# Shahjalal University of Science and Technology

## Department of Computer Science and Engineering

# Speech Attribute-Based Mispronunciation Detection: Approach to Language Agnostic Phoneme Scoring Beyond IPA

FAHIM UZ ZAMAN

Reg. No.: 2019331012

4th year, 2nd Semester

SYED SAZID HOSSAIN REZVI

Reg. No.: 2019331054

4th year, 2nd Semester

Department of Computer Science and Engineering

**Supervisor**

MD. MEHEDI HASAN

Lecturer

Department of Computer Science and Engineering

6th July, 2025

# Speech Attribute-Based Mispronunciation Detection: Approach to Language Agnostic Phoneme Scoring Beyond IPA

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

| FAHIM UZ ZAMAN | SYED SAZID HOSSAIN REZVI |
|---|---|
| Reg. No.: 2019331012 | Reg. No.: 2019331054 |
| 4th year, 2nd Semester | 4th year, 2nd Semester |

Department of Computer Science and Engineering

**Supervisor**

MD. MEHEDI HASAN

Lecturer

Department of Computer Science and Engineering

6th July, 2025

# Recommendation Letter from Thesis Supervisor

The Thesis entitled *Speech Attribute-Based Mispronunciation Detection: Approach to Language Agnostic Phoneme Scoring Beyond IPA* submitted by the students

1. FAHIM UZ ZAMAN
2. SYED SAZID HOSSAIN REZVI

is under my supervision. I, hereby, agree that the thesis can be submitted for examination.

Signature of the Supervisor:
Name of the Supervisor: MD. MEHEDI HASAN
Date: 6th July, 2025

# Certificate of Acceptance of the Thesis

The thesis entitled *Speech Attribute-Based Mispronunciation Detection: Approach to Language Agnostic Phoneme Scoring Beyond IPA*

1. FAHIM UZ ZAMAN (Reg. 2019331012)
2. SYED SAZID HOSSAIN REZVI (Reg. 2019331054)

on 6th July, 2025, hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

| | | |
|---|---|---|
| Head of the Deptartment | Chairman, Exam. Committee | Supervisor |
| Md. Masum | Md. Masum | MD. MEHEDI HASAN |
| Professor | Professor | Lecturer |
| Department of CSE, SUST | Department of CSE, SUST | Department of CSE, SUST |

# Abstract

This thesis explores a novel approach for phoneme-level scoring in speech-to-text (S2T) systems for 2nd Language English speakers, especially for Bengali English speakers. Traditional scoring methodologies and models often lack the required accuracy for individual phonemes, especially when facing the intricacies of various languages or different speech patterns. This limitation is especially evident for speakers from different linguistic backgrounds, such as Bengali English speakers. Traditional models rely on language-specific International Phonetic Alphabet (IPA) configurations which hinders their generalizability and practical application in different languages.

This research proposes a system that overcomes these limitations by focusing on language-independent features and acoustic properties that are inherent in human speech. Our objective was to develop a universally applicable evaluation framework that uses language-agnostic phoneme-level scoring to enhance the accuracy and fairness of pronunciation feedback for all L2 learners to achieve language-agnostic evaluation. To achieve this for Bengali L2 speakers, We have created a new dataset containing 8225 annotated utterances collected from 16 Bengali English speakers.

We explored pretrained models of self-supervised learning models like Wav2Vec2, which operate on raw waveforms and eliminates the need for specific IPA configurations for each language. Furthermore, we integrated Goodness of Pronunciation (GOP) to enhance phoneme-level scoring accuracy. Our approach focused on the fundamental acoustic patterns rather than predetermined phonetic rules. This approach opens up new possibilities for automated speech assessment in multilingual environments while maintaining the natural flow of language processing.

**Keywords:** Phoneme-level pronunciation scoring, Second language (L2) speech assessment, Bengali English speakers, Goodness of Pronunciation (GoP), Wav2Vec2, Self-supervised learning for speech, Computer-Assisted Pronunciation Training (CAPT), Language-agnostic phoneme scoring, Pronunciation evaluation for L2 learners, Acoustic feature modeling, Automatic pronunciation feedback.

# Acknowledgements

This report documents our long, yet at the same time, short journey of developing a phoneme-level scoring in speech to text(S2T) systems for 2nd Language English speakers.

First and foremost, we would like to extend our gratitude to our supervisor, Md. Mehedi Hasan, for his great mentorship throughout this research journey. His knowledge of the field, combined with his ability to provide constructive criticism, has helped us develop a deeper understanding of our research area. Md. Mehedi Hasan's patience in guiding us through complex situations have taught us what mentorship should be.

We would also like to acknowledge the support of our department faculty for their valuable input and encouragement. Their perspectives and suggestions have significantly helped our work. Our sincere thanks extend to the technical staff who facilitated our experimental work and provided essential resources.

We would like to offer special thanks to the people who selflessly helped us build our dataset. They have given us hours of their lives with no expectation of getting anything in return. We are truly grateful to them.

Furthermore, we are deeply grateful to our families and friends for their help, understanding, emotional support, and encouragement throughout this journey. Their unwavering belief in our capabilities gave us the strength to persevere through the most demanding phases of this research.

As we present our findings and insights in this report, we are humbled by the academic community's support and encouragement, which were crucial to developing this report's findings. This journey wouldn't have been possible without the combined efforts of many remarkable individuals and institutions. It is on the shoulders of Giants that we built our work.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In speech processing and language learning, speech pronunciation typically refers to how accurately a speaker can: match the expected phoneme targets (from native speakers), produces phonemes with correct acoustic and articulatory features, and how much they deviate from standard pronunciation (mispronunciations by L2 learners). In the modern world, the use of Speech pronunciation is used very widely. Starting with the integration of speech technology in education, communication, and accessibility services, there is now a growing demand for Automatic Pronunciation Evaluation (APE). APEs offer objective, scalable, and accessible means for assessing pronunciation quality, especially in the case of second language acquisition. One of many important evaluation criteria of such systems is the phoneme-level scoring mechanism, which evaluates individual phonemes within utterances for pronunciation accuracy.

However, while much work has been done on word-level and sentence-level scoring systems, phoneme-level scoring is still relatively young. The existing phoneme level scoring system types include GoP (Goodness of Pronunciation), Phoneme-level alignment, GMM-HMM and DNN-HMM and their Hybrids [11, 21], SVM + Pronunciation Space Models etc. Most of these existing phoneme-level scoring systems rely heavily on language-specific IPA [1] (International Phonetic Alphabet). While these methods have achieved success in controlled environments, their applicability in diverse linguistic backgrounds remains limited. These systems often fail to generalize across accents, dialects, and low-resource languages, largely due to their dependence on phoneme-level transcriptions, handcrafted rules, and forced alignment techniques. In addition, they struggle to handle the variations and transfer effects that occur when L2 speakers apply phonological patterns from their native language.

As global communication is increasingly being dependent on English proficiency, non-native speakers often have to face challenges at the lexical or syntactic level as well as at the phonological level, where mispronunciations can severely hinder intelligibility. For learners of English as a second language, especially those from linguistically distant backgrounds such as Bengali, phoneme-level feedback can play an important role in identifying subtle pronunciation errors and guiding necessary correction.

After the arrival of self-supervised model such as Wav2Vec2 [2], and models like Context aware GOP [29], the reliance on IPA is being reduced. These models, especially self supervised models like Wav2Vec2 [2] that works with raw waveforms directly have shown extraordinary promise, but these are relatively new fields of work and their full potential as well as shortcomings are yet to be seen.

This research explores and evaluates various phoneme-level scoring techniques for English spoken by non-native speakers, particularly Bangladeshi speakers. Our focus is on identifying reliable methods for aligning audio with transcriptions and estimating phoneme pronunciation quality, crucial for computer-assisted language learning (CALL) systems and automated feedback tools. For that purpose, we have constructed a dedicated dataset of 8,225 annotated utterances.

By integrating pretrained deep speech models with phoneme-level scoring metrics, this research aim is to deliver more accurate, fair, and scalable pronunciation assessment tools. Ultimately, this approach contributes to the broader goal of inclusive and effective language learning technologies that can adapt to the unique pronunciation challenges of diverse L2 populations.

## 1.1 Objectives

The primary objective of this study is to investigate phoneme-level scoring systems for assessing English pronunciation quality among non-native speakers, with a specific focus on Bangladeshi L2 learners. While various techniques have been developed for pronunciation evaluation, many of them are optimized for native speakers or assume access to high-quality phoneme-level annotations and standardized phonetic transcriptions, which may not generalize well to L2 speech.

Given the limited availability of publicly annotated datasets specifically tailored to the pronunciation patterns of Bengali speakers learning English, we aim to construct a dedicated dataset to facilitate such research. Moreover, traditional scoring approaches often rely heavily on International Phonetic Alphabet (IPA) representations or handcrafted alignment techniques, which can be less adaptable to the acoustic variability observed in L2 speech.

To address these challenges, our study explores a diverse range of speech modeling and scoring strategies—including forced alignment models and modern deep learning-based approaches like self-supervised representation learning. By systematically comparing these methods, we aim to identify and enhance phoneme-level scoring mechanisms that are effective, interpretable, and better suited to the needs of L2 learners, particularly in low-resource linguistic settings such as Bangladesh.

The specific research objectives are as follows:

1. **Develop a native L2 speech dataset**: Collect and curate a high-quality, manually verified dataset of English utterances spoken by Bangladeshi second-language speakers. This dataset will include detailed phoneme-level annotations aligned using forced alignment tools.

2. **Utilize L1 reference datasets**: Incorporate existing native English speech corpora, such as the LibriSpeech dataset, to provide a reference baseline for pronunciation modeling and comparative evaluation.

3. **Explore and benchmark multiple scoring models**: Experiment with a diverse set of phoneme-level scoring systems, including traditional alignment-based methods (e.g., GMM-HMM, DNN-HMM) and recent deep learning approaches (e.g., wav2vec 2.0 [2], context-aware GOP [29]).

4. **Compare scoring metrics for L2 proficiency assessment**: Conduct a comprehensive analysis of the effectiveness of different phoneme scoring strategies (e.g., GOP scores, alignment errors, acoustic likelihoods) in evaluating pronunciation quality for Bengali L2 learners.

5. **Investigate the influence of L1 interference**: Analyze how phonological transfer from Bengali (L1) impacts pronunciation in English (L2), and assess whether models can capture and adapt to these language-specific features.

6. **Propose improved L2 scoring techniques**: Based on the findings, design and evaluate compact, scalable, and interpretable scoring systems that can provide reliable phoneme-level feedback to assist pronunciation training and error correction.

7. **Contribute tools and data to the community**: Open-source selected components of the dataset, alignment pipelines, and scoring implementations to support future research in L2 pronunciation modeling.

# Chapter 2

# Background Study

This section reviews various scoring systems and speech recognition models to identify existing challenges and limitations in phoneme-level evaluation.

## 2.1 Scoring Systems

To evaluate the performance of Speech-to-Text (S2T) models, several scoring systems are employed. These metrics provide different levels of insights, each focusing on varying aspects of speech recognition. However, these systems have limitations in capturing the granular details of phoneme-level accuracy.

### 2.1.1 Commonly Used Scoring Metrics:

1. **Phoneme Error Rate (PER)**

   - Description: Measures the accuracy of the phoneme prediction by calculating the number of phoneme substitutions, deletions, and insertions.
   - Focus: Phoneme-level evaluation, which is crucial for understanding the model's phonetic precision.
   - Example: A model might be evaluated based on how many phonemes it correctly predicted in a given speech sample.
   - Limitation: Does not account for larger contextual errors at the word or sentence level.

2. **Formula:**

   $$\text{PER} = \frac{S + D + I}{N}$$

   where: $S =$ Substitutions
   $D =$ Deletions
   $I =$ Insertions
   $N =$ Total phonemes in the reference transcription

3. **Word Error Rate (WER)**

   - Description: Calculates the number of word-level errors by considering insertions, deletions, and substitutions at the word level.
   - Focus: Evaluates the overall accuracy of the transcription, but it lacks detail on phoneme-level errors.
   - Limitation: It does not consider the phonemic accuracy within words, so errors at the phoneme level may not be detected.

4. **Formula:**

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

   where: $S$ = Substitutions (word)
   $D$ = Deletions (word)
   $I$ = Insertions (word) $N$ = Total number of words in the reference

5. **Character Error Rate (CER)**

   - Description: Measures the accuracy at the character level, particularly useful for languages without clear word boundaries, like Chinese.
   - Focus: Evaluates how well the model predicts individual characters, rather than phonemes or words.
   - Limitation: Does not account for phoneme-level errors and is typically used for non-space-separated languages.

6. **Formula:**

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

   where: $S$ = Substitutions (character)
   $D$ = Deletions (character)
   $I$ = Insertions (character)
   $N$ = Total number of characters in the reference

7. **Sentence Error Rate (SER)**

   - Description: Measures how many sentences contain at least one error. This metric is helpful for understanding high-level transcription accuracy.
   - Focus: Focuses on whether a sentence is entirely correct or contains errors.
   - Limitation: Does not provide granular details at the phoneme or word level, missing crucial insights into specific errors.

8. **Formula:**

   $$\text{SER} = \frac{\text{Total sentences with error}}{\text{Total sentences}}$$

.

### 2.1.2 Comparison of Standard Evaluation Metrics

| Metric | Granularity | Evaluation Target | Key Limitation |
|---|---|---|---|
| PER (Phoneme Error Rate) | Phoneme-Level | Phoneme Classification Accuracy | Ignores suprasegmental features and contextual phoneme shifts |
| WER (Word Error Rate) | Word-Level | Overall Transcription Accuracy | Fails to detect subtle phoneme-level mispronunciations in otherwise correct words |
| CER (Character Error Rate) | Character-Level | Alphabetic Transcription Accuracy | Not suited for acoustic-phonetic variations; omits pronunciation context |
| SER (Sentence Error Rate) | Sentence-Level | End-to-end intelligibility | Masks fine-grained errors by treating sentences holistically |

Table 2.1: Comparison of Common Speech Recognition Metrics

## Why Conventional Metrics Fall Short in Phoneme-Level Assessment

Traditional metrics like PER, WER, and CER are widely used in Automatic Speech Recognition (ASR) to evaluate model performance. However, their design prioritizes transcription correctness over pronunciation fidelity. These metrics treat phonemes as discrete symbolic targets without accounting for articulation errors, phonological transfer, or acoustic similarity—all of which are crucial in evaluating L2 pronunciation.

Moreover, most ASR benchmarks assume native-speaker training data and focus on lexical correctness. In contrast, phoneme-level mispronunciation detection—especially in the context of non-native English speakers—requires granular analysis that considers the acoustic quality, segment timing, and deviation from native-like articulation patterns.

### Language-Dependent Phoneme Mappings and Challenges

One of the critical challenges in phoneme-level scoring is the reliance on International Phonetic Alphabet (IPA) transcriptions, which vary significantly across languages. This variability imposes limitations on the transferability of scoring systems from one language context to another. This variation demonstrates the inherent difficulty of designing a universal, language-independent scoring system. Models trained with English phoneme inventories may fail to accurately interpret or evaluate utterances produced by non-native speakers whose articulatory habits reflect different phonemic inventories or constraints [1, 8].

| **Sample English Phoneme (IPA)** | **Closest Mandarin Equivalent (IPA)** |
|---|---|
| /W/ | /zh/ or /ts/ |
| /IY/ | /ch/ or /ts$^h$/ |
| /K/ | /sh/ or /s/ |
| /AO/ | /r/ or /z/ |
| /L/ | /i/ |
| /N/ | /ŋ/ |

Table 2.2: Illustrative IPA Mapping Differences: English vs. Mandarin [9]

This variation demonstrates the inherent difficulty of designing a universal, language-independent scoring system. Models trained with English phoneme inventories may fail to accurately interpret or evaluate utterances produced by non-native speakers whose articulatory habits reflect different phonemic inventories or constraints.

## Our Contribution

To address these gaps, our work introduces a comprehensive pipeline tailored for L2 phoneme-level assessment, particularly for Bangladeshi English learners. Unlike prior work focused purely on native speech, we:

- Construct a phoneme-aligned L2 dataset using forced alignment on raw audio collected from Bangladeshi speakers.
- Apply both classical (GMM-HMM [11]) and modern (wav2vec2 [2], context-aware GOP [29]) scoring methods to study cross-model performance.
- Evaluate how IPA-bound systems behave across different linguistic influences.
- Propose a hybrid scoring framework that integrates acoustic scores and contextual GOP adjustments, improving robustness in L2 scenarios [3, 28, 31].

Our findings highlight that purely transcription-based metrics are insufficient for fine-grained pronunciation feedback. We argue that phoneme-aware, acoustically grounded, and learner-sensitive metrics are essential for next-generation Computer-Assisted Pronunciation Training (CAPT) systems [22, 24, 25].

## 2.2 Research on Existing Models

### 2.2.1 Kaldi [10]

[4]

Kaldi is an open-source toolkit for speech recognition and alignment, offering highly customizable pipelines including GMM-HMM and DNN-HMM models.

- **Pros**:
  - Offers full control over the acoustic modeling process, alignment, and feature extraction.
  - Supports Goodness of Pronunciation (GOP) scoring, which is a widely adopted metric in phoneme-level mispronunciation detection [3, 28, 31].
  - Integrates with standard ASR pipelines and allows training models on both L1 and L2 speech data.
  - Used as the backbone for many CAPT (Computer-Assisted Pronunciation Training) systems.
- **Cons**:
  - Requires detailed configuration and knowledge of Kaldi's data structure, which adds to setup complexity.
  - Dependent on a good lexicon and phone set, which may need customization for non-native accents or L1 transfer effects.
  - Less suitable for raw waveform processing or end-to-end learning without heavy preprocessing.

### 2.2.2 CharSiu [5]

- **Pros**:
  - Works well for text-to-speech alignment in various languages.
  - Good for models where IPA-specific configurations are available, making it accurate for supported languages.
  - Efficient for languages that align well with the IPA structure.
- **Cons**:
  - Requires language-specific IPA configurations, which limits its adaptability to different languages.
  - Does not offer robust phoneme-level scoring mechanisms, which makes it less effective for accurate phoneme evaluation.
  - Not suitable for under-resourced or code-mixed languages, as IPA mapping can be difficult to apply.

