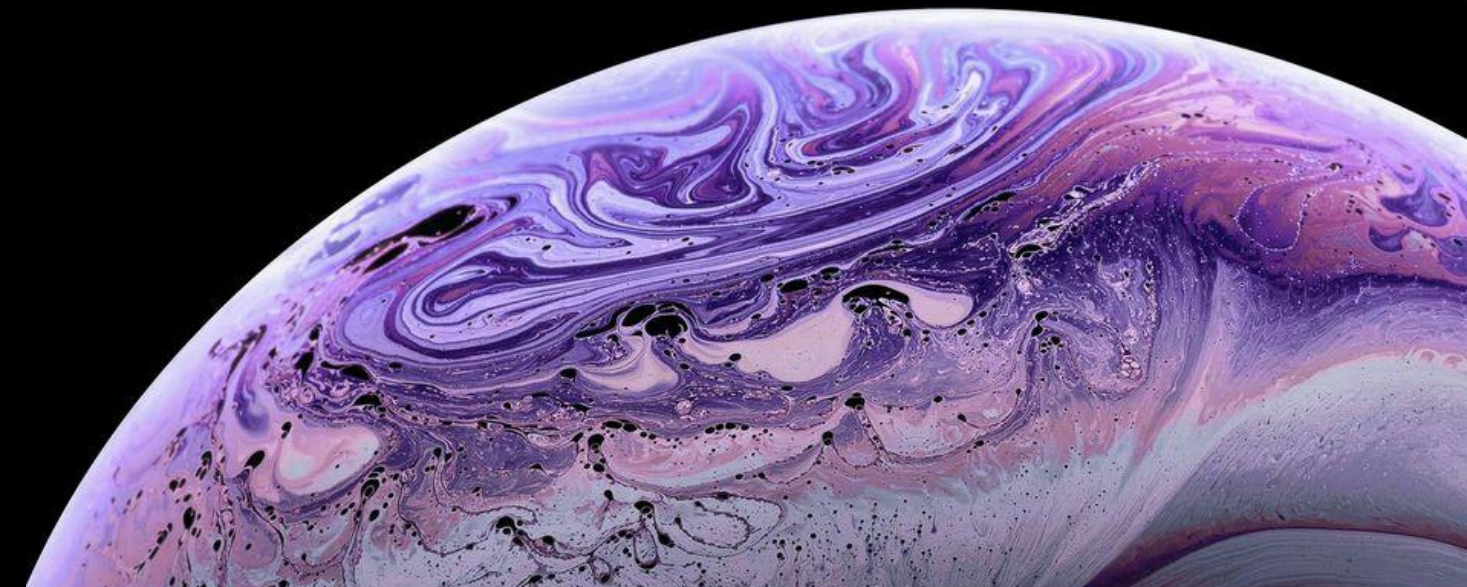
MACHINE LEARNING & PATTERN RECOGNITION

# Dhvani-187k

## Weakly-Supervised Word-Level Pronunciation Error Detection & Feedback System for Local Languages

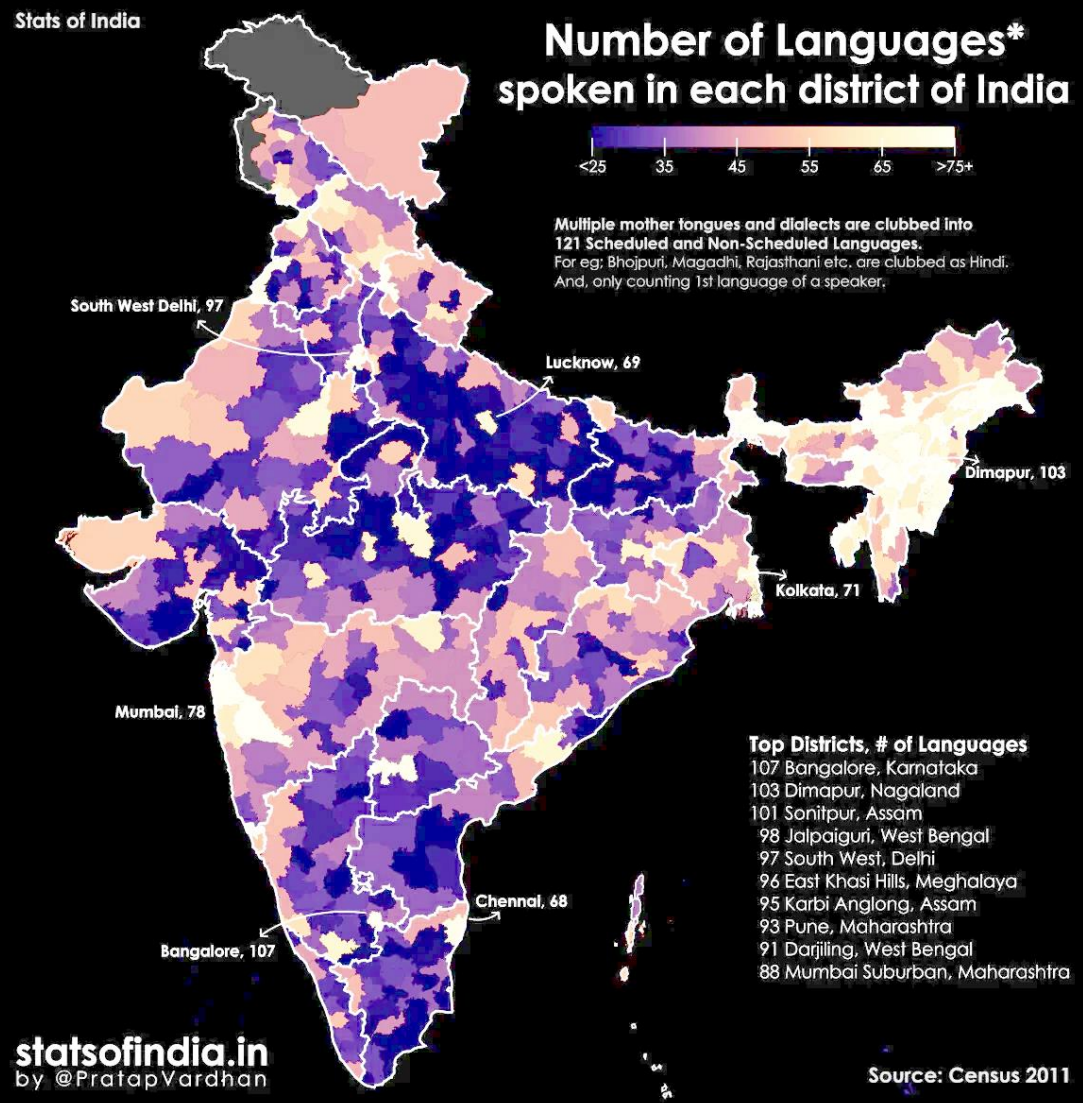
By

Arnav Rustagi,
Nimrat Kaur, &
Satvik Bajpai



# The Gap

- India has a rich linguistic landscape, but resources to learn pronunciation are lacking.
- Existing tools don't address regional variations or provide feedback.
- This hinders communication, cultural integration, and personal growth for non-native speakers.
- Innovative solutions are needed to bridge this gap and achieve fluency in learning local languages.




https://statsofindia.in/

# Problem Statement

India has a vast array of regional languages with diverse origins. This presents a steep pronunciation hurdle for non-native speakers. Current language learning tools offer no guidance for regional variations and lack crucial pronunciation feedback.

