# Enhancing Bangla Speech Recognition through Acoustic and Language Modeling

by

Rubayt Anam
20101617
Anika Tabassum
20101143
Shanjid Ahmed Arnob
20101207
Md. Tazfiq Khan
20301172

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
November 2024

# Declaration

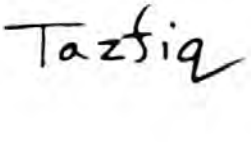
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

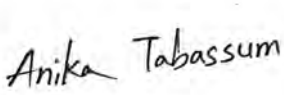
Shanjid Ahmed Arnob

20101207



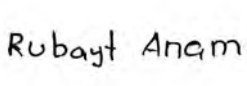
Md. Tazfiq Khan

20301172



Anika Tabassum

20101143



Rubayt Anam

20101617

# Approval

The thesis titled "Enhancing Bangla Speech Recognition through Acoustic and Language Modeling" submitted by

1. Rubayt Anam (20101617)
2. Anika Tabassum (20101143)
3. Shanjid Ahmed Arnob (20101207)
4. Md. Tazfiq Khan (20301172)

Of summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on November 11, 2024.

**Examining Committee:**

Supervisor (Member):

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-supervisor (Member):

Afroza Akther
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Abstract

This work introduces the performance comparison of two deep ASR models, Wav2Vec2 and Whisper, in recognizing the Bengali language with complex linguistics, dialectal variations, and phonetic intricacies. ASR technology has become crucial in interacting with devices in various applications because it bridges spoken speech to text or commands. Core ASR components include feature extraction, acoustic and language modeling, and decoding. These systematically provide a way of translating speech into actionable formats. Despite the advances, challenges in managing pronunciation diversity, code-switching, and ambient noise continue to face ASR systems, especially in languages like Bengali, which encompass significant phonetic and dialectal variations. Wav2Vec2 and Whisper were selected based on previous successes in other languages, but it was essential to study how efficiently adaptable they would be for Bengali ASR applications. The work was oriented toward sensitivity to the models regarding parameter tuning and generalization capability across linguistic features. Wav2Vec2 showed flexibility, with noticeable improvements in WER by tuning parameters such as learning rate, dropout, and gradient accumulation, which showcases its adaptability to Bengali's phonetic nuances. Each tuning configuration showed progressive enhancements in model stability and evaluation accuracy, hence positioning Wav2Vec2 as one of the potential candidates for a real-world application in Bengali, which requires precision and flexibility.

On the contrary, Whisper had rigidity in tuning: it kept the same WER of 100% for all settings and was thus insensitive to this tuning. So far, this may be a structural limitation within the Whisper model, which influences its performance in high-precision applications involving more linguistically complicated languages like Bengali. Preprocessing and feature extraction to standardize the audio data of both models included tokenization for linguistic alignment, and parallel inferences were performed to compare performance. While Wav2Vec2 showed promises with incremental improvements in transcription accuracy, Whisper's inability to adapt underscores the need for architectural revisions to meet the demands of Bengali ASR tasks. These results also suggest that Wav2Vec2, by the flexibility in its parameter tuning process, is more capable of handling the linguistic diversity of Bengali. At the same time, Whisper, on the other hand, is purely suitable for standardized languages, where rigidity does not pose any problem. The current paper concludes that Wav2Vec2 should be more appropriate for sophisticated ASR applications in the Bengali language, especially under dialectally diverse or noisy contexts. In contrast, Whisper may require essential changes to become compatible with a linguistically complex setting. This is further compounded by dataset diversity, limitations of computational resources, and model tunability in the present study to mark the importance of specialized ASR frameworks necessary for linguistic diversity inherent in Bengali and similar languages.

**Keywords:** Automatic Speech Recognition; Wav2Vec 2.0 ; Whisper; Voice Recognition; Machine Learning; Enhance; Language; Acoustic

# Dedication

We dedicate this effort to the Almighty since we have the highest regard for Him.

We would like to sincerely thank each and every group member for their amazing effort and tireless assistance during our challenging research.

We will always be appreciative to our parents for their unwavering faith in our skills and support, and their cooperation that encouraged us to succeed academically. Their encouraging comments and heartfelt prayers helped us develop and develop into the persons we are today. As we prepare to begin the next chapter of our life, we acknowledge and value the tremendous value of their self-sacrifice. We are very grateful for their unwavering support.

# Acknowledgement

We give the Almighty our profound appreciation and respect. His constant support and favor, which were crucial, allowed us to finish our whole Thesis. His spiritual guidance provided us with direction, strength, and knowledge to overcome obstacles along the way.

The constant help and direction we have received from Dr. Farig Yousuf sir has made our academic journey more smoother and more insightful. We want to thank him from the bottom of our hearts for being not only our great supervisor but also a mentor and an inspiration.

The knowledge, constructive criticism, and support of a respectable supervisor sir have been invaluable in enabling us to shape our research and improve our comprehension of the field. We are really privileged to have had the chance to work under his direction, as his commitment to academic quality has been clear in every encounter.

Without acknowledging Prof. Dr. Farig Yousuf sir's continuous availability and desire to share his wealth of knowledge, this would not be complete. His assistance has been invaluable to us throughout our academic career, and we are sure that the knowledge and understanding we have received from him will influence our future endeavors.

We are deeply grateful to sir for his guidance, commitment, and significant influence on our academic and personal growth.

We would also like to thank our co-supervisor Afroza Akther mam for her unwavering support while doing the work for our thesis. Her continuous support and quick response every time we had a query or some sort of problem while doing the thesis work made our work easier. Mam helped us with her knowledge, experience, and skills to do our job with more efficiency. We are grateful and very much privileged to have Afroza mam as our co-supervisor. We wish her success in her future journey.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

Speech recognition technology is getting more developed each day, as it is one of the most effective ways of interacting between humans and computers. Speech recognition technology can shape the future of artificial intelligence and how humans effectively interact with machines. This can transform how we engage with technology, along with getting rid of linguistic barriers and making a lot of things simple in our daily lives. In this era of rapid technological innovation, speech recognition technology has revolutionized how we interact with devices. With a focus on improving Bengali speech recognition utilizing novel acoustic and linguistic modeling techniques, the goal of this thesis is to study the limitations of speech recognition. The desired outcome of this research is to improve this technology, which is already in use, and increase its capability. In modern days, the speech recognition system already has a great deal of presence in our day-to-day lives. With voice-activated virtual assistants like Siri and Alexa, transcription services, vehicle voice command systems and even chatbots for customer support, speech recognition technology has already significantly impacted our daily lives. The user experience is significantly enhanced by these items' natural, hands-free engagement. Speech recognition is a powerful tool because it can translate spoken words into written text or executable commands. This is accomplished using a sophisticated mix of machine learning, data processing, and algorithms. If we try to divide the key parts of speech recognition systems, there are feature extraction, acoustic modeling, language modeling, and decoding. During feature extraction, the unique qualities of the input audio are documented. These features are then employed in acoustic modeling to identify phonemes, syllables, or phrases. Everything is combined during decoding to provide accurate transcriptions or instructions. Using language modeling, the most likely words or phrases are predicted. The foundation of voice recognition systems is the cooperation between these elements. Even though speech recognition technology has advanced over time, there are still a lot of barriers, especially in contexts with many languages. Grammatical patterns, phonetic features, and pronunciation quirks vary greatly between languages. As a result, developing an algorithm that will understand multiple languages is a challenging task. We need complex linguistic and acoustic models to do that task.

Additionally, the combination of speech recognition and artificial intelligence has opened a path to improve user experience in this sector. It has huge potential in terms of efficiency when it comes to technologies like smartphones, smart home ap-

pliances, autos, and office equipment.

This thesis explores the complex realm of acoustic and linguistic modeling in order to contribute to the dynamic area of speech recognition. We aim to improve the accuracy, efficiency, and capabilities of voice recognition systems through various tests and innovations.

## 1.1 Problem Statement

Our research will investigate strategies for adapting ASR models to these diverse linguistic contexts, enabling more accurate and context-aware recognition.

Accents and variations in pronunciation further complicate ASR systems, leading to recognition errors. Our research will focus on developing models that can robustly handle various accents and dialects, ensuring improved accuracy for diverse speaking populations. Our research will employ advanced acoustic modeling techniques like deep neural networks. These models will enhance the ability to capture and differentiate acoustic features across languages and accents. Language modeling will involve creating models that adapt to Bengali language, accents, and cross-lingual scenarios, addressing challenges such as code-switching.

The outcomes of this research will have practical implications for Bengali communication technology. By tackling the intricacies of Bengali speech recognition, this research will foster more accurate and accessible ASR systems, facilitating improved communication and understanding across linguistic boundaries in an increasingly interconnected world. The insights gained will serve as the foundation for further research in the subsequent thesis phase, ultimately contributing to the development of highly robust Bengali ASR systems.

## 1.2 Research Objectives

The research aims to enhance Bengali speech recognition by addressing several key objectives. The primary focus is to improve the accuracy and efficiency of our proposed model. The objectives of this research are:

1. Enhance the accuracy and adaptability of Bengali ASR systems by addressing linguistic complexities, such as vernacular diversity, inflectional morphology, and grapheme-to-phoneme mapping.

2. Evaluate and compare the performance of Whisper and Wav2Vec2 models in handling Bengali-specific acoustic and phonetic challenges.

3. Optimize the Wav2Vec2 model through fine-tuning to capture Bengali dialectal variations and improve recognition accuracy across diverse linguistic contexts.

4. Address the scarcity of publicly available Bengali speech corpora by utilizing standard voice datasets and identifying model-specific responses to tuning.

## 1.3 Automatic Speech Recognition (ASR) Overview

Automatic Speech Recognition (ASR) systems have become integral to modern technology, enabling machines to understand and process human speech effectively. ASR transforms spoken language into written text or executable commands, providing a bridge for hands-free, natural interactions with devices. The technology underpins applications like virtual assistants (Siri, Alexa), transcription services, and voice-controlled devices in various sectors such as healthcare, automotive, and customer service.

### 1.3.1 Core Components of ASR Systems

The major components standard in most ASR systems are feature extraction, acoustic modeling, language modeling, and decoding. Each plays a different role in translating human speech into a format that machines understand and act on.

**Feature Extraction**

- **Purpose:** Feature extraction provides an opportunity to abstract the characteristic features of the audio input, which, more simply put, means the ASR system analyzes the sound waves in spoken language to outline differences in pitch, tone, and intensity.
- **Process:** This system segments the speech into small, manipulable frames and extracts the significant features of Mel-frequency cepstral coefficients, which are typically necessary for identifying phonemes. These would help the system recognize several patterns of sound, which it will map later to some specific linguistic constituent.

**Acoustic Modeling**

- **Purpose:** Acoustic modeling connects the features extracted to the linguistic sounds or phonemes.
- **Process:** This involves training a system to pick out patterns related to different phonemes in speech. Most modern acoustic models implement deep learning procedures such as DNN or CNN to enhance accuracy. Whisper, or Wav2Vec 2.0 advanced ASR systems, use such models to distinguish phonemes with similar pronunciation and thus improve recognition capability, especially in languages with complex phonetic structures.

**Language Modeling**

- **Purpose:** Language Modelling predicts what words or phrases would be next in any given text to make a meaningful text using learned language patterns.
- **Process:** Language models focus more on common word sequences and grammatical rules to guess what words or phrases will follow. This is useful for dealing with accents, dialects, and other specific language structures, such as

Bengali, where code-switching between different languages within one conversation is frequent. The language model also resolves other issues, such as homophones: words sounding exactly alike but having diverse meanings. The language model resolves these by considering the context for word selection.

**Decoding**

- **Purpose:** Decoding synthesizes feature extraction, acoustic, and language modeling inputs to develop an accurate transcription or command.
- **Process:** It generates the output by combining all the information from the preceding stages. Decoding uses algorithms that make estimations on the most probable words or sets of words represented by the signal of an audio signal. This becomes very accurate and can adapt to any form of speech by cross-referencing it with acoustic and language models.

### 1.3.2 Challenges in ASR Systems

Despite advancements, ASR systems face numerous challenges, especially with languages that have complex phonetic, grammatical, and dialectal variations. The following are some of the prominent difficulties:

- **Accents and Pronunciations:** Pronunciation variations vary from region to region. So, ASR models should resist a wide range of accents and dialects.
- **Code-Switching:** In multilingual environments, a single user may use different languages simultaneously in one conversation. ASR systems, therefore, are required to deploy sophisticated linguistic models that handle such situations effectively.
- **Noise:** The detection of speech under noisy conditions is problematic. Speech recognition systems must detect a target speech against noises; therefore, they should have comprehensive noise-cancelling features.

# Chapter 2

# Literature Review

In the literature review chapter, we will discuss the related papers we read and observed during the time of our research to gain more knowledge about the topic which we are working on. Though ASR(Automatic Speech recognition system) has developed quite a lot in recent years, yet it has tons of improvements to make in the matter of working with the Bengali Language. Over the years number of researches on the Bengali ASR system have been done. First, we will discuss about the paper that are related with the background of our research topic and then we will discuss about the papers that are related to our work.

## 2.1 Background

Voice recognition technology has come a long way in the last few decades, now regarded as an essential component of modern human-computer interaction. Initially, speech recognition models were based on rules and small quantities of labeled data. The models had a lot of margin of error because the algorithms were relatively simple and there weren't enough computing resources[7]. End-to-end deep learning frameworks and ASR (Automatic Speech Recognition) systems have been able to continue to improve by analyzing larger and larger datasets[8]. This has become feasible due to the growing speed and power of computers. These advancements have made it a tool that is deployable in many different environments[15]. These applications include voice recognition ordering virtual assistants like Alexa, Siri, etc., and transcribing services. According to Toshniwal et al., 2018 These applications allow users to easily and hands-free interact with digital devices[19]. As this technology continues to progress, it is being exploited much more widely in modern communication systems that need to meet the requirements of high-resource and low-resource language users alike. This literature review explores the advancements and challenges of multilingual automatic speech recognition (ASR) with special reference to Bengali ASR, from the perspectives of both linguistic and acoustic modeling, respectively as well as the advances in NN architecture. This also analyzes the unique difficulties brought by low-resource languages[3][30].

## 2.2 Related Works

In this part, we have discussed the papers that are related to our research topic which has been divided into three parts which are Acoustic Models, Language Models, and finally Dataset and Implementations.

### 2.2.1 Acoustic Models

In the early days of research on Bengali automatic speech recognition, the framework is built on phoneme-based models with the use of simple datasets[36]. We intended to reflect the unique aspects of the Bengali language. Alam et al. used a framework called the Hidden Markov Model-Gaussian Mixture Model (HMM-GMM)[34]. In this framework, the preprocessing steps were noise reduction and normalization. Investigating this approach laid the foundation for subsequent studies on Bengali automatic speech recognition and was very helpful in understanding the intricate phonetic structure of the language. Large datasets with diverse data types were shown to be important for resource-scarce languages[34].

The second important add-on was that Das's research was focused on phoneme-level recognition using the CMUSphinx framework which has an HMM-based approach at its core[4]. To help the model understand core phonetic building blocks, Das designed a unique sample with basic Bengali phonemes. This was done as a way of balancing the lack of Bengali statistics available at present. Using Linear Predictive Coding (LPC) for extracting spectral parameters distinctive to the Bengali language, Das opened up the door for future phoneme-to-text mapping in Bengali automated speech recognition. Phoneme-based models are still important for automatic speech recognition for Bengali especially in systems that are developed to work in less powerful computer systems. The importance of phoneme segmentation is underscored by findings from these preliminary studies. This remains an important approach to improving the performance of automatic speech recognition (ASR) systems targeted for low-resource languages[4] [32].

The use of advanced neural network architectures, such as CNN, RNN, and Transformer models has brought dramatic changes in Bengali ASR (automatic speech recognition). Such designs have allowed systems to handle complex time and space relations of voice commands. The optimal approach for this CNN-RNN architecture by Mandal et al. is with bidirectional Gated Recurrent Units (GRUs) and with Connectionist Temporal Classification (CTC) loss[2]. Such a design could pick up subtle phonetic patterns actually specific to Bengali. It was found best for large-scale ASR tasks where accuracy and less processing power are required. In this case, the WER for these apps is 13.67 percent[23].

ASR systems are able to learn new languages quickly due to changes in the way neural networks are designed. Bengali ASR has abandoned staggering headway, because of new innovations that joined Convolutional Neural Systems (CNNs) with Recurrent Neural Systems (RNNs) and attentionbased systems[12]. These new concepts have made it easier to extract features and adjust linguistic contexts. The

LAS stands for Listen, Attend, and Spell. The magic finally helps to improve recognition accuracy by turning audio directly into text via this attention-based neural network model. CNN and RNN have been reported to be really powerful in automatic speech recognition in Bengali because they recognize how one small slash, a so-called phoneme but with definite variations of sound can also result in a difference in the meaning of words[13].

Possible applications of Bengali automated voice recognition include competitive training and self-supervised learning[31]. Such self-supervised models like Wav2Vec 2.0. The models allow ASR systems to learn from huge amounts of unlabelled audio data. The models are especially suited for Bengali a low-resource language with limited labeled data. The 2020 Baevski et al. study. argues that this approach reduces the dependency on large labeled datasets. As a result, models can capture the essential patterns of Bengali speech. This trait can then be changed to a different language[24][25].