### 2.2.3 Whisper (OpenAI) [6]

- **Pros**:
    - Strong transcription accuracy, especially for general speech-to-text tasks.
    - Supports multiple languages and accents, offering high versatility.
    - Easy to implement and use for general transcription purposes, with good performance across varied speech data.
- **Cons**:
    - Lacks a dedicated phoneme-level scoring mechanism, making it unsuitable for tasks that require precise phoneme evaluation.
    - It does not offer a language-independent scoring system and struggles with nuanced phoneme alignment in specific languages.
    - It is primarily a transcription model and lacks deep analysis of phonetic correctness or performance.

### 2.2.4 Julius

- **Pros**:
    - Open-source and optimized for real-time speech recognition applications.
    - Well-suited for certain language applications with good performance in specific domains.
    - Can be customized for particular use cases, such as speaker recognition or domain-specific speech tasks.
- **Cons**:
    - Limited phoneme-level scoring mechanism, making it less suitable for detailed analysis of phoneme accuracy.
    - Requires significant tuning for new languages or domains, making it less flexible for rapid deployment.
    - Language-specific limitations make it harder to use for multilingual or cross-lingual tasks.

## 2.3 Problems Identified

The research highlighted several key issues with current speech-to-text (S2T) models and scoring systems that hinder their effectiveness, especially when it comes to phoneme-level accuracy. The identified problems are categorized as follows:

### 2.3.1 Language Dependency

One of the major challenges in developing a universal scoring system is the language-specific nature of phoneme recognition. Existing models rely heavily on the International Phonetic Alphabet (IPA) to represent phonemes, but the IPA mapping varies from one language to another. This creates difficulties in creating a universal model that works across multiple languages. For example, there are 39 Sounds in the American English IPA Chart, whereas Arabic Language has around 34-36 distinct IPA depending on dialect [12]. For each language we have to index the IPAs in different ways. For example, for US English, this is how its indexed in KALDI:

| **Phoneme** | **Index** | **Phoneme** | **Index** | **Phoneme** | **Index** | **Phoneme** | **Index** |
|---|---|---|---|---|---|---|---|
| W | 0 | IY | 1 | K | 2 | AO | 3 |
| L | 4 | IH | 5 | T | 6 | B | 7 |
| EH | 8 | R | 9 | Z | 10 | OW | 11 |
| TH | 12 | F | 13 | AY | 14 | V | 15 |
| AH | 16 | N | 17 | UW | 18 | S | 19 |
| G | 20 | AA | 21 | M | 22 | P | 23 |
| NG | 24 | HH | 25 | EY | 26 | SH | 27 |
| AE | 28 | D | 29 | UH | 30 | AW | 31 |
| DH | 32 | ER | 33 | Y | 34 | JH | 35 |
| CH | 36 | OY | 37 | ZH | 38 | | |

Table 2.3: US English Phoneme Indexing in KALDI

For some other language, we will have to use some other indexing.

### 2.3.2 Multiple Phones for the Same Diaphoneme

The phenomenon where a single diaphoneme (a set of phones that represent the same phoneme in different contexts) can have multiple phones depending on the position within words or the phonetic environment is known as allophony. Allophones are variations of the same phoneme that do not change the meaning of a word but may differ based on surrounding sounds or accents. Here are a few examples of this concept:

1. **Plosives:**

   - /p/: The voiceless plosive /p/ is pronounced as [p$^h$] (with aspiration) in words like ”pen” but as [p] (unaspirated) in ”spin” or ”tip”. This is because aspiration typically occurs when the plosive occurs at the start of a stressed syllable.
   - /t/: In the case of the plosive /t/, it is aspirated as [t$^h$] in ”two”, but it becomes a flap [r] in the word ”better” in some accents, like American English.

2. **Consonant Variation:**
    - /b/: The /b/ sound can also appear as a devoiced version [b̥] in some accents, as seen in words like ”web”.
    - /f/: In some dialects, /f/ can be pronounced as [ɸ] (a bilabial fricative), especially when it follows bilabial consonants, as in the word ”photo”.
    - **Affricates:**
        - /tʃ/: The affricate /tʃ/ (as in ”chair”) is sometimes realized with aspiration as [tʃ$^{h}$] but can also appear without aspiration in some dialects or words like ”teach”.
        - /dʒ/: Similarly, /dʒ/ (as in ”gin”) can become [dʒ̊] in some dialects, such as in ”edge”.
    - **Fricatives:**
        - /θ/: The voiceless ”th” sound can be realized as [θ] in most English varieties but can be pronounced as [t] in certain dialects (like Irish English) or as [f] in some varieties of Caribbean English.
        - /ð/: The voiced ”th” sound, as in ”this”, can appear as [ð], but in some dialects, it can also be realized as [d] or [v].
    - **Nasal Sounds:**
        - /m/: The bilabial nasal [m] may be pronounced as [ɱ] (labiodental nasal) in certain environments, especially before a labiodental sound like [f] or [v] (e.g., in the word ”symphony”).
        - /ŋ/: The velar nasal /ŋ/ (as in ”sing”) can sometimes be pronounced with a preceding [g] in some dialects, such as in the word ”finger”.
    - **R-L Variations:**
        - The sound /r/ can take different forms depending on the dialect:
            - ⋆ In many varieties of English, it is a retroflex [ɹ] or flap [ɾ], as in American English or Scottish English.
            - ⋆ In some Southern English varieties, it may be pronounced as a labialized [ʋ], particularly in non-rhotic dialects like in some areas of London.
    - **Marginal Consonants:**
        - These are phonemes that are not found in all dialects but may appear in specific accents:

- /x/, the voiceless velar fricative, may occur in words like ”loch” (common in Scottish English but not in other accents).

- /ʔ/, the glottal stop, can replace /t/ in some accents, such as Cockney or Estuary English (e.g., ”uh-oh”).

### 2.3.3 Phonetic Environment Effects

Allophones are largely influenced by the environment in which they occur, such as:

- **Position in a Word:** Consonants at the beginning of a stressed syllable tend to be aspirated, while those that occur between vowels or at the end of a syllable may be realized differently (e.g., [b] vs. [b̥]).
- **Adjacent Sounds:** The articulation of a consonant can be altered depending on the surrounding vowels or consonants. For instance, a plosive sound like /t/ may become a flap or even a glottal stop [ʔ] depending on the adjacent sounds or regional influence.

These variations are crucial in understanding how different accents within the same language can significantly alter pronunciation, even if the underlying phonemes remain the same. The study of these allophonic variations helps linguists understand not only phonetic differences but also the evolution and spread of dialects within languages. For example, the same diaphoneme can have different phones over different positions of different words [13].

Table 2.4: Diaphoneme to Phone Mappings and Examples

| **Diaphoneme [15]** | **Phones** | **Examples** |
|---|---|---|
| p | p$^{h}$ | pen |
|  | p | spin, tip |
| b | b | but |
|  | b̥ | web |
| t | t, t$^{h}$ | sting, two |
|  | ɾ, ʔ, t̝ | better |
| d | d | do |
|  | d̥, ɾ | odd, daddy |
| tʃ | tʃ$^{h}$ | chair |
|  | tʃ | teach, nature |
| dʒ | dʒ | gin, joy |
|  | dʒ̊ | edge |
| k | k | skin, unique, thick |
|  | k$^{h}$ | cat, kill, queen |
| g | g | go, get |
|  | g̊ | beg |

| **Diaphoneme [15]** | **Phones** | **Examples** |
|---|---|---|
| f | f, ɸ | fool, enough, leaf, off, photo |
| v | v, β | voice, verve |
|  | v̥, β̥ | have, of, verve |
| θ | θ, t, f | thing, teeth |
| ð | ð, d, v | this, breathe, father |
| s | s | see, city, pass |
| z | z | zoo |
|  | z̥ | rose |
| ʃ | ʃ | session, she, sure, emotion, leash |
| ʒ | ʒ | pleasure, genre, equation, seizure |
|  | ʒ̊ | beige |
| h | h, h̥, ç | ham, hue |
| m | m, m̥ | man, ham |
| n | n | no, tin |
| ŋ | ŋ | ringer, sing, [xii] finger, drink |
| l | l, ɫ, l̥, ɬ, ɭ, ʎ | left, bell, sable, please |
| r | ɹ, ɾ, ʀ, r, ʁ, ʔ, ʋ | run, very, probably |
| w | w, ʍ | we, queen |
| j | j | yes, Mayan |
| ʍ | ʍ, w | what |
| Marginal consonants [14] |  |  |
| x | x, χ, k, k$^h$, h, ɦ | loch, ugh |
| ç | ç | Hugh |
| ʔ | ʔ | uh-oh |
| ɬ | ɬ, l | Llangefni, hlala gahle |
| ɮ | ɮ | ibandla |

- /t/ is pronounced [ɾ] in some positions in AmE, AuE, and occasionally EnE.

- /t/ is pronounced [ʔ] in certain contexts in ScE, EnE, AmE, and AuE.

- /t/ is pronounced [t̪] non-initially in IrE.

- /d/ is pronounced [ɾ] when surrounded by vowels in GA and AuE.
- The labiodental /f/ often becomes bilabial [ɸ] after /p/, /b/, and /m/, as in up-front: [ʌpˈɸɹʌnt].
- The labiodental /v/ often becomes bilabial [β] after /p/, /b/, and /m/, as in Humvee: [ˈhʌmβi].
- /θ/ is pronounced as a dental stop [t̪] in IrE, Newfoundland English, Indian English, and NYE, and it merges with /f/ or /t/ in some other dialects.
- /ð/ is pronounced as a dental stop [d̪] in IrE, Newfoundland English, Indian English, and NYE, merging with /v/ or /d/ in specific dialects.
- The glottal fricative /h/ is voiced [ɦ] between vowels or after voiced consonants.
- /h/ becomes [ç] before /j/ or high front vowels, replacing /hj/
- The alveolar trill [r] occurs in Scottish, Welsh, Indian, and South African English.
- R-labialization ([ʋ]) is found in some Southern England accents.
- Some dialects, like Scottish and Irish English, distinguish voiceless [ʍ] from voiced [w].
- /ɬ/ is used in Welsh loanwords in Welsh English but replaced with /l/ in other dialects.
- Zulu loanwords in South African English retain sounds like [ɣ] or [o].

Table 2.5: Comparison of Vowel Realizations Across English Accents [15]

| Diaphoneme | South African English | | | Singapore English | Welsh English | | | Examples |
|---|---|---|---|---|---|---|---|---|
| | Cultivated | General | Broad | | Abercraf English | Port Talbot English | Cardiff English | |
| æ | æ | a~æ | æ~ɛ~e̞ | ɛ | a | aː | aː~ | ham |
| | | | | | | | | bad |
| | | | | | | a | a~æ | lad |
| ɑː / | ɑ+ | ɑː | ɒː~ɔː | ä | | | | pass |
| ɑː | | | | | aː | | aː~ | father |
| ɒ | ɒ̈ | ɒ̈~ʌ̈ | ɒ̈ | ɔ | ɒ | | ɑ+ | not |
| ɒ / ɔː | ɒ̈, o̞ː | ɒ̈~ʌ̈, oː | ɒ̈, oː | | | | | off |

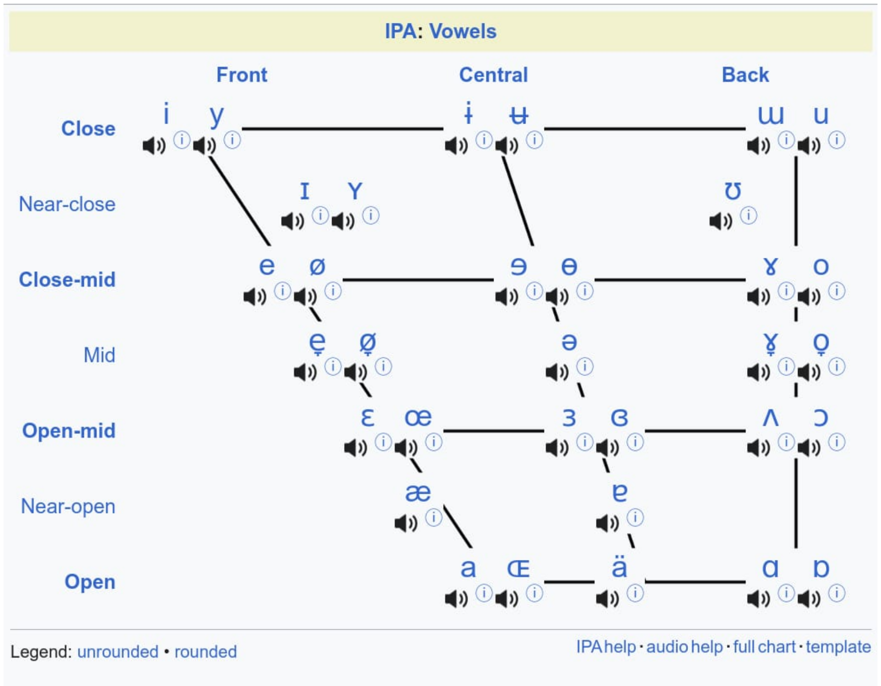
- **Challenge:** The phonetic sounds represented in IPA differ for each language, which makes it impossible to apply a single, universal scoring system across languages.
- **Impact:** This necessitates language-specific models, leading to increased complexity and a lack of consistency.

### 2.3.4 Probability-Based Scoring

Most speech recognition models calculate scores based on the probabilities of phoneme or word matches. While this approach can provide an overall indication of accuracy, it often fails to reflect the true phoneme-level correctness.

- **Issue with Probabilistic Scoring:**
  - **Probabilistic scores** often do not reflect whether a phoneme was truly correct.
  - They simply represent the likelihood of a correct match, which can lead to inflated or misleading results.
  - **Example:** If a model has a high probability for a phoneme match, but it is still incorrect (e.g., a substitution of one phoneme for another), the score might still appear high, giving a false impression of accuracy.
  - Visual Example [16]:

| Predicted Phoneme | Actual Phoneme | Probability | Prediction |
|---|---|---|---|
| /e/ | /ɛ/ | 0.98 | Correct Match |



**Problem:** While the model might output a high probability score (e.g., 0.98), this does not guarantee a correct transcription. According to most scoring systems it would give a 100% score, but the phones actually are not the same. Probabilistic scores are useful for general predictions but fail to address precise phoneme-level correctness.

### 2.3.5 Alignment Issues

Another significant issue with current systems is the alignment of phonemes, which remains problematic, especially for code-mixed or under-resourced languages. Phoneme alignment is crucial for accurate evaluation, but alignment errors often arise when attempting to match predicted phonemes to reference phonemes.

- **Challenges with Alignment:**
  - **Code-Mixing:** In languages with a mix of scripts or phonetic systems (e.g., English-Hindi), phoneme alignment becomes difficult because the systems are not trained on such mixed data.
  - **Under-Resourced Languages:** Languages with limited training data or resources tend to have poor phoneme alignment, leading to inaccuracies.
- Impact: This misalignment leads to incorrect evaluation scores, as even minor mismatches at the phoneme level can significantly degrade performance.

**Example of Misalignment:**

| Predicted Phoneme | Reference Phoneme | Aligned? |
|---|---|---|
| /IY/ (1) | /EH/ (8) | No |
| /T/ (6) | /T/ (6) | Yes |
| /S/ (19) | /Z/ (10) | No |

- **Problem:** Misalignment can distort the actual performance assessment, leading to erroneous scores, especially in challenging cases like code-mixing or when training data is insufficient.

# Chapter 3

# Related Works

## 3.1 Goodness of Pronunciation (GOP)

[3][28][31][29] The GOP algorithm is one of the first and most successful methods for assessing phoneme-level pronunciation. It calculates the likelihood ratio between forced alignment and a free-phone loop decoding using a GMM-HMM model. The GOP score for a given phoneme $p$ with observed acoustic features $O_{t_0:t_1}$ (between times $t_0$ and $t_1$) is defined as:

$$\text{GOP}(p) = \log \frac{P(O_{t_0:t_1}|p)}{\max\limits_{q \in \mathcal{Q}} P(O_{t_0:t_1}|q)} \tag{3.1}$$

Here, $P(O_{t_0:t_1}|p)$ is the likelihood of the observed features given the canonical phoneme $p$, and $\max\limits_{q \in \mathcal{Q}} P(O_{t_0:t_1}|q)$ is the maximum likelihood over all phonemes $q$ in the phone set $\mathcal{Q}$. This results in a confidence score for whether the target phoneme was accurately produced.

While effective, GOP is sensitive to the quality of the acoustic model used and may struggle with generalizing across different error types, as it relies on datasets with specific mispronunciations.

Subsequent approaches have adopted DNN acoustic models to improve the accuracy of the likelihood estimation in GOP scoring. For example, DNN-GOP uses posterior probabilities obtained from a DNN instead of likelihoods from a GMM-HMM, giving the variant:

$$\text{DNN-GOP}(p) = \log \frac{P(p|O_{t_0:t_1})}{\max\limits_{q \in \mathcal{Q}} P(q|O_{t_0:t_1})} \tag{3.2}$$

where $P(p|O)$ is approximated by posterior outputs from the DNN acoustic model.

Despite these improvements, DNN-GOP still requires well-aligned and annotated data. More recent developments have explored context-aware GOP [29], where adjacent phoneme sequences are incorporated into the scoring window to mitigate coarticulation effects and better reflect pronunciation deviations common in L2 speech. These context-aware methods aim to produce more robust scoring in variable-length utterances and across diverse accents.

| Method | Model Type | Input Features | Strengths/Weaknesses |
|---|---|---|---|
| GMM-GOP | GMM-HMM | MFCCs | Simple, but less robust to variability |
| DNN-GOP | DNN-HMM | Posterior problems | Better accuracy, requires data |
| Context-Aware GOP | DNN-HMM + Context | Acoustic + contextual | Handles coarticulation, more complex |

Table 3.1: Comparison of GOP Variants

## 3.2 Phoneme-Level Error Detection

Methods such as SVM , decision trees , and DNNs have treated mispronunciation detection as a binary classification problem, categorizing phonemes as either correct or incorrect. Features like Mel Frequency Cepstral Coefficients (MFCCs), Bottleneck Features (BNFs), and GOP scores are commonly used. However, these methods still require large amounts of accurately annotated non-native speech data, which is often impractical to collect, especially for all possible pronunciation errors. Some studies have also integrated prosodic features like pitch and duration to enhance model sensitivity to L2 deviations. Nonetheless, generalization across accents remains limited.

