Robust Speech Recognition System for Malayalam

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

in

Artificial Intelligence and Data Science

by

Kurian Benoy

(Roll No. 2021MCS120014)

to

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

KOTTAYAM-686635, INDIA

December 2023

DECLARATION

I, **Kurian Benoy** (**Roll No: 2021MCS120014**), hereby declare that, this report entitled **Robust Speech Recognition System for Malayalam** submitted to Indian Institute of Information Technology Kottayam towards partial requirement of **Master of Technology** in **Artificial Intelligence and Data Science** is an original work carried out by me under the supervision of **Dr. Manu Madhavan** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **Kurian Benoy**

December 2023

CERTIFICATE

This is to certify that the work contained in this project report entitled **Robust Speech Recognition System for Malayalam** submitted by **Kurian Benoy** (**Roll No: 2021MCS120014**) to Indian Institute of Information Technology Kottayam towards partial requirement of **Master of Technology** in **Artificial Intelligence and Data Science** has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635 (Dr. Manu Madhavan)

December 2023 Project Supervisor

ABSTRACT

This project focuses on developing an efficient, open-source automatic speech recognition (ASR) tool tailored to the Malayalam language. The goal is to construct a robust ASR system that can transcribe long-form audio accurately, with timestamps, while being adept at recognizing language nuances and various accents.

There are three primary objectives for this project. The first is to create an open-source ASR for Malayalam that allows open access to methodologies, datasets, model architectures, and source. The open-source ASR models are aiming to attain a word error rate (WER) of 0.15, which is close to human level performance. Secondly, the project aims to address a gap in the field of transcribing extended long-form audio content, which is a vital component in generating subtiltles for academic lectures, films, interviews etc. Lastly, the project seeks to establish a benchmark for assessing the performance of various Malayalam speech-to-text ASR models.

Open-source ASR models was created by fine-tuning Whisper model and using quantization techniques. The model weights have been released in open source via huggingface and a demo is available to try out our model in at https link. A benchmarking toolkit was created in Python, which helped in evaluating 17 ASR models in datasets like SMC MSC and Common Voice dataset. More ASR models will be assessed in the upcoming phase.

Upon concluding the initial phase of the project, the open-source ASR model has been developed with a word error rate of less than 0.15, as targeted and 17 ASR models have been evaluated. This achievement is a promising step towards enabling transcription of extended long-form audio speeches

and initial first steps has been taken in that objective as well. We hope our work will help advance research in ASR for Indic languages.

Contents

List of Figures **ix**

List of Tables **x**

1 Introduction **1**

1.1 Introduction to ASR 1

1.2 Challenges in Malayalam Speech-to-Text Processing 2

1.2.1 What is Long-form and short-form audios 3

1.3 What is meant by Robust speech recognition? 3

1.4 Motivation 4

1.5 Evaluating ASR Metrics 5

1.6 Outline of thesis 6

2 Literature Review **7**

2.1 Associated ASR Models in Malayalam 7

2.1.1 Hidden Markov Model-Based ASR for Malayalam . . . 7

2.1.2 Hybrid ASR for Malayalam 8

2.1.3 Multi-Lingual ASR 8

2.2 ASR Models in other Indian Languages 9

2.3 Benchmarking of ASR in English 11

3 Problem Definition and Objectives 13

3.1 Problem Definition 13

3.2 Project Objectives 14

4 Methodology 16

4.1 Open-Source ASR System 16

4.1.1 Fine-tuning Whisper 17

4.1.2 Quantization ASR 20

4.2 Supporting Long-Form Audio Speech Transcription 20

4.3 Benchmarking ASR Models 23

4.3.1 ASR Dataset in Malayalam 23

4.3.2 Approach for Benchmarking 24

5 Results and Discussion 26

5.1 Development of an Open-Source ASR System: 26

5.1.1 Model Weights Publication 27

5.1.2 Demos 28

5.2 Long-Form Audio Speech Transcription Support 29

5.3 Experimental Setup 30

5.4 Benchmarking of Various Malayalam ASR Models 31

5.4.1 Benchmarking Results in Common Voice Dataset . . . 32

5.4.2 Benchmarking Results in SMC MSC Dataset 35

6 Conclusion and Future Plans 38

Bibliography **40**

List of Figures

1.1 Phases of Automatic Speech Recognition from [1]. 2

4.1 Whisper Architecture from [2]. 18

4.2 Our novel approach inspired by [3] VAD Cut and Merge algorithm which supports long-form transcription of ASR for any model length. We introduced this approach for IndicSubtitler Demo website. 21

5.1 Demo interface for executing automatic speech recognition. . . 28

5.2 Code to get timestamps using WhisperX for Malayalam 30

5.3 Output of long-form transcription using WhisperX 30

5.4 Output generated with benchmarking library in the Common-Voice dataset. 32

5.5 WER in the common voice-11 dataset. 33

5.6 CER in the common voice-11 dataset. 34

5.7 Output generated with benchmarking library in the SMC MSC dataset. 35

5.8 WER in the SMC MSC dataset. 36

5.9 CER in the SMC MSC dataset. 37

List of Tables

2.1 Summary of Literature Survey 12

4.1 Architecture of various types of Whisper models 18

4.2 Dimension table of Whisper-small architecture 19

Chapter 1

Introduction

1.1 Introduction to ASR

Automatic Speech Recognition (ASR) is a technology that converts human speech into text using advanced algorithms and processing techniques. It has become an essential tool with a wide range of applications, from enabling hands-free interactions with mobile devices to transcribing spoken language into written form. While ASR has made significant strides in major languages like English, Chinese, and Spanish, developing ASR systems for under-resourced languages such as Malayalam presents unique challenges due to increased complexity and colloquial language issues.

Speech recognition research has experienced significant growth [4] thanks to the influence of military, academic, and commercial sectors. Tech giants like Google, Amazon, Facebook, and Apple have developed technologies such as Google Assistant, Amazon Alexa, and Siri, which utilize automatic speech recognition systems to convert voice input into text output in various phases

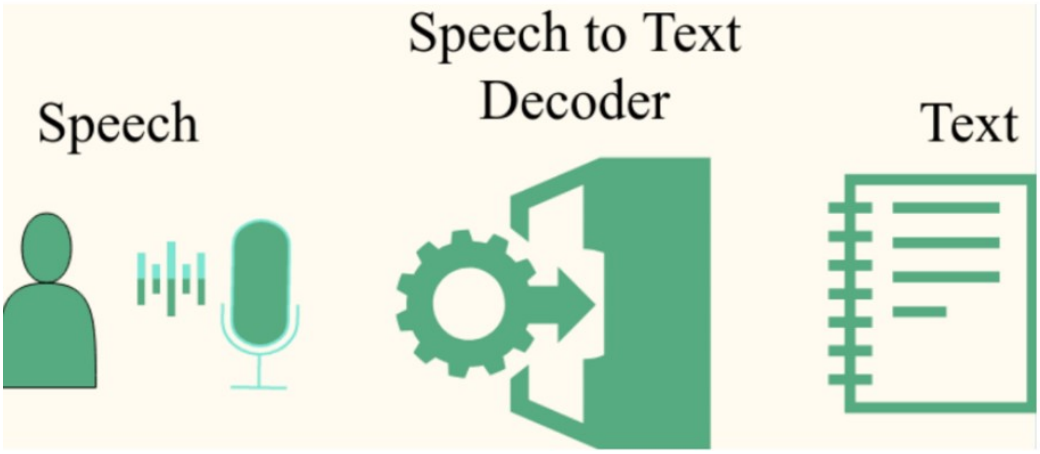


Figure 1.1: Phases of Automatic Speech Recognition from [1].

as shown in Figure 1.1. These companies share a common goal of making ASR more human-like.