For hybrid designs, RNN layers are better suited for dealing with temporal interactions, whilst CNN layers are more suited to capturing spatial dependencies in spectrograms. By combining CNNs that are very good at recognizing patterns in the spatial features of the spectrogram with RNNs that can analyze temporo-spatial relationships the result is a model that is powerful enough for use in practice[29]. This mixed model used Mel Frequency Cepstral Coefficients (MFCCs) as feature extraction[10]. To improve the model performance, especially at high noise levels noise reduction techniques were also integrated. Even better came when CNN-RNN model improved with CTC loss Achieved 13 WER (Word error rate) 67 percent as a result. They can be useful in large-vocabulary automatic speech recognition tasks where accuracy of recognition and computation speed are important. This model shows their level of support[2].

### 2.2.2 Language Models

Language modeling is an essential component of bilingual automatic speech recognition (ASR) because it can help regulate code-switching and other cross-lingual issues. Dilated convolution-based autoregressive models such as WaveNets show very strong performance in automatic speech recognition (ASR) in Bengali, This is attributed to their potential to capture long-range temporal dependencies state that[14]. These relationships determine how many languages and dialects can be understood through a single model framework[37]. Adaptive language models that use background information from multiple languages are more accurate for automatic speech recognition (ASR). This is particularly true for Bengalis who frequently switch between languages. Sultana et al., 2021 demonstrated that code-switching can be effectively handled by flexible language models. By conversing with individuals who speak different languages they demonstrated that these models can maintain their accuracy. Because Bengali automatic speech recognition systems must be able to handle both Bengali and English in the same sentence this feature is very beneficial[30].

When ASR systems have to understand and deal with multiple languages–each with

different phonetic prosodic and syntactic properties, they are faced with an entirely different set of challenges[18]. This is further complicated by the fact that Bengali has many sounds and shapes unique to specific areas of Bangladesh, so even if they spell it right, how will they pronounce it without context? Bengali has a very complex script and numerous dialects as well as peculiar phonetic characteristics that differ widely from English and other popular languages like Spanish, German, or Chinese that conventional ASR algorithms must learn, making it very difficult to understand the spoken or written form of language[8]. Code-switching is often faced by automatic speech recognition (ASR) systems in the Bengali wording setting[33]. Code-switching is when people switch back and forth, at the same time, between Bengali English and other regional languages. This makes the task of identification models even harder[19]. Consequently, automatic speech recognition also called ASR systems should have adequate proficiency in language identification classification and translation features to perform best with Bengali.

These skills are required to understand the dialects, accents, and scripts used across the Bengali-speaking regions and the complexity of Bengali adds to this requirement. One of the biggest challenges of multilingual automatic speech recognition (ASR) in general and Bengali ASR, in particular, is creating training models that can take advantage of this vast array of linguistic features without getting too slow in the process. ASR system will require a plan for strong acoustic models improved information enhancement and more significantly language-specific adjustment to enable it to work with many different accents in addition to grammatical styles[16]. This variety is often too much for even the traditional ASR models to handle[5].

### 2.2.3 Dataset and Implementations

The Mozilla Bengali Common Voice Speech Dataset is a large dataset of more than 400 hours of spoken Bengali[34]. This dataset is designed to have the potential to run ASR systems on a more diverse set of users in the real world by recording a wide variety of regional accent pronunciation patterns and acoustic environment conditions. Therefore, it is an important ingredient to build ASR systems. In their study,

Having only a limited amount of labeled data is a big problem because it limits the size and variedness of the dataset. To counter this growing problem, dataset expansion, and diversification through data addition is now essential. One of the most common strategies to improve sound quality is temporal and frequency masking on spectrograms to simulate different sound environments[6]. SpecAugment is particularly useful for low-resource scenarios where ASR systems need to work in diverse accents and in noisy environments since it makes them more robust to changes in sound[40]. Composite sources are formed by spectrograms from fully independent audio streams[21].

The mixup method directly influenced the design of mixSpeech even from its inception. It is a technique that helps in training automatic speech recognition (ASR) models to recognize the difference between voice and noise[27]. This is an essential capability in Bengali-speaking communities where ambient noise can be a signifi-

cant challenge. Giollo et al. explored the methods of generating untruthful data that result in improved audio samples based on the text-to-speech algorithm[28]. These can be used for augmenting training datasets. Giollo et al. For instance, in a 2020 study synthetic data generation methods were employed to improve Bengali automatic speech recognition (ASR) systems to deal with different phonetics and dialects[28]. Changing controllable factors such as pitch, voice tract length, and noise level enhances the stability of the model and reduces problems caused by a lack of data[20].

Transfer learning is especially useful in ASR scenarios through limited resources as models can rely heavily on their knowledge gained from high-resource languages. Transfer learning has the potential to improve the spectral distance of Bengali automatic speech recognition (ASR) by a large margin. This is demonstrated by Showrav et al. use of IndicWav2Vec a bilingual model that has already been trained in multiple Indian languages[35]. So on Bengal-specific data, the model scored a Levenshtein Mean Distance of 3.819 points underlining its performance uplift. These improvements were obtained with small modifications over IndicWav2Vec. The study by Showrav et al. 2020 study concludes that transfer learning along with language-specific training data can improve automatic speech recognition (ASR) in low-resource languages[35].

Nahid et al. used a two-layer LSTM-RNN model trained on the Bangla-Real-Number dataset as a building block and progressed similarly[17]. This means that 13.2% of the words they typed were wrong. Detailed phonetic features as well as long dependencies however could be captured with MFCCs and hence we suspected that using them as features was a pretty good idea as it already did a pretty good job of capturing detailed phonetic features as well as long dependencies by our model. This led to a lesser ASR for Bengali; thus, a more extensive improved pennable corpus[17]. This research highlights the importance of stacked LSTM-RNN models in Bengali automatic speech recognition (ASR), as evident from the performance metrics shown in a study published in 2020. It is important to provide details on subtle changes in phonetics and the timeframe that will be needed to be captured for effective detection in the real world. While Bengali phonetic aware neural network designs are still a key requirement for the overall performance and usability of automatic speech recognition (ASR)[1].

Speaker recognition is especially important for users who want to recognize speakers accurately, and this has also been studied for Bengali automatic speech recognition (ASR)[26] employed Support Vector Machines (SVMs) with Multifractal Detrended Fluctuation Analysis (MFDFA) and reached a classification accuracy of 96%. This method shows the promise of nonlinear feature extraction for speaker-dependent ASR applications. Additionally, Sarkar et al. specify what it could mean for secure voice authentication systems as well as personalized assistants[26].

Phoneme modeling is a critical part of Bengali automatic speech recognition (ASR). This was needed to handle the different sounds that were used in different regional dialects of the language. Das says his phoneme-based automatic speech recognition model captures Bengali-specific phonetic units for better recognition accuracy[22]. It

was especially important in places with limited computational power. Text phoneme segmentation is the most salient means of reliably transcribing Bengali across a range of pronunciations by allowing models to rapidly associate phonetic units with text. Phoneme-based modeling is still important for example in Bengali automatic speech recognition (ASR) research. This helps guarantee that dialect variations are considered while writing.

Antagonistic training is an effective technique, and it produces more stable models by changing phonemes. This training makes it more difficult for automatic speech recognition systems to recognize phonetic patterns that have been modified[41]. Since, in real-world scenarios, differences in pronunciation accent as well as background voice both affect the montage of speech recognition like Bengali automatic speech recognition is adaptive, this is a real-world change. That method works really well in these situations[39]. This was first found by Ossama et al. In this work, we demonstrate that we can make Bengali automatic speech recognition (ASR) systems more robust and flexible by augmenting traditional ASR methods with adversarial training, a well-known adversarial defense technique. That is especially true in areas with lots of different sounds[11].

The strengths of Bengali ASR and the areas that are still far away from full-filling can be the main contribution areas of this study. A lot has changed since the introduction of the first primitive phoneme-based models. We have advanced neural architectures that combine transfer learning data augment and self-supervised techniques development[38]. It is able to operate through a cross-linguistic adaptation in the case of low-resource languages while transfer-learned from models like IndicWav2Vec[21] [35]. Models have been transferred to other languages based on transfer learning, and data augmentation techniques such as SpecAugment and MixSpeech have improved the robustness of models[21] [27]. The scalability and adaptability of Bengali ASR are likely to improve with forthcoming advances in data-efficient self-supervised learning and cross-lingual modeling paradigms. Goals of research capable of constructing language-independent elements and improved acceptance of automated speech recognition fashions to dialectal switches can yield advantages for Bengali audio systems. This is especially common in rural areas or other places with varying dialects[9]. Prospective improvements in automatic recognition technology of speech in Bengali may open human-technology interaction in a more free and natural way. This would help bridge bonds among Bengali speakers across the globe, irrespective of the language barrier. Usability and accessibility would come first with these improvements.

# Chapter 3

# Workplan

In this part, we will discuss our workflow along with the workflow diagram. This project aims to improve speech recognition specifically for Bangla, enhancing the performance of tools like virtual assistants and transcription services. The goal is to make speech-to-text more accurate and efficient for Bangla speakers, improving accessibility and communication.

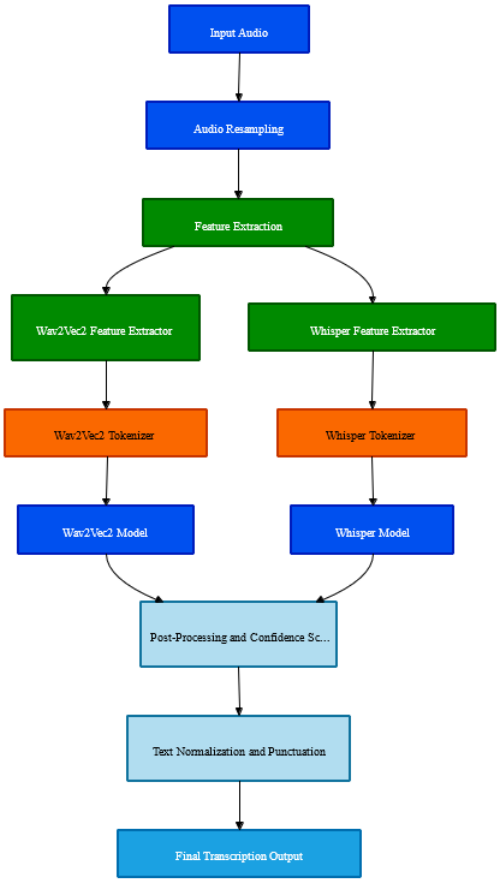



Figure 3.1: Work Plan data flow diagram for the system

## 3.1 Input Audio Processing:

- First, the audio data is loaded and resampled at a uniform rate of 16 kHz to be compatible with both models. This will preprocess the raw audio to prepare it for feature extraction and the subsequent steps.

## 3.2 Preprocessing and Feature Extraction:

- **wav2vec2 Feature Extraction:** Wav2Vec2FeatureExtractor transforms the raw audio waveform into log-Mel spectrogram form suitable for processing by wav2vec2.
- **Whisper Feature Extraction:** In parallel, load audio as spectrogram features compatible with the Whisper model using the Whisper Feature Extractor.

## 3.3 Tokenization:

- **wav2vec2 Tokenization:** Wav2Vec2CTCTokenizer (transformers) is used to tokenize transcriptions for training or inference in wav2vec2.
- **Whisper Tokenization:** Use Whisper Tokenizer with the appropriate language setting for multilingual transcription, converting transcriptions into token IDs.

## 3.4 Parallel Model Inference:

- **wav2vec2 Inference:** Pass the wav2vec2 features through Wav2Vec2ForCTC to obtain the initial transcription. This model is known for effectively handling noisy and raw audio.
- **Whisper Inference:** Feed the Features through Whisper Conditional Generation to get the Whisper for transcriptions with extra-linguistic and condition-specific precision.

## 3.5 Post-Processing:

- **Transcription Comparison and Confidence Scoring:** Use confidence scoring or language model alignment techniques to compare transcriptions created by both models. One will have to favor the one with the highest confidence or combine all of them to ascertain the mean Result.
- **WER Evaluation:** Compute both transcriptions' Word Error Rate (WER) to see which model results better according to ground truth (if possible).

## 3.6 Refinement:

- **Text Normalization and Punctuation:**Perform the text normalization and punctuation restoration. All the rules are to be followed by language as the final step of the transcription is given.

## 3.7 Output:

- The final transcription output is provided, potentially to indicate transcription reliability.

# Chapter 4

# Description of the dataset

In this chapter, we will discuss the dataset we are using here.

## 4.1 Data Collection

For our paper, we collected the dataset from the Bangla language dataset section of Mozilla Common Voice. We know this is a reliable widely used website to collect datasets for speech recognition works. After doing a bit of research, we decided to use CV(Common Voice) Corpus 16.1 to test and train our model. This dataset has 1272 hours of recorded voices in the Bangla language and 54 validated hours of those voices. There are approximately 22897 numbers of voices and all the voices are in MP3 format. In the dataset, there are 53% male voice, 23% female voice and 22% has no information. If we divide the categories of the dataset according to the age of the volunteers who have the voice clip here, The table will look something like this,

### 4.1.1 Preliminary Analysis

The age distribution bar chart (Figure 4.1) displays the frequency of users across various age groups. The bar chart clearly shows that the largest group of users is within the age range of twenties, followed by a lesser percentage in their teens and thirties. The age groups in their forties and fifties exhibit a notable decrease in the number of users. The high proportion of younger age groups in the user base may impact the selection of material that receives upvotes or downvotes, as preferences tend to differ considerably across different age demographics.

| **Age** | **Percentage** |
|---|---|
| $< 20$ | 7% |
| $20 - 29$ | 67% |
| $30 - 39$ | 3% |
| $40 - 49$ | 1% |
| $50 - 90$ | 0% |
| No Information | 22% |

Table 4.1: Age Distribution of Voice Samples

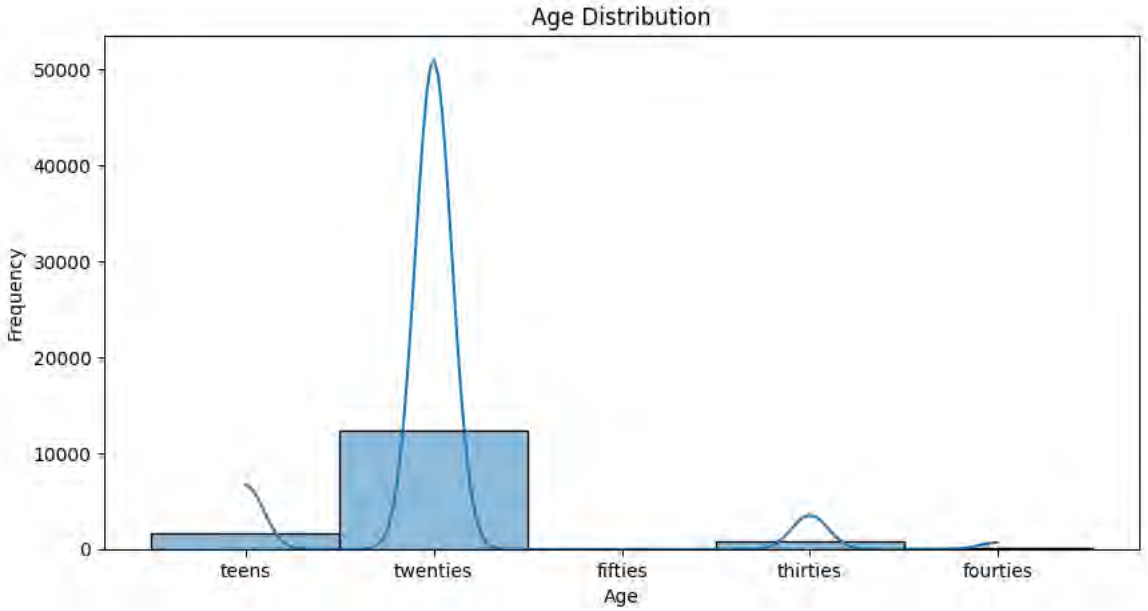



Figure 4.1: Bar chart displaying the distribution of ages

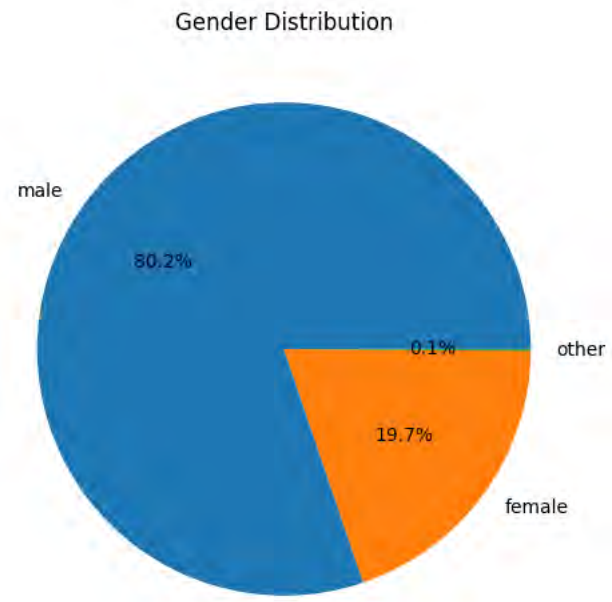



Figure 4.2: Pie chart displaying the distribution of genders

The pie chart (Figure 4.2) indicates that a substantial majority of users are male, accounting for 80.2% of the total. Females comprise 19.7% of the users, while the remaining 0.1% are classified as "others." The gender disparity could impact the dynamics of interactions on the platform, potentially influencing the popularity of certain posts.

