# Nepali To English Speech Translation With Prosody Prediction


**Team Members**

Pragyan Bhattarai (THA077BEI030)

Prashant Raj Bista (THA077BEI032)

Shakshi Kejriwal (THA077BEI044)

Sudipti Upreti (THA077BEI045)

**Supervised By**

Er. Kshetraphal Bohara

Department of Electronics and Computer Engineering

IOE, Thapathali Campus


July, 2024

# Presentation Outline

- Problem statement
- Objective
- Methodology
- Results
- Discussion and Conclusion
- References

# Problem Statement and Objectives

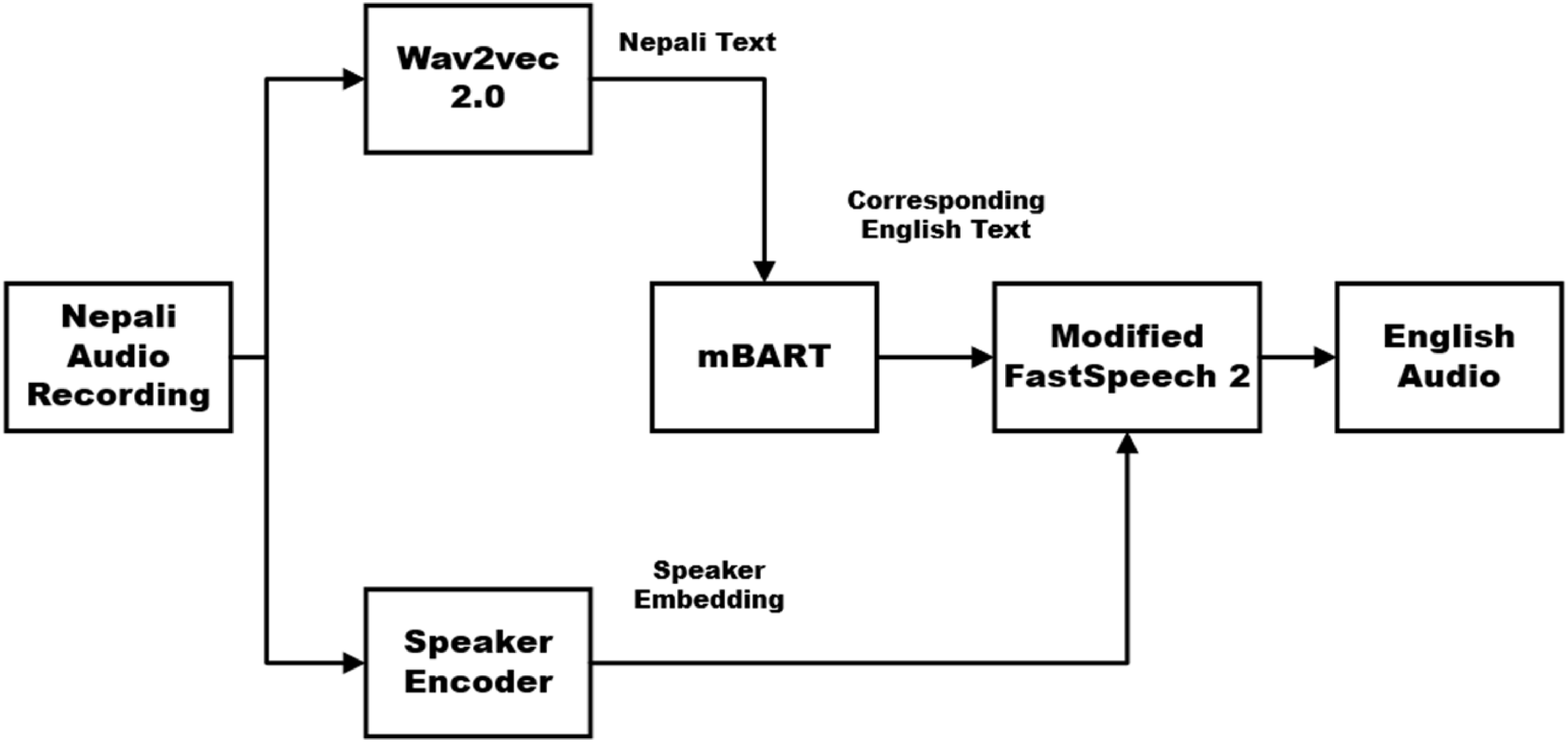
## Problem Statement

- Current translation systems lacks to convey the prosody and emotional nuances of spoken Nepali in English