This creates a barrier to communication, cultural integration, and hinders confidence development for learners.

A CAPT system designed specifically for India's local languages is essential to bridge this gap.

We propose

# Dhvani

A *Computer-Aided Pronunciation Training* (CAPT) system using deep neural networks to detect word-level pronunciation errors made by language learners.

Dhvani provides real-time feedback, guiding users on how to improve their pronunciation accurately for regional Indian languages like Hindi, Punjabi, Marathi, and others.

# Impact

- Democratize language learning for all non-native learners, regardless of location or resources.
- Boost learner confidence and enhance educational and career prospects
- Facilitate cultural integration and understanding across India's diverse communities.
- Promote and preserve India's rich linguistic diversity by empowering learners to speak local languages like natives.

# Literature Survey

Automated Detection Of Pronunciation Errors In Non-native English Speech employing Deep Learning

Structural Analysis of Hindi Phonetics and A Method for Extraction of Phonetically Rich Sentences from a Very Large Hindi Text Corpus

Speech Synthesis Is All You Need

Weakly Supervised Word Level Pronunciation Error Detection

# Automated Detection Of Pronunciation Errors In Non-native English Speech Employing Deep Learning

- Phonemes are the basic units of speech.
- Sound also has prosodic features like energy, frequency, and duration.
- Types of pronunciation errors:
  - Phoneme addition
  - Phoneme deletion
  - Phoneme modification
  - Lexical stress errors


*D. Korzekwa, "Automated detection of pronunciation errors in non-native English speech employing deep learning," arXiv.org, Sep. 13, 2022. https://arxiv.org/abs/2209.06265#:~:text=This%20Ph. (accessed May 10, 2024).*

# Challenges With Pronunciation Error Detection

- Availability of mispronounced speech is limited.
- Transcription errors in such mispronounced speech are very common.
- Lexical stress errors are a very important part of any CAPT system.
- Aligning canonical phonemes* with the recognized phonemes is hard.

*canonical phonemes -> the real phonemes of any sentence

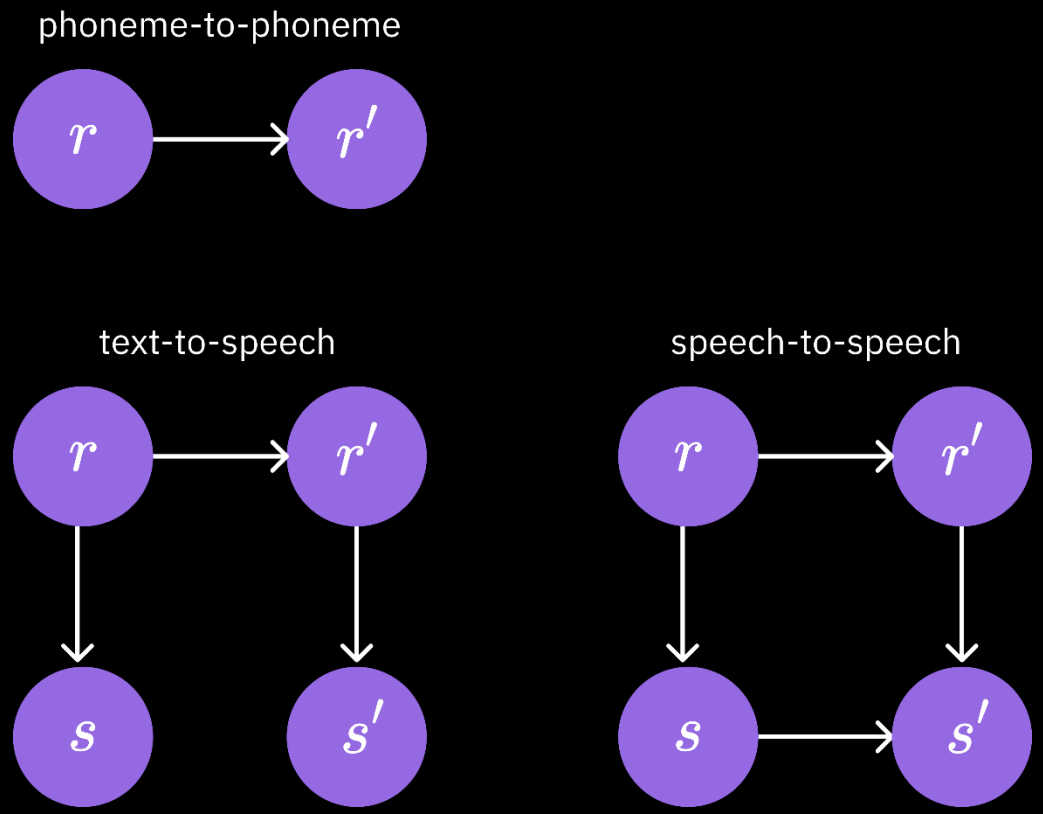
*Equation 1*

$$P(e|s,r) = \frac{P(e|r) \times P(s|e,r)}{P(s,r)}$$

**e -** array of phoneme size depicting errors
**s -** speech
**r -** sequences of phonemes

*D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, and B. Kostek, "Computer-assisted Pronunciation Training Speech synthesis is almost all you need"Accessed: Mar. 15, 2024. [Online]. Available: https://arxiv.org/pdf/2207.00774.pdf*

# Speech Synthesis Is All You Need




Three primary methods of speech synthesis:

replacing the phoneme in the word. (P2P)

changing the speech signal while retaining the original pronunciation (T2S)

changing the prosodic features of speech (S2S)

*D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, and B. Kostek, "Computer-assisted Pronunciation Training Speech synthesis is almost all you need"Accessed: Mar. 15, 2024. [Online]. Available: https://arxiv.org/pdf/2207.00774.pdf*

# Weakly Supervised Word Level Pronunciation Error Detection

The model is trained for 2 tasks, using 2 networks, while using the same decoder:

