhala speech datasets available in general use do not reflect banking industry terminology nor do they demonstrate banking sector speech patterns adequately. The scarcity of publicly accessible databases demanded researchers to develop their private dataset for the ASR model tuning process. Small dataset size-imposed constraints when training the model to achieve its best accuracy performance. The limited availability of extensive domain-specific datasets prevented the model from undergoing thorough training that could enhance its operational performance. Due to insufficient diversity in dataset availability which lacked widespread accents and speech styles and environmental

conditions the model showed limited capacity to apply across various speakers and genuine scenarios.

6.5.2. High Computational Demand

Large pre-trained models such as Whisper-medium demanded considerable computational resources that affected the study through increased computational requirements during fine-tuning. Fine-tuning models of this size needs computers with high-performance GPUs and requires long periods of training time. Focusing on models that feature extensive sizes adds intensive stress to the training procedure because it requires larger memory capacity and longer processing times. The available computational capacity remained insufficient for optimum training of larger models with all their requirements. The limitations forced careful management of training processes because researchers needed to balance their model sizes alongside training duration times and batch sizes. The high computational requirements for fine-tuning this massive model led to training time management conflicts between training steps and available computational resources that potentially impacted the model precision in its final state.

6.6. Implications for Future Research

The findings of this study offer valuable insights into the development of domain-specific Automatic Speech Recognition (ASR) systems for low-resource languages, specifically Sinhala, within the banking sector. Further exploration and multiple improvements in the field of ASR will significantly boost its capabilities particularly when dealing with restricted data access and computational resources.

6.6.1. Expansion of Domain-Specific Datasets

Future research must become a priority for creating extensive multilingual datasets which focus on individual domains such as banking. The main problem in this research was the insufficient amount of Sinhala banking sector speech data which was unavailable openly. Future research should concentrate on building extensive domain-focused public datasets because this problem hinders progress in this field. The competence of ASR systems will increase through obtaining datasets that incorporate various accents together with diverse dialects from real-life scenarios including noisy environments and diverse audio quality levels. In order to enhance ASR accuracy system developers should integrate domain-specific specialized language terminology alongside linguistic nuances into their methodologies.

Moreover, expanding the data collection process to include a larger and more diverse speaker base—such as 100 users, each contributing one scripted recording—would significantly enhance the generalizability of the model. This structured yet diverse dataset would provide broader linguistic coverage while maintaining consistency through controlled scripting, as well as better represent real-world speech variability.

6.6.2. Advancing Transfer Learning Techniques

The research findings demonstrate that transfer learning proves highly efficient for ASR systems mainly when using pre-trained models like Whisper for fine-tuning specifically when limited data exists. Future research will investigate sophisticated implementation methods of transfer learning that include domain adaptation and few-shot learning for optimizing the performance of specialized application ASR. The investigation of single pre-trained models alongside various pre-trained models and multiple training dataset applications could enable the development of advanced systems which address multiple domains through multiple inputs.

6.6.3. Optimization of Model Training

Future researchers need to develop training strategies for large models which consume less computational resources given their current high needs. The process known as knowledge distillation enables smaller models to learn from large ones to keep performance levels while decreasing costs. Research into gradient checkpointing and comparable memory-efficiency methods may enable the training of extensive additional model sizes that exceed current hardware constraints. The optimization of training algorithms together with hyperparameter adjustments in ASR systems would enhance both efficiency and effectiveness of the training process.

6.6.4. Human-Centered Evaluation

Future research should incorporate qualitative evaluations involving human assessments of the ASR application's performance in real-world scenarios. Gathering feedback from end-users can provide a deeper understanding of usability, perceived accuracy, and contextual effectiveness that purely quantitative metrics may overlook.

The research implications for low-resource language ASR development focus on developing larger datasets together with advanced transfer learning approaches and optimized training

methods. The accuracy and applicability along with operational efficiency of ASR systems will improve extensively for languages with limited resources such as Sinhala when researchers address study limitations while continuing to innovate in these fields in future research.

BIBLIOGRAPHY

Bahdanau, D., Cho, K. and Bengio, Y. (2014) *NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE*. Available at: https://doi.org/10.48550/arXiv.1409.0473.

Ghai, W. and Singh, N. (2012) *Literature Review on Automatic Speech Recognition*, *International Journal of Computer Applications*. Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bdcf7f42bc5db1842250b4dc3e5911a181dbd685.

Hannun, A. (2021) *The History of Speech Recognition to the Year 2030*. Available at: https://doi.org/10.48550/arXiv.2108.00084.

Huang, J. *et al.* (2020) 'Cross-Language Transfer Learning, Continuous Learning, and Domain Adaptation for End-to-End Automatic Speech Recognition'. Available at: http://arxiv.org/abs/2005.04290.