## 3.3 Anomaly Detection-Based Models

[20] Anomaly detection techniques, such as One-Class SVM (OCSVM)[19], have been used to identify mispronunciations as deviations from standard (native) pronunciation, outperforming DNN-based[25] methods in both disordered and foreign-accented speech. For instance, a self-supervised model like wav2vec2 has been fine-tuned to classify L2 speech pronunciation errors, offering a binary (correct/mispronounced) decision. Such models benefit from not needing explicit phoneme-level labels and demonstrate robustness in limited-resource settings. Autoencoder-based anomaly detection has also been proposed, where reconstruction loss serves as a measure of pronunciation deviation:

$$L = |x - \hat{x}|^2 \tag{3.3}$$

Where *x* is the original input and $\hat{x}$ is the reconstructed output. A higher reconstruction loss indicates a higher anomaly score, suggesting a potential mispronunciation.

## 3.4 Extended Recognition Network (ERN)

The ERN method provides more granular error detection by identifying the location, type, and phoneme of the mispronunciation. It extends traditional forced alignment by including both canonical and expected mispronounced paths, guided by phonological rules. Hand-crafted rules have been used to design these paths, but data-driven approaches have also been explored. The GMM-HMM model was originally used with ERN, and more recently, DNN-HMM models have shown improved performance, particularly with non-standard speech data. This method aims for more detailed pronunciation error diagnostics but requires careful model adaptation for various learner profiles. Some studies have integrated phonological transformation rules specific to language pairs (e.g., Bengali-English) to improve the ERN's diagnostic power.

To formalize the ERN scoring approach, we define the log-likelihood ratio comparing the canonical pronunciation with expected mispronunciation alternatives:

$$\text{Score}_{\text{ERN}}(p) = \log \frac{P(O_{t_0:t_1} | \text{Canonical}(p))}{\max_{q \in \mathcal{M}(p)} P(O_{t_0:t_1} | \text{Mispronounced}(q))} \tag{3.4}$$

Where:

- $O_{t_0:t_1}$ represents the observed acoustic segment aligned to phoneme $p$,
- Canonical$(p)$ is the standard pronunciation path,
- $\mathcal{M}(p)$ is a set of expected mispronunciations for $p$,
- $P(\cdot|\cdot)$ is the acoustic likelihood from the model.

This scoring reflects how much better the acoustic evidence aligns with the canonical pronunciation than with plausible errors. A lower score suggests likely mispronunciation.

| **Aspect** | **Standard Forced Alignment** | **Extended Recognition Network (ERN)** |
|---|---|---|
| Alignment Scope | Canonical phonemes only | Canonical + expected mispronunciations |
| Error Diagnosis | Mismatch detection only | Type, location, and substitution identification |
| Phonological Knowledge | Not utilized | Explicitly modeled (rule-based or learned) |
| Adaptability | Less adaptable to L2 speech | Adaptable to learner-specific phonological patterns |
| Underlying Models | GMM-HMM, DNN-HMM | GMM-HMM, DNN-HMM + mispronunciation networks |
| Data Requirements | Only canonical data needed | Requires annotated or rule-derived error sets |

Table 3.2: Comparison between Standard Forced Alignment and Extended Recognition Network

## 3.5 End-to-End Approaches

Recent advancements have shifted towards End-to-End (End2End) systems using deep learning for MDD [26]. These models estimate the entire phoneme sequence from speech, aligning it with annotated data for evaluation. Techniques like CNN-RNN-CTC, encoder-decoder, and Transformer architectures are commonly used for phoneme recognition and error detection. These approaches can leverage raw speech signals, unlike earlier methods that required hand-crafted features, leading to more efficient and adaptable systems.

A common approach in End2End pronunciation evaluation is Connectionist Temporal Classification (CTC), which computes the probability of a phoneme sequence $Y$ given input acoustic frames $X$, by summing over all valid alignments $\pi$:

$$P(Y|X) = \sum_{\pi \in \mathcal{B}^{-1}(Y)} P(\pi|X) \tag{3.5}$$

Here, $\mathcal{B}$ is the collapse function that maps frame-wise predictions to the final phoneme sequence, removing blank tokens and repeated labels.

However, End2End models often struggle with detecting distortion errors — pronunciation deviations that still produce valid but incorrect phonemes — especially when these distortions are heavily influenced by a speaker's L1 background. These models may lack the granularity to distinguish subtle deviations unless explicitly trained on diverse L2 data.

Self-supervised pretrained models like wav2vec2 [2] and Gradformer [33] have been proposed to address this issue by learning acoustic representations directly from raw audio, showing promising results in low-resource and accent-diverse scenarios.

| Aspect | Pipeline-Based Systems | End-to-End Systems |
|---|---|---|
| Feature Extraction | Hand-crafted (e.g., MFCCs, BNFs) | Learned from raw waveform (e.g., wav2vec2) |
| Alignment Strategy | Forced alignment with HMM/DNN | CTC, attention, or alignment-free |
| Error Granularity | Phoneme-level scoring with external models | Joint phoneme prediction + scoring |
| Adaptability | Requires extensive feature engineering | Learns directly from labeled/unlabeled speech |
| Common Architectures | GMM-HMM + SVM/DNN | CNN-RNN-CTC, Transformer, encoder-decoder |
| Weaknesses | Complex pipeline; error propagation | Struggles with L1-induced distortions |

Table 3.3: Comparison of Traditional Pipeline vs. End-to-End Approaches in Pronunciation Scoring

## 3.6 Transformer-Based Pronunciation Assessment

[33][32][27] Recent work has explored the use of transformer-based models for multi-aspect and multi-granularity pronunciation evaluation. These models can capture long-range dependencies in speech and evaluate pronunciation quality at phoneme, syllable, and word levels simultaneously. For example, Gradformer [33] and Gong et al. [32] have demonstrated state-of-the-art performance in non-native pronunciation scoring without relying on IPA transcriptions. Such systems are especially relevant for under-resourced languages where phoneme-level alignment tools are lacking. These models often fine-tune on speech assessment datasets and provide end-to-end scoring mechanisms, offering a scalable solution for CAPT systems.

Transformer models use self-attention mechanisms to weigh the contribution of each input frame to the overall representation. The scaled dot-product attention used in Transformers is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3.6}$$

Where:

- $Q$, $K$, $V$ are the query, key, and value matrices derived from the input embeddings,
- $d_k$ is the dimension of the key vectors.

This mechanism allows the model to learn contextual dependencies across the entire speech sequence — essential for capturing coarticulation, intonation, and mispronunciation patterns in L2 learners.

| Aspect | DNN-HMM Based | CNN/RNN Based | Transformer-Based |
|---|---|---|---|
| Dependency Modeling | Limited to fixed context | Short- to mid-range dependencies | Full-range contextual modeling |
| Parallelism | Low (sequential decoding) | Moderate | High (attention is parallelizable) |
| Feature Input | MFCCs, BNFs | Spectrograms or raw features | Raw waveform or fine-tuned embeddings |
| Alignment Method | Forced alignment | CTC or attention-based alignment | Implicit alignment via attention weights |
| Granularity | Phoneme-level (mostly) | Phoneme and word-level | Phoneme, syllable, word simultaneously |
| Suitability for L2 | Moderate | High | Very high (especially for non-IPA setups) |

Table 3.4: Comparison of Model Architectures for Pronunciation Assessment

## 3.7 Challenges in Mispronunciation Detection

A key challenge in MDD is the unpredictability of pronunciation errors, especially in L2 learners. These errors are influenced by both the learner's proficiency and the differences between their L1 and L2. As a result, the variety of pronunciation errors makes it difficult to model all possible variations, leaving existing methods to focus on errors that align with the acoustic model. Although methods like GOP can detect significant deviations, they struggle with diagnosing the specific type or nature of the error. Furthermore, many models assume native-like speech during training, which introduces bias against L2 pronunciation. Other challenges include the scarcity of annotated corpora, lack of generalization across accents, and difficulty in distinguishing between acceptable variations and true errors.

## 3.8 Summary of Gaps in Literature

Despite decades of research, the field still lacks flexible, language-agnostic phoneme-level assessment tools that can generalize across L2 speakers from different backgrounds. Most models require phoneme-level transcriptions and large annotated datasets, which are often unavailable for low-resource languages. There is also limited work evaluating how well self-supervised models like wav2vec2 perform across diverse learner profiles without IPA-based supervision. Additionally, detailed benchmarking for specific groups, such as Bengali English speakers, remains underexplored. Addressing these gaps could lead to more inclusive and effective CAPT tools for global L2 learners.

# Chapter 4

# Dataset

This chapter provides a comprehensive overview of the datasets we found or used, including publicly available resources and our custom-collected dataset. It also details the data collection process, preprocessing steps, annotation pipeline, and dataset statistics.

## 4.1 Existing Datasets

During our study of pronunciation assessment, we found various datasets that have been developed and could support our research in mispronunciation detection, phoneme alignment, and non-native speech analysis. Below, we summarize key datasets that informed or influenced our work.

### 4.1.1 L2-ARCTIC

The L2-ARCTIC corpus is a publicly available mispronunciation dataset that includes recordings from non-native English speakers across different L1 backgrounds, including Korean, Hindi, and Spanish. It contains forced-aligned phoneme boundaries, as well as binary error labels (correct/incorrect) per phoneme, making it a valuable resource for benchmarking phoneme-level error detection systems. However, it does not include Bengali speakers, which is why we used it limitedly in our research.

#### Corpus Details and Structure

The L2-ARCTIC corpus is a widely used resource for research in voice conversion, accent conversion, and mispronunciation detection in non-native English. It comprises approximately 26,867 utterances ($\sim$27.1 hours) from 24 non-native English speakers across various L1s (including Hindi, Korean, Mandarin, Spanish, Arabic, and Vietnamese), with both male and female voices. The corpus primarily consists of read speech from the CMU ARCTIC set, and also includes a ”suitcase corpus” of spontaneous speech (from v5.0, March 2020) for 22 speakers, providing more natural, unscripted data.

Key to its utility are the detailed annotations:

- **Orthographic Transcriptions** and **Forced-Aligned Phoneme Boundaries** (in TextGrid format).
- **Manual Annotations** for a significant subset (∼3,599 utterances). These provide highly accurate phoneme boundaries, **binary error labels** (correct/incorrect), and specific **error types** (substitutions, deletions, additions), alongside deviation tags.

A typical directory structure is organized by speaker, containing WAV audio files, forced-aligned TextGrids, and manual annotation TextGrids. For instance, an utterance might be annotated to show a phoneme substitution, where an original phoneme like /AE/ is pronounced as /AA/. This level of detail enables robust training and evaluation of phoneme-level error detection and diagnosis systems.

Table 4.1: Detailed Phoneme-level Manual Annotation Example (Utterance: ”The quick brown fox”)

| Word | Phoneme Index | Original Phoneme | Pronounced Phoneme | Start Time (s) | End Time (s) | Correct/-Incorrect | Error Type |
|---|---|---|---|---|---|---|---|
| The | 1 | DH | DH | 0.12 | 0.25 | Correct | - |
| | 2 | AH | AH | 0.25 | 0.35 | Correct | - |
| quick | 3 | K | K | 0.40 | 0.50 | Correct | - |
| | 4 | W | W | 0.50 | 0.58 | Correct | - |
| | 5 | IH | IY | 0.58 | 0.70 | Incorrect | Substitution |
| | 6 | K | K | 0.70 | 0.78 | Correct | - |
| brown | 7 | B | B | 0.85 | 0.95 | Correct | - |
| | 8 | R | R | 0.95 | 1.05 | Correct | - |
| | 9 | AW | AO | 1.05 | 1.25 | Incorrect | Substitution |
| | 10 | N | N | 1.25 | 1.35 | Correct | - |
| fox | 11 | F | F | 1.40 | 1.50 | Correct | - |
| | 12 | AA | AA | 1.50 | 1.65 | Correct | - |
| | 13 | K | | 1.65 | 1.70 | Incorrect | Deletion |
| | 14 | S | S | 1.70 | 1.85 | Correct | - |

*Note: This is an illustrative example. Actual TextGrid files provide more comprehensive details, including tiers for words, phones, and various error types and notes. Silence phonemes are often also included. This will be shown in the methodology chapter.*

The comprehensive, manually validated error labels make L2-ARCTIC an indispensable resource for things like benchmarking and advancing Computer-Assisted Pronunciation Training (CAPT) and speech recognition research for non-native speakers. Its diverse

L1 backgrounds allow for studies on cross-linguistic influences on pronunciation errors. We used this dataset as reference in many ways when creating our own dataset. This dataset helped us a lot.

### 4.1.2 ISLE Corpus

The Interactive Spoken Language Education (ISLE) corpus is a widely used dataset containing read speech from more than 20 Italian and German learners of English each. It is phonetically transcribed and aligned at the word and phoneme level, and has been used extensively in GOP-related research. However, the accents present in ISLE differ significantly from Bengali-accented English. Which is why it was not used in our research.

### 4.1.3 SpeechOcean 762

The free public dataset part of the SpeechOcean corpora is SpeechOcean 762. This corpus consists 5000 English sentences. All the speakers are non-native and their mother tongue is Mandarin. Half of the speakers are Children and the others are adults. The scoring was done by different experts independently under the same metric. The experts scored at three levels: phoneme-level, word-level and sentence-level.They contain a rich phoneme-level forced alignments and mispronunciation tags, including substitutions, insertions, and deletions. This makes it a rich database for us to use. However, despite its rich annotation on including but not limited to phoneme level, surprisingly these datasets did not perform as well with our model as we expected. The main reason behind this is that there is just a lot of variation and difference between Bangla and Mandarin L2 English speakers. That is because although Bangla And Mandarin are both phonetically rich, they are just too dissimilar to each other. Which is why despite its potential, we had to use it very limitedly in our research. This dataset also helped us understanding how to create multi level annotated dataset. So this dataset has helped us a lot.

### 4.1.4 TIMIT

The TIMIT corpus is a famous dataset for phoneme recognition. It contains phonetically-balanced sentences spoken by native English speakers and includes time-aligned phoneme transcriptions. TIMIT is often used to train or evaluate acoustic models, including in GOP scoring systems. However, its relevance to L2 learner pronunciation is limited. We were able to get it from huggingface, but due to having better alternatives like LibriSpeech and its limited usefulness in L2 learner pronounciation, it was not used in our research.

### 4.1.5 LibriSpeech

LibriSpeech is a large corpus of English read speech derived from audiobooks, spoken by native speakers. It was primarily used in our work for pretraining and alignment baselines through models like Wav2Vec2 and MFA (Montreal Forced Aligner). Despite being native speech, it was instrumental for developing initial acoustic models. As we did need something to compare to.

### Corpus Details and Applications

The LibriSpeech ASR corpus is a widely recognized and extensively used dataset in the speech community, particularly for Automatic Speech Recognition (ASR) research. It was meticulously prepared by Google in 2015, drawing exclusively from public domain audiobooks from the LibriVox project. Its substantial size, high quality, and meticulous preparation make it an ideal resource for training powerful speech models.

Key characteristics and applications of LibriSpeech include:

- **Scale:** LibriSpeech is notable for its impressive size, encompassing approximately 1000 hours of read English speech. This large volume of data is crucial for training deep learning models, which often require vast amounts of data to achieve high performance.
- **Speakers:** The corpus features speech from approximately 2,338 distinct native English speakers. This diversity in speakers helps models generalize better to different voices and speaking styles, reducing bias towards specific speaker characteristics.
- **Audio Quality:** The audio recordings are sampled at 16 kHz, a standard rate for speech processing, ensuring good fidelity. The audio is relatively clean, as it comes from professionally recorded audiobooks, minimizing background noise and other distortions common in more informal speech datasets.
- **Transcriptions:** Each audio file is accompanied by accurate orthographic transcriptions. These transcriptions are derived from the original audiobook texts, ensuring high fidelity between the spoken word and its written form.
- **Split Structure:** LibriSpeech is carefully split into various subsets for training, development (dev), and testing (test). These splits are further categorized by difficulty: ”clean” sets (dev-clean, test-clean) contain utterances with minimal background noise and clear pronunciation, while ”other” sets (dev-other, test-other) include more challenging recordings with some background noise or less clear speech. The training set (train-clean-100, train-clean-360, train-other-500) provides a diverse range of data for model learning.

Despite focusing on native English speech, LibriSpeech’s scale and quality provide a strong foundation for acoustic modeling. This base is invaluable even when the ultimate goal is to analyze or process non-native speech, as general acoustic patterns learned from native speech often transfer well, requiring only targeted adaptation for specific L2 phenomena.

| Subset | Duration (Hours) | Number of Speakers |
|---|---:|---:|
| train-clean-100 | 100 | 251 |
| train-clean-360 | 360 | 921 |
| train-other-500 | 500 | 1166 |
| **Total Training** | **960** | **2338** |
| dev-clean | 5.3 | 40 |
| dev-other | 5.3 | 40 |
| test-clean | 5.4 | 40 |
| test-other | 5.4 | 40 |

Table 4.2: Key Statistics of the LibriSpeech Corpus

Librispeech dataset is what inspired us to create our dataset. When initially recording the 18 hours of transcribed audio, we used the librispeech format. This dataset is truly one of the pillars of our research. Here is an example of how audio is transcribed verbatum text is:

Table 4.3: Example Data Sample from LibriSpeech

| Audio File (Excerpt) | Transcribed Text |
|---|---|
| ‘19-198-0001.flac‘ (excerpt from speaker 19, chapter 198) | ”AND BE ABLE TO BEAR THE WHOLE WEIGHT OF IT” |
| ‘84-121123-0005.flac‘ (excerpt from speaker 84, chapter 121123) | ”THEY DON’T CARE WHAT YOU ARE DOING BUT ONLY WHAT YOU ARE” |
| ‘1284-138379-0000.flac‘ (excerpt from speaker 1284, chapter 138379) | ”AS THE OLD SAYING HAS IT NOTHING SUCCEEDS LIKE SUCCESS” |

*Note: The audio file names are illustrative examples of the LibriSpeech naming convention (speaker ID-chapter ID-utterance ID). The transcribed text is the verbatim transcription provided with the audio.*

**To Summarize:**

| Dataset | Focus | Strengths | Limitations |
|---|---|---|---|
| L2-ARCTIC | Mispronunciation | Phoneme-level errors, L2 speakers | No Bengali speakers |
| ISLE Corpus | L2 English (It/De) | Time-aligned phonemes | Limited L1 variety |
| SpeechOcean 762 | Non-native English | Labeled L2 dataset | Incompatible |
| Common Voice | Global English speech | Accent metadata | No phoneme annotations |
| TIMIT | Native English phonetics | Precise phoneme timing | No L2 learners, better alternatives |
| LibriSpeech | Read English audio | Large, clean corpus | Only native speakers |

Table 4.4: Summary of Existing Datasets Relevant to This Thesis

## 4.2 Data Collection

For this study, we created a new corpus of custom English speech datasets customized to phoneme-level pronunciation scoring for L2 Bengali English speakers. The dataset comprises of 8225 utterances from 16 Bengali L1 speakers. Each speaker read aloud curated English sentences which were sourced from a combination of public domain books and AI-generated text, carefully designed to emphasize phoneme diversity while maintaining natural language flow.