## Phoneme Recognition Network (PRN)

Recognizes phonemes pronounced by the speaker

Architecture: RCNN encoder and same A-RNN decoder as MDN

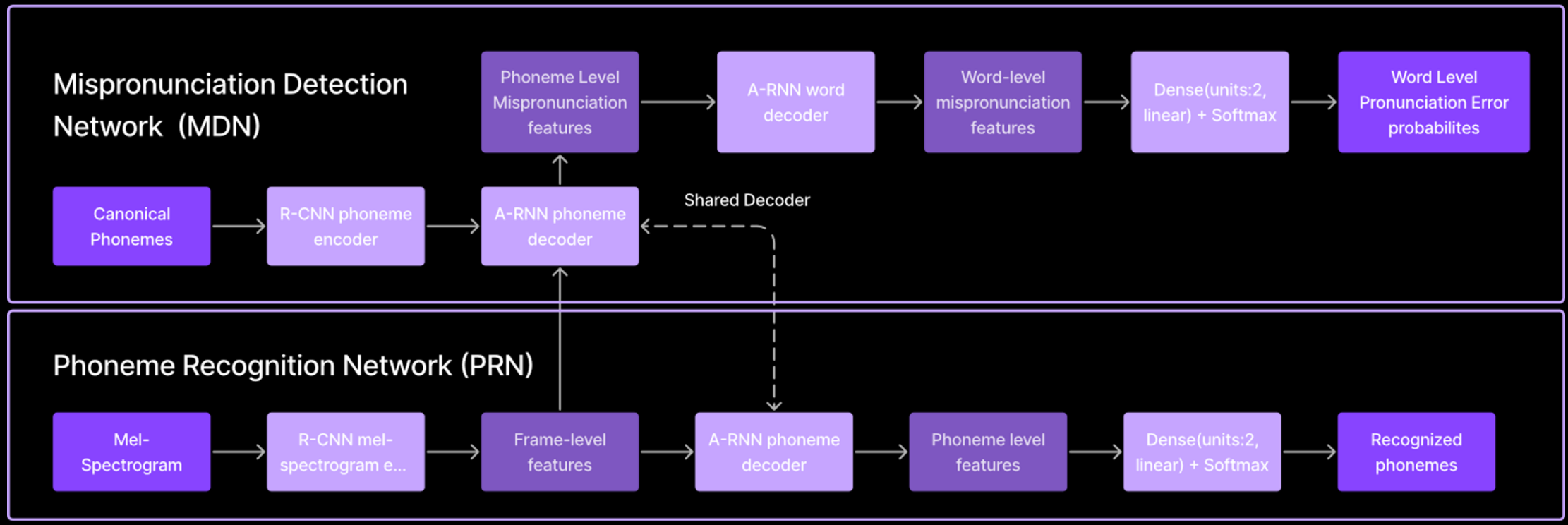
## Mispronunciations Detection Network (MDN)

Detects word-level pronunciation errors

Architecture: RCNN encoders for predicted phonemes
Attention-based RNN decoder for phoneme-level & word-level features

*[1]D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek, "Weakly-supervised word-level pronunciation error detection in non-native English speech." Accessed: Mar. 15, 2024. [Online]. Available: https://arxiv.org/pdf/2106.03494.pdf*

# Proposed Weakly-S Model




$$L(\theta) = \log(p(e|a, r_c, \theta)) + \log(p(ro|a, \theta))$$

**θ**: Parameters of the model.
**a**: Speech signal represented by mel-spectrogram.
**$r_c$**: Canonical phonemes that the speaker was expected to pronounce.
**$r_o$**: Phonemes recognized in a by PRN.
**e**: Probabilities of mispronounced words in spoken sentence.
**log**: Log Likelihood Loss

*[1]D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek, "Weakly-supervised word-level pronunciation error detection in non-native English speech." Accessed: Mar. 15, 2024. [Online]. Available: https://arxiv.org/pdf/2106.03494.pdf*

# Our Innovation

We aim to expand upon the present research in the following ways:

- Extend it to the language of Hindi first (and more later) which has 64 phonemes as compared to 45 in English.
- Implement a feedback system based on research.
- Trying different model architectures and hyperparameters.

# Structural Analysis of Hindi Phonetics and A Method for Extraction of Phonetically Rich Sentences from a Very Large Hindi Text Corpus

Velar: Made using the back part of your tongue with the roof of the back part of the mouth
Labial: Made with your lips.
Dental: Made by placing your tongue tip against your upper teeth.
Retroflex: They are made by curling your tongue tip back and touching the roof of your mouth behind the alveolar ridge.
Palatal: Made by touching the middle of your tongue to the hard palate (the bony roof of your mouth) behind the alveolar ridge.

| Group Name | Tongue Position | Group Members | | | | |
|---|---|---|---|---|---|---|
| | | VL* | VLA# | V¶ | VA§ | N† |
| क वर्ग | Velar | क | ख | ग | घ | ङ |
| च वर्ग | Palatal | च | छ | ज | झ | ञ |
| ट वर्ग | Retroflex | ट | ठ | ड | ढ | ण |
| त वर्ग | Dental | त | थ | द | ध | न |
| प वर्ग | Labial | प | फ | ब | भ | म |

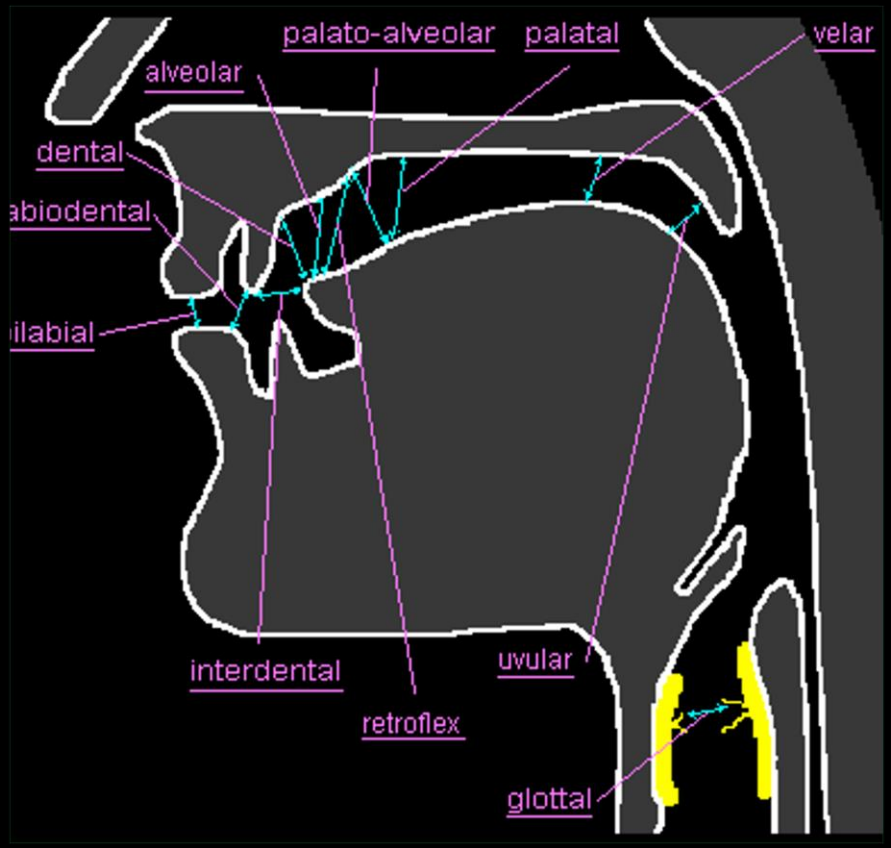
*VL = VOICE LESS
#VLA = VOICE LESS ASPIRATED
¶V = VOICED
§VA = VOICED ASPIRATED
†N = NASAL

Shrikant Malviya, Rohit Kumar Mishra, and Uma Shanker Tiwary, "Structural Analysis of Hindi Phonetics and A Method for Extraction of Phonetically Rich Sentences from a Very Large Hindi Text Corpus," arXiv (Cornell University), Jan. 2017, doi: https://doi.org/10.48550/arxiv.1701.08655.

# Feedback

The unvoiced palatal aspirated affricate 'छ' is pronounced like the 'cchh' in the English word 'catch-hay'. Produce this sound by raising the front part of the tongue towards the hard palate, releasing the air with a plosive sound, and following it with a strong aspiration.

The short vowel 'अ' is pronounced like the 'u' in the English word 'cut'. Produce this sound by spreading the lips slightly and keeping the tongue relaxed in the mouth.




Shrikant Malviya, Rohit Kumar Mishra, and Uma Shanker Tiwary, "Structural Analysis of Hindi Phonetics and A Method for Extraction of Phonetically Rich Sentences from a Very Large Hindi Text Corpus," arXiv (Cornell University), Jan. 2017, doi: https://doi.org/10.48550/arxiv.1701.08655.