Karunathilaka, N.A.K.H.S. (2020) *Low-resource Sinhala Speech Recognition using Deep Learning*. 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka. Available at: https://ieeexplore.ieee.org/document/9325468.

Kermanshahi, M.A., Akbari, A. and Nasersharif, B. (2021) 'Transfer Learning for End-to-End ASR to Deal with Low-Resource Problem in Persian Language', in *26th International Computer Conference, Computer Society of Iran, CSICC 2021*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.1109/CSICC52343.2021.9420540.

Levis, J. and Suvorov, R. (2012) 'Automatic Speech Recognition', in *The Encyclopedia of Applied Linguistics*. Oxford, UK: Blackwell Publishing Ltd, pp. 1–4. Available at: https://doi.org/10.1002/9781405198431.wbeal0066.

Liu, Y., Yang, X. and Qu, D. (2024) 'Exploration of Whisper fine-tuning strategies for low-resource ASR', *Eurasip Journal on Audio, Speech, and Music Processing*, 2024(1). Available at: https://doi.org/10.1186/s13636-024-00349-3.

Morbini, F. *et al.* (2013) *Which ASR should I choose for my dialogue system?* Association for Computational Linguistics. Available at: http://cmusphinx.sourceforge.net/wiki/tutorialadapt.

Nanayakkara, A.L. (2022) *Exploring Model Level Transfer Learning For Improving Sinhala Speech Recognition MCS3204*. Available at: https://www.researchgate.net/publication/377261579_Exploring_Model_Level_Transfer_Learning_For_Improving_Sinhala_Speech_Recognition.

Polat, H. *et al.* (2024) 'Implementation of a Whisper Architecture-Based Turkish Automatic Speech Recognition (ASR) System and Evaluation of the Effect of Fine-Tuning with a Low-Rank Adaptation (LoRA) Adapter on Its Performance', *Electronics (Switzerland)*, 13(21). Available at: https://doi.org/10.3390/electronics13214227.

Radford, A. *et al.* (2022) *Robust Speech Recognition via Large-Scale Weak Supervision*. Proceedings of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research. Available at: https://github.com/openai/.

du Simplon, R. *et al.* (2005) *I D I A P On the Use of Information Retrieval Measures for Speech Recognition Evaluation*. Available at: https://infoscience.epfl.ch/entities/publication/0d493723-4ca3-

4175-83f9-a1ff68c6cd48.

Song, Z. *et al.* (2024) 'LoRA-Whisper: Parameter-Efficient and Extensible Multilingual ASR'. Available at: http://arxiv.org/abs/2406.06619.

Twiefel, J. *et al.* (2014) *Improving Domain-Independent Cloud-Based Speech Recognition with Domain-Dependent Phonetic Post-Processing*. Available at: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Vaswani, A. *et al.* (2017) *Attention Is All You Need*. Available at: https://proceedings.neurips.cc/paper_files/paper/2017.

Wijesekera, H.D. and Alford, J. (2019) 'Bilingual Education Classrooms in Sri Lankan Schools: A Social Space for Ethnolinguistic Reconciliation', in, pp. 82–83. Available at: https://doi.org/10.1007/978-3-030-14386-2_5.

APPENDIX A: ASR QUALITATIVE ANALYSIS RESULTS

IshanSuga/whisper-small-si-bank-v3

{
  "model_name": "IshanSuga/whisper-small-si-bank-v3",
  "wer_score": 0.22388059701492538,
  "transcriptions": [
    {
      "audio_file": "bank_sin_01_00081.wav",
      "ground_truth": "ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න.",
      "transcription": "ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න.",
      "log": "[Audio File]: bank_sin_01_00081.wav\n[Reference]: ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න.\n[Hypothesis]: ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න."
    },
    {
      "audio_file": "bank_sin_01_00082.wav",
      "ground_truth": "විදේශ මුදල් තැන්පතු කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු දැනගන්න.",
      "transcription": "විද්ද්දුම්දර තැන්පත් කිරීමේ දිවිදිදිදිදිදිදෙන් චින්නාවිල ගාසත දෙන්න.",
      "log": "[Audio File]: bank_sin_01_00082.wav\n[Reference]: විදේශ මුදල් තැන්පතු කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු දැනගන්න.\n[Hypothesis]: විද්ද්දුම්දර තැන්පත් කිරීමේ දිවිදිදිදිදිදිදෙන් චින්නාවිල ගාසත දෙන්න."
    },
    {
      "audio_file": "bank_sin_01_00083.wav",
      "ground_truth": "මට විදේශ ගනුදෙනු සඳහා විශේෂිත විශ්වාස ලිපි ගිණුමක් අවශ්‍යයි.",
      "transcription": "මට විදේශ ගනුදෙනු සඳහා විශේෂිත විශ්වාස ලිපි ගිණුමක් අවශ්‍යයි.",
      "log": "[Audio File]: bank_sin_01_00083.wav\n[Reference]: මට විදේශ ගනුදෙනු සඳහා විශේෂිත විශ්වාස ලිපි ගිණුමක් අවශ්‍යයි.\n[Hypothesis]: මට විදේශ ගනුදෙනු සඳහා විශේෂිත විශ්වාස ලිපි ගිණුමක් අවශ්‍යයි."
    },
    {
      "audio_file": "bank_sin_01_00084.wav",
      "ground_truth": "අපගේ විශ්වාස ගිණුම් අංශය එයට අදාළ උපදේශනය ලබාදේ.",
      "transcription": "අපගේ විශ්වාස ගිණුම් අංශය එයට අදාළ උපදේශනය ලබාදේ.",
      "log": "[Audio File]: bank_sin_01_00084.wav\n[Reference]: අපගේ විශ්වාස ගිණුම් අංශය එයට