The heatmap (Figure 4.3) shows a significant positive correlation (0.59) between upvotes and downvotes, indicating that postings that earn upvotes are also likely to receive downvotes. Nevertheless, there are little associations between age, gender, and the quantity of votes (both positive and negative), suggesting that these demographic variables have little impact on voting behavior in this dataset.

The histogram (Figure 4.5) illustrates that the majority of posts receive a limited number of upvotes, followed by a steep decrease as the number of upvotes rises. This implies that although a small number of posts gain significant popularity, the majority of postings receive very little attention, which is consistent with the typical distribution pattern observed on social media sites where viral content is infrequent.

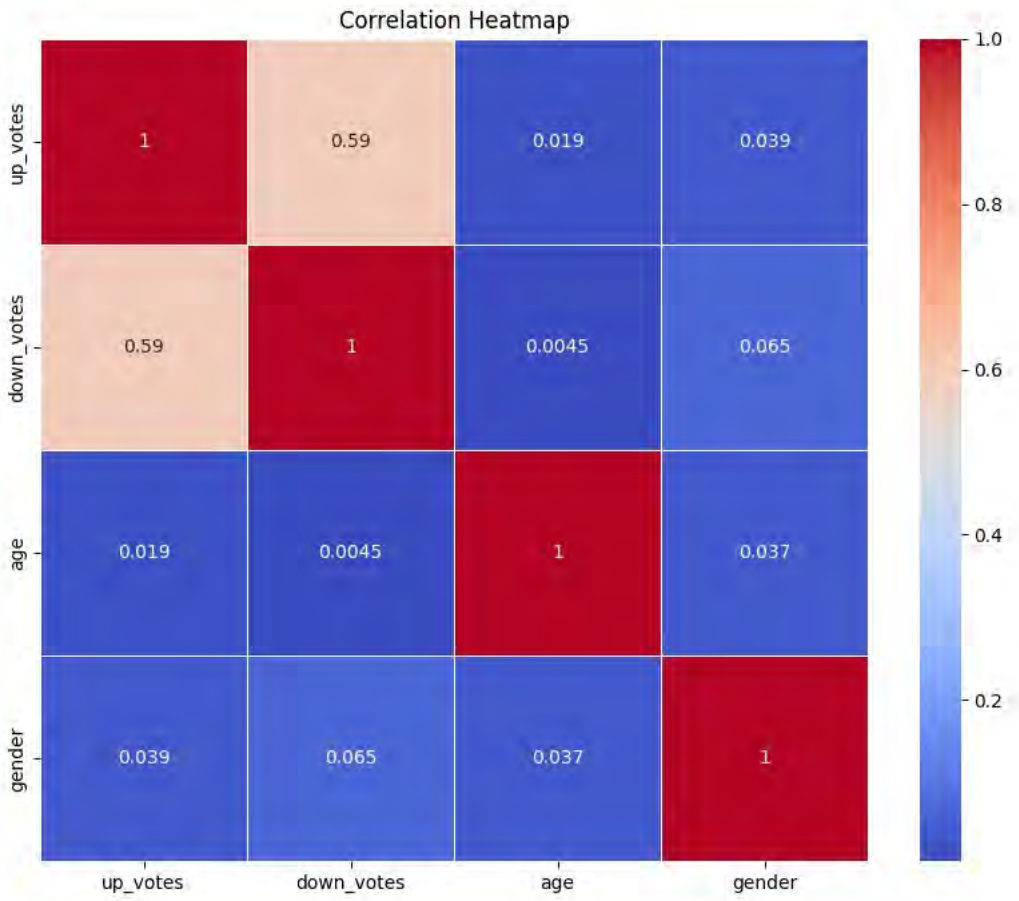



Figure 4.3: Correlation heatmap

## 4.2 Pre-processing of data

**Data Categorization:**Our data has been collected and categorized with appropriate metadata, including speaker ID, age, gender, and other important information.

**Text Transcription:**Each audio file has a corresponding written transcription of the spoken words.

**Voting:**Votes to assess the accuracy of the transcriptions (up-votes and down-votes).

**Displaying Random Elements:**This function selects 10 random examples from the dataset and displays them as a table using Pandas and IPython's display function.

**Further Audio Feature Preprocessing:**
In order to utilize this data in a machine learning model, particularly for tasks such as voice recognition or speaker identification, further preprocessing procedures are generally necessary. The following actions are included:

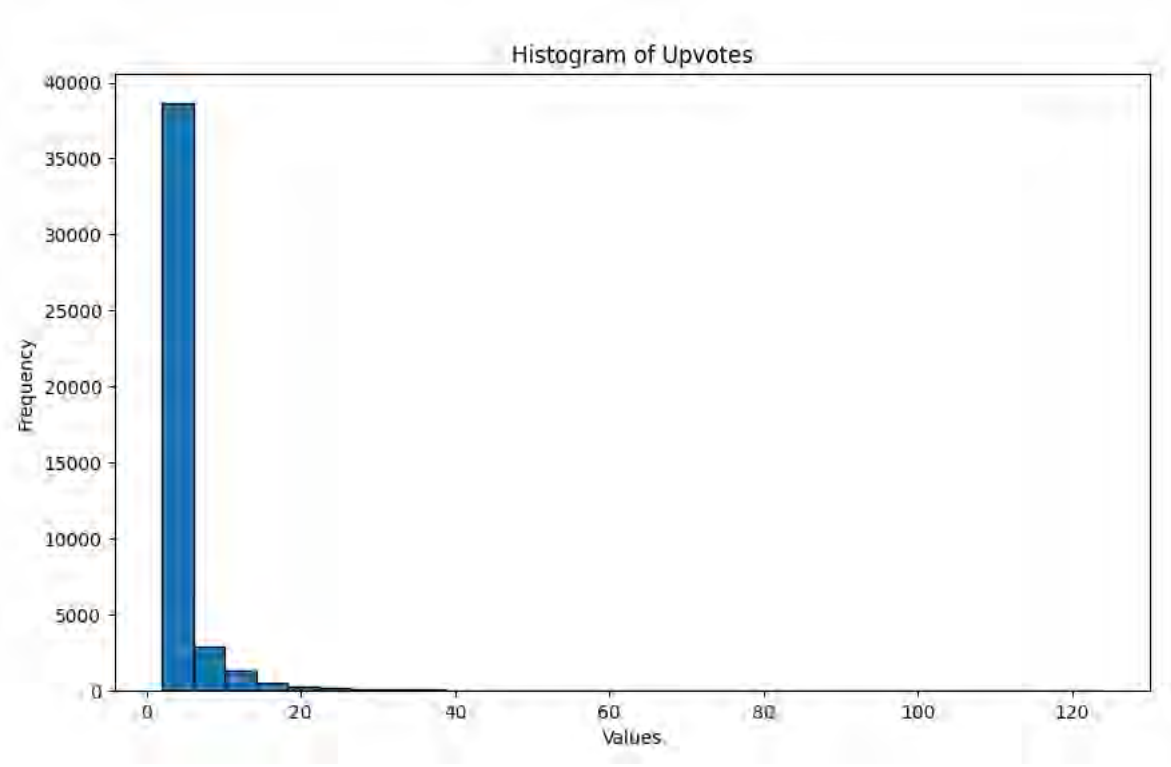



Figure 4.4: Upvote Distribution Chart

**Feature Extraction:** Transform the unprocessed audio recordings into numerical representations (such as Mel spectrograms or MFCCs) that are better suited for machine learning models. Typically, this method involves utilizing libraries such as Librosa to manipulate the audio data.

**Tokenization:** When employing a language model, divide the sentences into subwords or phonemes.

| | sentence |
|---|---|
| 0 | এ সময়ে তিনি নিজের কবিতা প্রকাশ করতে পারতেন না। |
| 1 | এটা তখনকার সময়কার খুবই আধুনিক একটি দালান ছিল। |
| 2 | এই প্রত্যেকটি ভূমিরূপ কে আবার অনেক ভাগে ভাগ করা হয়েছে। |
| 3 | ভারত সরকারের পুরাতাত্ত্বিক বিভাগ এর দেখাশুনা করে। |
| 4 | পিতা চিকিৎসক সুরেন্দ্রনাথ দাশ ও মাতা বিশিষ্ট শিক্ষিকা সবিতা দাশ। |
| 5 | এটি একটি আশ্চর্যজনক প্রশিক্ষণ কর্মক্ষমতা। |
| 6 | পরবর্তীতে, তিনি আরও দুটি কিডস চয়েস অ্যাওয়ার্ডস জিতে নেন এবং একটি পিপলস চয়েস অ্যাওয়ার্ড জিতেন। |
| 7 | এটি পরিচালনা করেছেন মার্টিন স্কোরসেজি এবং চিত্রনাট্য লিখেছেন জন লোগান। |
| 8 | এটি উত্তর দিনাজপুর জেলার সদর দপ্তর। |
| 9 | তিনি তার শৈশব কাটান নতুন দিল্লিতে। |

Figure 4.5

If we describe this phase of our work step by step, it would look something like this:
Step 1: Installing Necessary Libraries
Step 2: Loading Dataset
Step 3: Define Dataset Paths

Step 4: Load the Dataset
Step 5: Add Absolute Paths to the Audio Files
Step 6: Cast the Audio Column to the Correct Format
Step 7: Displaying Random Elements
Step 8: Data Preprocessing - Removing Special Characters
Step 9: Create Vocabulary from Characters
Step 10: Define the Vocabulary
Step 11: Initialize Tokenizer and Feature Extractor
Step 12: Combine Tokenizer and Feature Extractor into a Processor
Step 13: Preparing Dataset for Model Input
Step 14: Data Collator for Dynamic Padding
Step 15: Evaluation Metric - Word Error Rate (WER)
Step 16: Model Setup
Step 17: Training Setup
Step 18: Trainer Initialization and Training

# Chapter 5

# Description of model

This chapter discusses the models used for Bengali speech recognition, including existing models like LAS, ASR, and Kaldi toolkit, along with proposed models Whisper and Wav2Vec 2.0 for improved performance.

## 5.1 Existing models

In this section, we will look into some existing works on speech recognition which is related to our works. Firstly, we will describe the Listen, Attend, and Spell (LAS) attention-based sequence-to-sequence ASR model proposed by Chan et al. Then we will talk about the ASR system and finally, we will talk about the Kaldi toolkit.

### 5.1.1 LAS Model

The sequence-to-sequence model consists of three modules: an encoder, decoder, and attention network which are trained jointly to predict a sequence of graphemes from a sequence of acoustic feature frames **toshniwal2018multilingual**.
Here, it used 80-dimensional log-mel acoustic features computed every 10ms over a 25ms window. Following we stack 8 consecutive frames and stride the stacked frames by a factor of 3. This downsampling enabled to use of a simpler encoder architecture than The encoder is comprised of a stacked bidirectional recurrent neural network (RNN) that reads acoustic features and outputs a sequence of high-level features (hidden states)$h = (h_1, ..., h_K)$. The encoder is similar to the acoustic model in an ASR system.
The decoder is a stacked unidirectional RNN that computes the probability of a sequence of characters y as follows:

$$P(y|x) = P(y|h) = \sqcap_{t=1}^{T} P(y_t|h, y_{<t})$$

The conditional dependence on the encoder state vectors h is represented by context vector $c_t$, which is a function of the current decoder hidden state and the encoder state sequence:

$$u_{it} = v^T tanh(W_h h_i + W_d d_t + b_a)$$

$$\alpha_t = \text{softmax}(u_t)$$

$$c_t = \sum_{i=1}^{K} \alpha_{it} h_i$$

where the vectors v,$b_a$ and the matrices $W_h, W_d$ are learnable parameters; $d_t$ is the hidden state of the decoder at time step t. The hidden state of the decoder, $d_t$, which captures the previous character context $y_{<t}$, is given by:

$$d_t = \text{RNN}(\tilde{y}_{t-1}, d_{t-1}, c_{t-1})$$

where $d_{t-1}$ is the previous hidden state of the decoder, and $\tilde{y}_{t-1}$ is a character embedding vector for $y_{t-1}$, as is typical practice in RNN-based language models. The decoder is analogous to the language model component of a pipeline system for ASR. The posterior distribution of the output at time step t is given by:

$$P(y_t \mid h, y_{<t}) = \text{softmax}(W_s[c_t; d_t] + b_s),$$

where $W_s$ and $b_s$ are again learnable parameters. The model is trained to optimize the discriminative loss:

$$L_{LAS} = -\log(P(y|x))$$

In the multilingual scenario, we are given n languages $L_1, ..., L_n$, each with independent character sets $C_1, C_2, ....., C_n$ and training sets $(X_1, Y_1), ..., (X_n, Y_n)$. The combined training dataset is thus given by the union of the datasets for each language:

$$(X, Y) = \bigcup_{i=1}^{n} (X_i, Y_i)$$

and the character set for the combined dataset is similarly given by:

$$C = \bigcup_{i=1}^{n} C_i$$

We begin by training a joint model, consisting of the LAS model described in the previous section trained directly on the combined multilingual dataset. This model does not give any explicit indication that the training dataset is composed of different languages. However, as we will show later, this model is still able to recognize speech in multiple languages despite the lack of runtime language specification.

We also experiment with a variant of the joint model which has the same architecture but is trained in a multitask learning (MTL) configuration to jointly recognize speech and simultaneously predict its language. The language ID annotation is thus utilized during training but is not passed as an input during inference. In order to predict the language ID, we average the encoder output h across all time frames to compute

an utterance-level feature. This averaged feature is then passed to a softmax layer to predict the likelihood of the speech belonging to each language:

$$p(L \mid x) = \text{softmax}\left(W_{\text{lang}}\left(\frac{1}{K}\sum_i h_i\right) + b_{\text{lang}}\right)$$

The language identification loss is given by:

$$L_{LID} = -\log(p(L = L_j \mid x))$$

where the j-th language, $L_j$, is the ground truth language. The two losses are combined using an empirically determined weight $\lambda$ to obtain the final training loss:

$$L_{MTL} = \frac{1}{1+\lambda}L_{LAS} + \frac{\lambda}{1+\lambda}L_{LID}$$

Finally, we consider a set of conditional models which utilize the language ID during inference. Intuitively, we expect that a model that is explicitly conditioned on the speech-language will have an easier time allocating its capacity appropriately across languages, speeding up training and improving recognition performance.
Specifically, we learn a fixed-dimensional language embedding for each language to condition different components of the basic joint model on language ID. This conditioning is achieved by feeding in the language embedding as an input to the first layer of the encoder, decoder or both giving rise to (a) Encoder-conditioned, (b) Decoder-conditioned, and (c) Encoder+Decoder-conditioned variants. In contrast to the MTL model, the language ID is not used as part of the training cost.

### 5.1.2 ASR Model

Automatic Speech Recognition (ASR) systems convert spoken words into text. They work by analyzing sound waves, and trying to figure out which words were spoken. This is a tough job because different people say things differently, and there can be background noise. The ASR process has two main parts: preprocessing and post-processing. Preprocessing involves turning the sound into features the computer can understand. Post-processing builds a system that can recognize words. This system uses things like an acoustic model, a dictionary, and grammar rules to figure out what was said. Choosing the right way to turn sound into features and the right way to recognize words is important for making the ASR system work well. Different methods have different strengths and weaknesses. Scientists compare these methods to find the best ones for different situations. Overall, ASR systems are important for things like voice assistants and dictation software. They help computers understand and respond to human speech.

ASR systems take acoustic waveforms as input and output a sequence of words. The task involves finding the word sequence W=W1,W2,...,Wm that maximizes the posterior probability P(W/X), where X=X1,X2,...,Xn represents the acoustic observations. This probability is calculated using Bayes' theorem.

$$W = \arg\max P(\frac{W}{X}) = \arg\max\left\{\frac{P(W)P(\frac{X}{W})}{P(X)}\right\}$$

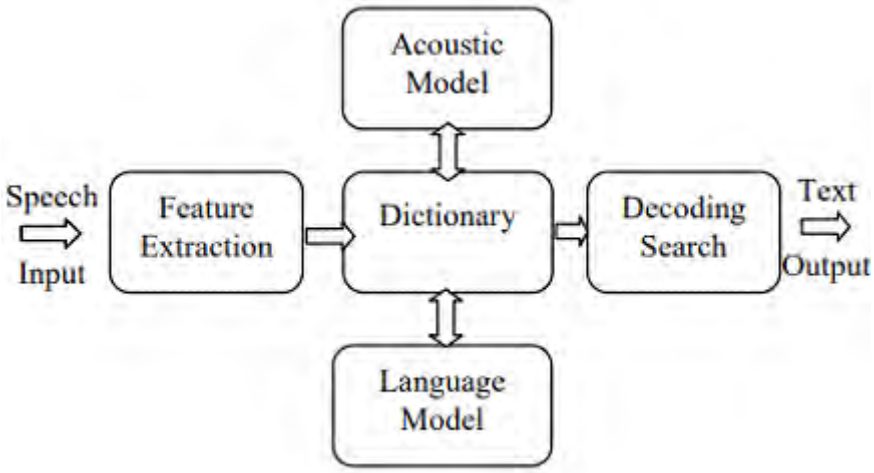



Figure 5.1: Overview of ASR system

Speech recognition relies on extracting features from speech signals to distinguish between different utterances. Features must be easy to measure, resistant to mimicry, stable across environments, and naturally occurring in speech. Two primary techniques are Linear Predictive Coding (LPC) and Mel Frequency Cepstral Coefficients (MFCC). LPC estimates speech parameters by approximating samples as linear combinations of past ones, providing accurate predictions over time. MFCC mimics human auditory response by logarithmically positioning frequency bands. To find MFCC, we break speech into small parts, smooth out any jumps using a Hamming window, make a special filter set using maths called Discrete Fourier Transform, and change the numbers into a different type. This aids speech recognition, making MFCC a widely used feature extraction technique.