# MUCS* Dataset (L1)

We are using the Multilingual & code-switching ASR challenge dataset from OpenSLR curated by the MUCS organization

- 95.05 hours of data for train, 5.55 hours for test
- 4506 unique sentences for train, 386 unique sentences for test
- 59 speakers in train set, 19 speakers in test set
- Audio files are sampled at 8kHz, 16bit encoding
- Total vocabulary of train + test set is 6542 words
- Each sample is of length 6-13 seconds, speaking a single sentence
- Several speakers are saying the same sentence
- The data set comprises of telephone quality speech data in Hindi. The recordings were collected through the Mobile Vaani platform having users from all across India
- Corresponding transcription was done by crowd workers
- This dataset is primarily made for the task of Automatic Speech Recognition
- Each individual samples consists of the audio recording & the text transcribed

https://www.openslr.org/104/

# Web Scraping (L2)

We manually scraped data from YouTube videos

- 1 hour of data for train, 0.2 hours for test
- We have 15 unique speakers across train & test
- Audio files was resampled to 8kHz, 16bit encoding
- Each sample is of length 6-8 seconds, speaking a sentence
- All speakers are saying different sentences
- The data is not transcribed
- This dataset is primarily made for the task of fine-tuning the Phoneme Recognition Network
- Each individual samples consists of the audio recording
- Since the Weakly-S model uses only 1% of data as the L2 non-transcribed data, this volume of data is satisfactory for finetuning.

# Speech Synthesis for Pronunciation Errors (L1)

We used suno/bark-small to synthetically generate mispronounced data as outlined by the paper speech synthesis is all you need.

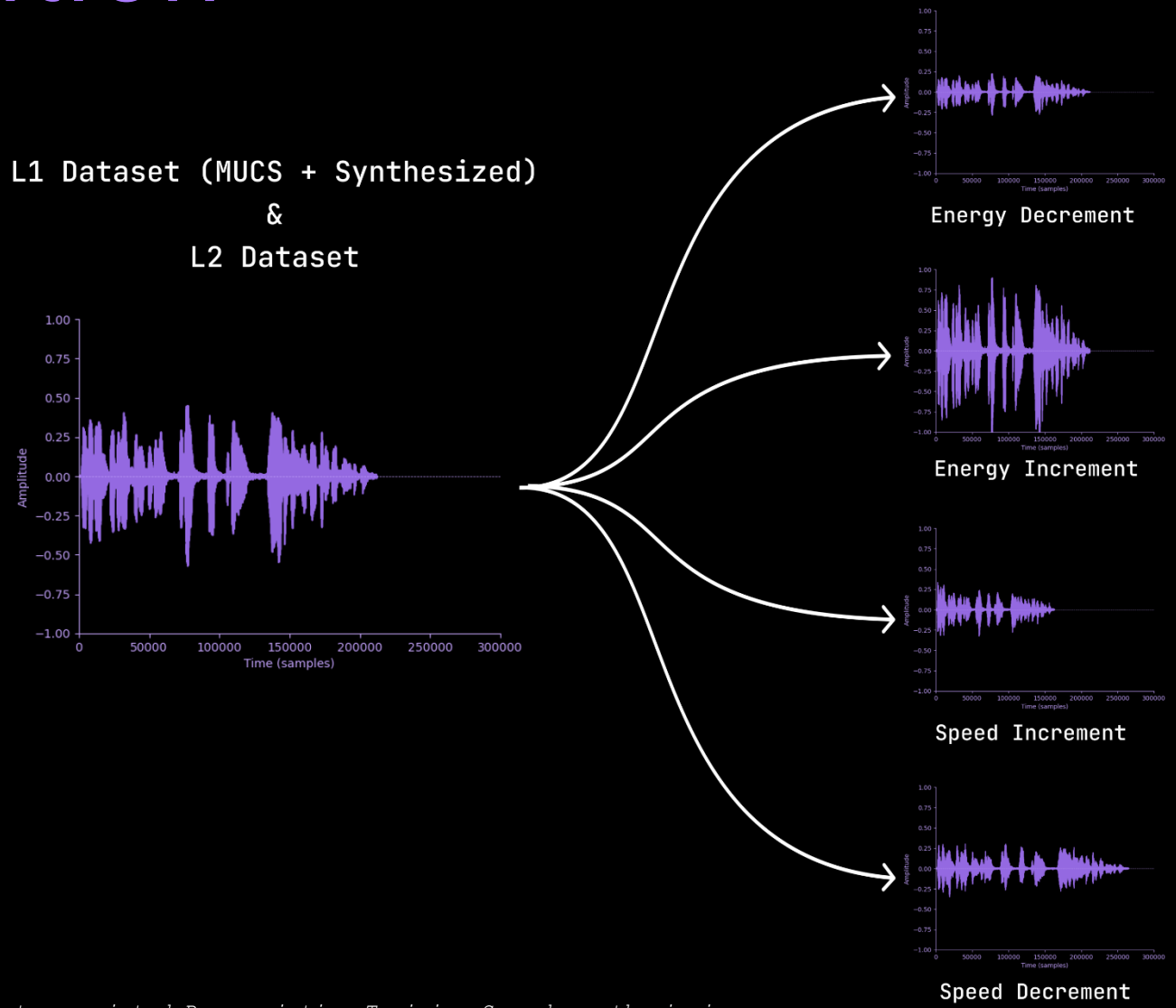
- 3+ hours of total data
- We had 10 unique speaker presets for hindi speech
- Our dataset has 1000 correct & 1000 incorrect sentences
- We introduced mispronunciation errors with a probability of 0.05
- Audio files was resampled to 8kHz, 16bit encoding
- Each sample is of length 6-8 seconds, speaking a sentence from a corpus of sentences from NCERT Hindi textbooks
- We generated a correct sample & an incorrect sample for each sentence using the same speaker
- The data is transcribed with the sentence being spoken
- Mispronounced data has the corresponding error probability vectors, denoting error probability per phoneme (0 or 1 for train set)
- This dataset is primarily made for the task of Automatic Speech Recognition
- Each individual samples consists of the audio recording and transcribed text

# Data Augmentation

We augmented each sample the following way based on the paper Speech Synthesis is All You Need:

- Original Audio Sample
- Energy Variation
  - Increasing energy of the audio sample
  - Decreasing energy of the audio sample
- Speed Variation
  - Increasing the speed of audio sample
  - Decreasing the speed of the audio sample

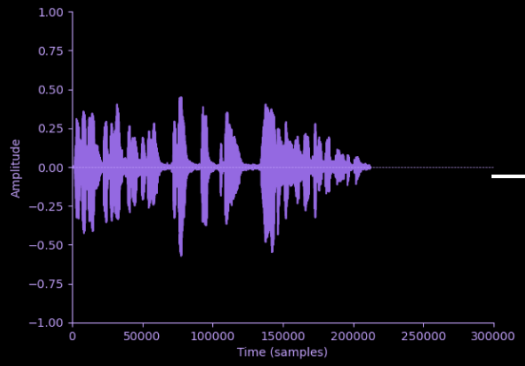
We used the following methodology to boost our dataset by 5 times.