අදාළ උපදේශනය ලබාදේ.\n[Hypothesis]: අපගේ විශ්වාස ගිණුම් අංශය එයට අදාළ උපදේශනය ලබාදේ."

},

{

"audio_file": "bank_sin_01_00085.wav",

"ground_truth": "මට අන්තර්ජාල බිල් ගෙවීම් සේවාව සක්‍රිය කරගැනීමට අවශ්‍යයි.",

"transcription": "මට අන්තර්ජාල බිල් ගෙවීම් සේවාව සක්‍රිය කරගැනීමට අවශ්‍යයි.",

"log": "[Audio File]: bank_sin_01_00085.wav\n[Reference]: මට අන්තර්ජාල බිල් ගෙවීම් සේවාව සක්‍රිය කරගැනීමට අවශ්‍යයි.\n[Hypothesis]: මට අන්තර්ජාල බිල් ගෙවීම් සේවාව සක්‍රිය කරගැනීමට අවශ්‍යයි."

},

{

"audio_file": "bank_sin_01_00086.wav",

"ground_truth": "ඔබගේ ගිණුම් ප්‍රවේශ පිටුවෙන්, බිල් ගෙවීම්, අන්තර්ජාල බිල් ගෙවීම් හරහා පහසුවෙන් ගෙවන්න.",

"transcription": "ඔබගේ ගිණුම් ප්‍රවේශ පිටුවෙන්, බිල් ගෙවීම්, අන්තර්ජාල බිල් ගෙවීම් හරහා පහසුවෙන් ගෙවන්න.",

"log": "[Audio File]: bank_sin_01_00086.wav\n[Reference]: ඔබගේ ගිණුම් ප්‍රවේශ පිටුවෙන්, බිල් ගෙවීම්, අන්තර්ජාල බිල් ගෙවීම් හරහා පහසුවෙන් ගෙවන්න.\n[Hypothesis]: ඔබගේ ගිණුම් ප්‍රවේශ පිටුවෙන්, බිල් ගෙවීම්, අන්තර්ජාල බිල් ගෙවීම් හරහා පහසුවෙන් ගෙවන්න."

},

{

"audio_file": "bank_sin_01_00087.wav",

"ground_truth": "ඔබගේ ගිණුමේ ආරක්ෂාව වැඩිදියුණු කිරීමට ද්විත්ව සත්‍යාපිරුක්ති ක්‍රමය සක්‍රිය කරන්න.",

"transcription": "ඔබගේ ගිණුමේ ආර්ෂවා වැදියුයු කිරීමට දිදෛෝසත් පිළික්නුමේ සක්වා කරන්න.",

"log": "[Audio File]: bank_sin_01_00087.wav\n[Reference]: ඔබගේ ගිණුමේ ආරක්ෂාව වැඩිදියුණු කිරීමට ද්විත්ව සත්‍යාපිරුක්ති ක්‍රමය සක්‍රිය කරන්න.\n[Hypothesis]: ඔබගේ ගිණුමේ ආර්ෂවා වැදියුයු කිරීමට දිදෛෝසත් පිළික්නුමේ සක්වා කරන්න."

},

{

"audio_file": "bank_sin_01_00088.wav",

"ground_truth": "අපගේ ගිණුම් කලමනාකරණ කණ්ඩායම ඔබට උදව් කිරීමට සූදානම්.",

"transcription": "අපගේ ගිණුම් කලමනාකරණ කණ්ඩායම ඔබට උදව් කිරීමට සූදාන්න.",

"log": "[Audio File]: bank_sin_01_00088.wav\n[Reference]: අපගේ ගිණුම් කලමනාකරණ කණ්ඩායම ඔබට උදව් කිරීමට සූදානම්.\n[Hypothesis]: අපගේ ගිණුම් කලමනාකරණ කණ්ඩායම ඔබට උදව් කිරීමට සූදාන්න."