1.2 Challenges in Malayalam Speech-to-Text Processing

This project aims to bridge the gap between technological advancements and linguistic diversity by focusing on the development of robust and accurate ASR systems for Malayalam. While major languages like English, Mandarin, and Spanish have received significant attention in ASR research and development, languages like Malayalam present distinct challenges due to their unique phonetic and morphological complexity [5].

Developing ASR systems for Malayalam is hindered by several obstacles. One significant challenge involves the lack of comprehensive comparative evaluations of existing ASR models for Malayalam. Most analyses are

limited to private datasets, making it difficult to accurately gauge the performance of different models. Additionally, the absence of an open-source methodology poses a substantial obstacle, hindering the systematic identification of suitable datasets and model architectures for Malayalam ASR. Furthermore, current ASR techniques for Malayalam do not cater specifically to the transcription of long-form audio with timestamps, a crucial requirement for applications such as transcribing lectures and interviews.

1.2.1 What is Long-form and short-form audios

For this project, long-form audio is defined as audio with at least a duration of 10 minutes and short-form audio as audio with duration less than 10 minutes.

1.3 What is meant by Robust speech recognition?

Robust speech recognition aims to accurately transcribe speech in various conditions, such as noisy environments or with speakers of different accents, by developing systems that can adapt to these challenges. In languages like English, in each region like US, India and Singapore people speak in different accents which can be quite difficult for ASR to recognize. Another aspect of robustness which is being planned is to support both long-form and short-form duration audio.

1.4 Motivation

- In Malayalam language, at the moment there are not any automatic speech recognition models which support long-form audio speech transcription, addressing the specific requirements for transcribing extended spoken content with timestamps. This is an essential component in creating subtitles for academic lectures, interviews, movies, serials etc.

- Even though there has been a lot of works in Malayalam Speech to text. They aren't open-source most of the time. This means leveraging open-source methodologies, the system intends to provide access to datasets, model architectures, and algorithms, promoting transparency, reproducibility, and collaboration in the development of Malayalam ASR technology. The open-source ASR is aiming to reach somewhat close to human-level reported accuracy of transcription(which is 0.7-0.10 WER according to [2]).

- Lot of works claim to have achieved 90 percentage accuracy in datasets, even in datasets which are not available in the public domain and kept properietary. Yet an apple to apple comparison will only ensure that whether model A or model B is better for Malayalam speech. This is where benchmarking to identify the true performance of any speech recognition model.

1.5 Evaluating ASR Metrics

ASR systems are evaluated by comparing the output speech (hypothesis) to the actual speech (ground truth). The evaluation metrics used to calculate ASR performance which are suitable [6] are as follows:

Word Error Rate (WER)

The WER is calculated using the following formula:

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

where:

- S represents the number of substitutions (words in the reference that are incorrectly replaced),
- D represents the number of deletions (words in the reference that are missing in the hypothesis),
- I represents the number of insertions (extra words in the hypothesis that are not in the reference),
- N represents the total number of words in the reference.

Character Error Rate (CER)

The CER is calculated using the following formula:

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

where:

- S represents the number of substitutions (characters in the reference that are incorrectly replaced),
- D represents the number of deletions (characters in the reference that are missing in the hypothesis),
- I represents the number of insertions (extra characters in the hypothesis that are not in the reference),
- N represents the total number of characters in the reference.

Both WER and CER aim to minimize the error rate, with a lower error rate indicating higher accuracy. These metrics provide a quantitative measure of system performance by comparing the system's output to a reference. They assist researchers and practitioners in assessing the quality of automatic recognition systems.

1.6 Outline of thesis

In the upcoming chapters, we will be going through the related works and then defined the problem statement and problem objectives of the thesis. We will discuss the methodology for solving each of the problem objectives and discuss the associated results with respect to achieving each of the defined project objectives.

Chapter 2

Literature Review

2.1 Associated ASR Models in Malayalam

Investigation into the pertinent literature revealed the following exemplary variants of Automatic Speech Recognition (ASR) systems developed for the Malayalam language:

- Models based on Hidden Markov Model
- Hybrid ASR models for Malayalam
- Multi-lingual ASR systems

2.1.1 Hidden Markov Model-Based ASR for Malayalam

Leveraging the capabilities of the Hidden Markov Model (HMM), Cini et al. [7] established that the Malayalam speech recognition of numerical content

is viable when trained over a corpus encompassing 420 sentences from 21 different speakers. Building on this, Anuj et al. [8] utilized the combinations of HMM and ANN to accomplish Malayalam speech recognition. In their respective internal test sets, [7] and [8] reported word-recognition accuracies of 91% and 86.67%.

2.1.2 Hybrid ASR for Malayalam

Kavya et al. [9] put forth an open vocabulary speech recognition system in Malayalam. Their strategy involved the construction of a hybrid ASR model that integrated an acoustic model ASR with one formed using language model and pronunciation lexicon.

The study scrutinized the Word Error Rate (WER) across moderate to large Out of Vocabulary (OOV) test sets of open source nature and posited a 10 to 7% enhancement over mere usage of an acoustic ASR system.

2.1.3 Multi-Lingual ASR

Several ASR systems originally tailored for multiple languages also extend compatibility with Malayalam. For instance, Alec et al. [2] utilized an encoder-decoder based model supporting speech recognition across 99+ languages. For navigating the Malayalam subset in the Common Voice 9 dataset, they reported a WER of 103.2 using a large-v2 model. Simultaneously, the CTC model deployed by Vineel et al. [10] supporting 1000+ languages reported a WER of 39.7 with the MMS L-1107 no LM checkpoint while analyzing the Malayalam subset in the Fleurs dataset. Both [2] and [10] provide facilities for fine-tuning of these models. Zhang et al. [11] trained a multi-

lingual ASR model based on YouTube audio data and successfully scaled to 100 language. Another new model named SeamlessM4T [12] from Meta is a Massively Multilingual and Multimodal Machine Translation—a single model that supports speech-to-speech translation, speech-to-text translation, text-to-speech translation, text-to-text translation, and automatic speech recognition for up to 100 languages. To build this, they used 1 million hours of open speech audio data to learn self-supervised speech representations with w2v-BERT 2.0 architecture. It comes with various model families like SeamlessM4T medium, large and large-v2.

2.2 ASR Models in other Indian Languages

Cross Lingual Speech Representations for Indic Languages, presents a self-supervised learning-based audio pre-trained model, CLSRIL-23 [13], which learns cross lingual speech representations from raw audio across 23 Indic languages. The model is built on top of wav2vec 2.0 and is trained on almost 10,000 hours of audio data. The paper compares the performance of multilingual pretraining with monolingual pretraining and shows that multilingual pretraining outperforms monolingual training in terms of learning speech representations that encode phonetic similarity of languages. The model's performance on downstream fine-tuning tasks for speech recognition is also evaluated, demonstrating a decrease of 5% in Word Error Rate (WER) and 9.5% in Character Error Rate (CER) when a multilingual pretrained model is used for finetuning in Hindi.

The second paper, "End-to-End Speech Recognition of Tamil Language"

[14] investigates the use of open-source speech recognition toolkits to build a speech recognition model for the Tamil language. The paper discusses the challenges of developing a corpus for under-resourced languages and presents the first results of developing a Tamil model using DeepSpeech. It also highlights the release of the trained Tamil ASR model and the training sets in the public domain. The paper aims to demonstrate a cost-effective approach for under-resourced languages in a financially constrained environment.