*D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, and B. Kostek, "Computer-assisted Pronunciation Training Speech synthesis is almost all you need"Accessed: Mar. 15, 2024. [Online]. Available: https://arxiv.org/pdf/2207.00774.pdf*

# Data Preprocessing

Prosodic Variations of Augmented Data




Error Probabilities → Padding

Standardization → Padding

To 256 Frames

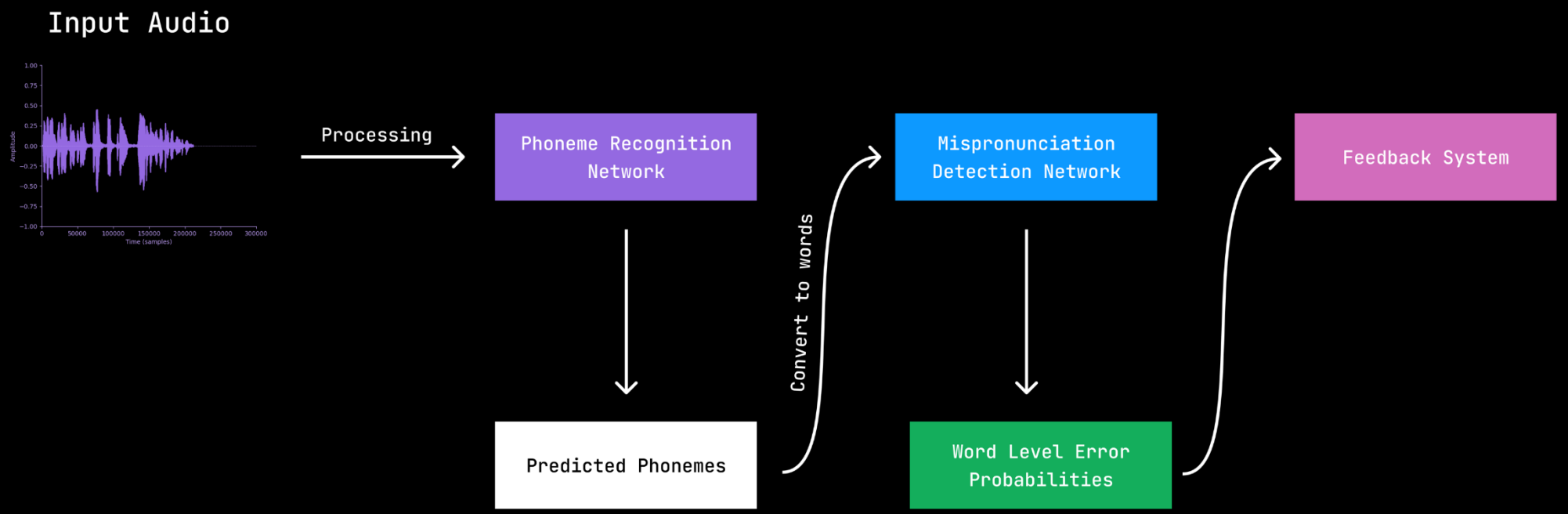
Transcript → Tokenization → Padding

To 256 Tokens

Data is ready!

64: unique phonemes
1: padding token
1: end of word token
1: end of sentence token

# ML Methodology

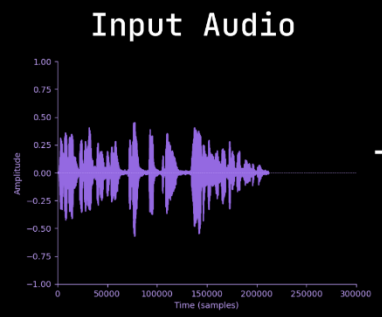


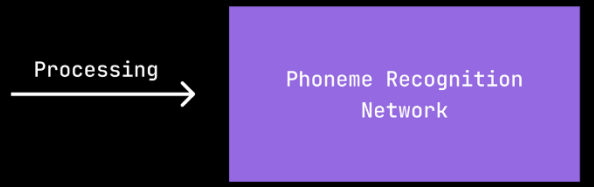
# Phoneme Recognition Network

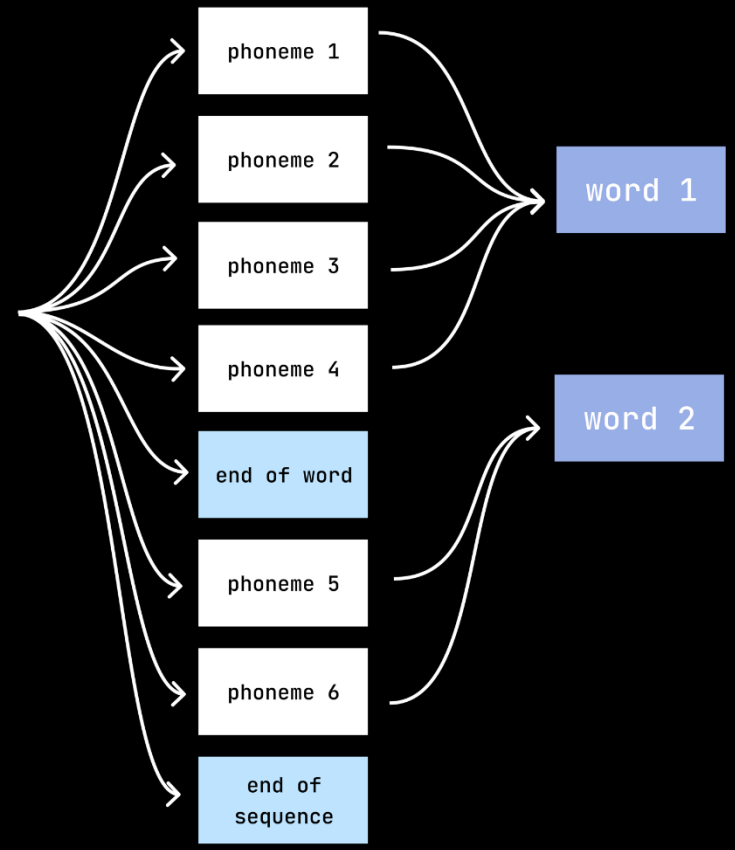
The Phoneme Recognition Network is the heart of our methodology.

Input: Mel Spectrogram
Output: Predicted phonemes

It is an encoder-decoder architecture
With a R-CNN encoder & A-RNN decoder.