},

{

"audio_file": "bank_sin_01_00089.wav",

"ground_truth": "ඔව්, අපගේ ඇප් එක භාවිතා කර ඔබගේ ඉතිරිකිරීමේ ගිණුමට සහ ක්‍රෙඩිට් ගිණුමට මුදල් යවා ගත හැක.",

"transcription": "ඔව්, අපගේ ඇප් එක භාවිතා කර ඔබගේ ඉතිරිකිරීමේ ගිණුමට සහ ක්‍රෙඩිට් ගිණුමට මුදල් යවා ගත හැක.",

"log": "[Audio File]: bank_sin_01_00089.wav\n[Reference]: ඔව්, අපගේ ඇප් එක භාවිතා කර ඔබගේ ඉතිරිකිරීමේ ගිණුමට සහ ක්‍රෙඩිට් ගිණුමට මුදල් යවා ගත හැක.\n[Hypothesis]: ඔව්, අපගේ ඇප් එක භාවිතා කර ඔබගේ ඉතිරිකිරීමේ ගිණුමට සහ ක්‍රෙඩිට් ගිණුමට මුදල් යවා ගත හැක."

},

{

"audio_file": "bank_sin_01_00090.wav",

"ground_truth": "ඔබගේ ගිණුමේ පරිශීලක නාමය සහ මුරපදය සැලකිලිමත්ව රහසිගතව තබන්න.",

"transcription": "ඔබගේ ගිණුමේ පරිශීලක නාමය සහ මුරපදය සැලකිලිමත්ව රහසිගතව තබන්න.",

"log": "[Audio File]: bank_sin_01_00090.wav\n[Reference]: ඔබගේ ගිණුමේ පරිශීලක නාමය සහ මුරපදය සැලකිලිමත්ව රහසිගතව තබන්න.\n[Hypothesis]: ඔබගේ ගිණුමේ පරිශීලක නාමය සහ මුරපදය සැලකිලිමත්ව රහසිගතව තබන්න."

},

{

"audio_file": "bank_sin_01_00091.wav",

"ground_truth": "මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.",

"transcription": "මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන්නෙන් පැහැදිලි කර දෙන්න.",

"log": "[Audio File]: bank_sin_01_00091.wav\n[Reference]: මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.\n[Hypothesis]: මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන්නෙන් පැහැදිලි කර දෙන්න."

},

{

"audio_file": "bank_sin_01_00092.wav",

"ground_truth": "ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.",

"transcription": "ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, ගනුදෙනු ඉතිහාසය පිටුව කියවා බලන්න.",

"log": "[Audio File]: bank_sin_01_00092.wav\n[Reference]: ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.\n[Hypothesis]: ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, ගනුදෙනු ඉතිහාසය පිටුව කියවා බලන්න."