## Objective

- To develop a Nepali-to-English speech-to-speech translation system  with prosody prediction on the translated language.

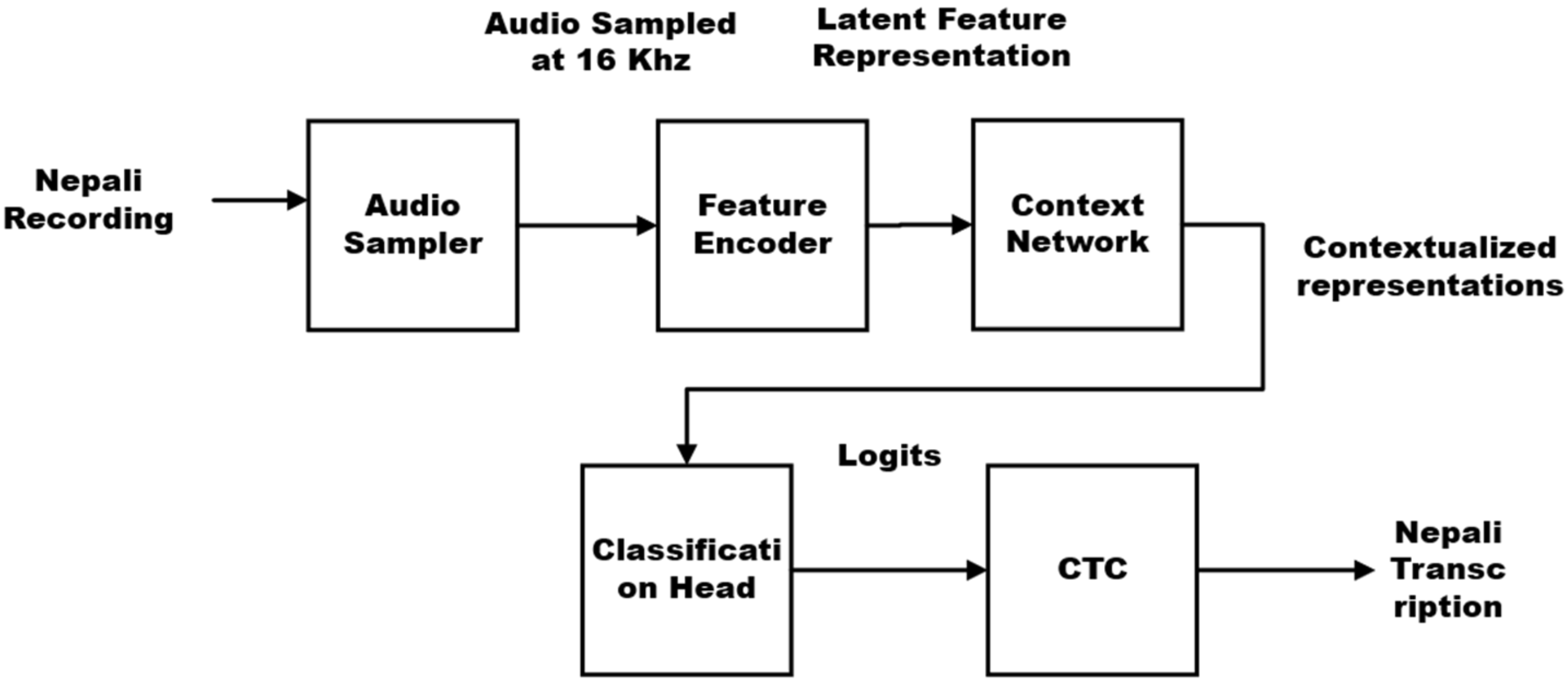
# Methodology- [1]