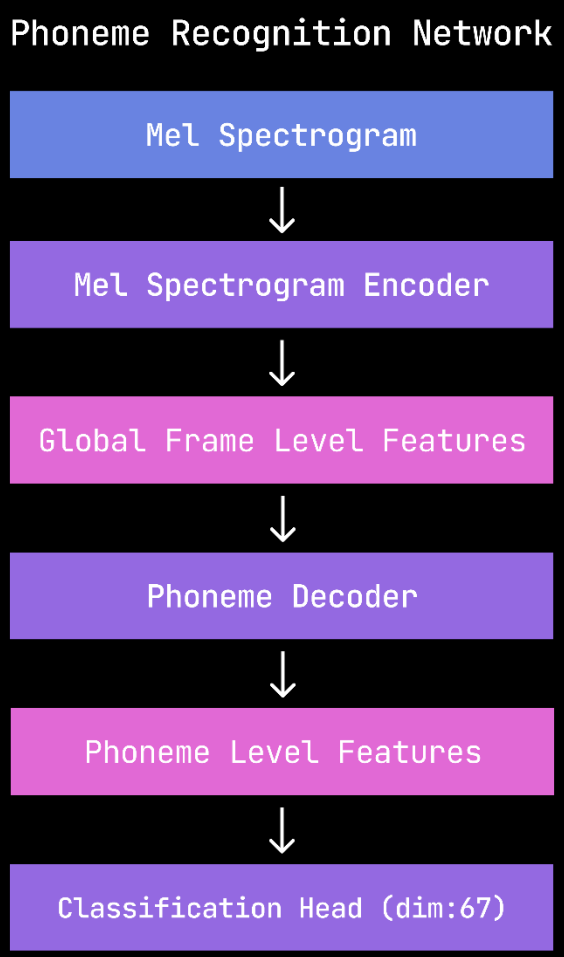


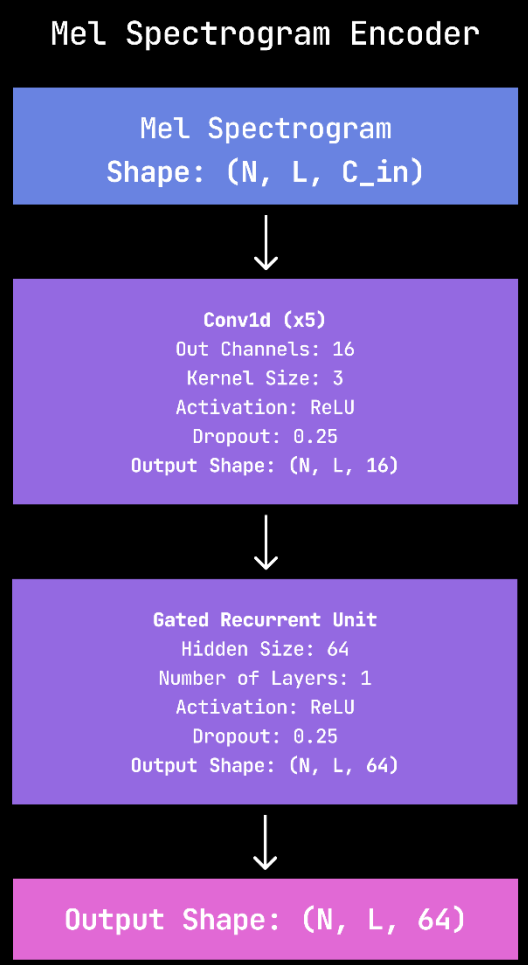



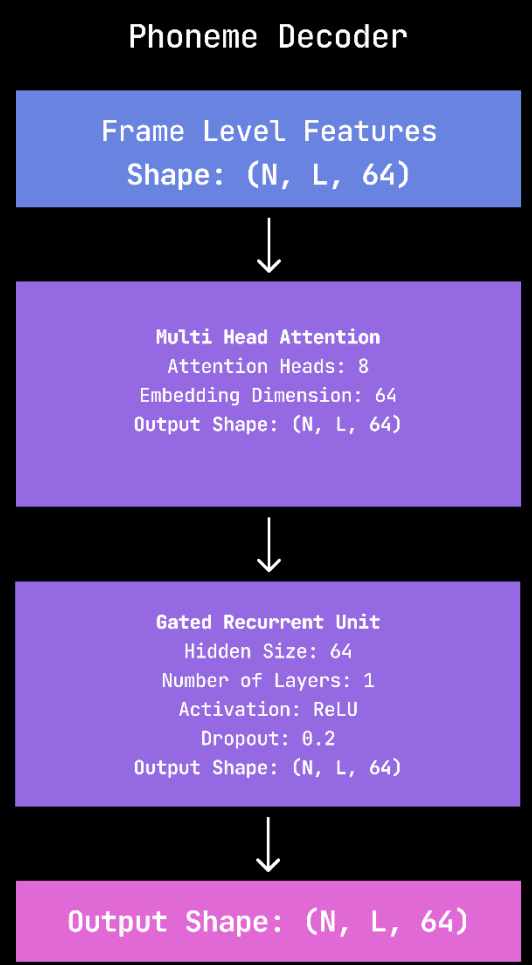




Novelty:
The Weakly-S model uses 3 kernels of size 5 to extract frame level features. Dhvani use 5 kernels of size 3 to extract the same features, boasting a better f1-score.

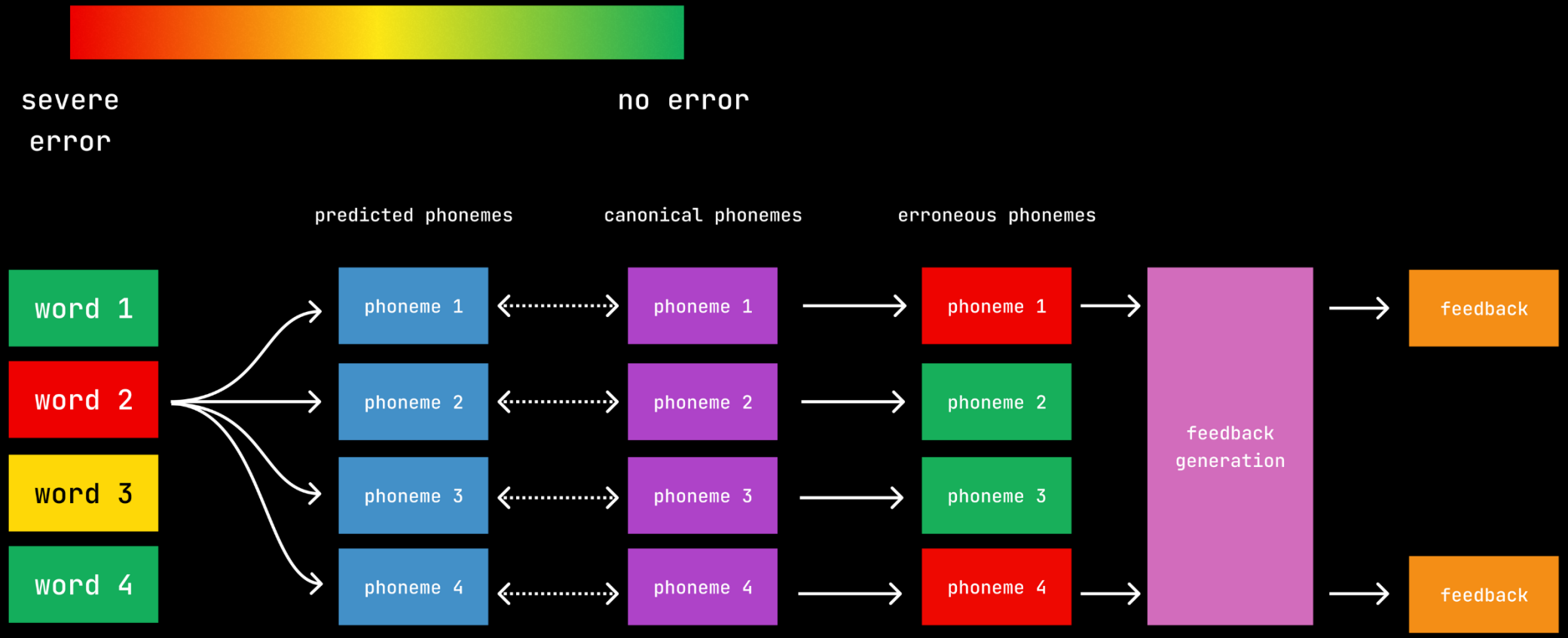
# Phoneme Recognition Network



# Feedback


severe error
no error
predicted phonemes
canonical phonemes
erroneous phonemes
word 1
word 2
word 3
word 4
phoneme 1
phoneme 2
phoneme 3
phoneme 4
phoneme 1
phoneme 2
phoneme 3
phoneme 4
phoneme 1
phoneme 2
phoneme 3
phoneme 4
feedback generation
feedback
feedback

# Training Methodology

## Training Data & Batching

- Data is batched with a batch size of 32
- Data is padded to the length of 256 tokens/frames
- We employ 10-Fold Cross Validation.