},

{

"audio_file": "bank_sin_01_00093.wav",

"ground_truth": "කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත.",

"transcription": "කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර

ඇත.",
"log": "[Audio File]: bank_sin_01_00093.wav\n[Reference]: කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත.\n[Hypothesis]: කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත."
},
{
"audio_file": "bank_sin_01_00094.wav",
"ground_truth": "මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?",
"transcription": "මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?",
"log": "[Audio File]: bank_sin_01_00094.wav\n[Reference]: මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?\n[Hypothesis]: මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?"
},
{
"audio_file": "bank_sin_01_00095.wav",
"ground_truth": "ඔබගේ ගෙවීම් නිසි වේලාවට කිරීමට හැකිනම් ක්‍රෙඩිට් අනුපාතය ඉහළ යනු ඇත.",
"transcription": "ඔබගේ ගෙවීම් නිශ්‍යාලකිරීමට කිරීමට කින්නෝ අනුපාතය යොනවිය යනුදෙන්න.",
"log": "[Audio File]: bank_sin_01_00095.wav\n[Reference]: ඔබගේ ගෙවීම් නිසි වේලාවට කිරීමට හැකිනම් ක්‍රෙඩිට් අනුපාතය ඉහළ යනු ඇත.\n[Hypothesis]: ඔබගේ ගෙවීම් නිශ්‍යාලකිරීමට කිරීමට කින්නෝ අනුපාතය යොනවිය යනුදෙන්න."
},
{
"audio_file": "bank_sin_01_00096.wav",
"ground_truth": "ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.",
"transcription": "ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.",
"log": "[Audio File]: bank_sin_01_00096.wav\n[Reference]: ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.\n[Hypothesis]: ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි."
},
{
"audio_file": "bank_sin_01_00097.wav",
"ground_truth": "අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්‍යතාවන් සලකා බලයි.",
"transcription": "අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්තයාවන් සලකා බලයි.",
"log": "[Audio File]: bank_sin_01_00097.wav\n[Reference]: අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්‍යතාවන් සලකා බලයි.\n[Hypothesis]: අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්තයාවන් සලකා බලයි."
},
{

```
      "audio_file": "bank_sin_01_00098.wav",
      "ground_truth": "ඔබේ ණය ගෙවීම් දිගු කර ගැනීම සඳහා මූල්‍ය උපදේශකයෙකු සම්බන්ධ කර ගන්න.",
      "transcription": "ඔබේ ණය ගෙවීම් දිකුකරනිනින් සඳහා මුරිල් උපදේශක් කළ සම්බන්ධ කරන්න.",
      "log": "[Audio File]: bank_sin_01_00098.wav\n[Reference]: ඔබේ ණය ගෙවීම් දිගු කර ගැනීම සඳහා මූල්‍ය උපදේශකයෙකු සම්බන්ධ කර ගන්න.\n[Hypothesis]: ඔබේ ණය ගෙවීම් දිකුකරනිනින් සඳහා මුරිල් උපදේශක් කළ සම්බන්ධ කරන්න."
    },
    {
      "audio_file": "bank_sin_01_00099.wav",
      "ground_truth": "ඔබට මුරපදය අමතක වී ඇත්නම් කරුණාකර ලහම බැංකු වෙත යන්න.",
      "transcription": "ඔබට මුරපදේ අමතක් ඇත්නුම් කරුණාකර ලගගැමු බැන්කු අන්නෙන්න.",
      "log": "[Audio File]: bank_sin_01_00099.wav\n[Reference]: ඔබට මුරපදය අමතක වී ඇත්නම් කරුණාකර ලහම බැංකු වෙත යන්න.\n[Hypothesis]: ඔබට මුරපදේ අමතක් ඇත්නුම් කරුණාකර ලගගැමු බැන්කු අන්නෙන්න."
    },
    {
      "audio_file": "bank_sin_01_00100.wav",
      "ground_truth": "ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.",
      "transcription": "ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.",
      "log": "[Audio File]: bank_sin_01_00100.wav\n[Reference]: ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.\n[Hypothesis]: ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න."
    }
  ]
}
```


openai/whisper-small


```
{
  "model_name": "openai/whisper-small",
  "wer_score": 3.613861386138614,
  "transcriptions": [
    {
      "audio_file": "bank_sin_01_00091.wav",
      "ground_truth": "මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.",
      "transcription":                                    "                                    අපි
```

දැතියි
යිය",
"log": "[Audio File]: bank_sin_01_00091.wav\n[Reference]: මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.\n[Hypothesis]: අපි දැතියි
යිය"
},
{
"audio_file": "bank_sin_01_00092.wav",
"ground_truth": "ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.",
"transcription": " අපින් අතින් කින්",
"log": "[Audio File]: bank_sin_01_00092.wav\n[Reference]: ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.\n[Hypothesis]: අපින් අතින් කින්"
},
{
"audio_file": "bank_sin_01_00093.wav",
"ground_truth": "කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත.",
"transcription": "
ඇතාන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්න"
,
"log": "[Audio File]: bank_sin_01_00093.wav\n[Reference]: කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත.\n[Hypothesis]:
ඇතාන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නන්‍්න්න්න්න්න්න්න්න්න්න්න්න්න්න"
},
{
"audio_file": "bank_sin_01_00094.wav",
"ground_truth": "මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?",

"transcription": "
අපින්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘ
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘ
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න"
,
"log": "[Audio File]: bank_sin_01_00094.wav\n[Reference]: මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?\n[Hypothesis]:
අපින්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘ
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘ
න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්නෘන්න්න්න්න්න්න්න්න්න්න්න්න්න්න"
},
{
"audio_file": "bank_sin_01_00095.wav",
"ground_truth": "ඔබගේ ගෙවීම් නිසි වේලාවට කිරීමට හැකිනම් ක්‍රෙඩිට් අනුපාතය ඉහළ යනු ඇත.",
"transcription": " අපින් දැන් කින් ක",
"log": "[Audio File]: bank_sin_01_00095.wav\n[Reference]: ඔබගේ ගෙවීම් නිසි වේලාවට කිරීමට හැකිනම් ක්‍රෙඩිට් අනුපාතය ඉහළ යනු ඇත.\n[Hypothesis]: අපින් දැන් කින් ක"
},
{
"audio_file": "bank_sin_01_00096.wav",
"ground_truth": "ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.",
"transcription": " අපින් දමට කර්න් කර්",
"log": "[Audio File]: bank_sin_01_00096.wav\n[Reference]: ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.\n[Hypothesis]: අපින් දමට කර්න් කර්"
},
{
"audio_file": "bank_sin_01_00097.wav",
"ground_truth": "අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්‍යතාවන් සලකා බලයි.",