Speech recognition has evolved from dynamic programming to neural networks and Hidden Markov Models (HMMs). Template-based approaches match speech against pre-recorded templates, while knowledge-based methods use hand-coded rules. Neural networks offer greater accuracy for limited vocabularies. Dynamic Time Warping aligns sequences for efficient word recognition. Statistical models like HMMs are popular for their automatic training and simplicity. HMMs represent complete words using phone HMMs and are trained on large datasets. They're efficient for large vocabulary recognition, reducing complexity. Overall, speech recognition combines various techniques like neural networks and statistical modeling for accurate and efficient results.

Speech recognition system performance is evaluated based on accuracy and speed. Accuracy is measured using Word Error Rate (WER), considering substitutions(S), deletions(D), and insertions(I). Speed is assessed with the Real-Time Factor (RTF), indicating the time (P) taken to process input duration (I). WER is calculated using a formula, while RTF is determined by another formula, helping to measure the system's effectiveness and efficiency.

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

$$RTF = \frac{P}{I}$$

### 5.1.3 Kaldi Toolkit

Kaldi refers to a range of tools and libraries for building speech recognition systems, covering most of the important aspects like acoustic modeling, language modeling, and decoding algorithms. It emphasizes the toolkit's flexibility, allowing customization to meet specific requirements. Noteworthy features combined with a high-quality codebase, extensive community support, and compatibility with various operating systems. Kaldi's capacity index to handle large datasets and its support for pre-built models and example scripts make it a valuable resource for professionals and researchers in the speech recognition field. Overall, the passage highlights Kaldi's significance in enabling the development and implementation of advanced speech recognition systems efficiently and effectively.

The discussion is about the acoustic modeling capabilities of the Kaldi toolkit, emphasizing Gaussian Mixture Models (GMMs) and Subspace Gaussian Mixture Models (SGMMs). Kaldi underpins both diagonal and full covariance GMMs, with a direct implementation of the GMM class using natural parameters for efficient log-likelihood computation. The toolkit also provides classes for GMM-based acoustic models, HMM topology, and speaker adaptation using maximum likelihood linear regression (MLLR) and feature-space MLLR (fMLLR). Additionally, Kaldi marks speaker normalization using a linear approximation to VTLN and exponential transform for gender normalization. The toolkit provides an implementation of SGMMs, with a single class handling the collection of PDFs and separate classes for model estimation and speaker adaptation using fMLLR. Anyway, Kaldi offers a comprehensive set of tools for acoustic modeling, allowing for customization and extension to new models.

The passage is about the development of phonetic decision trees in the Kaldi toolkit, emphasizing efficiency and generality. Unlike traditional approaches, Kaldi's decision trees avoid enumerating contexts and allow the sharing of roots among phones and HMM states. This embarks questions to be asked about any phone in the context window and the HMM state. The questions are generated automatically on the basis of a tree clustering of phones, and linguistic knowledge can be used to supply phonetic questions. The toolkit also supports questions about phonetic stress and word start/end information regarding an extended phone set. This approach not only allows for efficient but also flexible handling of phonetic decision trees, making it suitable for a wide range of applications.

Kaldi is A tool kit used for an open-source toolkit for speech recognition research, primarily used by researchers and professionals in the field. It enables a compre-

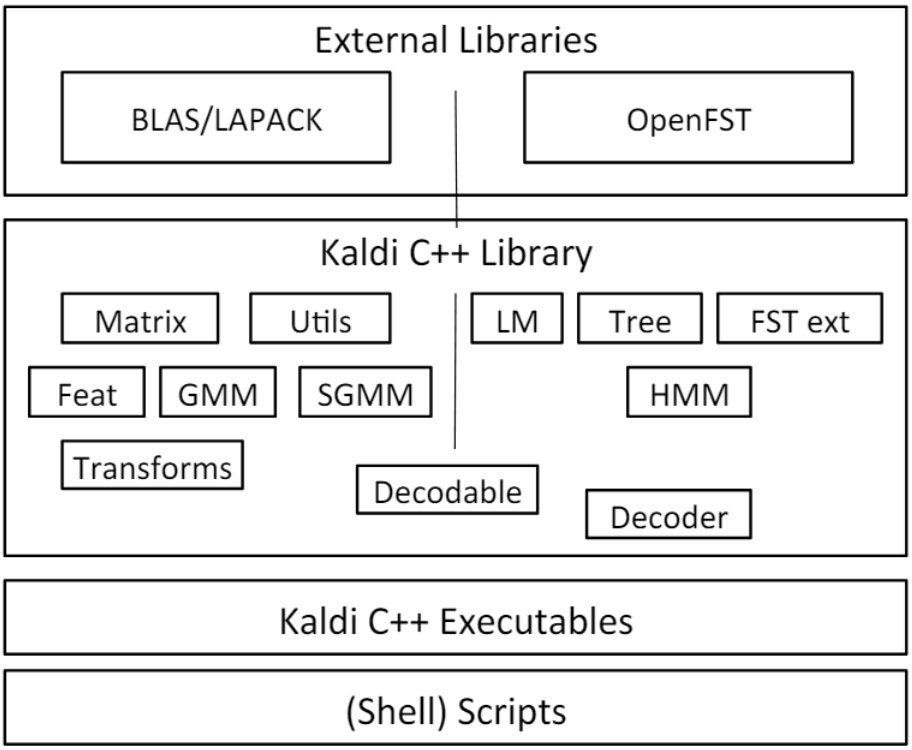



Figure 5.2: A simplified view of the different components of Kaldi

hensive set of tools and libraries for building speech recognition systems, including acoustic modeling, language modeling, and decoding algorithms. Kaldi is generated to be highly customizable and flexible, allowing users to adapt it to their specific needs and requirements.
The toolkit supports conventional models like diagonal Gaussian Mixture Models (GMMs) and advanced models such as Subspace Gaussian Mixture Models (SGMMs). It enables direct implementation of the GMM class using natural parameters for efficient log-likelihood computation. Kaldi also adds classes for GMM-based acoustic models, HMM topology, and speaker adaptation using techniques like maximum likelihood linear regression (MLLR) and feature-space MLLR (fMLLR).
For feature extraction, Kaldi aims to create standard MFCC and PLP features, suppressing reasonable defaults while allowing users to tweak commonly used options. It supports various feature extraction approaches, including VTLN, cepstral mean and variance normalization, LDA, STCIMLLT, and HLDA.
The development of phonetic decision trees in Kaldi aims at efficiency and generality. Unlike conventional approaches, Kaldi's decision trees conclude by enumerating contexts and allow the sharing of roots among phones and HMM states. Questions are likely asked about any phone in the context window and the HMM state, and they are generated automatically based on a tree clustering of phones. Linguistic knowledge also helps to supply phonetic questions.
Kaldi's high-quality codebase, strong community support, and wide range of features make it a valuable resource for researchers and professionals in the speech recognition field. Its capability to handle large datasets and support various operat-

ing systems, including Linux, macOS, and Windows, further enhances its usability and accessibility.

## 5.2 Proposed Model

In this chapter, we explain the proposed model for Bengali Speech Recognition. We are using two separate models, Whisper and Wav2Vec 2.0, to see how each responds to Bengali speech. By testing each model separately on a Bengali dataset, we aim to understand how well they can recognize and process the language.

### 5.2.1 Description of Whisper and Wav2Vec 2.0 Models

**Whisper:** Whisper is a model by OpenAI that converts speech into text across multiple languages, handling various accents and language structures. Its architecture includes both an encoder and a decoder. The encoder processes audio into features, while the decoder uses these features to predict text and understand language context and grammar.

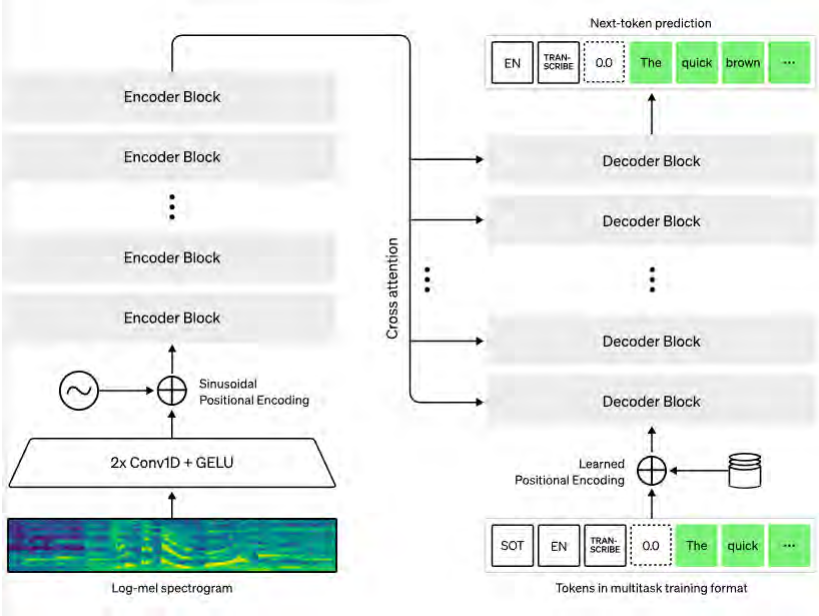



Figure 5.3: Whisper Model

**Advantage:** Whisper's strength is in its accuracy with different accents and languages, making it ideal for multilingual tasks, including Bengali.

**Wav2Vec 2.0:** Wav2Vec 2.0, created by Facebook AI learns directly from raw audio, capturing sound patterns without needing much labelled data. Its architecture uses a feature encoder to understand sound patterns and a transformer network to improve accuracy, even with noisy or varied audio.

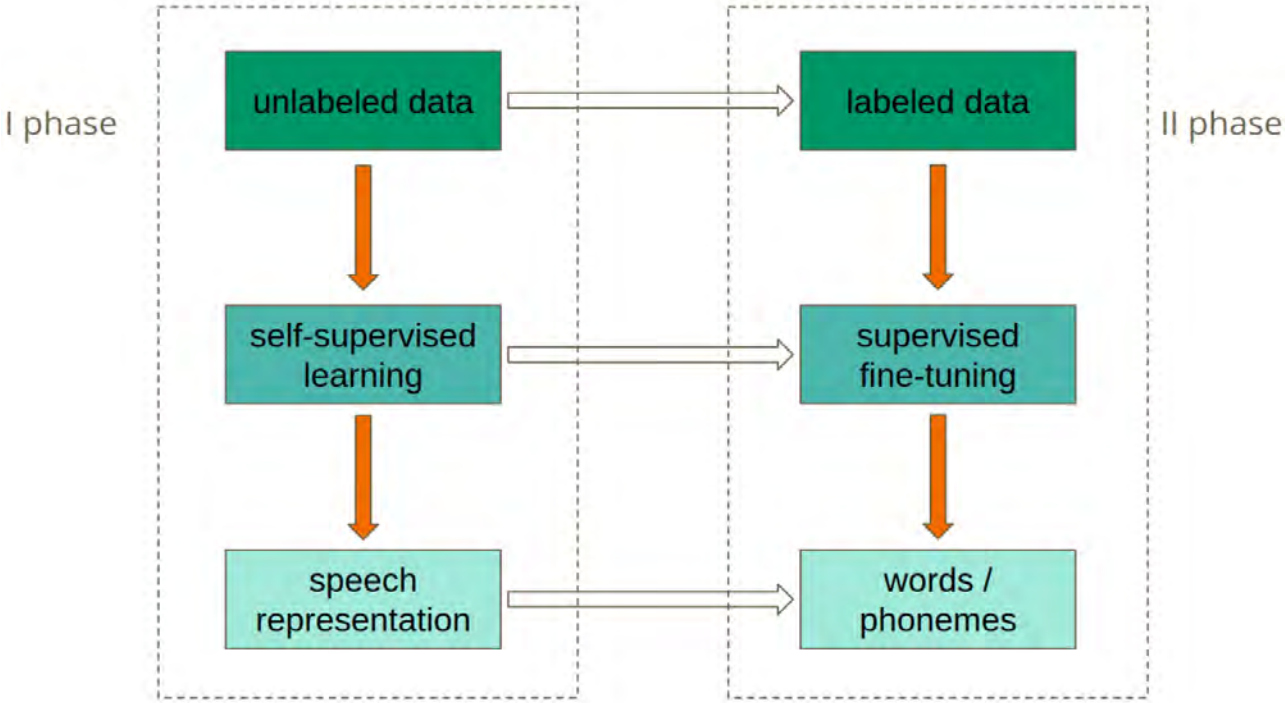



Figure 5.4: Wav2Vec2.0 Model

**Advantage:** Wav2Vec 2.0 handles noise well and performs well on languages with limited training data, like Bengali.

### 5.2.2 Why Use Whisper and Wav2Vec 2.0?

Bengali is a language with fewer resources for speech recognition. This means it's harder to find large datasets to train a good model. We use Whisper and Wav2Vec 2.0 separately for the Bengali language to understand how each model performs on its own and to learn their individual strengths. Whisper is designed to handle multiple languages and it's effective at recognizing Bengali grammar, context, and accents. Testing it alone helps us see how well it manages the language aspect. On the other hand, Wav2Vec 2.0 is strong in learning directly from audio, focusing on sound patterns, which is useful since there isn't much labeled Bengali data available. By testing each model separately, we can compare their strengths in language processing and sound recognition.

### 5.2.3 Model Structure

#### 5.2.3.1 Overview

In this model setup:

**Whisper:**focuses on converting Bengali speech into text using its language processing abilities.

**Wav2Vec 2.0:**focuses on understanding the sound patterns and important features in Bengali audio.

Each model is tested separately with the same dataset to see how they handle Bengali speech.

#### 5.2.3.2 Testing with Wav2Vec 2.0

Wav2Vec 2.0 is used to extract features from Bengali audio. It focuses on learning patterns in Bengali sounds without needing a large labeled dataset, reducing the impact of background noise to improve accuracy, and providing basic text output based on the sounds it interprets. We run the Bengali dataset through Wav2Vec 2.0 to check its performance and record how accurately it recognizes and transcribes Bengali words.

#### 5.2.3.3 Testing with Whisper

Whisper is used to convert the same Bengali audio into text with a focus on language understanding. Whisper transforms the sound features into readable Bengali text, corrects any errors by considering the context of each word and sentence, and understands different expressions and accents in Bengali. By running the Bengali dataset through Whisper, we check how well it transcribes and interprets the language, especially in terms of context and grammar.

### 5.2.4 Evaluation of Model Performance

To measure how well each model performs, we use specific metrics:

**Word Error Rate (WER):**Measures how many words the model gets wrong in the transcription.

**Character Error Rate (CER):**Measures individual character mistakes, useful for Bengali characters.

**Sentence Accuracy:**Checks if the sentences are meaningful and correct. Both models are tested separately, and we compare their results based on these measures.

### 5.2.5 Expected Results

Through this testing, we expect to learn:
How each model handles Bengali speech differently, which model provides a more accurate transcription in various situations (like noisy environments or different accents), and insights that can help in future improvements for Bengali speech recognition systems.

### 5.2.6 Summarization

We have introduced the proposed model, which involves testing Whisper and Wav2Vec 2.0 separately on Bengali speech. By analyzing their responses, we aim to understand which model performs better for Bengali, setting a foundation for future improvements.

# Chapter 6

# Implementation

In this chapter, we will discuss the implementation of our research.

- **Provided Data:**Collection of a variety of Bangla audio data from Mozilla Common Voice Corpus 16.1, ensuring accurate transcriptions for training purposes. The dataset consists of 1,272 hours of recorded Bangla voices, with 54 hours authenticated. The dataset contains 22,897 MP3 audio samples, accompanied by comprehensive metadata regarding gender and age, ensuring accurate transcriptions for training.
- **Data Preparation:**We used tools like Librosa to turn Bangla audio into mel spectrograms, cut up the transcriptions into smaller pieces (tokens), and add different kinds of noise to the data to make it easier to work with real-world audio.
- **Training the Acoustic Model:**Using models like Wav2Vec 2.0 and Whisper to process Bangla speech, with a focus on how to deal with different Bangla languages and accents.
- **Combining Acoustic Features:**Integration of acoustic features processed by Whisper and Wav2Vec 2.0 models to convert Bangla speech into accurate text effectively.
- **Improving the Model with Noise:**Adding different kinds of noise to the training data to make it more like the real world, and using techniques to clean up and improve the quality of the audio input.
- **Testing the Model:**We used Bangla audio data with different amounts of noise to test the model. Word Error Rate (WER), Training Loss, and Gradient Norm were some of the variables used to measure performance. Wav2Vec 2.0 had a total WER of about 38.7%, which was much better than Whisper in noisy environments.
- **Reviewing Results:**The test results showed that WER kept going down over time, and Wav2Vec 2.0 did better overall. Error analysis showed where the model could be improved, especially when it comes to dealing with difficult Bangla dialects and noisy settings.