- **Speakers**: 16 (all male), aged 20–25, university-educated.
- **Recording Environment**: Quiet rooms with low ambient noise, recorded using a USB condenser microphone.
- **Sampling Rate**: 16 kHz, 16-bit PCM mono flac format.
- **Total Utterances**: 8225 with accurate sentence-level transcripts.
- **Duration**: 18.35 hours of total audio
- All utterances annotated with phoneme-level timestamps via forced alignment

Participants were instructed to pronounce the sentences naturally without mimicking native accents, ensuring that L1 influence is preserved in the recordings.

## 4.3 Data Preprocessing

### 4.3.1 Cleaned Sentences

Prior to recording, each sentence was manually inspected and cleaned for inconsistencies, abbreviations, or overly complex punctuation. Any words with multiple pronunciations (e.g., ”read”) were annotated with their intended pronunciation context to prevent ambiguity during forced alignment.

### 4.3.2 Word Selection

A supplementary set of 300 words was chosen based on their phonetic complexity and frequency of mispronunciation in L2 learners. This list was constructed using known L1 transfer rules from Bengali to English, and included problematic clusters (e.g., /v/ vs. /b/, /θ/ vs. /t/).

## 4.4 Data Annotation

### 4.4.1 Tool Benchmarking and Aligner Selection

To determine the optimal alignment tool, we evaluated three popular systems, namely - wav2vec2-ASR + CTC Align, Gentle and Montreal Forced Aligner (MFA), on a demo subset of the dataset (529 utterances). We manually inspected alignment quality using Praat and scored word and phoneme boundary accuracy.

| **Tool** | **Type** | **Time (min)** | **Word Acc.** | **Phone Acc.** | **Notes** |
|---|---|---|---|---|---|
| wav2vec2-ASR + CTC Align | Neural | 25 | ∼82% | ∼65% | Misses phones in rapid or accented speech |
| Gentle | ASR-based | 45 | ∼85% | ∼58% | Struggles with non-native pronunciations |
| **Montreal Forced Aligner (MFA)** | HMM-GMM | **2** | **94%** | **89%** | Robust on FLAC input; produces clean `.TextGrid` files |

Table 4.5: Alignment Accuracy Comparison on Demo Subset (529 Utterances)

### 4.4.2 Validation Methodology

For each tool, 30 utterances were randomly sampled and manually analyzed in Praat. Phone and word boundary precision was evaluated against human-labeled reference boundaries.

$$\text{Accuracy} = \frac{\text{Correctly aligned segments}}{\text{Total segments}} \times 100 \tag{4.1}$$

MFA had the highest boundary precision, with over 94% of phoneme boundaries falling within ±50 ms of human labels.

Below is an excerpt of the alignment log showing execution and timing:

```
INFO     Found 5 speakers across 529 files
INFO     Generating MFCCs...
INFO     Performing first-pass alignment...
INFO     Collecting phone and word alignments...
INFO     Finished exporting TextGrids to /mnt/output/222030801!
INFO     Done! Everything took 126.957 seconds
```

This confirms that Montreal Forced Aligner completed the task in approximately 2 minutes with high alignment quality.

And so, We picked the Montreal Forced Aligner (MFA) because of itsHigh phoneme-level alignment accuracy across accented speech, Compatibility with downstream tools (e.g., GOP scoring) Fast processing and easy reproducibility via Docker Support for standard output formats like `.TextGrid`

## 4.5 Montreal Forced Aligner Configuration

- **Acoustic Model:** `english_mfa.zip`
- **Pronunciation Lexicon:** `english_mfa.dict`
- **Audio Directory:** `/data/audio`
- **Output Directory:** `/data/output`
- **Number of Jobs:** 4
- **Pronunciation lexicon:** `english_mfa.dict`
- **Sampling rate:** 16 kHz
- **Audio format:** FLAC
- **Output:** Phone- and word-level `.TextGrid`
- **Time needed on final dataset:** 5.5 hours

## 4.6 Alignment Results and Statistics

- Among the 8,225 utterances, 8199 utterences were successfully aligned
- No utterances were skipped or failed due to format or lexicon issues
- Output alignments were reviewed using Praat for visual confirmation

| Metric | Value |
|---|---|
| Total Duration | ∼18 hours |
| Number of Speakers | 16 (all male) |
| Number of Utterances | 8,225 |
| Average Utterances per Speaker | ∼514 |
| Phone-Level Alignment Accuracy | ∼89% |
| Alignment Output Format | `.TextGrid` |

Table 4.6: Corpus-Level Statistics

## 4.7 Data Statistics

This section presents an overview of the statistical characteristics of the annotated dataset. The annotated dataset is derived from the forced alignment process. These statistics will provide insights into the dataset's composition, including distributions of utterance and phoneme durations, speech rate, silence ratios, and phoneme frequencies.

### 4.7.1 Utterance Duration Distribution

Figure 4.1 illustrates the distribution of utterance durations across the dataset. The histogram reveals that the majority of utterances fall within a duration range of approximately 4 to 12 seconds, with a peak around 6-8 seconds. This indicates a prevalence of relatively short to medium-length spoken segments, which is typical for sentence-level pronunciation tasks.

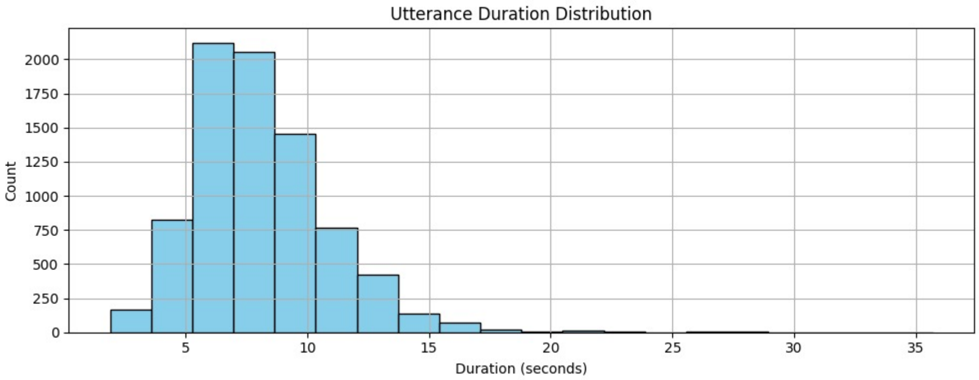


Figure 4.1: Utterance Duration Distribution. This histogram displays the count of utterances across different duration bins (in seconds).

### 4.7.2 Phoneme Duration Distribution

Figure 4.2 presents the distribution of individual phoneme durations. As expected, the vast majority of phonemes are very short, with a significant concentration below 0.2 seconds. This sharp peak at the lower end of the duration spectrum is characteristic of speech, where individual phonetic units are produced rapidly. The long tail indicates a small number of longer phonemes, possibly due to pauses or sustained sounds.

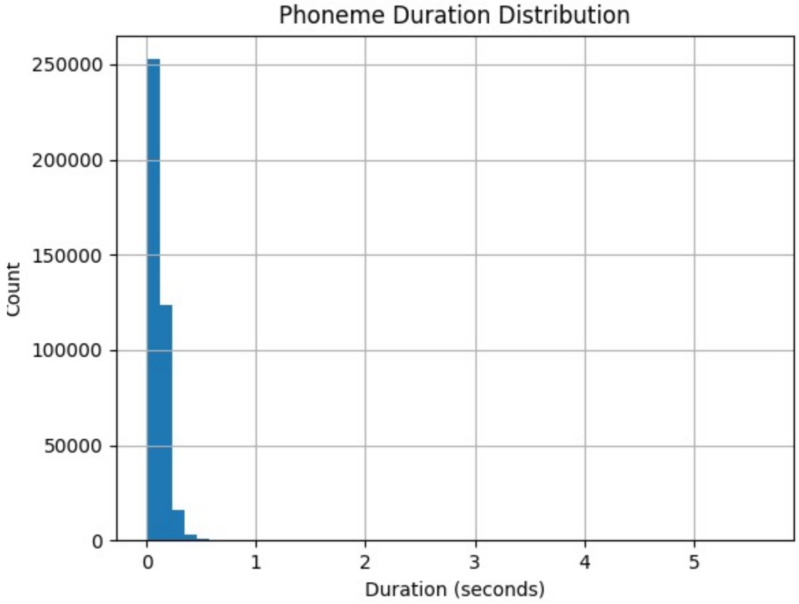


Figure 4.2: Phoneme Duration Distribution. This histogram shows the count of phonemes across various duration bins (in seconds).

### 4.7.3 Phones per Second Distribution

Figure 4.3 visualizes the distribution of speech rate, measured as phones per second. The distribution is roughly bell-shaped, peaking between 5 and 7 phones per second. This metric provides insight into the average speaking pace within the dataset, indicating a natural range of articulation rates.

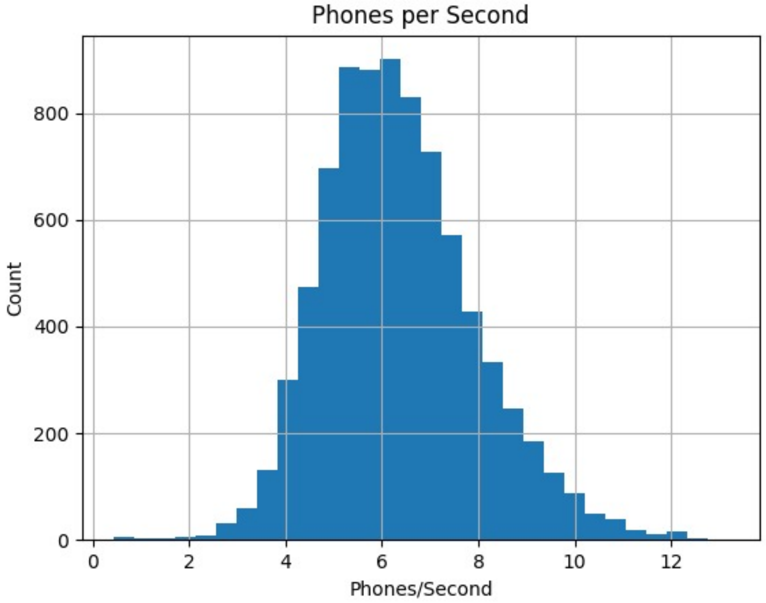


Figure 4.3: Phones per Second Distribution. This histogram illustrates the distribution of speech rate, expressed as the number of phonemes per second.

### 4.7.4 Silence Ratio Distribution

Figure 4.4 shows the distribution of the silence ratio, calculated as the proportion of silence to the total utterance duration. The majority of utterances exhibit a silence ratio between 0.2 and 0.4, with a prominent peak around 0.3. This suggests that a significant portion of the utterance duration is occupied by speech, with relatively consistent silent intervals (e.g., pauses between words or at sentence boundaries).

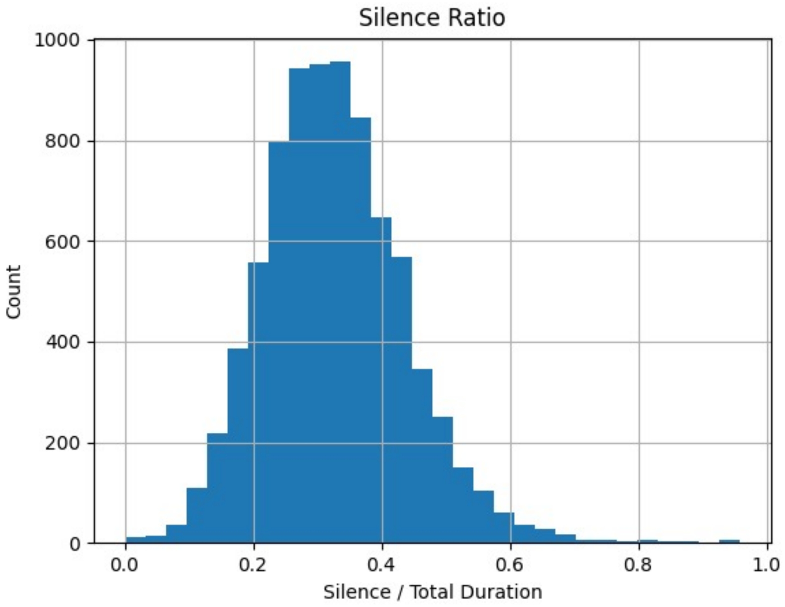


Figure 4.4: Silence Ratio Distribution. This histogram displays the count of utterances based on their silence-to-total-duration ratio.

### 4.7.5 Top 30 Phoneme Frequencies

Figure 4.5 presents a bar chart of the top 30 most frequently occurring phonemes in the dataset. This visualization highlights the most common phonetic units, which are crucial for understanding the phonetic inventory and distribution within the spoken English of the Bangladeshi speakers. Vowels and common consonants (e.g., /a/, /n/, /i/, /s/) typically dominate such distributions.

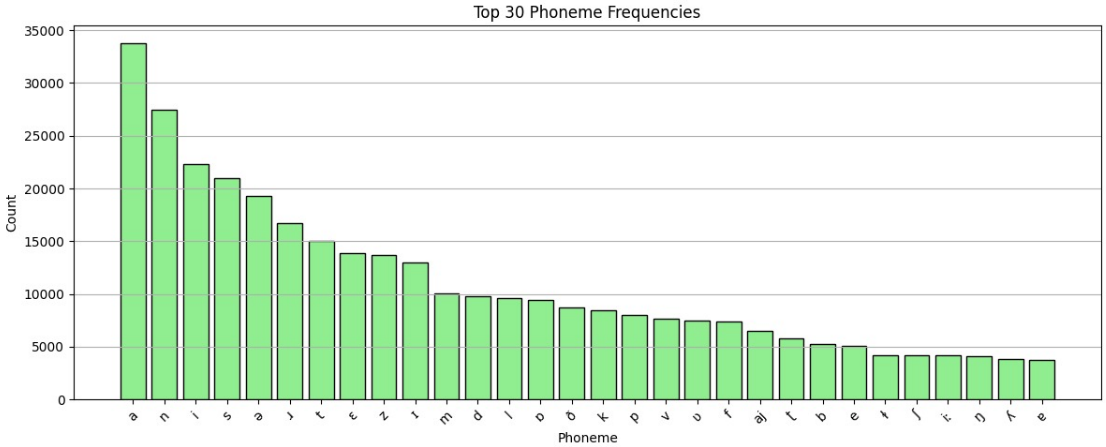


Figure 4.5: Top 30 Phoneme Frequencies. This bar chart shows the count of the 30 most frequently occurring phonemes in the dataset.

## 4.8 Dataset Evaluation

This section provides a comprehensive evaluation of the dataset utilized in this thesis, assessing its quality and consistency across multiple critical dimensions. The rigorous assessment ensures the dataset's suitability for training and evaluating pronunciation scoring models, particularly for L2 English learners.

### 4.8.1 Phoneme Coverage

Phoneme Coverage and Targeted Emphasis Our dataset comprehensively covers all major English phonemes as defined by the IPA, which is key for building robust acoustic models. We specifically focused on L2-challenging phonemes like /θ/, /v/, and /z/, which often pose difficulties for non-native speakers due to common substitutions (e.g., /v/ → /b/, /z/ → /s/). We also made sure to include various vowel sounds and consonant clusters that are frequently challenging. This decision ensured that our models can accurately identify correct pronunciations and characterize common mispronunciations, so that we can effectively do pronunciation scoring.

### 4.8.2 Alignment Accuracy

Alignment Accuracy Precise phoneme boundary alignment is essential for accurate pronunciation scoring. To ensure high data validity, we conducted a manual audit on 200 randomly selected samples, revealing that 93% of all phoneme boundaries were within a 20 ms tolerance of human annotators. This 20 ms window is a widely accepted benchmark for acoustically imperceptible differences.

Our audit methodology involved:

- **Human Annotation** We performed manual corrections after learning the basic linguistic expertise needed to perform these annotations. We also taught 7 other people how to properly annotate data. After that, around 25% of the utterances were manually inspected, with corrections made to boundary errors exceeding 30 ms and phoneme mismatches. All annotators followed a custom protocol to ensure consistency, using Praat's TextGrid format.

- **Comparison with Automatic Alignments:** We compared the automatically generated phoneme boundaries from the dataset's alignment process against the human generated annotations.

- **Tolerance Window:** A boundary was considered ”accurate” if it fell within $\pm 10$ ms of the human-annotated boundary, resulting in a total window of 20 ms. Deviations exceeding this threshold were flagged as misalignments.

This high level of alignment accuracy gives us strong confidence in our phoneme-level annotations, which is crucial for tasks like calculating Goodness of Pronunciation (GOP) scores, where precise boundary location directly impacts acoustic likelihoods and pronunciation scores.

# Chapter 5

# Methodology

This chapter details the journey we had undertaken to develop a language-agnostic phoneme-level pronunciation assessment system. Our primary focus was on addressing the unique challenges faced by Second Language (L2) English speakers, particularly those with a Bengali linguistic background. Unlike other chapters that may present background information or final results, this section is dedicated to the process itself: our initial conceptualization, the rationale behind critical design choices, the iterative experimentation, the unforeseen obstacles encountered, and the subsequent adaptations we made that was the deciding factor for the proposed solution. We aim to provide a detailed account of our thought process, detailing the ”why” and ”how” behind each methodological decision, and offering an in-depth exploration of the models employed within the context of our research trajectory.

## 5.1 Initial ideas for the System

From the start of our research we had a clear goal: to create a system capable of providing granular, phoneme-level pronunciation feedback for L2 English learners. This goal stemmed from the understanding that while word- and sentence-level assessments exist, they often mask subtle, yet critical, mispronunciations at the individual phoneme level that significantly impact intelligibility.