"transcription": " අපින් ක",

"log": "[Audio File]: bank_sin_01_00097.wav\n[Reference]: අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්‍යතාවන් සලකා බලයි.\n[Hypothesis]: අපින් ක"

},

{

"audio_file": "bank_sin_01_00098.wav",

"ground_truth": "ඔබේ ණය ගෙවීම් දිගු කර ගැනීම සඳහා මූල්‍ය උපදේශකයෙකු සම්බන්ධ කර ගන්න.",

"transcription": " අපි දැන්ශ්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ් න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ් න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ්න්ශ්න්න්න්න්න්න්න්න්න්න්න්න්න්න",

"log": "[Audio File]: bank_sin_01_00098.wav\n[Reference]: ඔබේ ණය ගෙවීම් දිගු කර ගැනීම සඳහා මූල්‍ය උපදේශකයෙකු සම්බන්ධ කර ගන්න.\n[Hypothesis]: අපි දැන්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ් න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ් න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්න්ශ්න්ශ්න්න්න්න්න්න්න්න්න්න්න්න්න්න"

},

{

"audio_file": "bank_sin_01_00099.wav",

"ground_truth": "ඔබට මුරපදය අමතක වී ඇත්නම් කරුණාකර ලඟම බැංකු වෙත යන්න.",

"transcription": " අපින් ක",

"log": "[Audio File]: bank_sin_01_00099.wav\n[Reference]: ඔබට මුරපදය අමතක වී ඇත්නම් කරුණාකර ලඟම බැංකු වෙත යන්න.\n[Hypothesis]: අපින් ක"

},

{

"audio_file": "bank_sin_01_00100.wav",

"ground_truth": "ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.",

"transcription": " අපින් දයස් ද",

"log": "[Audio File]: bank_sin_01_00100.wav\n[Reference]: ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.\n[Hypothesis]: අපින් දයස් ද"

}

]

}

openai/whisper-medium

{

"model_name": "openai/whisper-medium",

"wer_score": 3.208955223880597,

"transcriptions": [

{

"audio_file": "bank_sin_01_00081.wav",

"ground_truth": "ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න.",

"transcription": " අපි කිරින්න් කිරියික් කිරියික් කිරින් කිර",

"log": "[Audio File]: bank_sin_01_00081.wav\n[Reference]: ඔබගේ විදෙස් ගිණුම විවෘත කිරීමට පෙර විදේශ විනිමය නීති පිළිපදින්න.\n[Hypothesis]: අපි කිරින්න් කිරියික් කිරියික් කිරින් කිර"

},

{

"audio_file": "bank_sin_01_00082.wav",

"ground_truth": "විදේශ මුදල් තැන්පතු කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු දැනගන්න.",

"transcription": " विदेश मुदाल तम्पत्किरीमे दी अधाल विदम् सहा अवस्तर गास्तू दिनने गान्ना",

"log": "[Audio File]: bank_sin_01_00082.wav\n[Reference]: විදේශ මුදල් තැන්පතු කිරීමේදී අදාල වියදම් සහ අවසර ගාස්තු දැනගන්න.\n[Hypothesis]: विदेश मुदाल तम्पत्किरीमे दी अधाल विदम् सहा अवस्तर गास्तू दिनने गान्ना"

},

{

"audio_file": "bank_sin_01_00083.wav",

"ground_truth": "මට විදේශ ගනුදෙනු සඳහා විශේෂිත විශ්වාස ලිපි ගිණුමක් අවශ්‍යයි.",

"transcription": " මෙන්න්න් කිරියි යි",

"log": "[Audio File]: bank_sin_01_00083.wav\n[Reference]: මට විදේශ ගනුදෙනු සඳහා විශේෂිත විශ්වාස ලිපි ගිණුමක් අවශ්‍යයි.\n[Hypothesis]: මෙන්න්න් කිරියි යි"

},

{

"audio_file": "bank_sin_01_00084.wav",

"ground_truth": "අපගේ විශ්වාස ගිණුම් අංශය එයට අදාළ උපදේශනය ලබාදේ.",

"transcription": " अपके विश्वासगिनुं मांचे, एटा अधालु उपदेशन लबादे",

"log": "[Audio File]: bank_sin_01_00084.wav\n[Reference]: අපගේ විශ්වාස ගිණුම් අංශය එයට අදාළ උපදේශනය ලබාදේ.\n[Hypothesis]: अपके विश्वासगिनुं मांचे, एटा अधालु उपदेशन लबादे"