## Hardware & Schedule

- We trained our model on one machine with an M3 Pro Apple Silicon processor
- Each training step took about 1.59s.
- Each epoch had 370000 Training Steps & 42000 Validation Steps to train.

# Training Methodology

## Optimizer

- We used AdamW Optimizer
- We used a learning rate of 3e-6, beta-1 of 0.9 & beta-2 of 0.999 with an epsilon of 1e-8
- We employed 10-Fold Cross Validation.
- We used L2 Gradient Clipping of value 5 to address gradient explosion.

## Regularization

- Dropout: We applied different dropout values to different layers to prevent overfitting.
- Early Stopping: The model stops training at ~100 epochs, since the validation accuracy starts decreasing after that

# Performance Metrics

What performance metrics were used?

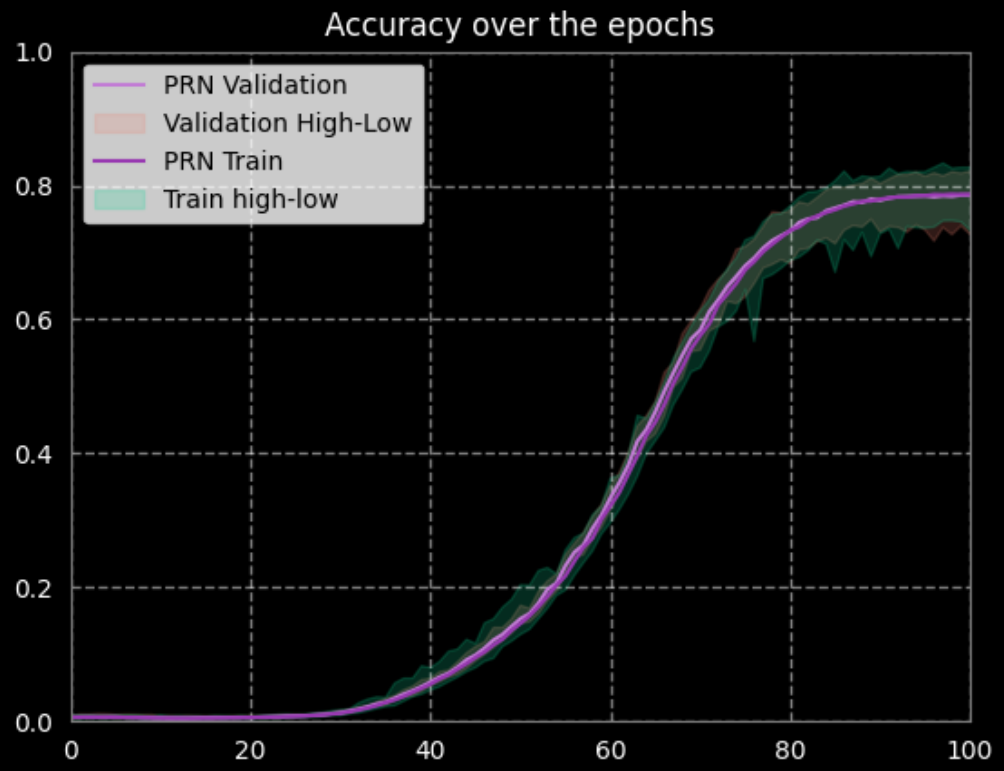
- We used Weighted F1-Score, Weighted Precision & Weighted Recall for our model.

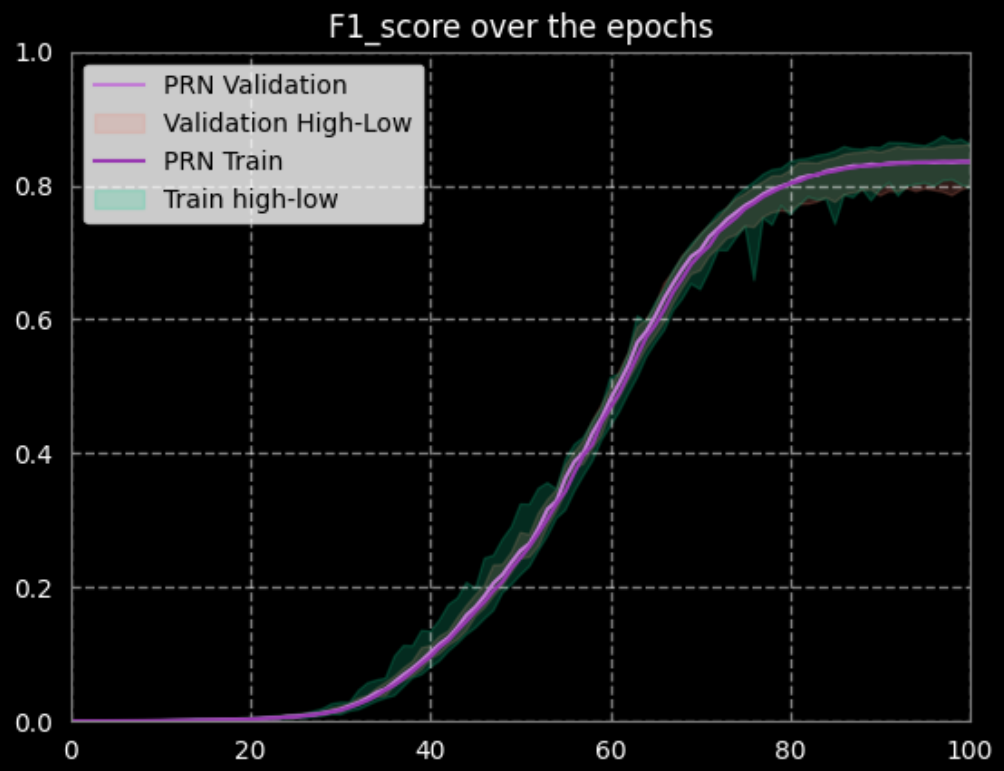
Why are these metrics relevant?