Towards Building ASR Systems for the Next Billion Users presents a significant contribution to the development of automatic speech recognition (ASR) systems for low-resource languages from the Indian subcontinent. The authors curated 17,000 hours of raw speech data for 40 Indian languages from various domains and used this data to pretrain several variants of wav2vec style models for 40 Indian languages. They analyzed the pretrained models and found that multilingual pretraining is an effective strategy for building ASR systems for the linguistically diverse speakers of the Indian subcontinent. The paper also discusses the fine-tuning of the model for downstream ASR for 9 languages, achieving state-of-the-art results on 3 public datasets, including very low-resource languages such as Sinhala and Nepali. The work demonstrates the effectiveness of multilingual pretraining in building ASR systems for the diverse languages of the Indian subcontinent [15]. The paper's findings are significant as they address the challenge of developing high-quality ASR models for a large and diverse pool of languages, particularly low-resource languages. The authors' approach of multilingual pretraining and fine-tuning has shown promising results, even for very low-resource languages. The availability of the curated speech data, pretrained models, and

state-of-the-art results on public datasets is a valuable contribution to the research and development of ASR systems for the Indian subcontinent, and it is hoped that this work will advance research in ASR for Indic languages

2.3 Benchmarking of ASR in English

In English, there exist benchmark models such as SUPERB [16]. It offers a unique framework for benchmarking speech processing models across an array of tasks including speaker verification, emotion detection, speech recognition, etc. It welcomes researchers to participate and share their results by providing a challenge and a leaderboard along with a benchmarking toolkit, thereby propelling the research frontier in representation learning and general speech processing.

The paper "ESB: A Benchmark For Multi-Domain End-to-End Speech Recognition" [17] introduces the End-to-end Speech Benchmark (ESB) aimed at evaluating the performance of a single automatic speech recognition (ASR) system across a broad spectrum of speech datasets. The speech datasets where specifically classified as Narrated, Spontaneous and Oratory types with each spectrum used for benchmarking. By specifically demonstrating how a unified speech system can be applied and evaluated across a wide range of data distributions, it establishes that E2E systems can approach within 2.6% of the performance of SoTA systems that are acutely tuned to a specific dataset.

Category	Model/Paper	Key Finding
ASR Models in Malayalam	Cini et al. [7]	Malayalam Numerical ASR is viable with HMM. Word-accuracy of 91%.
	Anuj et al. [8]	Used combinations of HMM and ANN. Word-accuracy of 86.67%.
	Kavya et al. [9]	Hybrid ASR model for open vocabulary speech recognition. Improved WER by 10-7%.
	Vineel et al. [10]	CTC model with a WER of 39.7 in Fleurs dataset.
	Alec et al. [2]	Whisper is an encoder-decoder based model with a WER of 103.2 in Common Voice 9 dataset.
	Barrault et al. [12]	A massively multilingual and multi-modal machine translation as a single model that supports tasks like speech-to-speech translation, speech-to-text translation, text-to-speech translation, text-to-text translation, and automatic speech recognition for up to 100 languages. It works in Malayalam but WER is not reported in paper.
ASR Models in Other Indian Languages	CLSRIL-23 [13]	Multilingual pretraining improves speech representations, Decreases WER by 5% and CER by 9.5% in Hindi.
	End-to-End Tamil ASR [14]	Cost-effective approach to build an ASR using Deep Speech for Tamil.
	Javed et al. [15]	Curated 17,000 hours of data for 40 Indian languages, achieved state-of-the-art results in low-resource languages.
Benchmarking ASR in English	SUPERB [16]	Framework for benchmarking speech processing models.
	ESB [17]	Evaluates performance of ASR systems across a broad spectrum of speech datasets.

Table 2.1: Summary of Literature Survey

Chapter 3

Problem Definition and Objectives

After going through the related research during the previous statements certain issues was identified, which are the basis of creation of this project.

3.1 Problem Definition

1. At the moment there is no ASR models which support long-form audio transcription in Malayalam which is a problem.

2. Even though there has been a lot of works in Malayalam Speech to text. They aren't open-source most of the time and there are not studies to compare these models performance one over the other in terms of metrics as it's not evaluated in same scale.

3.2 Project Objectives

The primary objectives of this project are as follows:

1. **Develop an Open-Source ASR System:** The project aims to design and implement an open-source ASR system for Malayalam that overcomes the limitations of existing speech-to-text techniques. By leveraging open-source methodologies, the system intends to provide access to datasets, model architectures, and algorithms, promoting transparency, reproducibility, and collaboration in the development of Malayalam ASR technology. It should achieve a key goal of the project is to achieve a Word Error Rate (WER) of less than 0.15 in the developed ASR system for speech to text model accuracy. We choose a target of 0.15 WER to reach close to human-level reported accuracy of transcription which at the moment on studies conducted in [2] paper is between 7.5 to 10 % WER.

2. **Support Long-Form Audio Speech Transcription:** In addressing the dearth of specialized provisions for transcribing long-form audio with timestamps in Malayalam, the project endeavors to develop features and capabilities that cater to the specific requirements of transcribing extended spoken content.

3. **Benchmark Various ASR Models:** The project seeks to compare and benchmark multiple ASR models to evaluate their performance in the context of Malayalam speech-to-text processing. By conducting systematic comparisons, the project aims to identify the strengths and

limitations of different ASR methodologies, leading to insights that can inform the selection of appropriate models for specific use cases.

Chapter 4

Methodology

4.1 Open-Source ASR System

The objective of our project was to create an open-source model with the aim of achieving state-of-the-art results in Malayalam speech recognition. To accomplish this, the project utilized the method of fine-tuning end-to-end speech recognition models, such as Whisper [2], in order to yield impressive outcomes and quantization. The general procedure for fine-tuning an ASR base model is as follows:

1. Gather a dataset for training the ASR model.
2. Choose a fitting initial architecture.
3. Train the selected model.
4. Evaluate the functioning of the model and repeat the training process if necessary.

4.1.1 Fine-tuning Whisper

Whisper functions as a Transformer-based encoder-decoder model as shown in Figure 4.1 and is also be referred to as a sequence-to-sequence model. It converts a sequence of audio spectrogram features into a text token sequence. Initially, the raw audio inputs are transformed into a log-Mel spectrogram by the feature extractor. Following this, the Transformer encoder encodes the spectrogram into a sequence of hidden encoder states. Ultimately, the decoder predicts text tokens in an autoregressive manner, taking into account both prior tokens and the hidden encoder states. It supports 100 languages and has the first token as language recognized in audio, followed by the task and text predicted.