},

{

"audio_file": "bank_sin_01_00085.wav",

"ground_truth": "මට අන්තර්ජාල බිල් ගෙවීම් සේවාව සක්‍රිය කරගැනීමට අවශ්‍යයි.",

"transcription": " අපි කිරින්න් කිරින් ක",

"log": "[Audio File]: bank_sin_01_00085.wav\n[Reference]: මට අන්තර්ජාල බිල් ගෙවීම් සේවාව සක්‍රිය කරගැනීමට අවශ්‍යයි.\n[Hypothesis]: අපි කිරින්න් කිරින් ක"

},

{

"audio_file": "bank_sin_01_00086.wav",

"ground_truth": "ඔබගේ ගිණුම් ප්‍රවේශ පිටුවෙන්, බිල් ගෙවීම්, අන්තර්ජාල බිල් ගෙවීම් හරහා පහසුවෙන් ගෙවන්න.",

"transcription": " ಸುಃ ಸುಃ",

"log": "[Audio File]: bank_sin_01_00086.wav\n[Reference]: ඔබගේ ගිණුම් ප්‍රවේශ පිටුවෙන්, බිල් ගෙවීම්, අන්තර්ජාල බිල් ගෙවීම් හරහා පහසුවෙන් ගෙවන්න.\n[Hypothesis]: ಸುಃ ಸುಃ"

},

{

"audio_file": "bank_sin_01_00087.wav",
"ground_truth": "ඔබගේ ගිණුමේ ආරක්ෂාව වැඩිදියුණු කිරීමට ද්විත්ව සත්‍යාපිරුක්ති ක්‍රමය සක්‍රිය කරන්න.",
"transcription": " අපි එය කිරීම් ක",
"log": "[Audio File]: bank_sin_01_00087.wav\n[Reference]: ඔබගේ ගිණුමේ ආරක්ෂාව වැඩිදියුණු කිරීමට ද්විත්ව සත්‍යාපිරුක්ති ක්‍රමය සක්‍රිය කරන්න.\n[Hypothesis]: අපි එය කිරීම් ක"
},
{
"audio_file": "bank_sin_01_00088.wav",
"ground_truth": "අපගේ ගිණුම් කලමනාකරණ කණ්ඩායම ඔබට උදව් කිරීමට සූදානම්.",
"transcription": " අපි කිරින්න් කිරින් ක",
"log": "[Audio File]: bank_sin_01_00088.wav\n[Reference]: අපගේ ගිණුම් කලමනාකරණ කණ්ඩායම ඔබට උදව් කිරීමට සූදානම්.\n[Hypothesis]: අපි කිරින්න් කිරින් ක"
},
{
"audio_file": "bank_sin_01_00089.wav",
"ground_truth": "ඔව්, අපගේ ඇප් එක භාවිතා කර ඔබගේ ඉතිරිකිරීමේ ගිණුමට සහ ක්‍රෙඩිට් ගිණුමට මුදල් යවා ගත හැක.",
"transcription": " ப बු ब",
"log": "[Audio File]: bank_sin_01_00089.wav\n[Reference]: ඔව්, අපගේ ඇප් එක භාවිතා කර ඔබගේ ඉතිරිකිරීමේ ගිණුමට සහ ක්‍රෙඩිට් ගිණුමට මුදල් යවා ගත හැක.\n[Hypothesis]: ப बු ब"
},
{
"audio_file": "bank_sin_01_00090.wav",

"ground_truth": "ඔබගේ ගිණුමේ පරිශීලක නාමය සහ මුරපදය සැලකිලිමත්ව රහසිගතව තබන්න.",
"transcription": " අපි",
"log": "[Audio File]: bank_sin_01_00090.wav\n[Reference]: ඔබගේ ගිණුමේ පරිශීලක නාමය සහ මුරපදය සැලකිලිමත්ව රහසිගතව තබන්න.\n[Hypothesis]: අපි"
},
{
"audio_file": "bank_sin_01_00091.wav",
"ground_truth": "මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.",
"transcription": " මිශ්රියි",
"log": "[Audio File]: bank_sin_01_00091.wav\n[Reference]: මට ගෙවීම් ඉතිහාසය බලන්න පුළුවන් ක්‍රමය පැහැදිලි කර දෙන්න.\n[Hypothesis]: මිශ්රියි"
},
{
"audio_file": "bank_sin_01_00092.wav",
"ground_truth": "ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.",
"transcription": " අපි කිරින්න් කිරින්න් කිරින්න් කිරින් කිර",
"log": "[Audio File]: bank_sin_01_00092.wav\n[Reference]: ඔබගේ අන්තර්ජාල ගිණුම් පරිශීලක පොතේ, 'ගනුදෙනු ඉතිහාසය' පිටුව කියවා බලන්න.\n[Hypothesis]: අපි කිරින්න් කිරින්න් කිරින්න් කිරින් කිර"
},
{
"audio_file": "bank_sin_01_00093.wav",
"ground_truth": "කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත.",
"transcription": " यटियन किवहोद, उभगे गिनुमत्त सियलुम विश्वास सहा आरक्षक क्रमस अकस्करेत्थ",