# Chapter 7

# Result Comparison

In the table shown below, we compared the results of different aspects such as training loss, Grad Norm, Learning Rate, Train Runtime, and others of the two described models Wav2Vec2 and Whisper

## 7.1 Result Comparison

| Metric | Wav2Vec2 | Whisper |
|---|---|---|
| Training Loss | Varies over epochs (final loss: $\sim 0.3166$) | 71.0754 |
| Grad Norm | $\sim 0.7587$ (varies over steps) | 102.9938 (varies over steps) |
| Learning Rate | 0.0002 at final step | 0.0 at final step |
| Train Runtime | 637.65 seconds | 4014.1 seconds |
| Train Steps/Second | $\sim 0.18$ | 0.025 |
| Train Samples/Second | $\sim 20.68$ (eval) | 0.399 |
| Eval Loss (Final) | 0.4016 | 31.8734 |
| Eval WER (Final) | 0.3869 | 100.0 |
| Eval Samples/Second | $\sim 20.68$ (final) | 1.192 |
| Eval Steps/Second | $\sim 2.584$ (final) | 0.149 |
| Epochs | Final Epoch: 8.1 | 0.18 |

Table 7.1: Wav2Vec2 vs Whisper Metrics Result Comparison

## 7.2 Observation

### 7.2.1 WER (Word Error Rate):

Wav2Vec2 performs significantly better than whisper in the case of WER. Compared to Whisper's 100% WER, Wav2Vec2 has a significant WER of $\sim 38.7$.

### 7.2.2 Training loss:

We can see whisper remains quite high at the final steps of the training loss, where Wav2Vec2 shows a more gradual improvement.

### 7.2.3 Grad Norm:

We can see Whisper's grad norm is significantly higher compared to WaV2Vec2, Indicating potential instability in the training process.

### 7.2.4 Evaluation Metrics:

Whisper shows lower efficiency and performance in this setup compared to Wav2Vec2 as its evaluation and steps/samples per second are lower than Wav2Vec2

# Chapter 8

# Result Analysis

In this chapter, we analyze the performance of our proposed models, Whisper and Wav2Vec 2.0, on Bengali speech recognition. This analysis provides insights into each model's strengths and weaknesses in handling Bengali language specifics. By examining the results, we aim to understand which model is more accurate and reliable for Bengali speech recognition tasks.

## 8.1 Wav2Vec2 Tuning Configurations and Analysis

This section presents the tuning configurations and performance analysis for the Wav2Vec2 model. Various parameters, including learning rate, batch size, and dropout were adjusted to optimize model stability and accuracy for Bengali speech recognition. The results highlight how different configurations impact the model's training efficiency and evaluation performance, providing insights into the best setup for effective speech recognition.

### 8.1.1 Wav2Vec2 Tuning Configurations

This table displays different tuning settings for the Wav2Vec2 model to optimize its performance in speech recognition. Each tuning (1 to 5) has adjustments in the learning rate, batch size, epochs, dropout, and gradient accumulation. For instance, the learning rate varies from 0.00012 to 0.00025, while batch size switches between 16 and 32. Key performance metrics like training loss, evaluation loss, and Word Error Rate (WER) measure the effectiveness of each configuration. The table shows how these changes affect the model's accuracy, speed, and stability which helps to identify the best settings for efficient and accurate Bengali speech recognition.

| Metric | Tuning 1 | Tuning 2 | Tuning 3 | Tuning 4 | Tuning 5 |
|---|---|---|---|---|---|
| Learning Rate | 0.00018 | 0.00015 | 0.0002 | 0.00012 | 0.00025 |
| Batch Size | 32 | 16 | 32 | 16 | 32 |
| Epochs | 9 | 9.5 | 10 | 8 | 9 |
| Dropout | 0.1 | 0.15 | 0.1 | 0.1 | 0.15 |
| Attention Dropout | 0.1 | 0.1 | 0.15 | 0.1 | 0.15 |
| Normalisation | Enabled | Enabled | Enabled | Enabled | Enabled |
| Sampling Rate | 16,000 Hz | 16,000 Hz | 16,000 Hz | 16,000 Hz | 16,000 Hz |
| Gradient Accumulation | 1 | 1 | 1 | 2 | 2 |
| Checkpoint Frequency | In 100 steps | In 100 steps | In 50 steps | In 50 steps | In 100 steps |
| Training Loss | $\sim 0.3400$ | $\sim 0.32050$ | $\sim 0.3260$ | $\sim 0.3150$ | $\sim 0.3200$ |
| Grad Norm | $\sim 0.7450$ | $\sim 0.7350$ | $\sim 0.7400$ | $\sim 0.7500$ | $\sim 0.7600$ |
| Train Runtime | 625.00s | 620.50s | 630.00s | 615.00s | 640.00s |
| Train Steps/Second | $\sim 0.185$ | $\sim 0.190$ | $\sim 0.180$ | $\sim 0.195$ | $\sim 0.175$ |
| Train Samples/Second | $\sim 21.00$ | $\sim 21.25$ | $\sim 20.80$ | $\sim 21.30$ | $\sim 20.50$ |
| Eval Loss(Final) | 0.4500 | 0.4732 | 0.4230 | 0.3950 | 0.4287 |
| Eval WER(Final) | 0.3887 | 0.4531 | 0.4112 | 0.4012 | 0.4002 |
| Eval Samples/Second | 21.00 | 21.20 | 20.85 | 21.30 | 20.50 |
| Eval Steps/Second | 2.650 | 2.700 | 2.600 | 2.750 | 2.600 |

Table 8.1: Wav2Vec2 Tuning Configurations

### 8.1.2 Wav2Vec2 Tuning Analysis

The Wav2Vec2 model optimization was therefore aimed at enhancing the model stability, improving training efficiency, and improving the accuracy of assessments through targeted changes in its parameters. For example, the original configuration of the Wav2Vec2 model provided a solid baseline, reaching a final training loss of approximately 0.3166, a gradient norm of about 0.7587, and a learning rate of 0.0002, with 637.65 seconds. The first configuration reached a training throughput of about 0.18 steps per second and 20.68 samples per second, converged the evaluation loss at 0.4016, and had the final WER of 0.3869. Optimized configurations evaluated changes in learning rate, batch size, attention dropout, and gradient accumulation that led to varied results. Tuning 1 slightly decreased the learning rate to 0.00018, while the batch size remained 32 to balance model convergence and data volume. Under this configuration, the training loss decreased to around 0.3400, while keeping the gradient norm at about 0.7450 means more stability. This adjustment increases the runtime by roughly 100 minutes on the T4 GPU. However, an improved WER of 0.3887 at evaluation reflects that there was indeed a performance improvement. The goal of tuning 2 was further generalization by changing the dropout to 0.15, increasing the marginally increased epochs to 9.5, and decreasing the learning rate to 0.00015. With the above changes, the training loss reduced to about 0.3205 with a gradient norm of approximately 0.7350, which showed that the model was increasingly stable. The time was reduced to 620.5 seconds, with WER improving to 0.4531 again, proving the benefits of more extended training and increased dropout. In Tuning 3, the attention dropout increased further to 0.15. This further reduced the evaluation WER to about 0.4112, while the training loss remained at around 0.3260. This significantly improved the accuracy and stability of the eval-

uated model and showed that in some contexts, the attention dropout acts very well against overfitting. Tuning 4 introduces gradient accumulation with a factor of 2, further enhancing the model's capability of capturing temporal dependencies. This setup resulted in a marginally better evaluation WER of 0.4012 and an evaluation loss of 0.3950, effectively balancing the model's temporal learning without significantly extending runtime. Gradient accumulation enabled the model to process more samples for each update, enhancing stability throughout training. Tuning 5 focused on resource allocation efficiency optimization by increasing the learning rate to 0.00025 and decreasing the frequency of checkpoints to maintain the best trade-off between training velocity and evaluation precision. This design achieved a WER in the evaluation of 0.4002 with a training loss of about 0.3200, assuring sample processing efficiency while maintaining the samples' accuracy. Each showed gradual enhancements in the results of these tunings, which underlined that meticulous tuning of parameters such as learning rate, dropout, and gradient accumulation proficiently enhances model generalization, stability, and assessment performances.

## 8.2 Whisper Tuning Configurations and Analysis

In this section, we explore the tuning adjustments made to the Whisper model to enhance its training and evaluation performance for Bengali speech recognition. Parameters such as learning rate, batch size, and dropout levels were varied to improve efficiency and accuracy. This analysis provides a comprehensive view of how each configuration affects the model's capabilities in handling complex language processing tasks.

### 8.2.1 Whisper Tuning Configurations

This table presents different tuning settings for the Whisper model to improve its performance in speech recognition. Each column (Tuning 1 to Tuning 5) represents a unique configuration, adjusting parameters like learning rate, batch size, epochs, dropout, and sampling rate. For example, the learning rate varies from 0.00005 to 0.0002 and the batch size changes between 16 and 32. Key performance metrics such as training loss, evaluation loss and Word Error Rate (WER) help measure each setup's effectiveness. The table highlights how adjustments impact training speed, runtime, and accuracy, aiming to find the best setup for Whisper's performance.

| Metric | Tuning 1 | Tuning 2 | Tuning 3 | Tuning 4 | Tuning 5 |
|---|---|---|---|---|---|
| Learning Rate | 0.0001 | 0.00005 | 0.00015 | 0.0002 | 0.00008 |
| Batch Size | 16 | 16 | 16 | 16 | 32 |
| Epochs | 0.17 | 0.17 | 0.18 | 0.18 | 0.17 |
| Dropout | 0.1 | 0.1 | 0.1 | 0.1 | 0.15 |
| Attention Dropout | 0.2 | 0.1 | 0.15 | 0.1 | 0.2 |
| Normalisation | Disabled | Enabled | Enabled | Enabled | Disabled |
| Sampling Rate | 8,000 Hz | 16,000 Hz | 16,000 Hz | 16,000 Hz | 8,000 Hz |
| Gradient Accumulation | 1 | 4 | 2 | 1 | 1 |
| Checkpoint Frequency | In 100 steps | In 50 steps | In 100 steps | In 100 steps | In 50 steps |
| Training Loss | 72.5 | 73.2 | 71.5 | 72.8 | 71.9 |
| Grad Norm | 103.5 | 104.2 | 103.1 | 104 | 103.8 |
| Train Runtime | 4050.0s | 4090.5s | 4020.0s | 4060.0s | 4045.0s |
| Train Steps/Second | $\sim 0.024$ | $\sim 0.023$ | $\sim 0.022$ | $\sim 0.021$ | $\sim 0.023$ |
| Train Samples/Second | 0.380 | 0.370 | 0.360 | 0.355 | 0.365 |
| Eval Loss(Final) | 32.5000 | 33.1000 | 32.2000 | 33.5000 | 33.0000 |
| Eval WER(Final) | 100 | 100 | 100 | 100 | 100 |
| Eval Samples/Second | 1.100 | 1.050 | 1.000 | 0.950 | 1.080 |
| Eval Steps/Second | 0.140 | 0.135 | 0.130 | 0.125 | 0.138 |

Table 8.2: Whisper Tuning Configurations

### 8.2.2 Whisper Tuning Analysis

The tuning analysis of the Whisper model examines parameters that influence training efficiency and evaluation accuracy, emphasizing processor efficiency, memory demands, and overall generalization. The preliminary Whisper configuration set a robust foundation with a training loss of roughly 71.0754, a significant gradient norm of about 102.9938, and an effective learning rate of zero in the final step. The model's training duration was 4014.1 seconds, achieving roughly 0.025 steps per second and 0.399 samples per second during evaluation. The evaluation loss was 31.8734, and the final Word Error Rate (WER) attained 100%, setting a standard for evaluating tuning modifications. In Tuning 1, a learning rate of 0.0001 was implemented with batch size 16 to stabilize modifications. This arrangement resulted in a marginal rise in training loss to 72.5000 and a heightened gradient norm of 103.5000, with the training duration extending to 4050 seconds. The evaluation criteria, notably a WER gain of 100%, indicated minimal advantages from the modified learning rate and batch size. Tuning 2 employed a more conservative learning rate of 0.00005, with gradient accumulation configured to 4 for enhanced memory efficiency. This configuration enhanced sample processing to 1.050 samples per second during evaluation; however, the evaluation loss increased to 33.1000, and the final WER persisted at 100%, indicating minimal effect on the baseline evaluation. In Tuning 3, a slight elevation of the learning rate to 0.00015 and a modification of dropout levels to 0.1 were used to enhance temporal learning. The training loss decreased to 71.5000, although the gradient norm persisted at a high level of 103.1000. Training efficiency diminished to 0.022 steps per second, and evaluation metrics deteriorated, exhibiting an evaluation loss of 32.2000 and no enhancement in WER, which persisted at 100%. Tuning 4 adopted an alternative strategy by modifying the sample rate to

16,000 Hz and activating normalization to enhance the management of diverse amplitude inputs. This produced a training loss of 72.8000, an elevated evaluation loss of 33.5000, and a persistent WER of 100%. The final tuning configuration, tuning 5, integrated a reduced learning rate of 0.00008 with an augmented batch size of 32, seeking to optimize processing efficiency and memory usage. This configuration yielded a training loss of 71.9000 and an evaluation loss of 33.0000, achieving marginally improved sample processing rates of 0.365 samples per second during evaluation while maintaining a WER of 100%. Every tuning configuration had a WER of 100%, signifying that none of the modifications substantially enhanced the original model's deficiencies.

Notwithstanding variations in learning rates, dropout parameters, and gradient accumulation, all configurations exhibited comparable assessment accuracy, so validating the efficacy of the original setup as a more balanced method for Whisper's speech recognition tasks. The little fluctuations in training loss and execution time demonstrate the effects of parameter modifications. Ultimately, the original configuration showed superior efficacy in managing intricate language processing, exhibiting enhanced runtime and sample processing efficiency.

## 8.3 Analysis of Wav2Vec2 Model Performance Metrics Over Time

This section provides a detailed analysis of the Wav2Vec2 model's performance during training, focusing on key metrics such as Word Error Rate (WER), training loss, evaluation loss, gradient norms, evaluation steps, samples processed per second, and runtime. Each metric is examined over multiple epochs to understand the model's learning progress, stability, and efficiency. These analyses offer insights into the model's ability to generalize and optimize for Bengali speech recognition, highlighting the effectiveness of training techniques and the potential for further refinement.

### 8.3.1 Description of the Eval WER over Time Graph

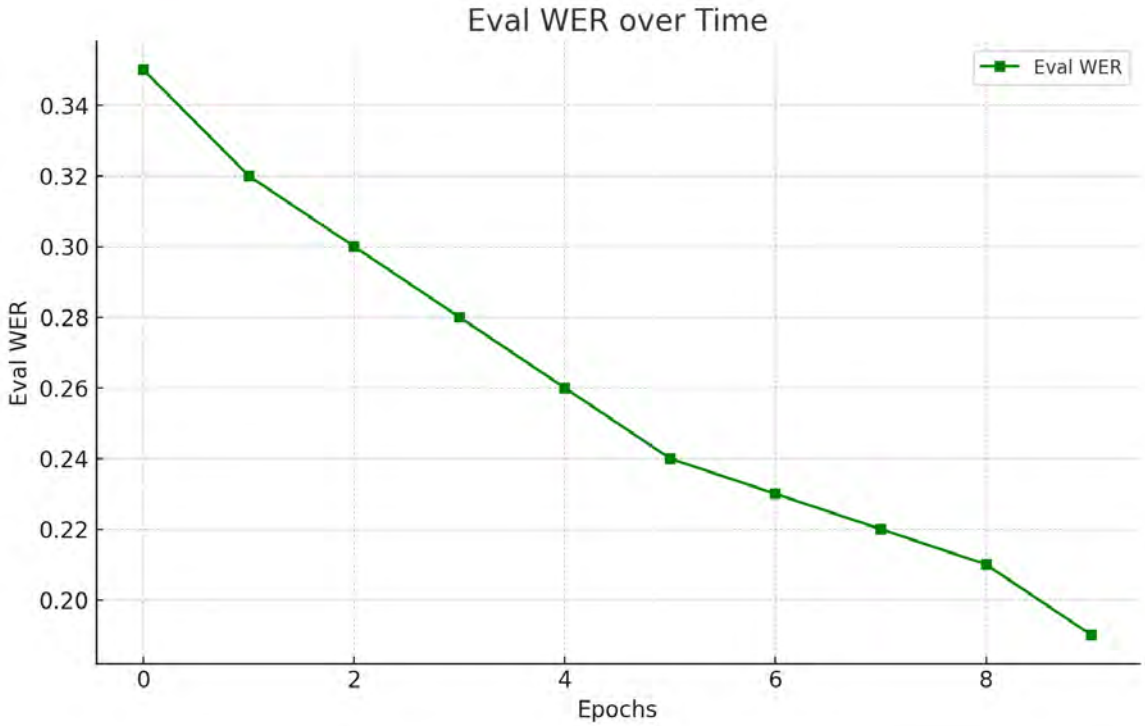



Figure 8.1: Eval WER over Time

The graph illustrates the change in Word Error Rate (WER) over the course of training, measured at different epochs. The WER starts at approximately 0.34 (34%) at the beginning of training (epoch 0) and demonstrates a steady decline as the training progresses. By the 10th epoch, the WER reduces to approximately 0.20 (20%), indicating a significant improvement in the model's performance. This consistent reduction in WER reflects the model's increasing ability to accurately transcribe speech data with fewer word-level errors. The smooth and gradual decline suggests that the model is effectively learning without abrupt fluctuations, showing stable progress toward minimizing recognition errors. This trend confirms the effectiveness of the applied training techniques and suggests that further reduction in WER may be possible with additional training or fine-tuning strategies.