Whisper model consists of six different model types:

- Tiny
- Base
- Base
- Medium
- Large

In December 2022, an improved large model named large-v2, and large-v3 in November 2023 was released. The architecture details of various model types are:

The model dimension table for Whisper-small architecture is listed as below:

Models	Heads	Width	Parameters
Tiny	4	384	39M
Base	6	512	74M
Small	12	768	244M
Medium	24	1024	769M
Large	32	1280	1.5B

Table 4.1: Architecture of various types of Whisper models

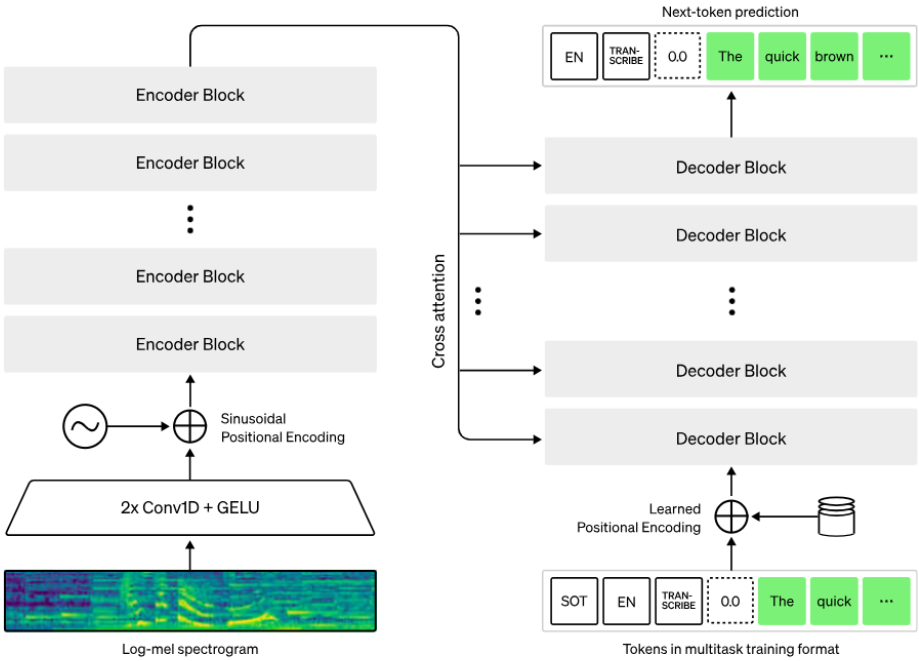


Figure 4.1: Whisper Architecture from [2].

Layer Name	**Output Dimension**
Conv1D	80 x 768
Conv1D	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperEncoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
WhisperDecoder	768 x 768
Linear	768 x 51865

Table 4.2: Dimension table of Whisper-small architecture

The steps involved in the fine-tuning process are:

1. Load the dataset.
2. Preprocess the dataset, and prepare the feature_extractor and tokenizer.
3. Train the selected model.
4. Evaluate the functioning of the model and repeat the training process if necessary.

When fine-tuning the model, the architecture remains same but only the model's weights change based on the input data it's being fine-tuned upon.

4.1.2 Quantization ASR

Using quantization, it's possbile to optimize one of the best available ASR model for efficiency and high performance. The Whisper [2] models supports 'int8float16', float16, int8 and int6 quantization formats, ensuring efficient processing and transcription of speech data without compromising accuracy much using faster-whisper [18] framework.

4.2 Supporting Long-Form Audio Speech Transcription

According to Max et al. [3], the support for long-form audio transcription is feasible, and can accommodate a multitude of languages, including English,

Chinese, and French. This approach is proposed to be effective for Malayalam, provided adequate number of Malayalam base models is created.

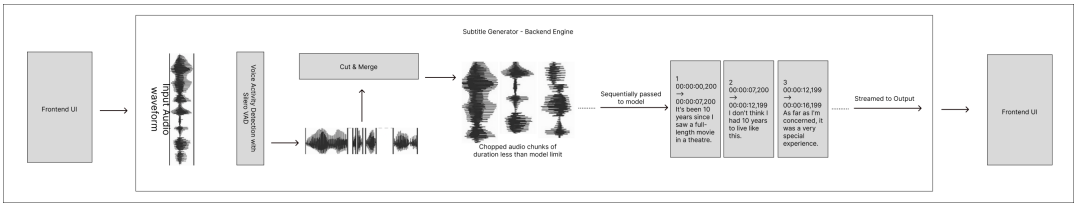


Figure 4.2: Our novel approach inspired by [3] VAD Cut and Merge algorithm which supports long-form transcription of ASR for any model length. We introduced this approach for IndicSubtitler Demo website.

Our Novel approach for handling long-form speech transcription for any ASR models

Voice Activity Detection (VAD), also recognized as Speech Activity Detection or merely Speech Detection, delineates the technique of distinguishing whether human speech is present or absent within an audio segment. VAD stands as a critical component across a spectrum of audio processing applications, ranging from telecommunication setups to voice-activated interfaces, significantly contributing to the minimization of computational and resource expenditures by prioritizing segments that contain speech.

The primary objective of VAD centers around segregating speech from non-speech portions in an audio signal, which may encompass noise, music, or silence. The advent of proficient VAD algorithms is vital for augmenting the efficiency of various subsequent processes like speech recognition, voice encoding, and surveillance frameworks, among others.

A pivotal source of inspiration for this novel approac was [3] paper, which

was particularly advocating for extended-duration audio transcription, derived from examining the VAD Cut and Merge Algorithm discussed in the WhisperX [3] paper. WhisperX paper[3] introduces an algorithm where the lengths of active speech segments can vary significantly, being either substantially shorter or longer than the model time limit of an Automatic Speech Recognition (ASR) model. To mitigate this challenge, a minimum cut operation is implemented during the binary post-processing smoothing phase to constrain the duration of active speech segments. Furthermore, a merging technique is suggested, whereby neighboring segments, subsequent to the min-cut operation, are amalgamated to ensure they do not exceed the maximal input duration permissible by the ASR model. For a deeper understanding, sections 2.1 and 2.2 in the referenced WhisperX paper[3] offer comprehensive insights.

We split the audio with VAD Cut and Merge algorithm discussed above and pass sequentially to the ASR model. This was as shown in Figure 4.2 the results can be shown in a streaming manner very quickly rather than waiting for the whole audio to be completely processed. This helps in It introducing innovative concepts such as a Generative UI for ASR which was using in IndicSubtitler, inspired by ChatGPT[19], commencing the subtitling process within the first 15 seconds instead of the typical 5-10 minutes processing time for an hour-long audio.

4.3 Benchmarking ASR Models

Research papers such as ESB [17] emphasize that the type of data chosen for Malayalam is crucial when benchmarking ASR models. For a proper ASR evaluation dataset, it should include the speaking style, such as the Narrated, Oratory, or Spontaneous formats. However, the only speaking style available in Malayalam was the Narrated Speech, which was collected either in a studio environment or natural settings, typically crowd-sourced.

4.3.1 ASR Dataset in Malayalam

The following data sets were available for being use in Open-source during our research:

- Open SLR 63 - This is a crowdsourced high-quality Malayalam multi-speaker speech dataset containing transcribed audio of Malayalam sentences recorded by volunteers.
- Indic TTS, IITM - The Indic TTS Speech Corpus by IITM is a special corpus of Indian languages, including Malayalam, covering 13 major languages of India. It comprises over 10,000 spoken sentences/utterances for each language.
- IMaSC(ICFOSS Malayalam Speech Corpus) dataset - IMaSC is a Malayalam text and speech corpus containing approximately 50 hours of recorded speech with 8 speakers. The corpus contains 34,473 text-audio pairs of Malayalam sentences spoken by 8 speakers, totalling in approximately 50 hours of audio.

- GMaSC dataset - The corpus contains 2,000 text-audio pairs of Malayalam sentences spoken by 2 speakers, totalling in approximately 139 minutes of audio. Each sentences has at least one English word common in Malayalam speech.

- SMC Malayalam Speech Corpus - This test set consists of about 1500+ audio which is collected as crowd-sourced dataset spoken by 50+ speakers totalling in approximately 1 hour 30 minutes dataset.

- Mozilla Common Voice - Mozilla Common Voice is a multilingual dataset that includes speech data for various languages, including Malayalam.