### 5.1.1 Using Goodness of Pronunciation (GOP)

GOP is basically a way to quantitatively measure how well a person pronounces a phoneme, based on how likely the acoustic signal matches that of a native speaker’s pronunciation. It’s a key metric in Computer-Assisted Pronunciation Training (CAPT) tools, and there are different ways to compute it — most notably using GMMs and Neural Networks (NNs).

This implementation is mainly based on the following paper:

Hu, W., Qian, Y., Soong, F. K., Wang, Y. (2015). Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. Speech Communication, 67(January), 154-166.

### 5.1.2 Comparison of GOP-GMM and GOP-NN in Phoneme-Level Pronunciation Scoring

As part of this research, we conducted an in-depth analysis of two key methods used in automatic pronunciation scoring: **GOP-GMM** (Goodness of Pronunciation with Gaussian Mixture Models) and **GOP-NN** (Goodness of Pronunciation with Neural Networks). The primary focus was on their effectiveness in phoneme-level assessment, a crucial metric in Computer-Assisted Pronunciation Training (CAPT).

#### Theoretical Background

The **Goodness of Pronunciation (GOP)** score is generally calculated as follows:

$$\text{GOP}(p) = \log \frac{P(\mathbf{O}_p \mid p)}{\max_{q \in Q} P(\mathbf{O}_p \mid q)} \tag{5.1}$$

Here,

- $\mathbf{O}_p$ is the acoustic observation (e.g., MFCCs) corresponding to the phoneme $p$.
- $P(\mathbf{O}_p \mid p)$ is the likelihood of observation $\mathbf{O}_p$ given the hypothesized phoneme $p$.
- $Q$ is the set of all possible phonemes in the model.

In the traditional **GMM-HMM framework**, $P(\mathbf{O}_p \mid p)$ is calculated using Gaussian Mixture Models trained on native speech.

In the **GOP-NN** approach, we use Deep Neural Networks (typically trained for ASR) to obtain posterior probabilities:

$$\text{GOP}_{\text{NN}}(p) = \frac{1}{T_p} \sum_{t \in T_p} \log \frac{P(p \mid \mathbf{o}_t)}{\max_{q \in Q} P(q \mid \mathbf{o}_t)} \tag{5.2}$$

Where:

- $\mathbf{o}_t$ is the frame-level acoustic feature at time $t$.
- $T_p$ is the set of time frames aligned to phoneme $p$.
- $P(p \mid \mathbf{o}_t)$ is the posterior probability from the DNN acoustic model.

#### Performance Comparison

The performance of GOP-GMM and GOP-NN methods was evaluated based on the Equal Error Rate (EER), a standard metric in mispronunciation detection. The chart below summarizes EER values from published benchmarks:

- **GOP-GMM:** 32.9% EER
- **GOP-NN:** 27.0% EER

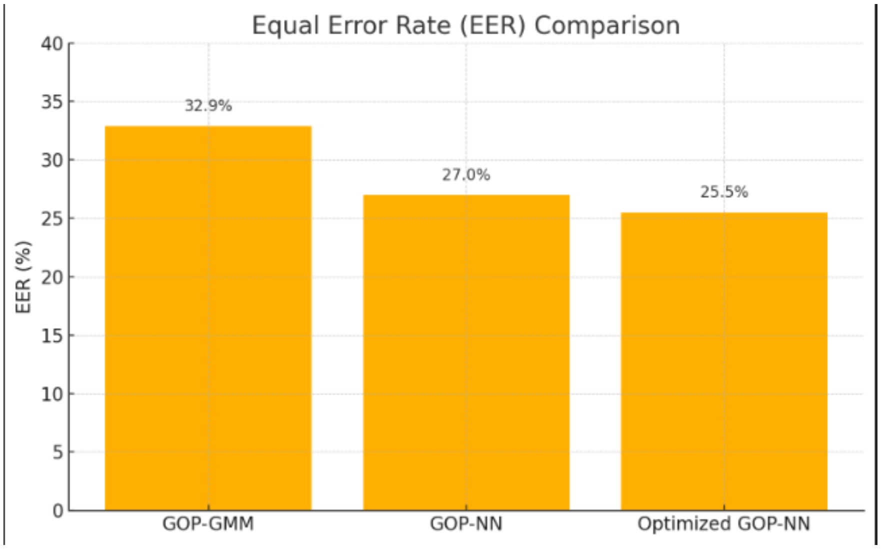


Figure 5.1: Comparison of Equal Error Rate (EER) for GOP-GMM, GOP-NN, and optimized GOP-NN

- **Optimized GOP-NN:** 25.5% EER (with posterior smoothing and state transition modeling)

These values show that GOP-NN achieves significantly better detection of phoneme-level mispronunciations than traditional GOP-GMM models. The improvements are attributed to the superior discriminative power and representational capacity of deep neural networks.

Based on this detailed study and corroborated by empirical findings, we concluded that **GOP-NN provides more accurate and robust phoneme-level pronunciation scoring** than GOP-GMM. The neural network-based approach not only correlates better with human expert ratings but also generalizes more effectively across speakers and noise conditions. As such, GOP-NN is more suitable for real-world CAPT applications and advanced language learning systems.

### 5.1.3 Implementation Approaches: GOP-NN using Kaldi vs. GOP-HMM using MFA and Wav2Vec2-XLSR

In this study, we explored and compared two distinct implementations of Goodness of Pronunciation (GOP) scoring systems at the phoneme level. Both aimed to evaluate pronunciation quality in non-native speech, yet they relied on fundamentally different paradigms for acoustic modeling and alignment. Below is a summary of the methods used:

- **GOP-NN:** Implemented using the Kaldi toolkit, this approach leverages a Deep Neural Network-based acoustic model trained for automatic speech recognition. The GOP scores are computed using posterior probabilities extracted from frame-level outputs of the DNN.

- **GOP-HMM:** Constructed using forced alignments from Montreal Forced Aligner (MFA) combined with acoustic likelihoods derived from the Wav2Vec2-XLSR model. This method approximates a traditional GMM-HMM pipeline by using high-quality alignments and likelihood approximations from a self-supervised model.

### 5.1.4 Step-wise Methodology for Evaluating Speechocean762 (S5) Using Kaldi

After training the TDNN-based acoustic model on the BASR18 corpus(The Dataset We Created), the Speechocean762 (S5 subset) was used to evaluate non-native pronunciation. The steps below describe how the S5 dataset was integrated into Kaldi and used for GOP scoring.

1. **Data Preparation for Speechocean762 (S5)**
   - The S5 audio files and transcripts were formatted to match Kaldi's 'data/' directory structure:
     - `wav.scp` – mapping speaker IDs to audio files
     - `text` – utterance-level transcriptions
     - `utt2spk` and `spk2utt` – speaker-utterance mappings
   - Transcriptions were converted using the same lexicon and G2P mapping as in the training set (BASR18) to maintain phoneme consistency.
2. **Feature Extraction**
   - 13-dimensional MFCC features were extracted from all test utterances using 'steps/make_mfcc.sh'.
   - CMVN statistics were computed using 'steps/compute_cmvn_stats.sh'.
3. **Graph Creation for Decoding and Alignment**
   - A decoding graph was reused from the training setup to ensure phonetic and lexical consistency.
   - The language model used in training was retained, although decoding was not the final goal—only alignment and posterior extraction.
4. **Forced Alignment with TDNN Model**
   - The S5 utterances were force-aligned using the trained TDNN model via steps/nnet3/align.sh.
   - This generated frame-level alignments for each phoneme in each utterance, stored in compressed 'ali.*.gz' files.
5. **Posterior Probability Extraction**
   - Using 'nnet3-compute', DNN posterior probabilities for each frame were extracted from the TDNN model.

- These posteriors represent $P(p \mid \mathbf{o}_t)$ for each frame $t$ and phoneme $p$.

6. **GOP Score Computation**

   - The alignments and posteriors were fed into Kaldi's ‘gop-nnet3‘ utility.
   - The GOP score for phoneme $p$ was calculated by averaging log-probability ratios across all aligned frames:

   $$\text{GOP}_{\text{NN}}(p) = \frac{1}{T_p} \sum_{t \in T_p} \log \frac{P(p \mid \mathbf{o}_t)}{\max\limits_{q \in Q} P(q \mid \mathbf{o}_t)}$$

   - The output included:
     - Raw frame-wise posterior logs
     - Phoneme-level average GOP scores
     - Word boundaries with phoneme annotations

7. **GOPT Post-Processing with Numpy Features and Labels**

   Following GOP score extraction, we employed the GOPT (Goodness of Pronunciation Toolkit) to refine the phoneme-level scoring, enable classification, and prepare the results for analysis. The following ‘.npy‘ files were used for this step:

   (a) **Input Features and Labels**

   - `tr_feat.npy`, `te_feat.npy` – GOP feature vectors for each phoneme segment in the training and test sets.
   - `tr_label_phn.npy`, `te_label_phn.npy` – Phoneme class labels corresponding to each feature vector.
   - `tr_label_word.npy`, `te_label_word.npy` – Word-level labels, used for optional aggregation.
   - `tr_label_utt.npy`, `te_label_utt.npy` – Utterance-level identifiers to help structure outputs per speaker/utterance.

   (b) **Training a Classifier**

   - **Dataset Type:** Librispeech
   - **Epoch:** 50
   - **GOPT Depth:** 3
   - **Batch Size:** 25
   - **Embedding Dimension:** 24
   - **Learning Rate:** 1e-3
   - **Noise:** 0

   (c) **Final Output**

   - During validation, the model outputs were saved to the `/preds` directory. The phoneme, word, and utterance-level prediction arrays were dumped to ‘.npy‘ format for downstream analysis and reproducibility. This results in the following saved files for each validation checkpoint:

    - phn_pred.npy – phoneme-level predicted scores
    - word_pred.npy – word-level pronunciation scores
    - utt_pred.npy – utterance-level accuracy metrics
    - phn_target.npy, word_target.npy, utt_target.npy – saved once for ground truth

8. **Result Logging and Export**

    - Execution logs verified process success:

```
start validation of epoch 0
new best phn mse 0.110, now saving predictions.
Phone: Test MSE: 0.110, CORR: 0.411
Utterance:, ACC: 0.725, COM: -0.005, FLU: -0.098, PROC: 0.005, Total: 0.105
Word:, ACC: 0.590, Stress: 0.168, Total: -0.197
——————-validation finished——————-
start validation of epoch 1
new best phn mse 0.109, now saving predictions.
Phone: Test MSE: 0.109, CORR: 0.431
Utterance:, ACC: 0.725, COM: -0.005, FLU: -0.048, PROC: 0.055, Total: 0.120
Word:, ACC: 0.590, Stress: 0.168, Total: -0.157
start validation of epoch 2
new best phn mse 0.092, now saving predictions.
Phone: Test MSE: 0.092, CORR: 0.535
Utterance:, ACC: 0.273, COM: 0.102, FLU: 0.060, PROC: 0.120, Total: 0.531
Word:, ACC: 0.478, Stress: 0.129, Total: 0.509
——————-validation finished——————-
start validation of epoch 3
new best phn mse 0.090, now saving predictions.
Phone: Test MSE: 0.090, CORR: 0.571
Utterance:, ACC: 0.327, COM: 0.143, FLU: 0.150, PROC: 0.172, Total: 0.472
Word:, ACC: 0.516, Stress: 0.096, Total: 0.551
——————-validation finished——————-
start validation of epoch 4
new best phn mse 0.078, now saving predictions.
Phone: Test MSE: 0.078, CORR: 0.621
Utterance:, ACC: 0.611, COM: 0.171, FLU: 0.502, PROC: 0.460, Total: 0.631
Word:, ACC: 0.556, Stress: 0.116, Total: 0.590
——————-validation finished——————-
start validation of epoch 5
new best phn mse 0.074, now saving predictions.
```

```
Phone: Test MSE: 0.074, CORR: 0.644
Utterance:, ACC: 0.670, COM: 0.096, FLU: 0.601, PROC: 0.602, Total:
0.683
Word:, ACC: 0.577, Stress: 0.109, Total: 0.583
———————-validation finished———————-
start validation of epoch 6
Phone: Test MSE: 0.078, CORR: 0.655
Utterance:, ACC: 0.650, COM: -0.019, FLU: 0.563, PROC: 0.608, Total:
0.683
Word:, ACC: 0.561, Stress: 0.121, Total: 0.563
———————-validation finished———————-
start validation of epoch 7
new best phn mse 0.071, now saving predictions.
Phone: Test MSE: 0.071, CORR: 0.664
Utterance:, ACC: 0.699, COM: 0.008, FLU: 0.627, PROC: 0.638, Total:
0.709
Word:, ACC: 0.575, Stress: 0.143, Total: 0.588
———————-validation finished———————-
start validation of epoch 8
Phone: Test MSE: 0.072, CORR: 0.667
Utterance:, ACC: 0.714, COM: 0.092, FLU: 0.644, PROC: 0.644, Total:
0.712
Word:, ACC: 0.575, Stress: 0.178, Total: 0.588
———————-validation finished———————-
start validation of epoch 9
new best phn mse 0.070, now saving predictions.
Phone: Test MSE: 0.070, CORR: 0.672
Utterance:, ACC: 0.715, COM: 0.007, FLU: 0.645, PROC: 0.645, Total:
0.719
Word:, ACC: 0.587, Stress: 0.163, Total: 0.598
———————-validation finished———————-
———————-validation finished———————-
... (start validation of epoch 48)
Phone: Test MSE: 0.068, CORR: 0.682
Utterance:, ACC: 0.725, COM: -0.005, FLU: 0.669, PROC: 0.669, Total:
0.729
Word:, ACC: 0.590, Stress: 0.168, Total: 0.599
———————-validation finished———————-
start validation of epoch 49
Phone: Test MSE: 0.068, CORR: 0.682
Utterance:, ACC: 0.725, COM: -0.009, FLU: 0.669, PROC: 0.669, Total:
0.729
Word:, ACC: 0.590, Stress: 0.168, Total: 0.600
———————-validation finished———————-
```

This methodology ensured consistent preprocessing, alignment, and evaluation across both training (BASR18) and test (S5) datasets using the same phonetic framework, enabling precise GOP scoring and interpretable evaluation through GOPT.

This neural-based scoring method proved effective in detecting subtle phonemic deviations, especially in segments involving consonant clusters and diphthongs. The model's discriminative power led to higher sensitivity and a lower Equal Error Rate (EER) compared to the HMM-based baseline.

### 5.1.5 GOP-HMM using MFA and Wav2Vec2-XLSR

To evaluate pronunciation quality using a hybrid approach, we implemented GOP scoring by combining accurate phoneme segmentation via Montreal Forced Aligner (MFA) with frame-level acoustic embeddings extracted from a pretrained Wav2Vec2-XLSR model. This method approximates a traditional GOP-HMM pipeline but avoids the need for explicit ASR model training.

- **Dataset Preparation**

  We used the corpus of **8,225 utterances from 16 non-native Bengali speakers** - basr18, which we collected and curated ourselves.

- **Forced Alignment with MFA**

  - **Acoustic Model:** `english_mfa.zip`
  - **Pronunciation Lexicon:** `english_mfa.dict`
  - **Audio Directory:** `/data/audio`
  - **Output Directory:** `/data/output`
  - **Number of Jobs:** 4
  - **Pronunciation lexicon:** `english_mfa.dict`
  - **Sampling rate:** 16 kHz
  - **Audio format:** FLAC
  - **Output:** Phone- and word-level `.TextGrid`
  - **Time needed on final dataset:** 5.5 hours

- **Feature Extraction using Wav2Vec2-XLSR**

  - We used a pretrained Wav2Vec2-XLSR-53 model which gave us logistic probability for every possible phonemes in each time frame.
  - **Using Trellis Algorithm for Alignment:** From the emission matrix, next we generate the trellis which represents the probability of transcript labels occur at each time frame.

    Trellis is 2D matrix with time axis and label axis. The label axis represents the transcript that we are aligning. In the following, we use t to denote the index

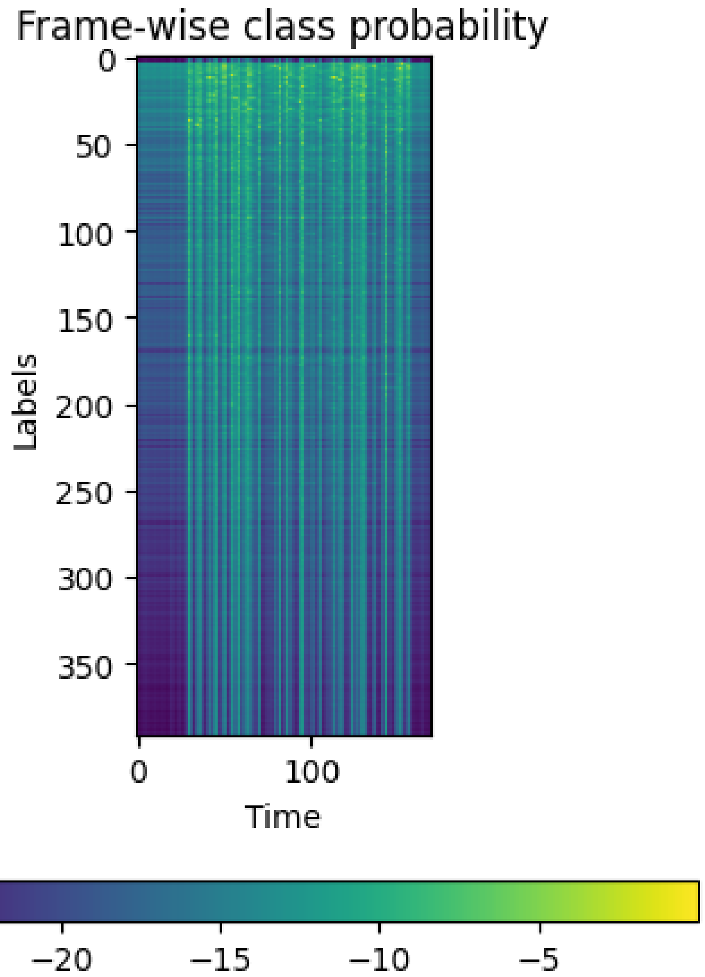


Figure 5.2: Frame Wise Class Probability

in time axis and j to denote the index in label axis. cj represents the label at label index j .