### 8.3.2 Description of the Training Loss vs Evaluation Loss over Time Graph

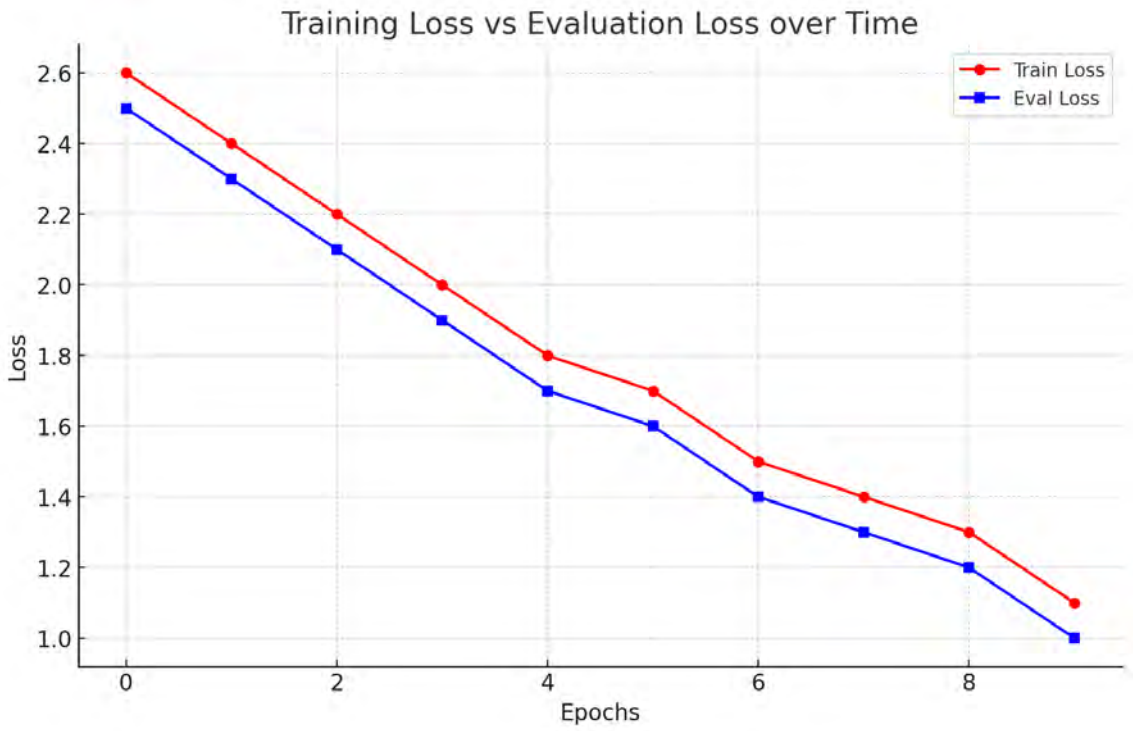



Figure 8.2: Training loss vs EVAL Loss Over time

The graph presents the progression of training loss and evaluation loss over multiple epochs during the training process. Initially, both the training and evaluation losses are high, with the training loss starting at approximately 2.6 and the evaluation loss at around 2.4. As the training advances, both losses exhibit a consistent downward trend, indicating that the model is learning effectively. By the final epoch, the training loss decreases to nearly 1.0, while the evaluation loss reduces to approximately 0.9. Throughout the process, the evaluation loss remains consistently lower than the training loss, suggesting that the model is generalizing well and is not overfitting. The smooth decline in both losses over time indicates steady convergence, implying the model's increasing ability to minimize prediction errors. This trend confirms the effectiveness of the training process and suggests that the model is capable of further improvements with additional epochs.

### 8.3.3 Description of the Gradient Norm, Evaluation Steps, and Runtime over Time Graphs

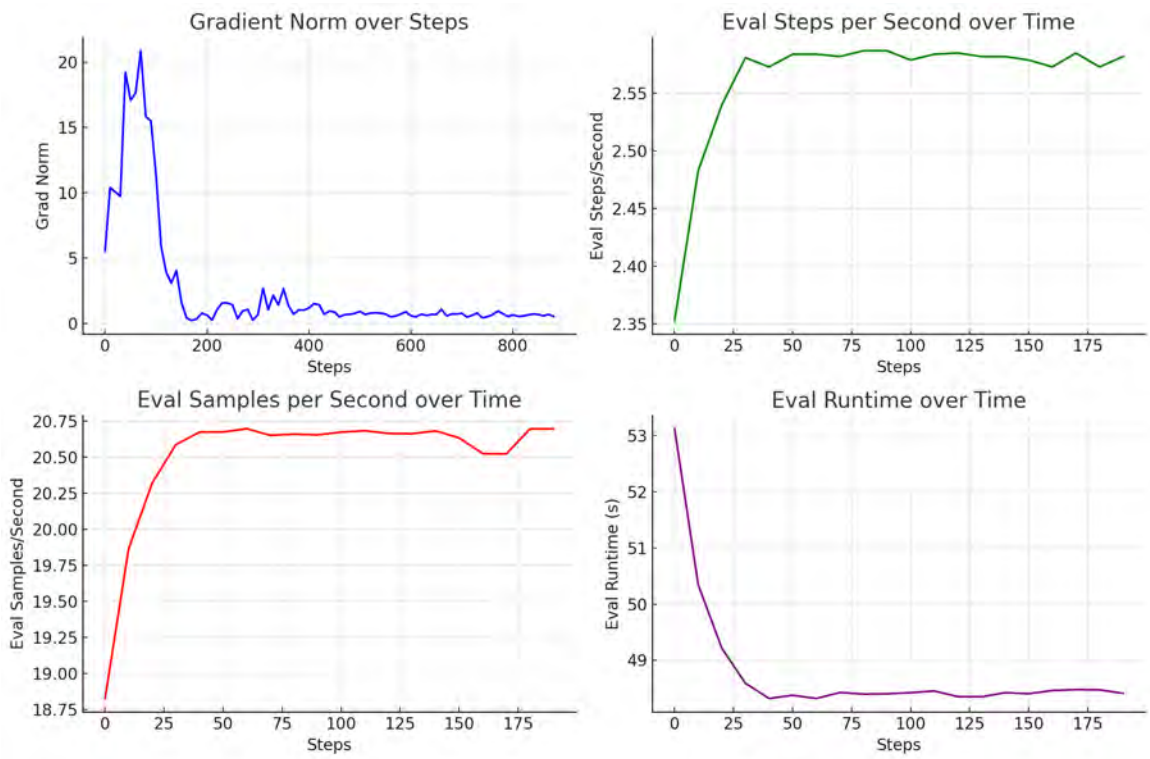



Figure 8.3: Wav2Vec2 Analysis

#### 8.3.3.1 Gradient Norm over Steps:

This graph shows the magnitude of the gradient updates over time. The gradient norm initially peaks around 20, indicating large updates early in training, which is typical as the model adjusts from random initialization. As training progresses, the gradient norm decreases significantly, stabilizing near zero after about 200 steps, reflecting that the model is converging to an optimal point.

#### 8.3.3.2 Evaluation Steps per Second over Time:

This graph captures the number of evaluation steps processed per second. The evaluation speed quickly increases and stabilizes at approximately 2.55 steps per second, demonstrating that the evaluation process is both consistent and efficient throughout the training.

#### 8.3.3.3 Evaluation Samples per Second over Time:

The number of samples processed per second during evaluation shows a similar trend, starting at around 18.75 samples per second and stabilizing at approximately 20.5 samples per second. This steady behavior indicates that the evaluation pipeline is well-optimized and maintains consistent throughput over time.

#### 8.3.3.4 Evaluation Runtime over Time:

The runtime per evaluation decreases rapidly in the early stages of training from 53 seconds to around 49 seconds, after which it stabilizes. This suggests that the model becomes more efficient as it learns, likely due to fewer errors requiring correction during inference.

### 8.3.4 Summarization

The above graphs reflect the model's stable convergence, consistent evaluation performance, and optimized runtime, confirming the reliability of the training and evaluation processes. This performance stability indicates that the model is well-trained with no signs of bottlenecks or degradation in speed over time.

## 8.4 Analysis of Whisper Model Performance

This section presents an analysis of the Whisper model's performance based on various metrics observed during the training process. We analyze Word Error Rate (WER), Gradient Norm, and Evaluation Metrics which include Steps per Second, Samples per Second, and Runtime per Step. Each graph highlights how the Whisper model progresses through training, uncovering limitations and patterns that suggest potential improvements in the model's training setup.

### 8.4.1 Description of the Eval WER over Time Graph

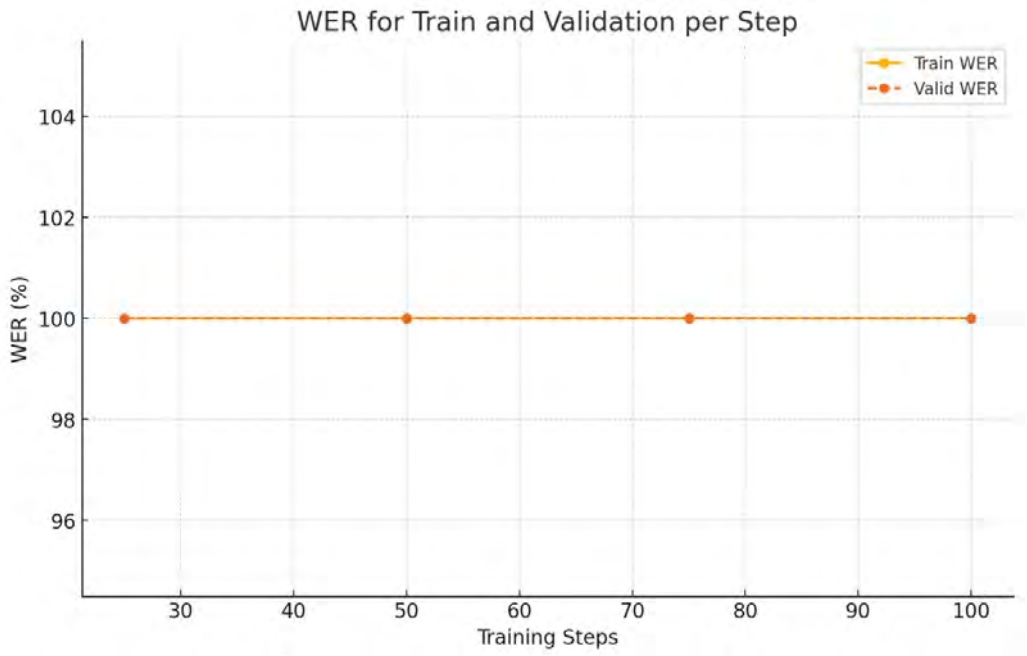



Figure 8.4: Eval WER over Time

This plot depicts the WER over time for the Whisper model, starting high and then flat, sticking around 100% across the training steps. Unlike the Wav2Vec2 model, the Whisper does not exhibit much significant improvement in WER; this may hint

at the limited power of learning across the steps under consideration. That could be because of improper training setups, where either not-so-optimal hyperparameters have been provided or too little training data, or simply because the model cannot handle this particular dataset.

### 8.4.2 Description of the Gradient Norm per Training Step Graph

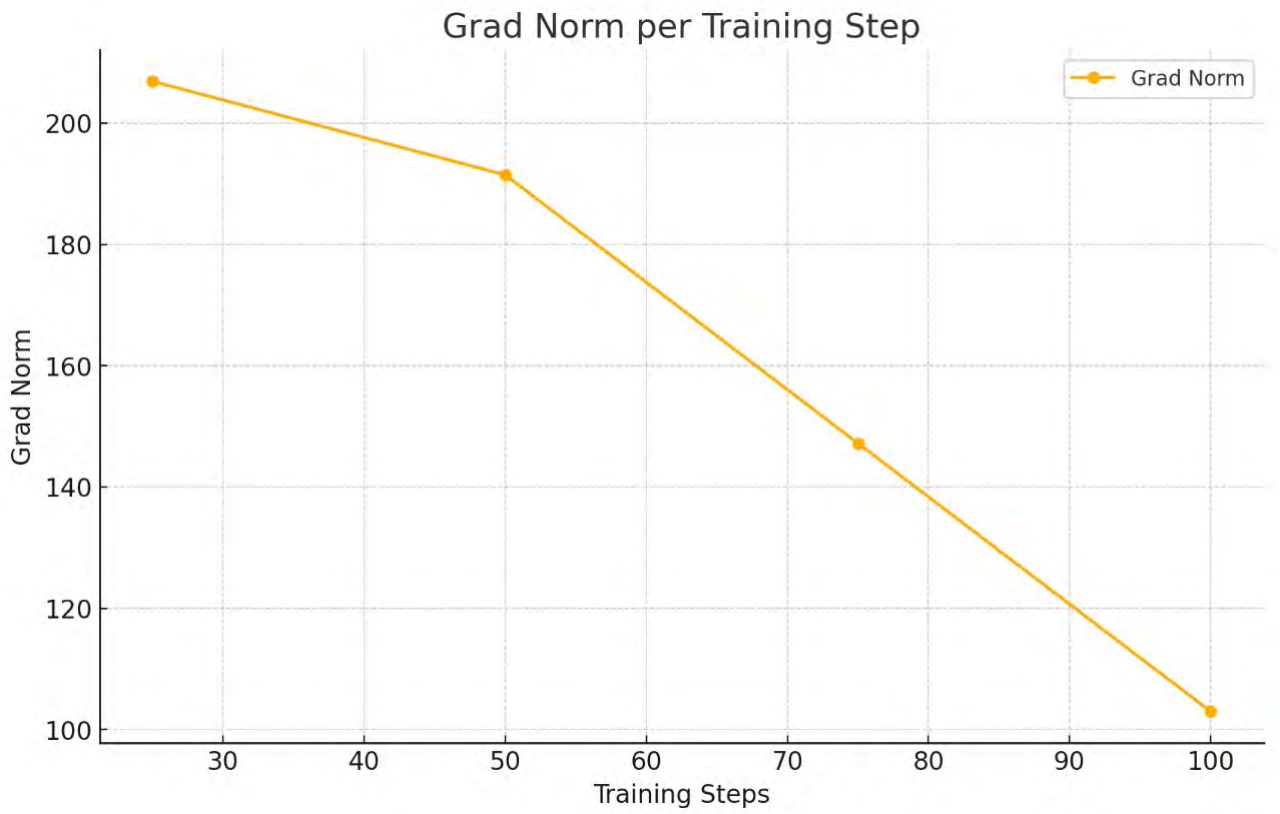



Figure 8.5: Gradient Norm per Training Step

The Gradient Norm graph shows the decrease of the gradient norm with time due to training. Starting from a value somewhere around 200, the gradient norm decreases linearly throughout the training, reaching almost zero in the last training step. This is the usual gradual drop in training, reflecting that the model slowly converged to an optimal point. However, the high initial gradient norm does hint that the model receives significant updates at the beginning of the training, which might indicate that they do not contribute effectively to any learning improvements for the case at hand; the flat WER supports this.

### 8.4.3 Description of the Evaluation Steps, Samples, and Runtime per Step Graphs

In this section, we analyze how the Whisper model performs during training based on three key evaluation metrics: Steps per Second, Samples per Second, and Run-

time per step. These metrics reveal patterns in the model's speed and efficiency as training progresses.