- AI4Bharath Kathbath - Kathbath is an human-labeled ASR dataset containing 1,684 hours of labelled speech data across 12 Indian languages from 1,218 contributors located in 203 districts in India.

Given the availability of many ASR datasets, the datasets SMC Malayalam Speech Corpus dataset and the CommonVoice dataset are choosen due to their adherence to the narrated style and their access to a broad array of speakers. Constraints in available datasets forced us to go with a single speaking style.

4.3.2 Approach for Benchmarking

The approach included:

1. Establishment as a Python library for further benchmarking of whisper-based transformer models.

2. Conducting calculations for WER, CER, model size, and the time required to benchmark the model on selected datasets.

3. Development of a reproducible methodology so the results of benchmarking can be saved as a dataset.

Chapter 5

Results and Discussion

5.1 Development of an Open-Source ASR System:

The outcomes of the final year project significantly align with the initially established objectives. The designed and developed open-source ASR system for Malayalam resulted in the formulation of two models: "Whisper-small-ml-gmasc" and "Whisper-small-ml-imasc." These models were evaluated using both the Common Voice dataset and the MSC dataset. Particularly, the latter model achieved a Character Error Rate (CER) of 12.84 and a Word Error Rate (WER) of 24.83 in the Common Voice dataset. Likewise, it demonstrated a CER of 14.64 and a WER of 27.28 in the MSC dataset. This achievement indicates the success of the ASR system in achieving a CER of less than 0.15.

The model "vegam-whisper-medium-ml" presents a significant improve-

ment within the Open-source ASR build using quantization ASR techniques. This model represents a rapid version of Malayalam ASR, designed for efficiency and high performance. Optimized for speed and functionality, the "vegam-whisper-medium-ml" model supports 'int8float16', float16, int8 and int6 quantization formats, ensuring efficient processing and transcription of speech data without compromising accuracy.

5.1.1 Model Weights Publication

The model weights were published on Hugging Face for the below models in following links:

1. Whisper-small-ml-imasc - https://huggingface.co/kurianbenoy/whisper-small-ml-imasc
2. Whisper-small-ml-gmasc - https://huggingface.co/kurianbenoy/whisper-small-ml-gmasc
3. Vegam Whisper Medium ML - https://huggingface.co/kurianbenoy/vegam-whisper-medium-ml
4. Vegam Whisper (FP16 model only) - https://huggingface.co/kurianbenoy/vegam-whisper-medium-ml-fp16
5. Vegam Whisper (INT8 model only) - https://huggingface.co/kurianbenoy/vegam-whisper-medium-ml-int8
6. Vegam Whisper (INT16 model only) - https://huggingface.co/kurianbenoy/vegam-whisper-medium-ml-int16

7. Vegam Whisper (INT8_FLOAT16 model only) - https://huggingface.co/kurianbenoy/vegam-whisper-medium-ml-int8_float16

5.1.2 Demos

A user interface(UI) demo as shown in Figure 5.1 to showcase the practical operation of our model, which transcribes the speech from an audio file or input audio recording. The interface was build using gradio [20] and supports input formats by uploading file or recording voice. This demo is now hosted in huggingface spaces and can be tried out by checking the link in `https://huggingface.co/spaces/kurianbenoy/Pallakku`.

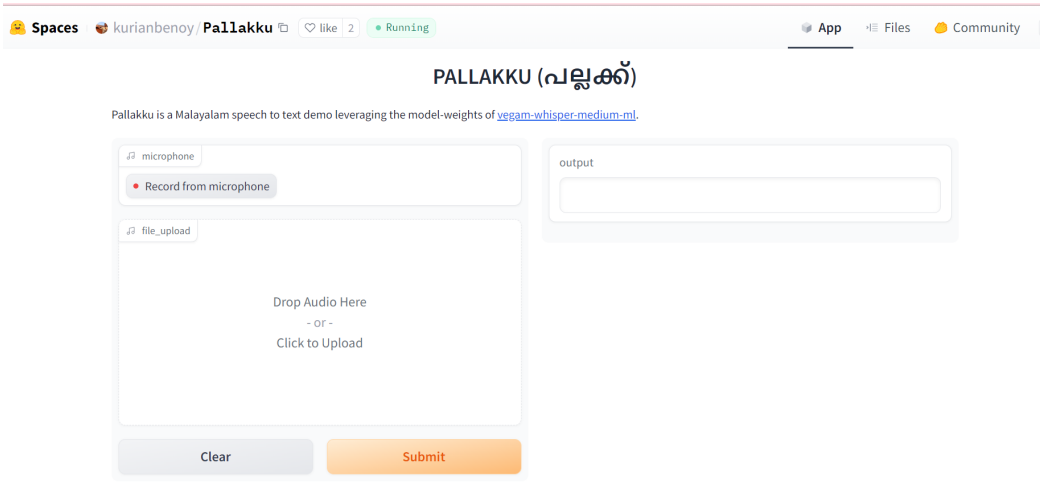
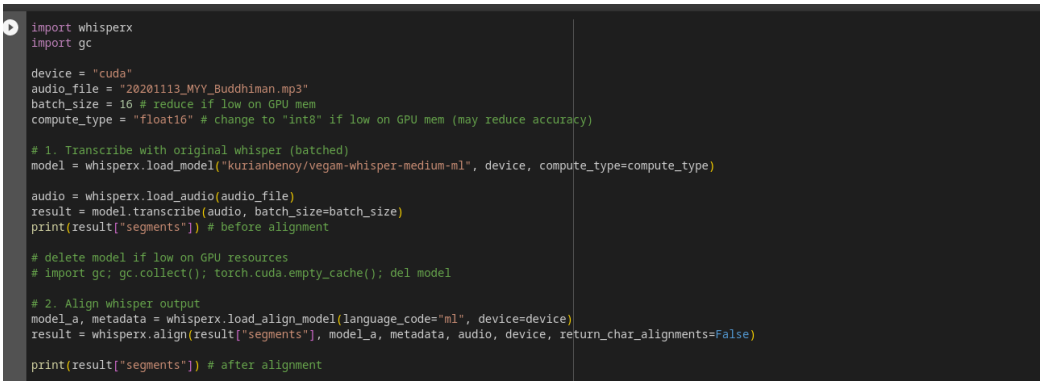


Figure 5.1: Demo interface for executing automatic speech recognition.

5.2 Long-Form Audio Speech Transcription Support

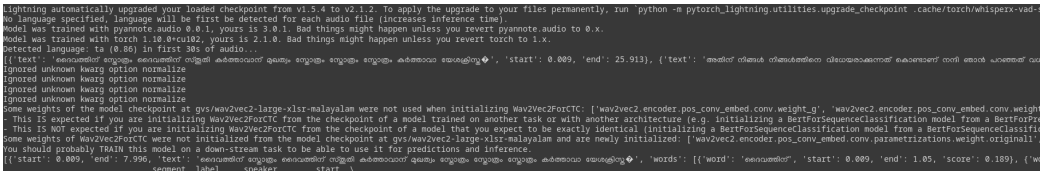
Moreover, the project advanced in the provision of long-form audio speech transcription support in Malayalam, fulfilling the specific demand for transcribing extensive spoken content. While the initial prototype for long-form communication might not be entirely functional at the moment, the project has established a foundation for accommodating this significant component, which is particularly vital for applications like academic lectures, interviews, and linguistic analysis.