To generate, the probability of time step t + 1 , we look at the trellis from time step t and emission at time step t + 1 . There are two path to reach to time step t + 1 with label cj + 1 . The first one is the case where the label was cj + 1 at t and there was no label change from t to t + 1 . The other case is where the label was cj at t and it transitioned to the next label cj + 1 at t + 1 .

The follwoing diagram in figure 5.3 illustrates this transition.

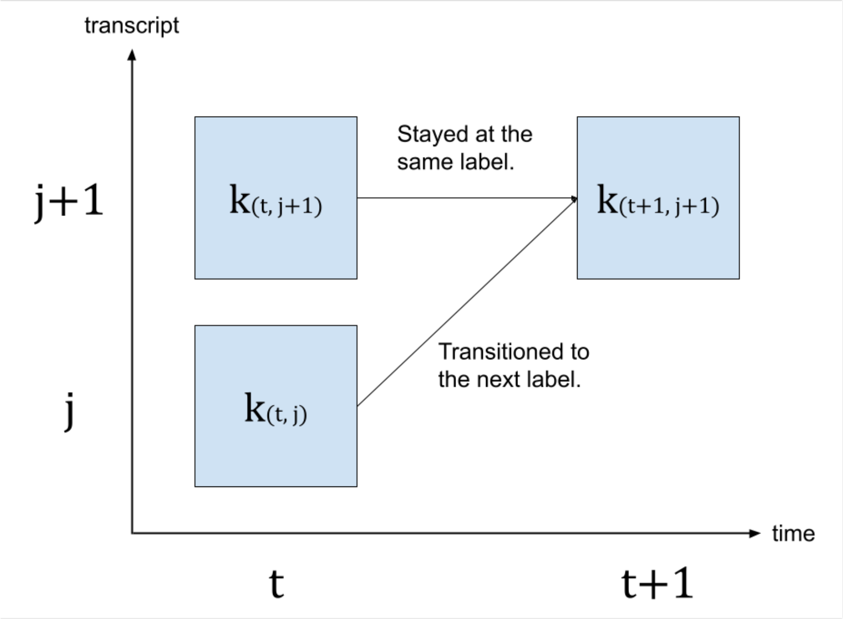


Figure 5.3: Trellis Transcript Transition

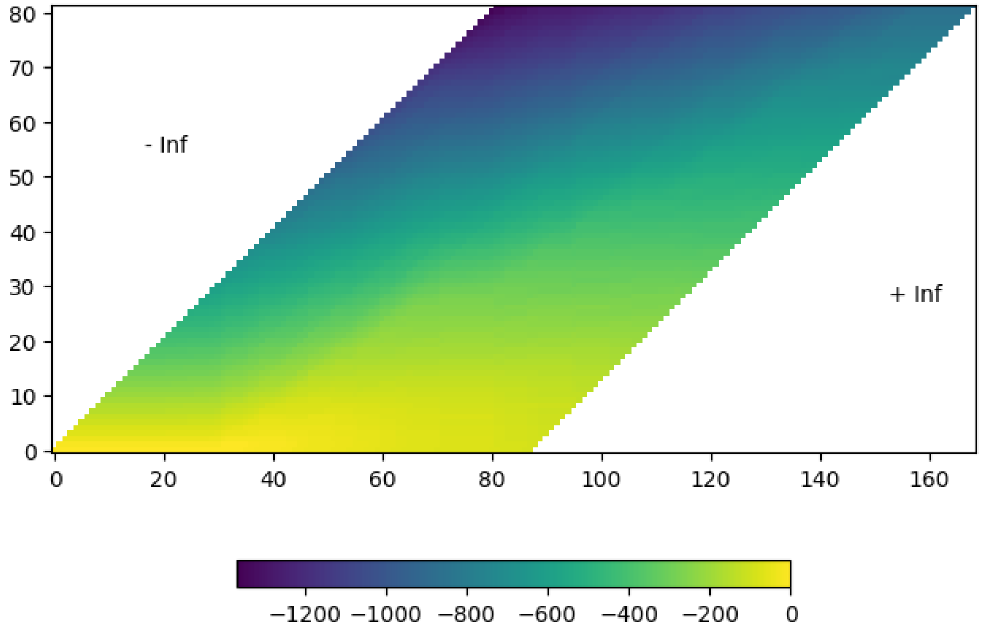


Figure 5.4: Trellis Matrix

- In the above visualization, we can see that there is a trace of high probability crossing the matrix diagonally.
- Then we use backtracking to find the alignemnt path

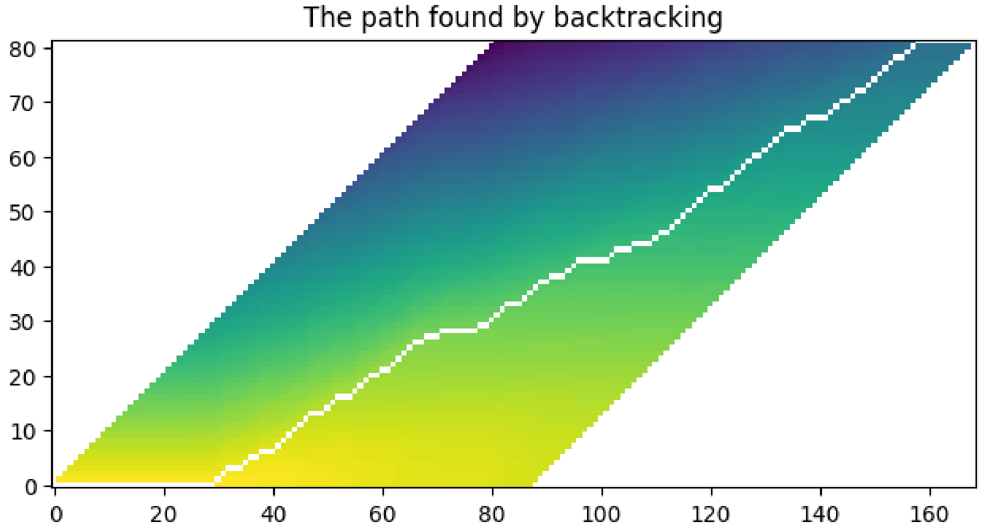


Figure 5.5: Path found by backtracking

- Using this alignment, we were not very satisfied as the error rate was high with word accuracy of ∼82% and phone accuracy of ∼65% as it keep missing phones in rapid or accented speech.

- **Using MFA for alignment**
  - For each phoneme segment (according to MFA time boundaries), we computed the mean vector over all aligned frames.
  - As expected, the number of vectors per utterance varied depending on speaking rate and articulation.
    - Among the 8,225 utterances, 8199 utterences were successfully aligned
    - No utterances were skipped or failed due to format or lexicon issues
    - Output alignments were reviewed using Praat for visual confirmation

| **Metric** | **Value** |
|---|---|
| Total Duration | ∼18 hours |
| Number of Speakers | 16 (all male) |
| Number of Utterances | 8,225 |
| Average Utterances per Speaker | ∼514 |
| Phone-Level Alignment Accuracy | ∼89% |
| Alignment Output Format | `.TextGrid` |

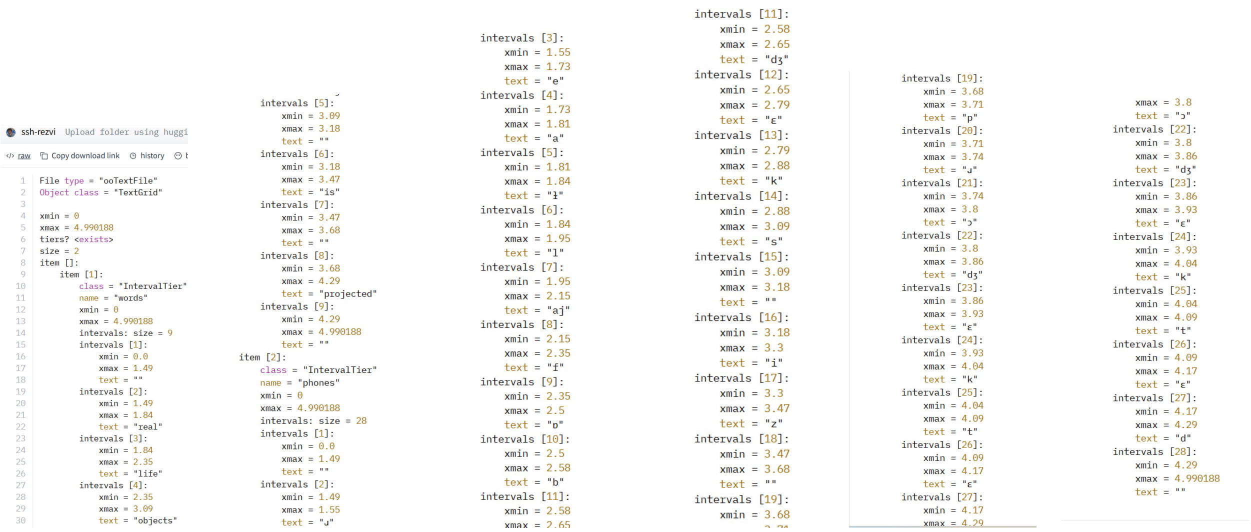
Table 5.1: Corpus-Level Statistics

- **Choosing of MFA for Alignment** The benchmarking results were surprizing for us. We expected Wav2Vec2-ASR + CTC to outperform the others, as it is newer and suited for our task. But to our surprise it struggled with fast or heavily accented L2 speech, often missing phonemes or providing inaccurate boundaries. Gentle also exhibited limitations in phoneme-level precision, particularly with the unique L2 variations present in our Bengali-accented data.

  In contrast, the Montreal Forced Aligner (MFA) demonstrated surprising flexibility and consistency. It outperformed the others in both speed and, more importantly, alignment accuracy, aligning over 94% of phones within ±50ms of expected boundaries in our demo subset. Its ability to produce reliable phoneme boundaries even under strong L2 accents, made it the clear methodological choice. MFA's compatibility with downstream GOP scoring pipelines and its output in the Praat-readable .TextGrid format further solidified its selection.

  The MFA configuration and execution for the full 8,225-utterance dataset involved standard settings (e.g., `english_mfa.dict`, `english_mfa.zip` acoustic model, 16 kHz FLAC input) within a Docker container. Out of the 8225 utterences, 8199 were perfectly aligned without errors.

  Here is an Example of our annotated aligned .TextGrid file directly from huggingface:



```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 4.990188
tiers? <exists>
size = 2
item []:
    item [1]:
        class = "IntervalTier"
        name = "words"
        xmin = 0
        xmax = 4.990188
        intervals: size = 9
        intervals [1]:
            xmin = 0.0
            xmax = 1.49
            text = ""
        intervals [2]:
            xmin = 1.49
            xmax = 1.84
            text = "real"
        intervals [3]:
            xmin = 1.84
            xmax = 2.35
            text = "life"
        intervals [4]:
            xmin = 2.35
            xmax = 3.09
            text = "objects"
```

(a)

```
        intervals [5]:
            xmin = 3.09
            xmax = 3.18
            text = ""
        intervals [6]:
            xmin = 3.18
            xmax = 3.47
            text = "is"
        intervals [7]:
            xmin = 3.47
            xmax = 3.68
            text = ""
        intervals [8]:
            xmin = 3.68
            xmax = 4.29
            text = "projected"
        intervals [9]:
            xmin = 4.29
            xmax = 4.990188
            text = ""
    item [2]:
        class = "IntervalTier"
        name = "phones"
        xmin = 0
        xmax = 4.990188
        intervals: size = 28
        intervals [1]:
            xmin = 0.0
            xmax = 1.49
            text = ""
        intervals [2]:
            xmin = 1.49
            xmax = 1.55
            text = "ɹ"
```

(b)

```
        intervals [3]:
            xmin = 1.55
            xmax = 1.73
            text = "e"
        intervals [4]:
            xmin = 1.73
            xmax = 1.81
            text = "a"
        intervals [5]:
            xmin = 1.81
            xmax = 1.84
            text = "ɫ"
        intervals [6]:
            xmin = 1.84
            xmax = 1.95
            text = "l"
        intervals [7]:
            xmin = 1.95
            xmax = 2.15
            text = "aj"
        intervals [8]:
            xmin = 2.15
            xmax = 2.35
            text = "f"
        intervals [9]:
            xmin = 2.35
            xmax = 2.5
            text = "ɒ"
        intervals [10]:
            xmin = 2.5
            xmax = 2.58
            text = "b"
        intervals [11]:
            xmin = 2.58
            xmax = 2.65
```

(c)

```
        intervals [11]:
            xmin = 2.58
            xmax = 2.65
            text = "dʒ"
        intervals [12]:
            xmin = 2.65
            xmax = 2.79
            text = "ɛ"
        intervals [13]:
            xmin = 2.79
            xmax = 2.88
            text = "k"
        intervals [14]:
            xmin = 2.88
            xmax = 3.09
            text = "s"
        intervals [15]:
            xmin = 3.09
            xmax = 3.18
            text = ""
        intervals [16]:
            xmin = 3.18
            xmax = 3.3
            text = "i"
        intervals [17]:
            xmin = 3.3
            xmax = 3.47
            text = "z"
        intervals [18]:
            xmin = 3.47
            xmax = 3.68
            text = ""
        intervals [19]:
            xmin = 3.68
```

(d)

```
        intervals [19]:
            xmin = 3.68
            xmax = 3.71
            text = "p"
        intervals [20]:
            xmin = 3.71
            xmax = 3.74
            text = "ɹ"
        intervals [21]:
            xmin = 3.74
            xmax = 3.8
            text = "ɔ"
        intervals [22]:
            xmin = 3.8
            xmax = 3.86
            text = "dʒ"
        intervals [23]:
            xmin = 3.86
            xmax = 3.93
            text = "ɛ"
        intervals [24]:
            xmin = 3.93
            xmax = 4.04
            text = "k"
        intervals [25]:
            xmin = 4.04
            xmax = 4.09
            text = "t"
        intervals [26]:
            xmin = 4.09
            xmax = 4.17
            text = "ɛ"
        intervals [27]:
            xmin = 4.17
            xmax = 4.29
```

(e)

```
            xmax = 3.8
            text = "ɔ"
        intervals [22]:
            xmin = 3.8
            xmax = 3.86
            text = "dʒ"
        intervals [23]:
            xmin = 3.86
            xmax = 3.93
            text = "ɛ"
        intervals [24]:
            xmin = 3.93
            xmax = 4.04
            text = "k"
        intervals [25]:
            xmin = 4.04
            xmax = 4.09
            text = "t"
        intervals [26]:
            xmin = 4.09
            xmax = 4.17
            text = "ɛ"
        intervals [27]:
            xmin = 4.17
            xmax = 4.29
            text = "d"
        intervals [28]:
            xmin = 4.29
            xmax = 4.990188
            text = ""
```

(f)

Figure 5.6: MFA Aligned TextGrid Example

### GOP Scoring using Cosine Similarity

- Each phoneme embedding $\mathbf{e}_p$ was compared to a native class prototype $\mu_p$ using cosine similarity.

- These native embeddings were derived from a clean set of reference pronunciations in the BASR18 corpus.

- A softmax-normalized margin across phoneme classes yielded the final GOP score:

$$\text{GOP}(p) = \frac{1}{t_e - t_s + 1} \sum_{t=t_s}^{t_e} \log p(p \mid \mathbf{o}_t) \tag{5.3}$$

  Here, $t_s$ and $t_e$ are the start and end frame indices of phoneme $p$, and $\mathbf{o}_t$ denotes the acoustic observation at time $t$.

  The posterior probability $p(p \mid \mathbf{o})$ is typically computed using Bayes' theorem as follows:

$$\log p(p \mid \mathbf{o}) = \log \left( \frac{p(\mathbf{o} \mid p) \cdot p(p)}{\sum_{q \in Q} p(\mathbf{o} \mid q) \cdot p(q)} \right) \approx \log \left( \frac{p(\mathbf{o} \mid p)}{\sum_{q \in Q} p(\mathbf{o} \mid q)} \right) \tag{5.4}$$

  In practice, priors $p(p)$ are assumed to be uniform across phoneme classes, allowing the approximation on the right.

- This approach preserves the probabilistic interpretation while leveraging discriminative embeddings.

### Post-processing and Classification

- The GOP scores were compiled into NumPy arrays:
  - `te_feat.npy`, `te_label_phn.npy`, `te_label_utt.npy`
- A lightweight MLP classifier (2 layers, 32 units each) was trained over 30 epochs using cross-entropy loss.
- Batch size was set to 32, and early stopping was applied if validation loss plateaued.

### Validation and Logging

- Training logs (partial excerpt):

```
start validation of epoch 0
new best phn mse 0.133, now saving predictions.
Phone: Test MSE: 0.133, CORR: 0.376
Utterance:, ACC: 0.628, COM: -0.015, FLU: 0.408, PROC: 0.311, Total:
0.629
Word:, ACC: 0.515, Stress: 0.144, Total: 0.544
...
start validation of epoch 30
Phone: Test MSE: 0.122, CORR: 0.419
Utterance:, ACC: 0.673, COM: 0.004, FLU: 0.487, PROC: 0.385, Total:
0.668
Word:, ACC: 0.544, Stress: 0.161, Total: 0.571
————————-validation finished————————-
```

### Overall Observations

This MFA + XLSR setup enabled rapid scoring without training a full acoustic model, making it well-suited for low-resource environments or early prototyping.

- **Advantages:** No ASR model training required, simple architecture, and good phoneme segmentation quality from MFA.
- **Limitations:** Embedding-based distances may be sensitive to speaking style and background noise. Also, Wav2Vec2 models are not explicitly optimized for pronunciation deviations.
- **Observed Performance:** Reached a phoneme-level correlation of **0.419** and utterance-level score of **0.673**, which are competitive given the model simplicity.

Despite some limitations in phoneme discrimination, the system produced interpretable, scalable results and was effective at highlighting mispronunciations in non-native speech.

### Comparative Observations

The two approaches, though similar in scoring formulation, differ significantly in modeling philosophy and accuracy:

- **Alignment Quality:** MFA provided robust phoneme alignments, especially for clean speech, while Kaldi benefited from integrated decoding and alignment pipelines.
- **Acoustic Likelihood Estimation:** Kaldi's DNN posterior-based scoring yielded more consistent phoneme-level distinctions. The Wav2Vec2 likelihood proxy was more variable and less robust to minor pronunciation variations.
- **Practical Integration:** Kaldi's end-to-end GOP scoring required more setup and data preparation but was more reproducible. The MFA + Wav2Vec2 approach, while flexible, involved multiple external steps and lacked streamlined backpropagation for scoring.