"log": "[Audio File]: bank_sin_01_00093.wav\n[Reference]: කෙටියෙන් කිවහොත්, ඔබගේ ගිණුමට සියලුම විශ්වාස සහ ආරක්ෂක ක්‍රම සකස් කර ඇත.\n[Hypothesis]: यटियन किवहोद, उभगे गिनुमत्त सियलुम विश्वास सहा आरक्षक क्रमस अकस्करेत्थ"
},
{
"audio_file": "bank_sin_01_00094.wav",
"ground_truth": "මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?",
"transcription": " මිශ්රින් කිරීම කිරි",
"log": "[Audio File]: bank_sin_01_00094.wav\n[Reference]: මගේ ක්‍රෙඩිට් අනුපාතය වැඩිදුර නංවා ගැනීමට උපකාරයක් ලැබේද?\n[Hypothesis]: මිශ්රින් කිරීම කිරි"
},
{
"audio_file": "bank_sin_01_00095.wav",
"ground_truth": "ඔබගේ ගෙවීම් නිසි වේලාවට කිරීමට හැකිනම් ක්‍රෙඩිට් අනුපාතය ඉහළ යනු ඇත.",
"transcription": " අපි",
"log": "[Audio File]: bank_sin_01_00095.wav\n[Reference]: ඔබගේ ගෙවීම් නිසි වේලාවට කිරීමට හැකිනම් ක්‍රෙඩිට් අනුපාතය ඉහළ යනු ඇත.\n[Hypothesis]: අපි"
},
{
"audio_file": "bank_sin_01_00096.wav",
"ground_truth": "ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.",
"transcription": " අපි එය කිරින්න් කිරින් කිර් ක",
"log": "[Audio File]: bank_sin_01_00096.wav\n[Reference]: ඔබගේ ගිණුම් සුරක්ෂිතභාවය තහවුරු කිරීම සඳහා ඊමේල් තහවුරු කිරීම අවශ්‍යයි.\n[Hypothesis]: අපි එය කිරින්න් කිරින් කිර් ක"

},
{
"audio_file": "bank_sin_01_00097.wav",
"ground_truth": "අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්‍යතාවන් සලකා බලයි.",
"transcription": " अपके मूल्ली अपदेशिकिं अपबगे आयोजन अवश्षतावान सल्काबलाई",
"log": "[Audio File]: bank_sin_01_00097.wav\n[Reference]: අපගේ මූල්‍ය උපදේශකයින් ඔබගේ ආයෝජන අවශ්‍යතාවන් සලකා බලයි.\n[Hypothesis]: अपके मूल्ली अपदेशिकिं अपबगे आयोजन अवश्षतावान सल्काबलाई"
},
{
"audio_file": "bank_sin_01_00098.wav",
"ground_truth": "ඔබේ ණය ගෙවීම් දිගු කර ගැනීම සඳහා මූල්‍ය උපදේශකයෙකු සම්බන්ධ කර ගන්න.",
"transcription": " මිශ්රින්න් කිරියි යි",
"log": "[Audio File]: bank_sin_01_00098.wav\n[Reference]: ඔබේ ණය ගෙවීම් දිගු කර ගැනීම සඳහා මූල්‍ය උපදේශකයෙකු සම්බන්ධ කර ගන්න.\n[Hypothesis]: මිශ්රින්න් කිරියි යි"
},
{
"audio_file": "bank_sin_01_00099.wav",
"ground_truth": "ඔබට මුරපදය අමතක වී ඇත්නම් කරුණාකර ලඟම බැංකු වෙත යන්න.",
"transcription": " අපි කිරීම කිර",
"log": "[Audio File]: bank_sin_01_00099.wav\n[Reference]: ඔබට මුරපදය අමතක වී ඇත්නම් කරුණාකර ලඟම බැංකු වෙත යන්න.\n[Hypothesis]: අපි කිරීම කිර"
},
{
"audio_file": "bank_sin_01_00100.wav",
"ground_truth": "ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිබෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.",
"transcription": " අපි කිරීම

කිරීම් කිරීම් කිරීම් කිරීම් කිරීම් කිර",

"log": "[Audio File]: bank_sin_01_00100.wav\n[Reference]: ඔබගේ බැංකු කාඩ්පත නැතිවී ඇත්නම් කරුණාකර පාරිභෝගික සේවා සහාය ඉක්මනින්ම ලබාගන්න.\n[Hypothesis]: අපි කිරීම් කිර"

}

]

}