The transcription of long-form audio files is facilitated using the WhisperX [3] framework by capitalizing on our own model weights. The initial code to pass our model weights with WhisperX to generate timestamps is shown in Figure 5.2. The output in Figure 5.3 shows the associated language detected as ta(tamil), which is inaccurate as Whisper language identification is not always correct. It also shows the sentence chunks with it's associated start and end time, followed by words with it's own associated start time, end time as shown in Figure 5.3.



```python
import whisperx
import gc

device = "cuda"
audio_file = "20201113_MYY_Buddhiman.mp3"
batch_size = 16 # reduce if low on GPU mem
compute_type = "float16" # change to "int8" if low on GPU mem (may reduce accuracy)

# 1. Transcribe with original whisper (batched)
model = whisperx.load_model("kurianbenoy/vegam-whisper-medium-ml", device, compute_type=compute_type)

audio = whisperx.load_audio(audio_file)
result = model.transcribe(audio, batch_size=batch_size)
print(result["segments"]) # before alignment

# delete model if low on GPU resources
# import gc; gc.collect(); torch.cuda.empty_cache(); del model

# 2. Align whisper output
model_a, metadata = whisperx.load_align_model(language_code="ml", device=device)
result = whisperx.align(result["segments"], model_a, metadata, audio, device, return_char_alignments=False)

print(result["segments"]) # after alignment
```

Figure 5.2: Code to get timestamps using WhisperX for Malayalam



```
Lightning automatically upgraded your loaded checkpoint from v1.5.4 to v2.1.2. To apply the upgrade to your files permanently, run `python -m pytorch_lightning.utilities.upgrade_checkpoint .cache/torch/whisperx-vad-
No language specified, language will be first be detected for each audio file (increases inference time).
Model was trained with pyannote.audio 0.0.1, yours is 3.0.1. Bad things might happen unless you revert pyannote.audio to 0.x.
Model was trained with torch 1.10.0+cu102, yours is 2.1.0. Bad things might happen unless you revert torch to 1.x.
Detected language: ta (0.86) in first 30s of audio...
[{'text': ' ഒൌവത്തിന് തുടക്കം ഒൌവത്തിന് തുടക്കി കർത്താവാണ് ഉലകം തുടക്കം തുടക്കം തുടക്കം കർത്താവാ യേശുക്രിസ്തു', 'start': 0.009, 'end': 25.913}, {'text': ' അതിന് നിങ്ങൾ നിങ്ങൾക്കിടെ വിലയാകാനാണ് കൊണ്ടാണ് നന്നി ഞാൻ പറഞ്ഞത് വ
Ignored unknown kwarg option normalize
Ignored unknown kwarg option normalize
Ignored unknown kwarg option normalize
Ignored unknown kwarg option normalize
Some weights of the model checkpoint at gvs/wav2vec2-large-xlsr-malayalam were not used when initializing Wav2Vec2ForCTC: ['wav2vec2.encoder.pos_conv_embed.conv.weight_g', 'wav2vec2.encoder.pos_conv_embed.conv.weigh
- This IS expected if you are initializing Wav2Vec2ForCTC from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPr
- This IS NOT expected if you are initializing Wav2Vec2ForCTC from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassifi
Some weights of Wav2Vec2ForCTC were not initialized from the model checkpoint at gvs/wav2vec2-large-xlsr-malayalam and are newly initialized: ['wav2vec2.encoder.pos_conv_embed.conv.parametrizations.weight.originalli
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
[{'start': 0.009, 'end': 7.996, 'text': ' ഒൌവത്തിന് തുടക്കം ഒൌവത്തിന് തുടക്കി കർത്താവാണ് ഉലകം തുടക്കം തുടക്കം തുടക്കം കർത്താവാ യേശുക്രിസ്തു', 'words': [{'word': 'ഒൌവത്തിന്', 'start': 0.009, 'end': 1.05, 'score': 0.189}, {'w
                        segment  label  speaker      start  e
```

Figure 5.3: Output of long-form transcription using WhisperX

5.3 Experimental Setup

Fine-tuning and benchmarking Whisper-based ASR models can be computationally intensive tasks that often require significant GPU resources. The exact requirements will depend on the size of the Whisper model variant being used (e.g., small, medium, large, or extra-large), the size of the dataset, and the desired speed of the process. For the thesis the fine-tuned models were trained on Whisper small architecture on datasets like IMaSC and GMaSC dataset. Inorder to train these models a VM with RTX 5000 GPU which has 16 GB GPU memory, the VM needed to have a storage capacity of 50GB to store the dataset and associated model metafiles was required. A GPU with at least 16 GB of VRAM, such as an NVIDIA V100, A100, or

an equivalent, would be suitable for handling most tasks. I

In-order to do benchmarking of 17 ASR models, T4 and RTX 5000 GPUs. It require a minimum of 10 GB of storage and a mid-range GPU with at least 8 GB of VRAM, such as an NVIDIA RTX 2080 or 3080, could suffice for many benchmarking tasks. The same configuration was also enough to run long-form audio ASR transcription as well.

5.4 Benchmarking of Various Malayalam ASR Models

Additionally, the project set benchmarks for various Malayalam ASR models, thereby contributing to the field by comparing and assessing the performance of different models. This process will offer valuable insights for choosing suitable models for specific use cases and will advance ASR technology within the context of the Malayalam language.

Our aim was to benchmark open-source Malayalam ASR models and document the results. The project have primarily benchmarked Whisper-based models [2]. For benchmarking the evaluation dataset, two datasets was identified:

1. Common Voice
2. SMC Malayalam Speech Corpus

5.4.1 Benchmarking Results in Common Voice Dataset

MODEL NAME	WER	CER	MODEL SIZE	TIME(second
openai/whisper-tiny	154.21	180.45	37.76M	22.277158
openai/whisper-base	118.39	131.08	72.59M	22.352587
openai/whisper-small	100.06	95.04	241.73M	25.442846
openai/whisper-medium	127.97	136.43	763.86M	53.880491
openai/whisper-large	125.73	139.62	1.54B	82.74608
openai/whisper-large-v2	100.26	93.6	1.54B	71.14292622
thennal/whisper-medium-ml	11.56	5.41	763.86M	924.979711
parambharat/whisper-tiny-ml	38.31	21.93	37.76M	59.535259
parambharat/whisper-base-ml	30.33	16.16	72.59M	96.419609
parambharat/whisper-small-ml	21.65	11.78	241.73M	273.555688
anuragshas/whisper-large-v2-ml	24.46	11.64	1.54B	1779.561592
DrishtiSharma/whisper-large-v2-malayalam	26.25	13.17	1.54B	1773.661774
kurianbenoy/whisper-small-ml-gmasc	41.12	21.24	241.73M	53.09780836
kurianbenoy/whisper-small-ml-imasc	24.27	12.84	241.73M	50.89170098
kurianbenoy/vegam-whisper-medium-ml	23.75	17.13	763.86M	115.083
kurianbenoy/vegam-whisper-medium-ml-int8	25.1	19.24	763.86M	139.30652
kurianbenoy/vegam-whisper-medium-ml-int8_float16	22.09	16.47	763.86M	117.4155335

Figure 5.4: Output generated with benchmarking library in the Common-Voice dataset.

Figure 5.4 shows the 17 ASR models results being calculated in the Common Voice Dataset. The result shows the associated WER, CER, model parameters size and time taken to obtain results. Some of the findings are:

- The models which are the fastest in terms of getting results openai/whisper-tiny and openai/whisper-base. Yet these models performance are not so good, which will be investigated further in Figures 5.5 and 5.6. Among the models which have a decent performance of less than 0.5 WER, the fastest ASR model is ‘kurianbenoy/whisper-small-ml-imasc‘.