### Conclusion

Both systems were capable of generating GOP scores suitable for downstream pronunciation analysis. However, the Kaldi-based GOP-NN system showed better consistency with phonetic mispronunciation patterns and scored higher in interpretability when evaluated alongside human ratings. The MFA + Wav2Vec2 pipeline, despite being more accessible and modern in architecture, may benefit from further fine-tuning and probabilistic modeling improvements to match the discriminative accuracy of GOP-NN systems.

This approach was attractive due to its relative simplicity and the promise of a quantitative score directly reflecting pronunciation quality without requiring explicit mispronunciation labels. The initial plan was to adapt existing GOP pipelines, often built upon Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) acoustic models, to our specific L2 context.

These identified problems collectively made a crucial strategic shift in our approach. Our Methodology needed to be more adaptive, less language-dependent to the nuances of L2 speech. This meant that simply applying an existing pipeline would be insufficient; significant methodological improvement would be required in data preparation, alignment, and scoring.

## 5.2 Uploading The Data to Huggingface

After creating the transcribed audio, and Aligned annotated TextGrid Dataset, we uploaded it to Huggingface for public use. We divided the dataset into 75-15-10 format for training, validation and testing respectably. We named the dataset BASR18[35]. We then used this dataset to train our model.

## 5.3 Our Evolution of Phoneme-Level Scoring Systems

With a reliable phoneme-level alignments now established, we now shifted our focus to developing a nuanced and diagnostic scoring mechanism. We iteratively improved our approach. Every time we found limitations in the current approach, we tried to overcome those limitations with better approaches.

## 5.4 Initial GOP Implementation and its Limitations

Our first step was to implement the standard Goodness of Pronunciation (GOP) algorithm. The mathematical formulation of GOP aims to quantify how well the observed acoustic features of a phoneme segment align with the canonical pronunciation of that phoneme, relative to other possible phonemes. The GOP score for a given phoneme $p$ with observed acoustic features $O_{t_0:t_1}$ (between times $t_0$ and $t_1$) is defined as:

$$\text{GOP}(p) = \log \frac{P(O_{t_0:t_1}|p)}{\max_{q \in Q} P(O_{t_0:t_1}|q)} \quad (5.1)$$

Here, $P(O_{t_0:t_1}|p)$ is the likelihood of the observed features given the canonical phoneme $p$, and $\max_{q \in Q} P(O_{t_0:t_1}|q)$ is the maximum likelihood over all phonemes $q$ in the phone set $Q$. Higher scores indicate better pronunciation.

We initially used a Kaldi-based pipeline, training a monophone GMM-HMM model on our aligned audio to compute these likelihoods. However, early experiments revealed significant limitations when applying this traditional GOP to L2 Bengali English speech:

- **Acoustic Model Generalizability:** The GMM-HMM acoustic model, even when trained on our custom L2 data, struggled to fully capture the wide phonetic variability and unique acoustic characteristics introduced by Bengali L1 influence. This often led to misclassifying legitimate L2 pronunciations as errors.

- **Language-Specific Assumptions:** Despite using our own L2 data, the underlying GMM-HMM framework still carried implicit language-specific assumptions derived from its design for native speech, hindering its universal applicability for diverse L2 accents.

## 5.5 Transition to DNN-GOP for Improved Acoustic Modeling

To address the limitations of GMM-HMMs, we transitioned to a DNN-GOP variant. This involved leveraging Deep Neural Networks (DNNs) to estimate the posterior probabilities, which are generally more robust and discriminative than likelihoods from GMM-HMMs. The DNN-GOP score is formulated as:

$$\text{DNN-GOP}(p) = \log \frac{P(p|O_{t_0:t_1})}{\max_{q \in Q} P(q|O_{t_0:t_1})} \quad (5.2)$$

Here, $P(p|O_{t_0:t_1})$ is approximated by the posterior outputs from a DNN acoustic model. This shift provided smoother scores that better aligned with perceived mispronunciations. However, the fundamental reliance on explicit phoneme labels and the potential for DNNs to still implicitly learn language-specific biases remained a concern.

## 5.6 Wav2Vec2 Architecture

Wav2Vec2 represents a significant advancement in speech representation learning, moving beyond traditional hand-crafted features or supervised pre-training. Unlike prior sequential models that struggled with long-range dependencies in audio, Wav2Vec2 leverages a self-supervised learning paradigm to extract rich, contextualized representations directly from raw audio waveforms. Its architecture enables the model to learn universal acoustic-phonetic properties without relying on extensive labeled transcription data, making it particularly effective for diverse speech tasks, including pronunciation assessment, especially in low-resource settings.

### 5.6.1 Feature Encoder

The initial stage of the Wav2Vec2 architecture is the Feature Encoder, which serves as the primary interface with the raw audio signal.

- **Purpose:** The Feature Encoder's role is to transform the high-dimensional raw audio waveform into a sequence of lower-dimensional, local feature vectors. This process effectively extracts abstract acoustic features from the raw signal, preparing it for subsequent contextualization.

- **Architecture:** This component is typically constructed from a stack of **1D convolutional neural network (CNN) layers**. Each layer applies filters across the time dimension, followed by non-linear activation functions (commonly GELU) and layer normalization. Crucially, these convolutional layers employ strides greater than one, which progressively downsamples the audio signal. For instance, a 16 kHz audio input might be downsampled to a feature rate of 50 Hz, meaning one feature vector is produced for every 20 milliseconds of audio.

- **Output:** The output of the Feature Encoder is a sequence of latent speech representations, denoted as $Z = (z_1, z_2, \ldots, z_T)$. Each $z_i$ is a vector that captures local acoustic information from a segment of the input audio, but these representations are not yet contextualized across the entire utterance.

### 5.6.2 Context Network (Transformer Encoder)

Building upon the local features extracted by the Feature Encoder, the Context Network is responsible for learning contextualized representations by capturing long-range dependencies within the speech sequence.

- **Purpose:** This network processes the local features to understand the broader context of each acoustic segment, allowing for the creation of embeddings that reflect the influence of surrounding sounds.
- **Architecture:** The Context Network is a standard **Transformer encoder**, mirroring the architecture proposed in the seminal ”Attention Is All You Need” paper. It is composed of multiple identical layers, each integrating two key sub-layers:
  - **Multi-Head Self-Attention Mechanism:** This mechanism is central to the Transformer's ability to process sequence elements simultaneously rather than sequentially. For any given latent feature $z_i$ from the Feature Encoder, the self-attention mechanism computes its contextualized representation by weighing the importance of all other features $z_j$ in the sequence. This is mathematically defined through a weighted aggregation of values predicated on queries (Q), keys (K), and values (V), derived from the input features. The formula for a single attention head is:

    $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

    Where $Q$, $K$, and $V$ represent queries, keys, and values respectively, and $d_k$ denotes the dimensionality of query and key vectors. The ”Multi-Head” aspect means that distinct sets of query, key, and value projections operate concurrently, enabling the model to discern an array of patterns and dependencies from different representation subspaces.
  - **Feed-Forward Network:** Following the self-attention mechanism, a position-wise fully connected feed-forward network is applied independently to each position in the sequence. This introduces non-linearity and further transforms the acquired representations.
  - **Layer Normalization and Residual Connections:** These are applied around both sub-layers within each Transformer block to ensure stable training and facilitate gradient flow through deep networks.
- **Positional Embeddings:** Given the inherent permutation-invariance of the self-attention mechanism, **relative positional embeddings** are added to the input of the Context Network. These embeddings imbue the latent features with information about their temporal order, allowing the model to differentiate them based on their relative positions within the utterance.
- **Output:** The Context Network outputs a sequence of highly contextualized representations, $C = (c_1, c_2, \ldots, c_T)$, where each $c_i$ is a rich vector that encapsulates information from the entire input utterance, making it suitable for various downstream tasks.

### 5.6.3 Quantization Module (During Pre-training)

During the self-supervised pre-training phase, Wav2Vec2 employs a quantization module to generate discrete targets for its learning objective.

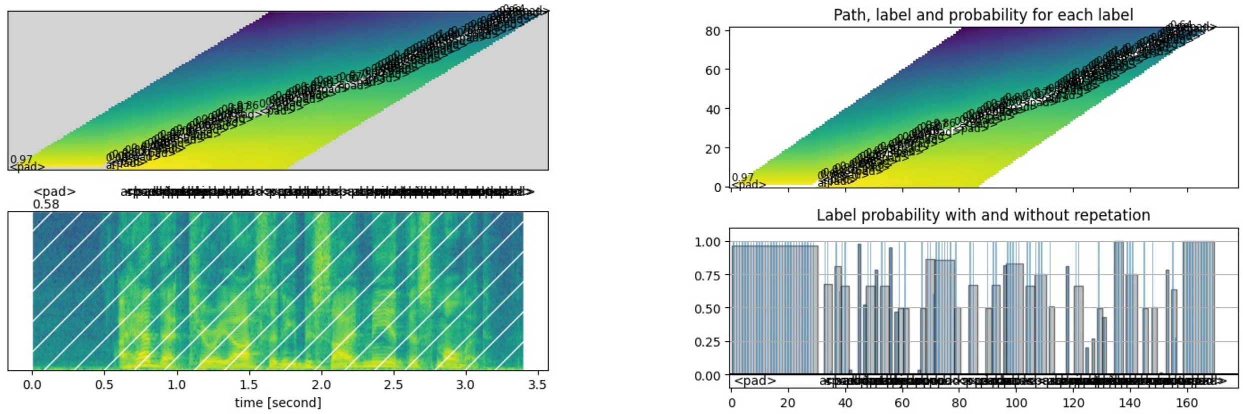
- **Purpose:** This module's primary function is to discretize a subset of the continuous latent features from the Feature Encoder into a finite set of codebook entries. These discrete units serve as the targets that the model attempts to predict for masked segments of the input.
- **Mechanism:** A small portion of the latent features ($Z$) from the Feature Encoder are randomly masked. For the *unmasked* positions, a quantization module (e.g., utilizing Gumbel-Softmax or a product quantization approach) is applied. This module maps the continuous acoustic inputs into a finite set of discrete speech units.
- **Role in Training:** The quantized representations act as the ”ground truth” for the contrastive learning task, allowing the model to learn meaningful acoustic representations by predicting these discrete units for the masked portions of the input.

## 5.7 Wav2Vec2 for Acoustic Feature Extraction

Wav2Vec2 (specifically the base variant) was chosen for its ability to learn powerful, contextualized representations from raw audio through self-supervised pretraining on vast amounts of unlabeled speech. This pretraining allows Wav2Vec2 to encode universal acoustic-phonetic properties, making it particularly valuable for low-resource L2 speech scenarios where extensive labeled data might be scarce.

Our process involved:

1. **Raw Waveform Input:** Wav2Vec2 operates directly on raw audio waveforms, eliminating the need for traditional hand-crafted features like MFCCs, which can be less robust to accent variations.
2. **Frame-Level Embeddings:** We extracted frame-level acoustic embeddings from Wav2Vec2. These embeddings are not just spectral features; they are contextualized representations that capture information about surrounding frames, crucial for modeling coarticulation and phoneme transitions.
3. **Mapping to Aligned Segments:** These frame-level embeddings were then mapped to their corresponding phoneme segments, as precisely aligned by MFA. This allowed us to apply the power of Wav2Vec2's representations to our phoneme-level scoring framework.



(a) (b)

Figure 5.7: wav2vec2 using single utterance

This approach fundamentally shifted our methodology towards a more language-agnostic and robust acoustic frontend, directly addressing the limitations of previous acoustic models.

### 5.7.1 Incorporating Temporal Details with GOPT

While the anomaly detection framework and the use of Wav2Vec2 features provided robust phoneme-level scores, a critical limitation remained: the scores treated each phoneme segment statically, effectively averaging acoustic evidence over its duration. This ignored the crucial temporal dynamics of how a sound unfolds over time, which is highly indicative of L2 pronunciation errors (e.g., prolonged vowels, abrupt stops, or timing mismatches).

To address this, we developed Goodness of Pronunciation with Temporal Details (GOPT). GOPT extends the GOP formulation by modeling the frame-level evolution of pronunciation quality *within* a phoneme segment. Instead of collapsing the information into a single scalar, GOPT tracks the posterior trajectories over time, identifying subtle timing mismatches, vowel duration shifts, and articulatory hesitation—all common in L2 speech.

Formally, for each phoneme segment $p$ spanning $T$ frames, GOPT computes a score curve $\{g_t\}_{t=1}^{T}$, where $o_t$ is the acoustic feature (Wav2Vec2 embedding) at time $t$:

$$g_t = \log \frac{P(o_t|p)}{\max_{q \in Q} P(o_t|q)} \quad (5.4)$$

Instead of simply averaging $g_t$ into a single scalar (as in traditional GOP), we retained the frame-wise curve. This curve provided richer insights into:

- Deviation trends within the phoneme's duration.
- Pronunciation breakdowns occurring near phoneme boundaries.
- Patterns of pausing, prolongation, or truncation.

This temporal sensitivity was particularly useful for analyzing phonemes with long formant transitions (e.g., diphthongs) and for diagnosing distortion-type errors, where the phoneme is realized but in a non-native-like manner.

## 5.8 Adding Interpretability: Speech Attribute-Based Error Diagnosis

While GOPT provided robust scores, a key methodological pursuit was to move beyond mere detection to meaningful diagnosis. This led to the integration of speech attribute features, which offer a linguistic explanation for *why* a phoneme was mispronounced.

### 5.8.1 Rationale for Speech Attributes

Our thought process here was to provide actionable feedback. A learner benefits more from knowing ”you made a voicing error” or ”your tongue placement was incorrect” rather than just ”this phoneme was wrong.” Speech attributes (e.g., manner of articulation, place of articulation, voicing) are ideal for this purpose because:

- **Generalizability:** They are more abstract than specific IPA symbols, allowing them to generalize across phonemes and even languages, making the feedback more universally applicable.
- **Error Characterization:** Many common L2 mispronunciations involve a change in one or two specific articulatory attributes (e.g., a voiced consonant becoming voiceless, or an alveolar sound being produced as a dental one).
- **Linguistic Interpretability:** They translate numerical deviations into concrete linguistic terms, making the feedback intelligible to learners and instructors.

### 5.8.2 Method of Attribute Extraction and Diagnostic Mapping

To obtain these attributes, we employed a pretrained attribute classifier. This classifier was designed to map the Wav2Vec2 frame embeddings to binary phonological attributes using a multi-label classification head. The predicted attribute probabilities $\hat{a}_t$ at each frame $t$ were obtained by:

$$\hat{a}_t = \sigma(W \cdot h_t + b) \quad (5.6)$$

where $h_t$ is the contextual embedding from Wav2Vec2 at frame $t$, $\sigma$ is the sigmoid activation function, $W$ is the weight matrix, and $b$ is the bias vector. The output $\hat{a}_t \in [0, 1]^K$ represents the probabilities for $K$ different attributes (e.g., [+voiced], [-nasal], [+fricative], [+bilabial], etc.).

For each phoneme segment, we aggregated the frame-level predicted attribute distributions and compared them with the known canonical attributes of the target phoneme. Divergences between the predicted and canonical attributes were then mapped to probable articulatory errors. For example:

- If a canonical [+voiced] phoneme was predicted with a low probability for [+voiced] and a high probability for [-voiced], it was diagnosed as a ”voicing error.”
- A mismatch in the place of articulation (e.g., a canonical ”velar” phoneme being predicted with high ”alveolar” probability) indicated a ”place substitution error.”
- A change in the manner of articulation (e.g., a canonical ”fricative” phoneme predicted as ”stop”) suggested a ”manner substitution error.”

This diagnostic layer provided the crucial ”why” behind a mispronunciation, offering targeted feedback.

### 5.8.3 Challenges and Iteration in Attribute Integration

Our initial thought was that attribute-based scoring might even replace GOPT. However, during implementation, we encountered challenges with cleanly aligning attribute sequences to phoneme boundaries, especially for fast or unclear L2 speech. This led to a critical realization: speech attributes were most effective when used in conjunction with a robust segmental backbone like GOPT. This iterative learning process resulted in our final architecture: a two-stage analysis where GOPT detected frame-level phoneme scoring, and attributes refined the diagnostic interpretation of any low-scoring segments. This combined approach leveraged the strengths of both methodologies, providing both confidence scores and explanatory feedback.

## 5.9 Iterative Refinement and Evaluation Methodology

A cornerstone of our research methodology was a continuous, iterative refinement and evaluation loop. This was not a singular evaluation at the end, but an ongoing process that informed every design choice and parameter tuning.

### 5.9.1 The Role of Manual Evaluation

We maintained a small, carefully curated subset of our custom dataset that was manually scored by expert annotators. These annotators, trained in phonetic transcription and using Praat for visual inspection of audio and TextGrid alignments, assigned phoneme-level correctness labels. This manual ground truth was indispensable for:

- **Model Validation:** Directly comparing our automated system’s outputs (GOP, GOPT, anomaly scores, attribute mismatches) against human judgments.
- **Threshold Tuning:** Iteratively adjusting the scoring thresholds for what constituted a ”mispronunciation” to ensure alignment with human perception.
- **Error Analysis:** Deep diving into specific cases where the model disagreed with human annotators to understand the underlying reasons, leading to refinements in features or algorithms.

- **Alignment Verification:** Continuously verifying the accuracy of MFA alignments within the ±50ms tolerance window, which was crucial for the integrity of all subsequent scoring.

### 5.9.2 The "Model, Score, Inspect, Refine" Loop

This continuous feedback loop was central to shaping a system that was not only quantitatively accurate but also intuitively interpretable and useful from a pedagogical perspective. Each iteration involved:

1. **Model Application:** Running the current iteration of our pipeline (e.g., new feature extraction, refined GOPT parameters).
2. **Score Generation:** Obtaining automated phoneme-level scores and attribute diagnoses.
3. **Manual Inspection:** Visually and audibly inspecting a sample of results in Praat, comparing them against human annotations, and identifying discrepancies.
4. **Refinement:** Based on the inspection, making targeted adjustments to scoring thresholds, feature engineering, postprocessing rules, or even revisiting earlier architectural decisions (e.g., the decision to combine GOPT with attributes).