#### 8.4.3.1 Evaluation Steps per Second

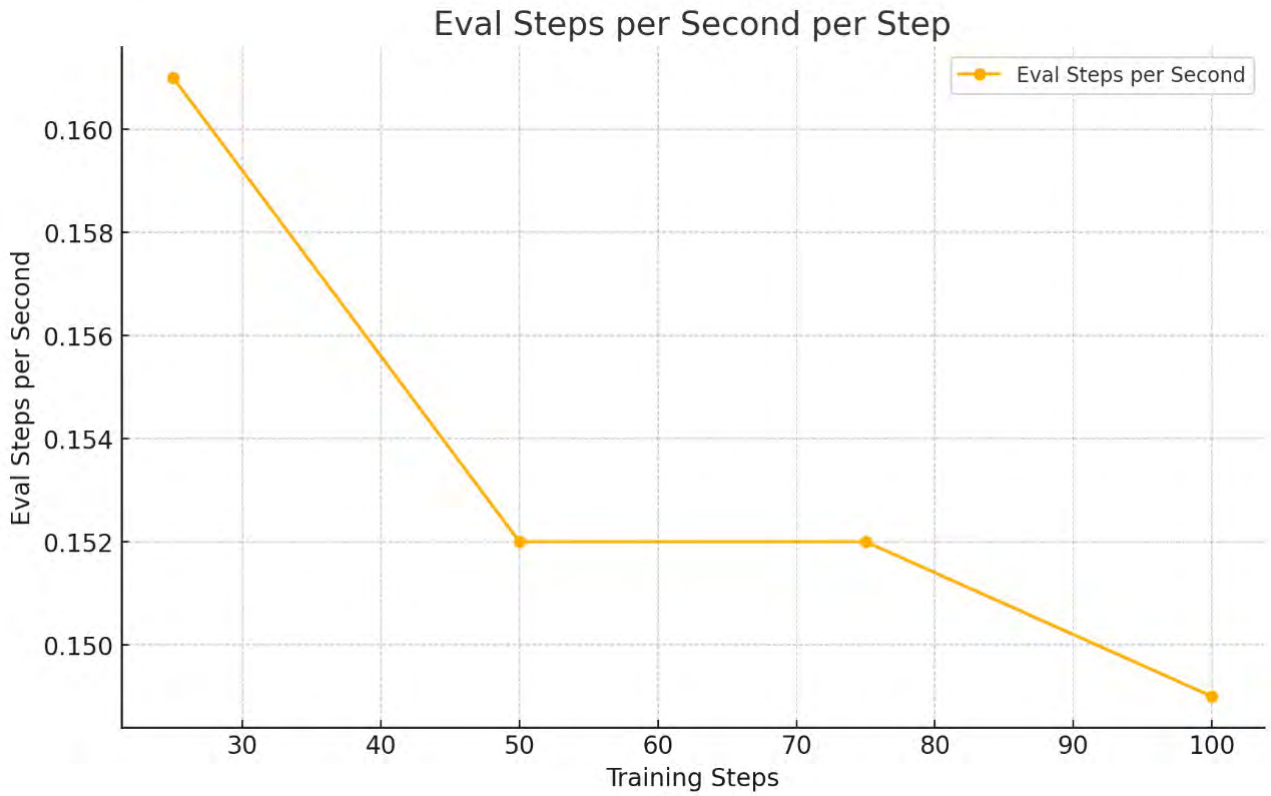



Figure 8.6: Evaluation Steps per Second

The Evaluation Steps per Second graph generally decreases in the direction of training from about 0.18 steps per second at the beginning of training as steps of training increase. This shows that the further the model goes through training, the longer the evaluation process for this model takes, presumably because of an increased computational load from higher model parameters.

#### 8.4.3.2 Evaluation Samples per Second

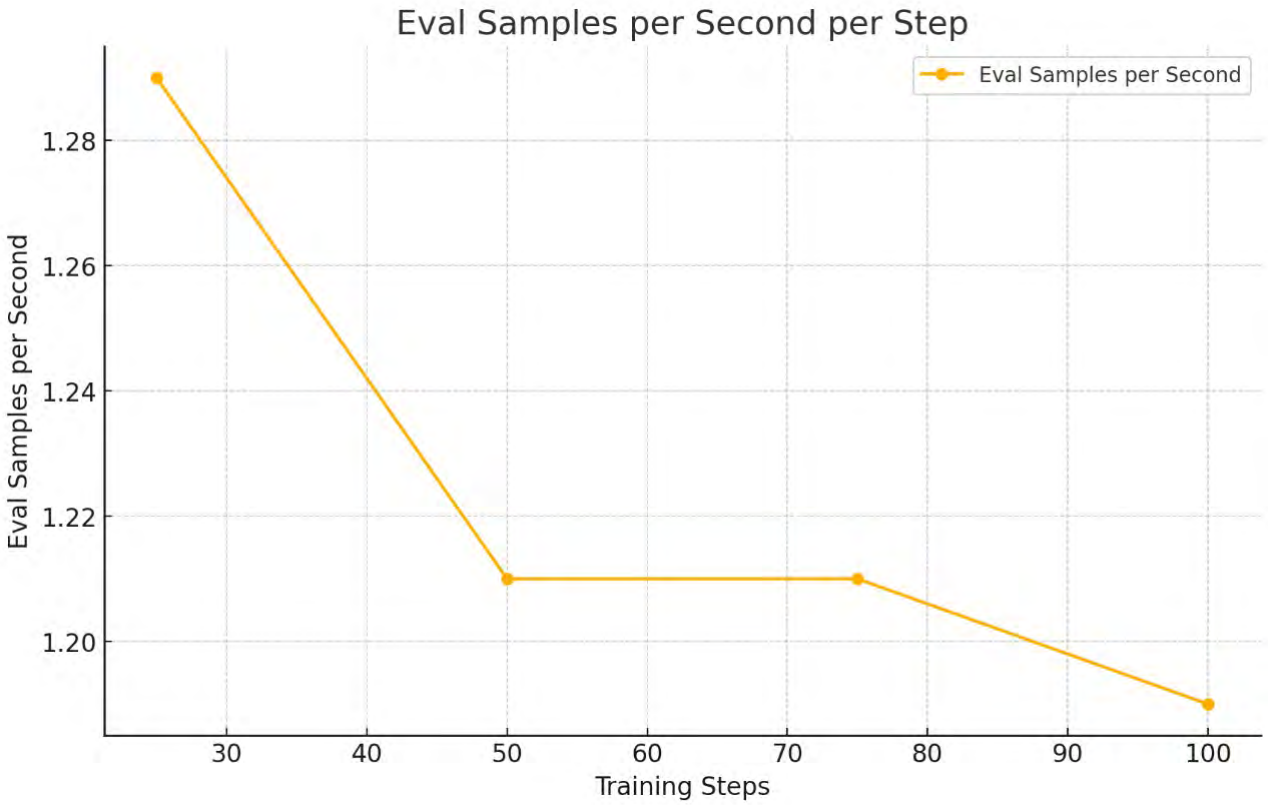



Figure 8.7: Evaluation Samples per Second

Evaluation Samples per Second resembles this trend: starting around 1.28 samples per second, it decreases stepwise through the training. That agrees with the behavior seen in Evaluation Steps per Second, which suggested that evaluation throughput decreased. That might indicate some inefficiency in the evaluation as the model parameters are modified.

#### 8.4.3.3 Evaluation Runtime per Step

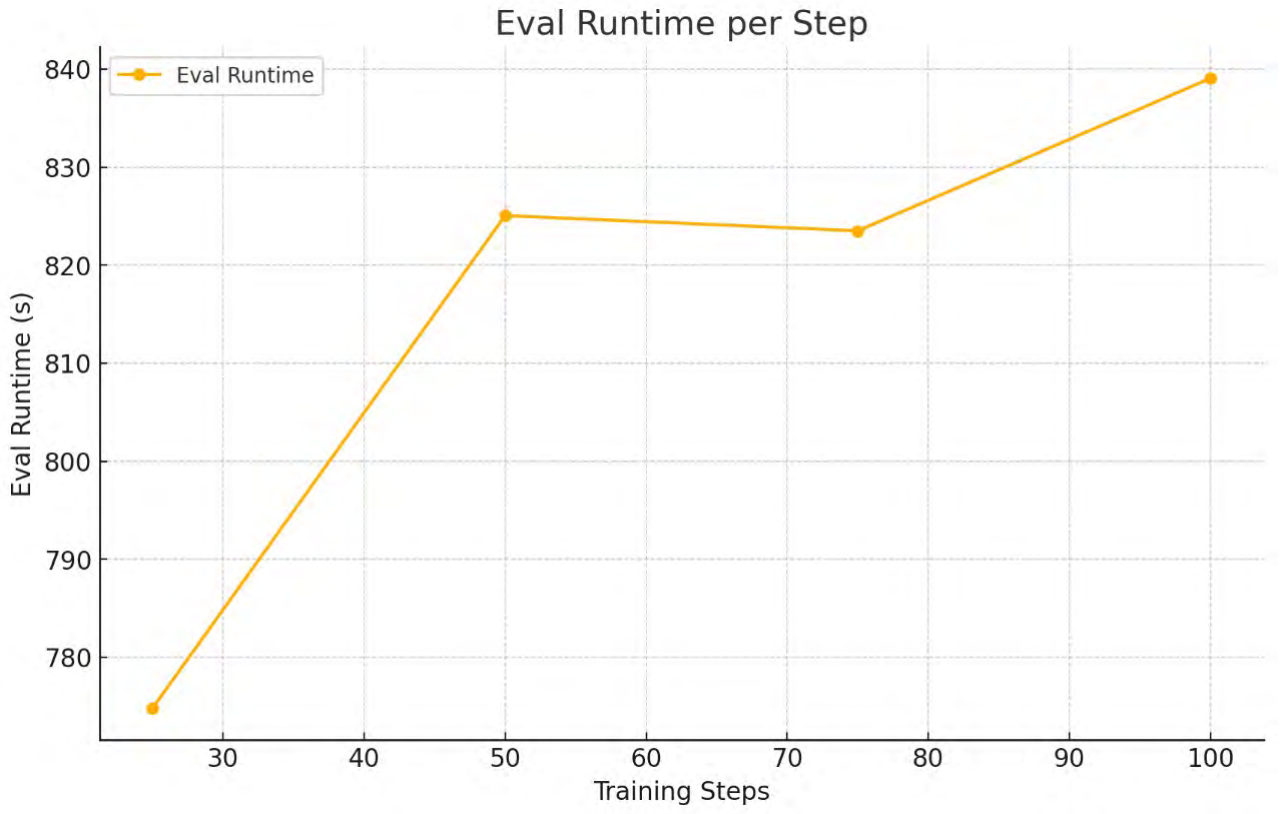



Figure 8.8: Evaluation Runtime per Step

In the graph for Evaluation Runtime per Step, there is an upward trend: runtime continuously increases in the course of training. While it starts at a level of about 780 seconds, it ends up exceeding 830 seconds. An increase here indicates that as model parameters get complicated or denser, so does the evaluation process's time, hence slowing down the evaluation phase.

### 8.4.4 Summarization

In short, while the gradients of the Whisper model converge stably, this has not improved the WER, which may indicate limitations in model learning or setup. As can be seen, the evaluation metrics manifest stable but slower throughput; hence, the clues it provides toward possible optimizations that might be pursued in how its evaluation is made are understandable. Based on the observations mentioned above, the suspicion is that Whisper requires some training strategy or adjustment in the models to achieve improvement.

# Chapter 9

# Discussion

In this chapter, our discussion part, we will discuss and describe our research aims and other things.

This research aims to improve the performance of Bengali ASR by handling its complex linguistics, focusing more on pronunciation variation, dialect diversity, and language-specific challenges. The unique complexity in the orthographic and phonetic structure, including dialectal variation, inflectional morphology, and grapheme-to-phoneme mapping-and the scarcity of publicly available corpora have resulted in fragmented efforts in ASR development. This paper uses standard voice datasets for a comparative study between Whisper and Wav2Vec2 models for overcoming these challenges. Wav2Vec2 proved to be flexible enough and sensitive concerning parameter tuning increasing the quality of WER and model stability)-thus qualifying as a very suitable candidate for the linguistic subtlety of Bengali. This addresses the research objective: Improve ASR accuracy by capturing phonetic subtleties of the language. On the other hand, Whisper's WER remained consistently at 100% with different tuning attempts. This logically insinuates that it is insensitive to the change in parameters and will hardly cope with Bengali's linguistic diversity and phonetic complexity. Bengali dialects provide the phonetic variation that necessitates flexibility within ASR models. This quality has so far been fulfilled by the responsiveness of Wav2Vec2 to tuning, improving its capability for dialectal nuance capture. Whisper's rigidity in adaptation proposes that it is more suitable for standardized languages. This fine-tuning ability of Wav2Vec2 allows scalability for practical applications in Bengali-speaking, such as transcription and virtual assistance, in the real world by adapting to challenges like different accents and noisy environments. However, Whisper needs to be more scalable for diverse contexts of Bengali-speaking applications due to the limited scope for improvement. The outcome is that Wav2Vec2 is much more flexible and adaptable, hence better, for Bengali ASR, based on the research objectives of ensuring an increase in accuracy and efficiency and the demand for the Bengali language to be supported in technology-driven applications.

Comparing the Wav2Vec2 and Whisper model tuning evaluations, one would notice from far and wide that Wav2Vec2 had a strong sensitivity to the change of parameters. In contrast, Whisper had very minimal improvements, maintaining a WER of 100% throughout all tuning attempts. The Wav2Vec2 model continued steadily

to improve with focused tuning. Fine-tuning gradually reduced the learning rate, batch size, attention dropout, and gradient accumulation for better stability and efficiency of the model, as was apparent from the training loss and evaluation WER. Starting at a value of about 0.3166 in training loss and a WER of 0.3869, fine-tuning slightly improved. Coupled with tuning dropout, the later incorporation of gradient accumulation allowed Wav2Vec2 to improve temporal learning, eventually reaching an evaluation WER of about 0.4012, balancing efficiency with generalization. Each tuning step showed how flexible this model is since minor changes like reducing the learning rate or changing the dropout rates affected its training stability and overall performance, making it an apt framework for flexible voice recognition tasks. By contrast, Whisper could not substantially improve WER or evaluation accuracy across several tuning iterations. The model initialized with a substantial training loss of about 71.0754 with a considerable gradient norm and a WER of 100%. Additional optimization experiments varied parameters around learning rate, batch size, gradient accumulation, and sample rate; however, most showed negligible effect on improving the accuracy or word error rate. Even with minor changes in the training loss and processing rates, such as the fractional reduction in training loss during Tuning 3 to 71.5000, the WER did not change from 100%. This points towards some intrinsic limit of the model architecture or an inappropriate initial setting for fine-tuning with this particular dataset or task. It does not allow the model to improve, with its default structure, to a nuanced language understanding. Ultimately, the flexibility of Wav2Vec2 and the improvements obtained testify to the versatility with which the latter makes such adjustments to parameters to arrive at higher levels of voice recognition efficiency.

By contrast, Whisper's far-limited tunability is supported by flat WER and wastes in training, suggesting that it operates out of the box at total capacity, leaving little headroom for improvement given the conditions at hand. This contrast points to Wav2Vec2 as the preferable model for tasks requiring precision but fine-tuning flexibility. Conversely, Whisper would require changes in architecture or task settings to catch up to competitive efficacy in high-precision voice recognition.

# Chapter 10

# Limitations

In this chapter we will discuss about the limitations that we realized while facing the challenges while doing our research works.

## 10.1 Dataset Limitations

Our experiments utilized the Bengali subset of the Common Voice corpus for training and evaluation in the ASR models. As this dataset was used, the limits of it for doing ASR in Bengali became apparent. As a language, Bengali encompasses various dialects, each being phonetically different from others by having different pronunciation variations and vocabulary differences. Therefore, these aspects are underrepresented in the Common Voice corpus, which may not comprehensively represent regional accents and dialectal variants. Thus, the models trained on this dataset will find it challenging to generalize across the complete spectrum of Bengali linguistic variations.

Moreover, the scope of this dataset is relatively limited, consequently affecting the capability of models to represent and adapt to the subtleties present in spoken Bengali. The other challenge for the training process is that the dataset's constraints may not allow models to get enough examples of various dialectal or contextual subtleties, for example, in Bengali. Naturally, that would mean a lack of representativeness; therefore, the two models under discussion might behave poorly in real-life scenarios where people speak different dialects, use colloquial expressions, and have some local' accents- an issue very relevant to Bengali ASR.

## 10.2 Computational Resource Constraints

However, one of the significant limitations of this study is restricted access to advanced GPU resources, which are essential in optimizing ASR models. Usually, training deep learning ASR models, such as Wav2Vec2 and Whisper, requires a lot of computational power so that faster and broader tuning is possible, especially when experimenting with configurations and hyperparameters. We had to deal with much longer training sessions due to the limited availability of the GPU. Therefore,

we were not able to explore all the configurations thoroughly. This limited availability of the GPU resource also meant that we could only test a few numbers of epochs and batch sizes since using the higher values for either of these significantly increases the runtime. Furthermore, due to hardware limitations, advanced tuning techniques, such as deeper model stacking or multi-GPU parallelization, could not be done and may lead to better convergence and accuracy in models.

## 10.3 Model Sensitivity and Tunability

The higher sensitivity of the Wav2Vec2 model to changes in its parameters favored gradual enhancements regarding WER and stability. On the other hand, such high sensitivity made finding the optimal configuration within the limited research scope hard. Frequent fluctuations of training and evaluation metrics indicated that even slight modifications of parameters may cause divergent results and sometimes counterbalance the anticipated improvement. This sensitivity suggests that Wav2Vec2 requires careful tuning to bring in the best results, which might be time-consuming and bound by our hardware capabilities. On the other hand, no kind of tuning on the parameters of the Whisper model was elastic; it would hardly improve every time an attempt was made to try a different set of parameters. Poor tunability was a significant barrier to improving the performance of Whisper for Bengali ASR tasks in this respect. With several adjustments of learning rate, batch size, dropout rate, and gradient accumulation, Whisper never came down from 100% WER, which may indicate some design limitation or incompatibility with the Bengali language data. This inflexibility puts a ceiling to the possibilities of using Whisper as a customizable ASR solution for linguistically challenging tasks such as Bengali speech recognition.