- The models are all based on the Whisper architecture, but they vary in size and complexity. The largest model being large-v2 model architectures like DrishtiSharma and anuraghas models with 1.5 billion

parameters and smallest being openai/tiny model with a model size of 37.76 million parameters.

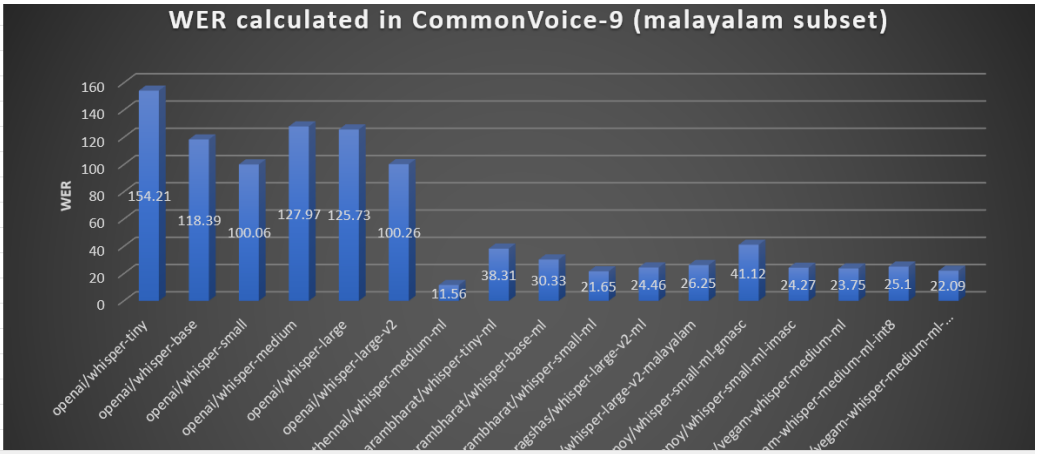


Figure 5.5: WER in the common voice-11 dataset.

Figure 5.5 shows the performance in Word Error Rate(WER) in CommonVoice the ASR models evaluated to get the performance. The findings from this figure are:

- The best WER performance is by thennal/whisper-medium-ml model with a WER of 11.56 .
- The worst WER performance is by openai/whisper-tiny.
- Out of the models produced as part of the work in the thesis, the best performance was using vegam-whisper-medium-ml model with a WER of 22.09.

Figure 5.6 shows the performance in Character Error Rate(CER) in CommonVoice the ASR models evaluated to get the performance. The findings from this figure are:

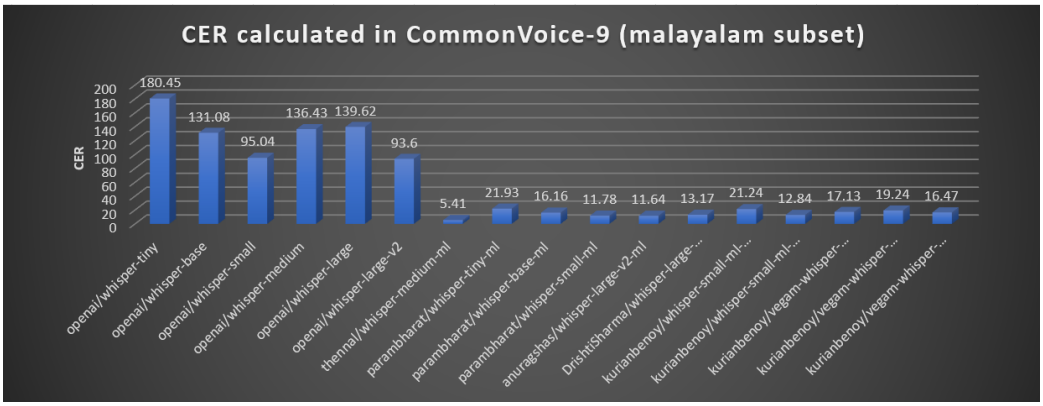


Figure 5.6: CER in the common voice-11 dataset.

- The best CER performance is by thennal/whisper-medium-ml model with a CER of 5.41 .
- The worst CER performance is by openai/whisper-tiny.
- Out of the models produced as part of the work in the thesis, the best performance was using whisper-small-ml-imasc model with a CER of 12.84.

5.4.2 Benchmarking Results in SMC MSC Dataset

Figure 5.7 shows the 17 ASR models results being calculated in the SMC MSC Dataset. The result shows the associated WER, CER, model parameters size and time taken to obtain results. Observations are:

MODEL NAME	WER	CER	MODEL SIZE	TIME(seconds)
openai/whisper-tiny	139.63	177.3	37.76M	375.532
openai/whisper-base	155.97	200.05	72.59M	448.95
openai/whisper-small	111.57	123.7	241.73M	479.736
openai/whisper-medium	101.45	104.23	763.86M	672.291
openai/whisper-large	107.01	113.62	1.54B	1067.557
openai/whisper-large-v2	100.27	102.4	1.54B	1040.25
thennal/whisper-medium-ml	2.244	1.247	763.86M	8736.731
parambharat/whisper-tiny-ml	43.96	25.78	37.76M	727.57
parambharat/whisper-base-ml	37.185	21.389	72.59M	1124.314
parambharat/whisper-small-ml	28.265	15.379	241.73M	2893.445
anuragshas/whisper-large-v2-ml	23.57	12.33	1.54B	10467.876
DrishtiSharma/whisper-large-v2-malayalam	30.53	19.81	1.54B	10067.01
kurianbenoy/whisper-small-ml-gmasc	32.07	16.89	241.73M	498.5966506
kurianbenoy/whisper-small-ml-imasc	24.27	12.9	241.73M	577.12132
kurianbenoy/vegam-whisper-medium-ml	10.7	9.38	763.86M	1512.018722
kurianbenoy/vegam-whisper-medium-ml-int8	18.92	16.91	763.86M	1647.418209
kurianbenoy/vegam-whisper-medium-ml-int8_float16	19.53	17.75	763.86M	1452.718654

Figure 5.7: Output generated with benchmarking library in the SMC MSC dataset.

- The models which are the fastest in terms of getting results openai/whisper-tiny. Yet these models performance are not so good, which will be investigated further in Figure 5.8 and 5.9. Among the models which have a decent performance of less than 0.5 WER, the fastest ASR model is ‘kurianbenoy/whisper-small-ml-gmasc‘.

- The models are all based on the Whisper architecture, but they vary in size and complexity. The largest model being large-v2 model architectures like DrishtiSharma and anuraghas models with 1.5 billion

parameters and smallest being openai/tiny model with a model size of 37.76 million parameters.