- Weighted precision, recall and F1 Score are appropriate metrics when the dataset is imbalanced, i.e., each class has a different number of samples. They provide a more reliable indicator of the model's performance compared to global accuracy
- Weights are defined as the proportion of occurrences of each class in “target”. To calculate the weighted metrics, we calculate the metrics for each class separately and then take their weighted sum

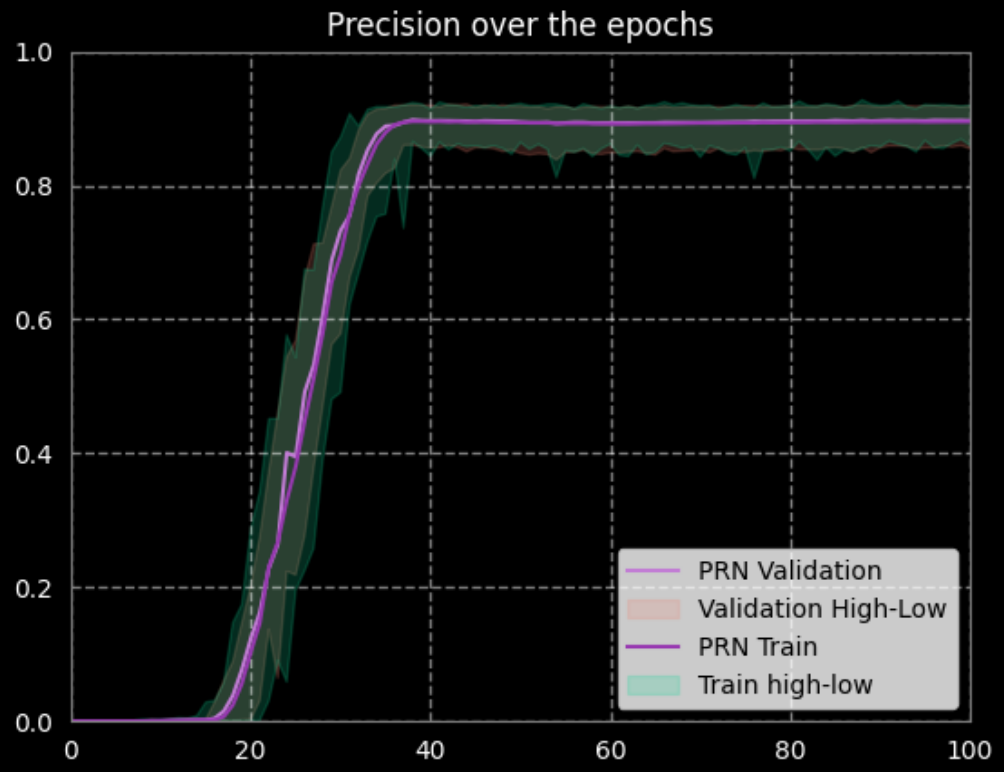
https://pytorch.org/torcheval/stable/generated/torcheval.metrics.functional.multiclass_f1_score.html

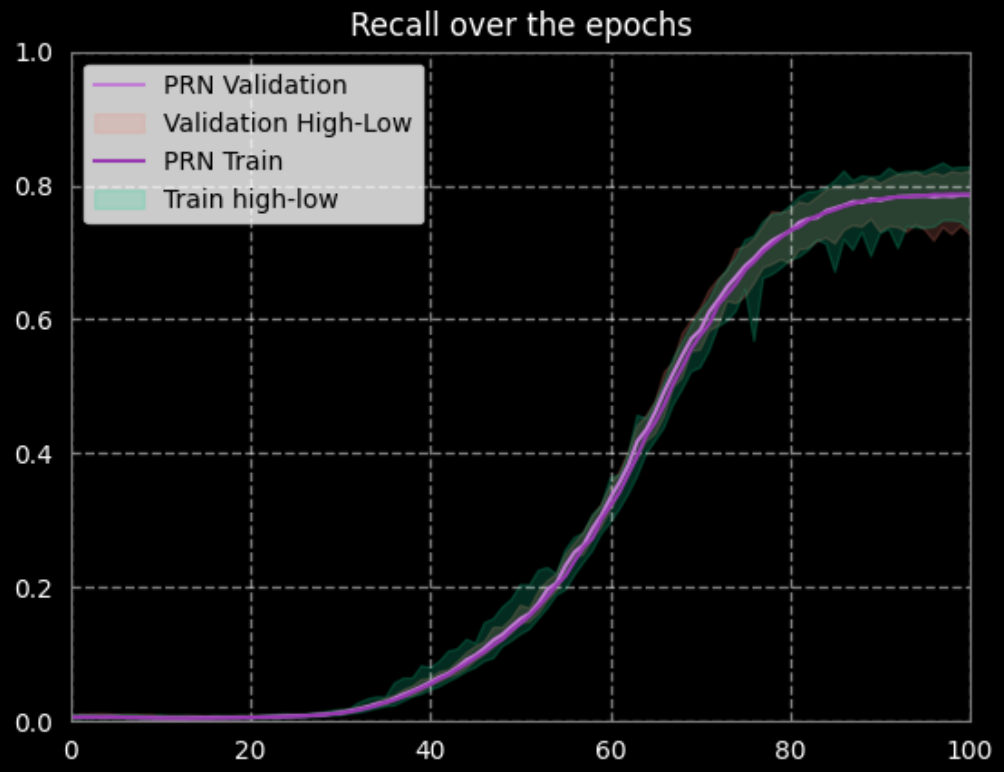
# Results - Training



# Results - Training



# Test Metrics

| Metric | Value |
| --- | --- |
| Accuracy | 76.32% |
| Precision | 89.64% |
| Recall | 75.48% |
| F1-Score | 82.24% |

# Comparisons

| Model | Paper | Accuracy |
|---|---|---|
| PR-PM | D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, S. Calamaro, and B. Kostek, "Weakly-supervised word-level pronunciation error detection in non-native English speech." Available: https://arxiv.org/pdf/2106.03494 | 55.52% |
| Multi Task DNN-HMM | M. L. Seltzer and Jasha Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," May 2013, doi: https://doi.org/10.1109/icassp.2013.6639012. | 79.75% |
| Bidirectional LSTM | A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," *IEEE Xplore*, 2005. https://ieeexplore.ieee.org/document/1556215 | 69.5% |
| Bidirectional RNN | A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," *IEEE Xplore*, 2005. https://ieeexplore.ieee.org/document/1556215 | 64.7% |
| ASRG (Attention-based Recurrent Sequence Generator) | J. Chorowski and D. Bahdanau, "Attention-Based Models for Speech Recognition." Accessed: May 10, 2024. [Online]. Available: https://arxiv.org/pdf/1506.07503 | 88.5% |
| Dhvani-187k | Team Dhvani | 76.3% |

# Challenges Faced

1. Encoder Issue and Solution
    - Encoder not extracting relevant features
    - Replaced 5x5 conv layer with two 3x3 conv layers
2. Validation-Training Metric Discrepancy
    - Increased dropout rate from 0.045 to 0.25 in encoder
3. Gradient Explosion
    - Slow learning and erratic losses across epochs
    - Implemented gradient clipping and small learning rate
4. Incorrect Micro-Average F1-Score
    - Model predicting same values repeatedly
    - Used weighted averages for computing F1-score
5. Potential Bias
    - Data based on synthetic mispronunciations
    - Susceptible to biases in real-world scenarios

# Deployability

- Integrate CAPT into Language Courses
  - Our CAPT system can be integrated into language and communication courses such as those offered by CTLC (Center for Thinking and Language Communication)
  - Provide personalized pronunciation feedback and practice for students
  - Install the CAPT system as a self-learning tool for students and staff in the library.
- Mobile App/Web Platform
  - Develop a mobile app or web platform version of the CAPT system
  - Enable access to the system for remote learning and practice
- Collaborative Research
  - Attract research collaborations with linguistic experts and ML researchers
  - Continuously enhance and expand the CAPT system for more languages

# Challenges in Deployment

- Handling Accent and Dialect Variations
- Continuous Learning and Model Updates
- Model Optimization and Deployment
- Reliance on Synthesized Data

# Thank You!