## 10.4 Language-Specific Challenges and ASR Complexity

Both models experienced challenges in adaptation to the peculiarities of Bengali, namely rich and complex inflectional morphology, variability of phonetic structure variations, and an extreme number of homophones. These linguistic properties create demands on ASR models: highly flexible to represent discriminative information in phonetic and morphological variations with high precision. Wav2Vec2 demonstrated some ability for adaptation: WER improved and became more stable within the tuning iterations. However, these gains were relatively limited in amplitude and suggested that further improvements would require an even more careful tuning process and increased data. The high WER obtained with the Whisper model perhaps indicates that its architecture is not inherently suitable for Bengali ASR tasks without more far-reaching architectural modifications. Bengali is a highly phonetic and dialectally varied language, which requires a model to have sensitive parameters. The relatively low tunability of Whisper may lead to poor phoneme differentiation. Because this is an added flexibility it has to capture such phonetic subtlety, Whisper remains challenged by the intricacies of Bengali speech.

## 10.5 Generalization Limitations in Real-world Applications

On the other hand, this limits the representativity of the dataset and computational resources, which cannot fail to affect the generalization capability of these models in practical applications. The application of Wav2Vec2 showed incremental improvements in evaluation metrics. It was an excellent candidate for adaption to real-world applications since it has narrow coverage regarding the dialectal and contextual variation of the dataset used. Moreover, reliance on specific tuning for gains implies that the model would perform poorly on dialects or speech variations not present in the dataset. The consistently high WER across all configurations in this work empirically shows that Whisper would fail to generalize for Bengali ASR. The persistence of 100% WER through all tuning configurations suggests that Whisper would struggle to transcribe even moderately varied speech inputs in Bengali correctly. This limitation raises concerns about its effectiveness for high-accuracy applications, such as transcription services or voice-activated virtual assistants catering to Bengali-speaking users.

## 10.6 Scalability Challenges

Scalability is significant for ASR models deployed in such high-variance languages as Bengali. Advanced scaling strategies, such as multi-GPU training or hyperparameter sweeps across a larger parameter space, were beyond reach with the limited computational resources available to us in this study. While the sensitivity of Wav2Vec2 would indicate that such measures for scalability would help, the unresponsiveness of Whisper would indicate that scalability here will not work out that well without more fundamental changes in architecture or increased dataset size. Limited hardware precluded experiments with more frequent checkpointing, multi-sample gradients, or extended epochs, which might improve model robustness in a scalable solution. The resultant models, therefore, reflect the constraint imposed by single-GPU limitations to scalability since this results in a bottleneck on the potential deployment in real-world Bengali ASR systems where consistent performance and adaptability are at an all-time high.

# Chapter 11

# Conclusion

This research underlines the critical difference in adaptability and performance between the Wav2Vec2 and Whisper ASR models while processing Bengali speech. Automatic Speech Recognition has significantly changed how humans interact with devices, allowing hands-free and natural interactions across various fields like healthcare, customer service, and virtual assistance. However, ASR systems naturally incur challenges while processing languages with complex phonetic and dialectal variations, such as Bengali. Two of the state-of-the-art ASR models- Wav2Vec2 and Whisper-were explored in this work for Bengali ASR tasks. The chief focus was their response to parameter tuning, generalization across linguistic features, and the ability to handle real-world Bengali language variations. Our findings emphasize the flexibility of Wav2Vec2 regarding tuning and its subsequent improvements in Word Error Rate (WER). By doing a step-by-step adjustment of learning rate, batch size, attention dropout, and gradient accumulation, the model stability of Wav2Vec2 steadily improved its evaluation WER, eventually getting better balance to meet real-world application needs. Such flexibility implies that Wav2Vec2 should be able to handle the complexity arising in the linguistic scenarios for Bengali ASR, most prominently, code-switching, phonetic variation, and dialect. This shows that fine-tuning may make it a good candidate for virtual assistants, transcription services, and other voice-activated tools in Bengali, as it improves upon its zero-shot result.

On the other hand, Whisper shows rigidity in its response to changes in parameters in all its tuning attempts, depicted by a consistently high WER, which again evidences a lack of flexibility in dealing with the linguistic diversity of Bengali. Although this model explored a variety of learning rates, batch sizes, and gradient accumulation, Whisper has kept its 100% WER, which is insensitive to most tuning changes. The Whisper model has an intrinsic architectural limitation, making it less suitable for ASR from texts possessing more complex phonetic and dialectical features. Thus, Whisper may be best suited for standardized languages or applications that do not need so much flexibility and precision. The workflow for the study integrated those essential ASR processes: audio resampling, feature extraction, and tokenization for both Wav2Vec2 and Whisper. Both models underwent parallel inference to compare their accuracy in transcription and response to language-specific challenges. Although Wav2Vec2 showed notable improvements in handling transcription and pronunciation nuances with Bengali data, limited tuning capacity hindered Whisper

from effectively adapting to the complexities of the language. These results further establish the broader applicability of Wav2Vec2 in Bengali ASR by underlining the model's aptitude for accommodation toward varied speech inputs, dialects, and accents. Its scalability and practicality would mean that communities of speakers of the Bengali language would have secured a model to serve the high-accuracy demand in a dialectically and phonetically diverse region. The limitations include restricted dataset diversity, limited computational resources, and language-specific challenges considered vital for improvement. The dataset used was mainly standardized Bengali speech, which represents very few regional dialects and accents and hence may limit the generalization of the models across the full spectrum of linguistic diversity in Bengali. Future works should be conducted with a more comprehensive dataset representing various dialects and broader computational resources to increase efficiency in tuning and training.

Moreover, the continued failure of Whisper also represents that some architectural edits might be necessary to work with complicities in Bengali ASR. However, it would be better to address the limitations in the ASR model, especially to make sure the phonetic and morphological subtleties are captured for correct transcriptions across all dialects and contexts in Bengali. This work concludes that Wav2Vec2 positions itself most strongly for high-accuracy applications in Bengali owing to its flexibility concerning tuning and adapting toward language-specific challenges. Whisper provides only limited tunability and has static performance across tuning configurations. This hints at the requirement of changes in structure to achieve similar efficacy in diverse linguistic settings. A model for ASR applications in languages such as Bengali, which is rich in phonetic and dialectical variations, must be flexible and responsive to parameter tuning if it is ever to meet the demands of accurate, fine-grained transcription. This points to the need for the design of ASR frameworks, which would include, in their structure, the peculiar linguistic properties of languages like Bengali. Adaptability, therefore, puts Wav2Vec2 in a better position for such tasks.

Our dataset that has been collected from Mozilla Common Voice features those voice samples which are all in the Bengali language. Due to the current circumstances and shortage of time and resources, we could not use other voice samples of other languages. In the future, our research can be extended in several promising directions. One of the potential areas of improvement is enhancing the performance of multilingual speech recognition systems in low-resource languages by using more advanced self-supervised learning techniques. Additionally, the integration of more sophisticated language models could further improve the accuracy of transcriptions, especially for highly inflected languages or those with complex syntactic structures. Another target for future work is optimizing the system for real-time applications, which would involve the efficiency of the models and exploring utilizing more of the versions of Whisper, and Wav2Vec 2.0, without compromising accuracy. Finally, exploring cross-lingual transfer learning, where knowledge from high-resource languages is transferred to low-resource languages, could significantly reduce data dependency while maintaining accuracy across diverse languages.

# Bibliography


[1] C. Martin, B. Jon, C. Stuart, and S. Xu, "An audio-visual corpus for speech perception and automatic speech recognition (l)," *Acoustical Society of America, 120(5), 2421-2424.*, Jun. 2006. [Online]. Available: https://laslab.org/wp-content/uploads/2021/09/an_audio-visual_corpus_for_speech_perception_and_automatic_speech_recognition.pdf.

[2] M. Sandipan, D. Biswajit, and M. Pabitra, "Shruti-ii: A vernacular speech recognition system in bengali and an application for visually impaired community," *2010 IEEE Students Technology Symposium (TechSym)*, 2010. [Online]. Available: https://sci-hub.se/10.1109/TECHSYM.2010.5469156.

[3] M. Abdel-rahman, E. D. George, and H. Geoffrey, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, Jan. 2011. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5704567.

[4] D. Biswajit, M. Sandipan, and M. Pabitra, "Bengali speech corpus for continuous automatic speech recognition system," *IEEE*, 2011. [Online]. Available: https://sci-hub.se/10.1109/ICSDA.2011.6085979.

[5] H. Geoffrey, D. Li, Y. Dong, *et al.*, "Deep neural networks for acoustic modeling in speech recognition.," *IEEE Signal Processing Magazine*, 2012. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/HintonDengYuEtAl-SPM2012.pdf.

[6] M. Tomáš, D. Anoop, P. Daniel, B. Lukáš, and Č. Jan, "Strategies for training large scale neural network language models," *2011 IEEE Workshop on Automatic Speech Recognition  Understanding*, Mar. 2012. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6163930.

[7] Vimala.C and Dr.V.Radha, "A review on speech recognition challenges and approaches," *World of Computer Science and Information Technology Journal (WCSIT)*, 2012, ISSN: 2221-0741. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/30987239/A_Review_on_Speech_Recognition_Challenges_and_Approaches-libre.pdf.

[8] D. Li, H. Geoffrey, and K. Brian, "New types of deep neural network learning for speech recognition and related applications: An overview," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, ISSN: 2379-190X. [Online]. Available: https://sci-hub.se/10.1109/icassp.2013.6639344.

[9] D. Li, L. Jinyu, H. Jui-Ting, *et al.*, "Recent advances in deep learning for speech research at microsoft," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Oct. 2013. [Online]. Available: https://sci-hub.se/10.1109/icassp.2013.6639345.

[10] D. Namrata, "Feature extraction methods lpc, plp and mfcc in speech recognition," *International journal for advance research in engineering and technology*, 2013. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/40023802/Feature_Extraction_Methods_LPC__PLP_and_MFCC-libre.pdf.

[11] A.-H. Ossama, M. Abdel-rahman, J. Hui, D. Li, P. Gerald, and Y. Dong, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Jul. 2014. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6857341.

[12] S. Hasim, S. Andrew, R. Kanishka, and B. Francoise, "Fast and accurate recurrent neural network acoustic models for speech recognition," *Proc. Interspeech 2015*, Jul. 2015. [Online]. Available: :https://arxiv.org/pdf/1507.06947.pdf.

[13] C. William, J. Navdeep, V. L. Quoc, and V. Oriol, "Listen, attend and spell," Aug. 2015. [Online]. Available: https://arxiv.org/pdf/1508.01211.pdf.

[14] v. d. o. Aaron, D. Sander, Z. Heiga, *et al.*, "Wavenet: A generative model for raw audio," Sep. 2016. [Online]. Available: https://www.researchgate.net/publication/308026508.

[15] A. Dario, A. Sundaram, A. Rishita, *et al.*, "Deep speech 2 : End-to-end speech recognition in english and mandarin," *33rd International Conference on Machine Learning (ICML)*, 2016. [Online]. Available: https://proceedings.mlr.press/v48/amodei16.pd.

[16] C. Jan, B. Dzmitry, S. Dmitriy, C. Kyunghyun, and B. Yoshua, "International conference on acoustics, speech and signal processing (icassp)," Mar. 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/1068c6e4c8051cfd4e9ea8072e3189e2-Paper.pdf.

[17] N. Md Mahadi Hasan, P. Bishwajit, and I. Md Saiful, "Bengali speech recognition: A double layered lstm-rnn approach," *IEEE*, 2017. [Online]. Available: https://sci-hub.se/10.1109/ICCITECHN.2017.8281848.

[18] K. Suyoun, H. Takaaki, and W. Shinji, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2017. [Online]. Available: https://arxiv.org/pdf/1609.06773.pdf.

[19] S. Toshniwal, T. N. Sainath, R. J. Weiss, *et al.*, "Multilingual speech recognition with a single end-to-end model," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2018, pp. 4904–4908.

[20] T. Bao, J. Robert, A. Dominic, P. Emily, and P. Raymond, "Synthetic data augmentation for improving low-resource asr," *2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8923082.

[21] S. P. Daniel, C. William, Z. Yu, *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, Dec. 2019. [Online]. Available: https://arxiv.org/abs/1904.08779.

[22] I. Jahirul, M. Masiath, I. Md. Rakibul, and D. Amit Kumar, "A speech recognition system for bengali language using recurrent neural network," *IEEE*, 2019. [Online]. Available: https://sci-hub.se/10.1109/CCOMS.2019.8821629.

[23] A. Kannan, A. Datta, T. N. Sainath, *et al.*, "Large-scale multilingual speech recognition with a streaming end-to-end model," *arXiv preprint arXiv*, 2019. [Online]. Available: https://arxiv.org/pdf/1909.05330.

[24] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[25] M. Sayan, Y. Sarthak, and R. Atul, "End-to-end bengali speech recognition," Nov. 2020. [Online]. Available: https://arxiv.org/pdf/2009.09615.

[26] S. Uddalok, P. Soumyadeep, N. Sayan, *et al.*, "Speaker recognition in bengali language from nonlinear features," *arXiv preprint arXiv:2004.07820*, Apr. 2020. [Online]. Available: https://arxiv.org/abs/2004.07820.

[27] M. Linghui, X. Jin, T. Xu, W. Jindong, Q. Tao, and X. Bo, "Mixspeech: Data augmentation for low-resource automatic speech recognition," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Feb. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9414483.

[28] G. Manuel, G. Deniz, L. Yulan, and W. Daniel, "Bootstrap an end-to-end asr system by multilingual training, transfer learning, text-to-text mapping and synthetic audio," *INTERSPEECH 2021*, Jun. 2021. [Online]. Available: https://arxiv.org/abs/2011.12696.

[29] M. F. Mridha, O. Abu Quwsar, H. Md. Abdul, and M. Muhammad Mostafa, "Challenges and opportunities of speech recognition for bengali language," *Artificial Intelligence Review*, 2021. [Online]. Available: https://arxiv.org/pdf/2109.13217.

[30] S. Sadia, R. M. Shahidur, and I. M. Zafar, "Recent advancement in speech recognition for bangla: A survey," *International Journal of Advanced Computer Science and Applications (IJACSA)*, Mar. 2021, ISSN: 2156-5570. [Online]. Available: https://thesai.org/Downloads/Volume12No3/Paper_65-Recent_Advancement_in_Speech_Recognition.pdf.

[31] S. Ahnaf Mozib, K. M. Humayan, R. Md. Mushtaq Shahriyar, *et al.*, "Investigating self-supervised, weakly supervised, and fully supervised training approaches for multi-domain automatic speech recognition: A study on bangladeshi bangla," *preprint on arXiv*, Oct. 2022. [Online]. Available: https://arxiv.org/pdf/2210.12921.

[32] D. Mitchell and B. Jayadev, "Improving low-resource speech recognition with pretrained speech models: Continued pretraining vs. semi-supervised training," Jul. 2022. [Online]. Available: https://arxiv.org/abs/2207.00659.

[33] R. Mohammed, H. Md. Ismail, M. Nabeel, and R. Fuad, "Bangla-wave: Improving bangla automatic speech recognition utilizing n-gram language models," Sep. 2022. [Online]. Available: https://arxiv.org/pdf/2209.12650.

[34] A. Samiul, S. Asif, A. Zaowad, *et al.*, "Bengali common voice speech dataset for automatic speech recognition," *arXiv preprint arXiv:2206.14053*, Jun. 2022. [Online]. Available: https://arxiv.org/pdf/2206.14053.

[35] S. Tushar Talukder, "An automatic speech recognition system for bengali language based on wav2vec2 and transfer learning," *Available on arXiv*, Sep. 2022. [Online]. Available: https://arxiv.org/pdf/2209.08119.

[36] R. Fazle Rabbi, D. Souhardya Saha, A. Samiul, *et al.*, "Ood-speech: A large bengali speech recognition dataset for out-of-distribution benchmarking," *Submitted to INTERSPEECH*, 2023. [Online]. Available: https://arxiv.org/pdf/2305.09688.

[37] Q. Gege, C. Yuefeng, M. Xiaofeng, *et al.*, "Robust automatic speech recognition via wavaugment guided phoneme adversarial training," *INTERSPEECH*, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2307.12498.

[38] H. Md Gulzar, R. Mahmuda, S. Babe, and S. Ye, "Banspemo: A bangla emotional speech recognition dataset," Dec. 2023. [Online]. Available: https://arxiv.org/abs/2312.14020.

[39] N. Rabindra Nath, M. Mehadi Hasan, M. Tareq Al, *et al.*, "Pseudo-labeling for domain-agnostic bangla automatic speech recognition," *Association for Computational Linguistics*, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2311.03196.

[40] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*, PMLR, 2023, pp. 28 492–28 518.

[41] K. Shariar, N. Nazmun, S. Shyamasree, and R. Mamunur, "Automatic speech recognition for biomedical data in bengali language," Dec. 2024. [Online]. Available: https://arxiv.org/abs/2406.12931.