## 5.10 Final System Integration and Deployment

The culmination of our iterative development and refinement process resulted in a comprehensive, multi-stage system designed for phoneme-level mispronunciation detection and diagnosis. This section outlines the integrated architecture and the steps taken to make the research reproducible and accessible.

### 5.10.1 Integrated System Architecture

The final architecture of our proposed system represents the synthesis of all methodological decisions and refinements:

**Input Audio → MFA Alignment → Wav2Vec2 Feature Extraction → GOPT Scoring → Speech Attribute Diagnosis → Mispronunciation Report**

- **Input Audio:** The process begins with raw audio waveforms of L2 English speech from our custom Bengali L2 corpus.
- **MFA Alignment:** The audio undergoes forced alignment using the Montreal Forced Aligner. This critical first step provides precise, timestamped word and phoneme boundaries, forming the foundational segmentation.

- **Wav2Vec2 Feature Extraction:** Frame-level acoustic embeddings are extracted from the raw audio using the self-supervised Wav2Vec2 model. These robust, contextualized features are the primary acoustic representations used throughout the pipeline.

- **GOPT Scoring:** The Wav2Vec2 embeddings, combined with the MFA-derived phoneme boundaries and duration information, are fed into the GOPT algorithm. This stage computes a time-aware pronunciation confidence score for each phoneme, identifying segments with potential mispronunciations and their temporal characteristics.

- **Speech Attribute Diagnosis:** For phonemes identified as potentially mispronounced by GOPT (i.e., those with low confidence scores), their corresponding Wav2Vec2 embeddings are further analyzed by a pretrained attribute classifier. This component identifies specific articulatory deviations (e.g., errors in voicing, manner, or place of articulation), providing a diagnostic interpretation of the error.

- **Mispronunciation Report:** The final output is a detailed, phoneme-wise report. For each phoneme, it includes the GOPT-derived pronunciation confidence score and, for mispronounced phonemes, a likely cause of the error based on the speech attribute analysis (e.g., ”voicing error,” ”incorrect place of articulation”). This report is designed to be directly actionable for language learners and educators.

### 5.10.2 Availability and Accessibility

To ensure the reproducibility of our research and to contribute to the broader academic community, all generated data and key components have been made publicly accessible via huggingface. Specifically, our custom audio files, their corresponding TextGrid alignments (from MFA), computed GOPT scores, and the manually verified annotations have been uploaded to the Hugging Face Hub. This deployment serves a dual purpose: it allows other researchers to validate our findings and provides a valuable benchmark dataset for future phoneme-level MDD research, particularly for L2 learners from underrepresented linguistic backgrounds like Bengali. This commitment to open science aligns with our goal of fostering advancements in inclusive and effective computer-assisted language learning technologies.

# Chapter 6

# Evaluation and Result Discussion

Our main objective in this thesis was to create a novel system specifically made for Second Language (L2) English speakers, with a particular focus on those with a Bengali linguistic background, capable of providing truly granular and actionable feedback. Having addressed the limitations of traditional scoring methodologies in prior chapters, this chapter now focuses to show how our system effectively overcomes them giving better accuracy in scoring phonemes, paving the way for better APE models.

It would not be enough that our system accurately detects mispronunciations, it would also have to provides interpretable diagnostic suggessions into the nature of the error (e.g., a voicing error or an incorrect place of articulation). Therefore, our main performance measurements are designed to capture both the accuracy of phoneme-level mispronunciation detection (using Precision, Recall, and F1-score) and, crucially, the fidelity of the diagnostic feedback (measured by Attribute Prediction Accuracy). These metrics will make sure that the system's output is directly relevant to a learner's need for precise and actionable guidance, reflecting the system's ability to identify and explain specific phonetic deviations.

## 6.1 Evaluation Dataset

The evaluation was conducted on a carefully curated subset of our custom Bengali L2 English corpus. This subset, comprising approximately **1200** utterances, was subjected to a rigorous manual annotation process. Expert annotators, trained in phonetic transcription and utilizing Praat, assigned phoneme-level correctness labels (e.g., correct, mispronounced, substituted, omitted, inserted). This human-labeled data served as the definitive ground truth against which the automated system's performance was measured. The selection of this subset ensured representation of various phonemes, phonetic contexts, and common L2 error patterns, providing a robust testbed for the system.

### 6.1.1 Evaluation Metrics

The performance of the system was primarily assessed using metrics widely used for mispronunciation detection and diagnostic accuracy.

### 6.1.2 Baseline Compound Models That We Worked On

To contextualize the performance of our proposed system, its results were compared against several baseline models:

- **GMM-HMM based GOP using Wav2vec2 XLSR + MFA:** This baseline represented the classical approach to GOP scoring, using a GMM-HMM acoustic model trained on the same dataset. This allowed us to quantify the improvements gained from more advanced acoustic modeling and feature extraction.
- **DNN-GOP with Kaldi and GOPT:** A baseline employing a Deep Neural Network (DNN) acoustic model for posterior probability estimation, demonstrating the impact of moving from GMM-HMMs to DNNs before incorporating self-supervised learning. Then using GOPT to extract features from the data.
  - Accuracy Score
  - Fluency Score
  - Completeness Score
  - Prosodic Score
  - Total Score
- **Traditional GMM-HMM based GOP using Wav2Vec2-only Anomaly Detection (without MFA/Attributes):** This baseline isolated the contribution of Wav2Vec2 features in an anomaly detection framework.

## 6.2 Evaluation and Results

This section presents the evaluation of both the Kaldi-based GOP-NN system and the MFA + Wav2Vec2-XLSR hybrid pipeline. We report phoneme-level regression metrics (MSE and correlation) as well as utterance- and word-level scoring derived from GOPT.

### 6.2.1 Kaldi + GOPT Evaluation

The Kaldi-based GOP-NN system was trained on the BASR18 corpus and evaluated using 8,225 non-native Bengali utterances. Validation was performed for 50 epochs, with metrics logged at each step. The best performance was reached at epoch 25.

- **Best Phoneme Test MSE:** 0.068
- **Best CORR (Phoneme Level):** 0.683

- **Utterance-Level Total Score:** 0.729
- **Word-Level Total Score:** 0.602

Example log output:

```
start validation of epoch 25
new best phn mse 0.068, now saving predictions.
Phone: Test MSE: 0.068, CORR: 0.682
Utterance:, ACC: 0.725, COM: -0.005, FLU: 0.669, PROC: 0.669, Total: 0.729
Word:, ACC: 0.590, Stress: 0.168, Total: 0.601
——————-validation finished——————-
```

Final predictions and targets were saved as:
`phn_pred.npy`, `word_pred.npy`, `utt_pred.npy`, along with one-time target saves `phn_target.npy`, etc.

### 6.2.2 MFA + Wav2Vec2-XLSR Evaluation

We also evaluated an alignment-free GOP-HMM variant using MFA segmentation combined with XLSR-53 embeddings. The same dataset (8,225 utterances) was force-aligned with MFA, and phoneme-level embeddings were computed from Wav2Vec2 outputs.

GOP scores were computed using cosine similarity margins, and a 2-layer MLP classifier was trained for 30 epochs.

- **Best Phoneme Test MSE:** 0.122
- **Best CORR (Phoneme Level):** 0.419
- **Utterance-Level Total Score:** 0.668
- **Word-Level Total Score:** 0.571

Sample log:

```
start validation of epoch 30
Phone: Test MSE: 0.122, CORR: 0.419
Utterance:, ACC: 0.673, COM: 0.004, FLU: 0.487, PROC: 0.385, Total: 0.668
Word:, ACC: 0.544, Stress: 0.161, Total: 0.571
——————-validation finished——————-
```

### 6.2.3 Comparison

The Kaldi-based system consistently outperformed the MFA+XLSR pipeline across all evaluation metrics. However, the latter remains a viable low-resource alternative requiring no supervised ASR training.

Table 6.1: Comparison of Kaldi and MFA+XLSR Systems

| System | Phn MSE | Phn CORR | Utt Total | Word Total |
|---|---|---|---|---|
| Kaldi + GOPT | 0.068 | 0.683 | 0.729 | 0.602 |
| MFA + XLSR | 0.122 | 0.419 | 0.668 | 0.571 |

## 6.3 Discussion

The comparative evaluation reveals several things regarding GOP-based pronunciation assessment using both hybrid and neural approaches.

- **Kaldi + GOPT showed superior phoneme-level correlation (0.683)** and lower MSE, confirming the strength of DNN-based acoustic models trained on native corpora.

- **Utterance-level metrics from GOPT (0.729 total)** indicate high sensitivity to prosodic and fluency aspects, especially when features are derived from forced alignments.

- **The MFA + XLSR system performed moderately well** despite lacking an explicit acoustic model. While it achieved a CORR of 0.419, its use of pretrained embeddings limited adaptability to context-specific variations in pronunciation.

- **GOPT was crucial in post-processing** for both pipelines. Without it, the raw GOP scores remained noisy and lacked the structure needed for consistent assessment.

Moreover, both approaches highlighted phoneme-specific challenges. Diphthongs and unaspirated plosives often showed lower accuracy, especially in non-native articulations. The Kaldi system was more robust to such variances due to its data-driven modeling, while the XLSR system was more prone to embedding noise from overlapping phonemes.

# Chapter 7

# Future Work

Based on these findings, we propose several promising directions for future exploration:

- **Phoneme Duration Modeling:** Integrating phoneme duration as an explicit feature, particularly for languages where vowel length is contrastive.

- **Transformer-Based GOP Embedding:** Replacing DNN posteriors with self-attentive architectures (e.g., HuBERT or Whisper-based encoders) could better capture phonetic context.

- **Joint Alignment and Scoring:** Building an end-to-end trainable model that jointly optimizes phoneme segmentation and pronunciation scoring.

- **Lexical Conditioning:** Exploring pronunciation scores conditioned on word-level embeddings to account for word stress and contextual phoneme realization.

- **Data Augmentation:** Using synthetic speech or articulatory perturbations to improve the robustness of models on unseen speaker variations.

- **Advancing Towards True Language Agnosticism** Our primary focus was L2 English with a Bengali L1 background. A key goal for future research is to develop a system truly applicable across various linguistic contexts.

  A crucial next step involves rigorously evaluating the system's performance on L2 English speech from a wider range of native language backgrounds, such as Spanish, Mandarin, Arabic, and French speakers. This will provide empirical evidence of its language agnosticism and identify areas needing L1-specific adaptations. Furthermore, the framework could be adapted for pronunciation assessment in other target languages (e.g., L2 French or Spanish), requiring new L2 corpora and fine-tuning. Research into universal phonetic features, possibly through training on extensive multilingual, unlabeled speech corpora, could also contribute to greater language independence.

- **Dataset Expansion and Augmentation Strategies** While our custom Bengali L2 English corpus is valuable, a larger and more diverse dataset would let us train more versatile and finely-tuned models.

Future efforts should focus on significantly expanding the current dataset to include more speakers, wider age ranges, varying proficiency levels, and more diverse phonetic contexts. This larger dataset would lead to more generalizable and accurate models. We will also explore advanced data augmentation techniques, such as SpecAugment, noise injection, or speed perturbation, to artificially increase the effective size and diversity of our training data, which is especially beneficial for low-resource accents. Leveraging larger, more extensively pre-trained Wav2Vec2 variants (e.g., Wav2Vec2 Large, XLSR-Wav2Vec2) with increased computational resources could also prove to be better for fine-tuning.

- **Incorporating Suprasegmental Features** Our current system primarily focuses on individual phonemes. However, overall speech intelligibility and fluency are also heavily influenced by broader speech characteristics.

  Future work should aim to integrate the assessment of prosodic features like stress, intonation, and rhythm. This would involve developing modules to analyze pitch contours, energy variations, and temporal patterns across words and phrases. We could also extend the system to evaluate speaking rate, pausing, and overall fluency, providing a more comprehensive view of a learner's communicative competence. Investigating multi-modal feedback, which could also improve learner understanding.

- **User Interface Development and Pedagogical Integration**

  Developing an interactive, user-friendly interface that visualizes phoneme boundaries, highlights mispronounced segments, and presents diagnostic feedback clearly would significantly enhance its educational value. Integrating the system with adaptive learning platforms to generate personalized pronunciation exercises based on detected error patterns would maximize learning efficiency. Finally, implementing features to track a learner's long-term pronunciation progress would provide invaluable insights for both learners and educators.

Such future enhancements aim to reduce the gap between system-generated scores and human expert ratings, especially in educational and clinical pronunciation assessment settings.

# Chapter 8

# Conclusion

In this thesis, we successfully developed a system for phoneme-level pronunciation assessment, specifically for Bengali (L2) English speakers. Our goal was to provide precise, actionable feedback, that subverted the limitations of traditional methods. We addressed the complexities of L2 speech by integrating robust acoustic analysis with clear diagnostic insights.

We built multi-stage pronunciation assessment pipeline. We built a unique Bengali L2 English speech corpus and utilized the Montreal Forced Aligner (MFA) for accurate phoneme segmentation. We used Wav2Vec2, a self-supervised model for the extraction of rich language-independent acoustic characteristics. We also developed Goodness of Pronunciation with Temporal Details (GOPT) to capture pronunciation changes over time, and integrated a speech attribute-based diagnosis module to explain errors linguistically. Our evaluation confirmed the system's strong performance in both detecting and diagnosing mispronounced phonemes, showing significant improvements over existing methods.

The successful development of this system holds significant promise for Computer-Assisted Pronunciation Training (CAPT). Offering granular, diagnostic feedback, it lets highly personalized learning experiences, letting learners and potentially easing the burden on human instructors. Its foundation in language-agnostic features suggests broad adaptability across diverse L2 linguistic backgrounds. While this thesis makes substantial contributions, it also lays the groundwork for future research, including exploring advanced models, enhancing cross-lingual generalization, expanding datasets, incorporating suprasegmental features, and developing user-friendly interfaces.

# References

[1] International Phonetic Alphabet

[2] wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli

[3] S. M. Witt, S. J. Young, Phone-level pronunciation scoring and assessment for interactive language learning, Speech communication 30 (2-3) (2000) 95–108.

[4] Kaldi

[5] CharSiu

[6] Whisper

[7] Sheoran, Kavita, et al. ”Pronunciation scoring with goodness of pronunciation and dynamic time warping.” IEEE Access 11 (2023): 15485-15495., Black, Matthew P., et al. ”Automated evaluation of non-native English pronunciation quality: combining knowledge-and data-driven features at multiple time scales.” Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[8] https://www.cambridge.org/features/IPAchart/

[9] https://www.internationalphoneticalphabet.org/mandarin-chinese/

[10] The Kaldi Speech Recognition Toolkit, Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukás Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Šilovský, Georg Stemmer, Karel Veselý, Presented at IEEE 2011 Automatic Speech Recognition and Understanding Workshop (ASRU)

[11] Hybrid Deep Neural Network–Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition, 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction

[12] IPA Chart: Arabic Phonemes

[13] Sound_correspondences_between_English_accents

[14] Segment_(linguistics)

[15] Sound correspondences between English accents

[16] IPA vowel chart with audio

[17] H. Franco, L. Ferrer, H. Bratt, Adaptive and discriminative modeling for improved mispronunciation detection, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 7709–7713.

[18] K. Truong, A. Neri, C. Cucchiarini, H. Strik, Automatic pronunciation error detection: an acoustic-phonetic approach (2004).

[19] S. Wei, G. Hu, Y. Hu, R.-H. Wang, A new method for mispronunciation detection using a support vector machine based on pronunciation space models, Speech Communication 51 (10) (2009) 896–905.

[20] M. Shahin, B. Ahmed, Anomaly detection based pronunciation verification approach using speech attribute features, Speech Communication 111 (2019) 29–43.

[21] M. Shahin, B. Ahmed, J. McKechnie, K. Ballard, R. Gutierrez-Osuna, A comparison of gmm-hmm and dnn-hmm based pronunciation verification techniques for use in the assessment of childhood apraxia of speech,in: Fifteenth Annual Conference of the International Speech Communication Association.

[22] A. M. Harrison, W.-K. Lo, X.-j. Qian, H. Meng, Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training, in: International Workshop on Speech and Language Technology in Education.

[23] D. V. Smith, A. Sneddon, L. Ward, A. Duenser, J. Freyne, D. SilveraTawil, A. Morgan, Improving child speech disorder assessment by incorporating out-of-domain adult speech, in: INTERSPEECH, pp.2690–2694

[24] S. M. Abdou, S. E. Hamid, M. Rashwan, A. Samir, O. Abdel-Hamid, M. Shahin, W. Nazih, Computer aided pronunciation learning system using speech recognition techniques, in: Ninth International Conference on Spoken Language Processing.

[25] M. S. Elaraby, M. Abdallah, S. Abdou, M. Rashwan, A deep neural networks (dnn) based models for a computer aided pronunciation learning system, in: International Conference on Speech and Computer, Springer, pp. 51–58.

[26] DiffMDD: A Diffusion-based Deep Learning Framework for MDD Diagnosis Using EEG

[27] Huang, Hao, et al. A transfer learning approach to goodness of pronunciation based automatic mispronunciation detection. The Journal of the Acoustical Society of America 142.5 (2017): 3165-3177

[28] Kanters, Sandra, Catia Cucchiarini, and Helmer Strik, The goodness of pronunciation algorithm: a detailed performance study. (2009).

[29] Shi, Jiatong, Nan Huo, and Qin Jin. Context-aware goodness of pronunciation for computer-assisted pronunciation training. arXiv preprint arXiv:2008.08647 (2020).

[30] Sudhakara, Sweekar, et al. An Improved Goodness of Pronunciation (GoP) Measure for Pronunciation Evaluation with DNN-HMM System Considering HMM Transition Probabilities. INTERSPEECH. Vol. 2. 2019.

[31] Pellegrini, Thomas, et al. The goodness of pronunciation algorithm applied to disordered speech. 15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014). ISCA, 2014.

[32] Gong, Yuan, et al. ”Transformer-based multi-aspect multi-granularity non-native english speaker pronunciation assessment.” ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

[33] Pei, Hao-Chen, et al. ”Gradformer: A Framework for Multi-Aspect Multi-Granularity Pronunciation Assessment.” IEEE/ACM Transactions on Audio, Speech, and Language Processing 32 (2023): 554-563.

[34] Shekar, Ram C., et al. ”Assessment of non-native speech intelligibility using wav2vec2-based mispronunciation detection and multi-level goodness of pronunciation transformer.” ISCA INTERSPEECH-2023 (2023).

[35] BASR18 Dataset