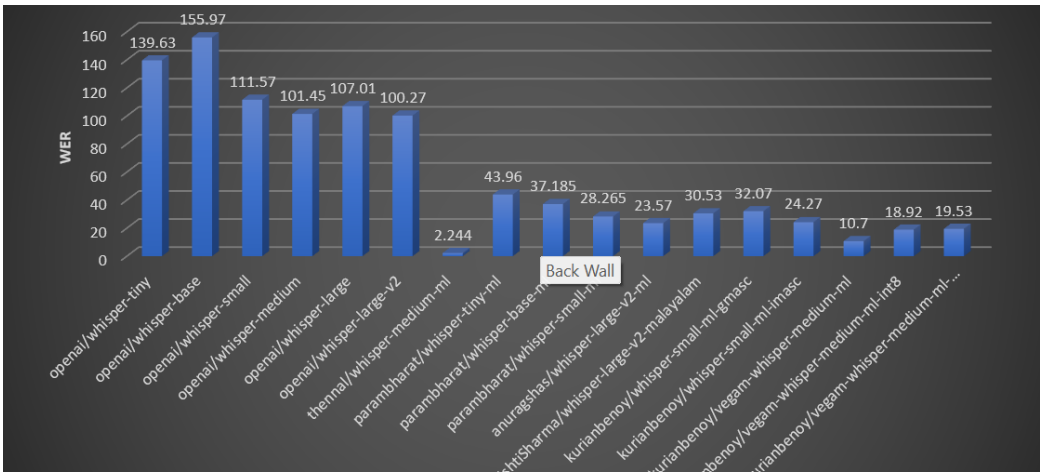


Figure 5.8: WER in the SMC MSC dataset.

Figure 5.8 shows the performance in Word Error Rate(WER) in the SMC MSC dataset among the ASR models evaluated to get the performance. The findings from this figure are:

- The best WER performance is by thennal/whisper-medium-ml model with a WER of 2.24 .
- The worst WER performance is by openai/whisper-base.
- Out of the models produced as part of the work in the thesis, the best performance was using vegam-whisper-medium-ml model with a WER of 10.7.

Figure 5.9 shows the performance in CER in the SMC MSC dataset among ASR models evaluated to get the performance. The findings from this figure are:

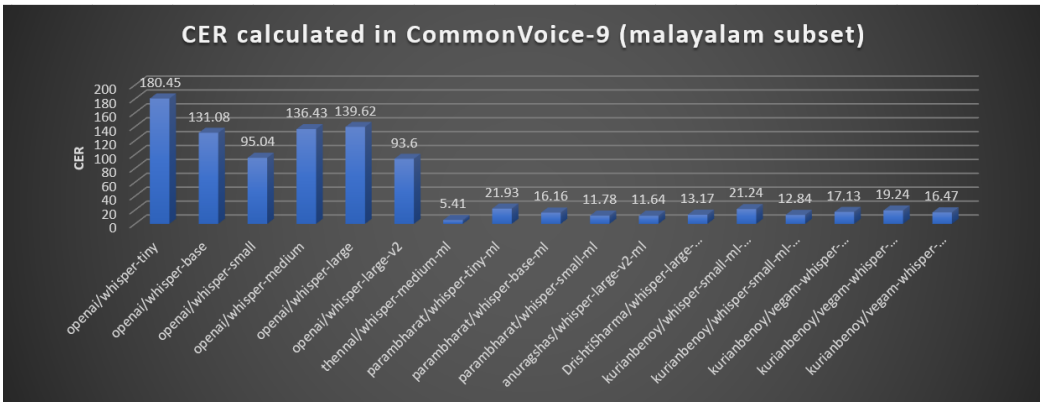


Figure 5.9: CER in the SMC MSC dataset.

- The best CER performance is by thennal/whisper-medium-ml model with a CER of 1.24 .
- The worst CER performance is by openai/whisper-base.
- Out of the models produced as part of the work in the thesis, the best performance was using vegam-whisper-medium-ml model with a CER of 9.38.

Chapter 6

Conclusion and Future Plans

At present, we have initiated work on all the three stated project objectives. Our primary aim of creating open-source Automated Speech Recognition (ASR) model weights has been achieved. The goal of obtaining a Word Error Rate (WER) of less than 0.15 with our model weight - kurianbenoy/vegam-whisper-medium-ml has been achieved in the dataset SMC MSC speech corpus. Though the same score was not attained using the Common Voice dataset, we have reached a Character Error Rate (CER) of less than 0.15. As part of the next phase, we aim to further improve these models by refining the architecture and incorporating additional data to push the boundaries of the current state of the art.

The project has embarked on the task of long-form speech transcription where the initial findings are encouraging. Much work is still required to assess the performance of long-form audio transcription and fine-tune ASR to be suitable for transcription purposes.

We have successfully benchmarked 17 ASR models using our benchmark-

ing tool and aim to construct a leaderboard and increase the number of models assessed in the near future.

To conclude, the project has made substantial progress towards its set objectives, contributing significantly to progress in ASR technology for the Malayalam language and paving a path for enhanced future improvements.

Bibliography

[1] K. Manohar, A. Jayan, and R. Rajan, "An open framework for malayalam speech to text," 2023. https://kavyamanohar.com/post/kerala-science-congress-2023/.

[2] A. Radford, K. Jong Wook, and X. Tao, "Robust speech recognition via large-scale weak supervision," *International Conference on Machine Learning*, pp. 28492–28518, 2023.

[3] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," *Interspeech conference*, 2023.

[4] N. Donges, "A brief history of asr: Automatic speech recognition," 2019. https://towardsdatascience.com/a-brief-history-of-asr-automatic-speech-recognition-95de6c014187.

[5] K. Manohar, A. Jayan, and R. Rajan, "Quantitative analysis of the morphological complexity of malayalam language," *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic*, pp. 71–78, September 2020.

[6] S. Seyfarth and P. Zhao, "Evaluating an automatic speech recognition service," 2020. https://aws.amazon.com/blogs/machine-learning/evaluating-an-automatic-speech-recognition-service/.

[7] C. Kurian and K. Balakrishnan, "Speech recognition of malayalam numbers," *World congress on nature and biological inspired computing(NaBIC), IEEE*, pp. 1475–1479, 2009.

[8] Mohammed, Anuj, and N. K.N, "Hmm/ann hybrid model for continuous malayalam speech recognition," *Procedia Engineering*, vol. 30, pp. 616–622, 2012.

[9] K. Manohar, A. Jayan, and R. Rajan, "Syllable subword tokens for open vocabulary speech recognition in malayalam," *NSURL*, pp. 1–7, 2022.

[10] A. T. Pratap, Vineel, "Scaling speech technology to 1,000+ languages," *Facebook Research publication*, 2023.

[11] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei., "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," *arXiv, abs/2303.03926*, 2023.

[12] C. Y. A. M. M. C. e. a. Barrault, L., "Seamlessm4t-massively multilingual multimodal machine translation," *AI Meta Publications*, August 2023.

[13] G. A., C. H.S., and S. P. etl., "Clsril-23: Cross lingual speech representations for indic languages.," *arXiv, abs/2107.07402.*, 2021.

[14] M. Changrampadi, S. A., M. B. Narayanan, and N. Khan, "End-to-end speech recognition of tamil language," *Intelligent Automation and Soft Computing*, vol. 32, p. 1309–1323, 11 2021.

[15] T. Javed, S. Doddapaneni, A. Raman, K. S. Bhogale, G. Ramesh, A. Kunchukuttan, P. Kumar, and M. M. Khapra, "Towards building asr systems for the next billion users," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 10813–10821, 2022.

[16] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, pp. 1194–1198, 2021.

[17] S. Gandhi, P. Von Platen, and A. M. Rush, "Esb: A benchmark for multi-domain end-to-end speech recognition," *arXiv arXiv:2210.13352*, 2022.

[18] K. Guillaume Klein and K. H. K. et al., "faster-whisper," 2022-onwards. `https://github.com/SYSTRAN/faster-whisper/`.

[19] O. Team, "Chatgpt," 2022-onwards. `https://chat.openai.com/`.

[20] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, "Gradio: Hassle-free sharing and testing of ml models in the wild," *arXiv preprint arXiv:1906.02569*, 2019.