# Methodology - [2] (Description of System Block Diagram)

- wav2vec 2.0 processes the recording.
- wav2vec 2.0 produces the corresponding Nepali text as output.
- mBART processes the Nepali text and outputs with an English Text.
- FastSpeech 2 takes English text and speaker embedding.
- FastSpeech 2 synthesizes the corresponding English audio with desired prosody.

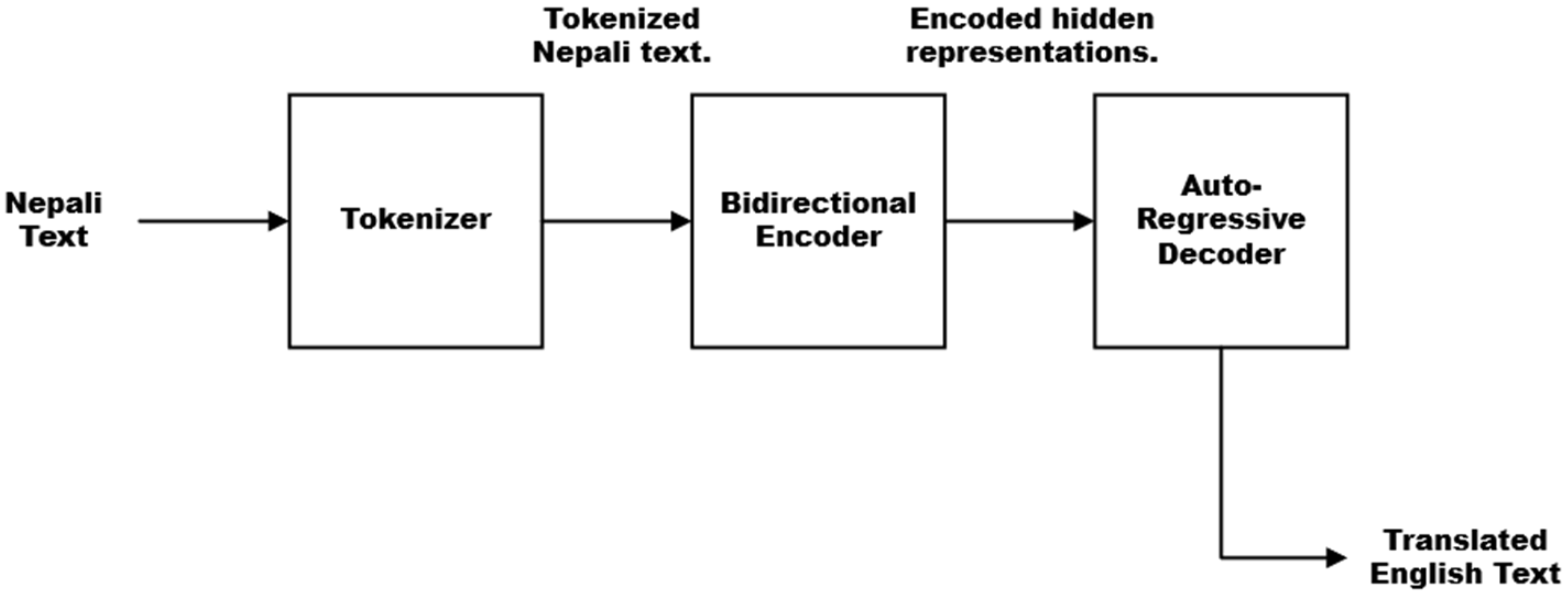
# Methodology - [3]
# (Data Flow in wav2vec 2.0)



# Methodology - [4] (wav2vec 2.0 Description)

- Feature encoder extracts the features and converts the data into the latent feature representation from recording sampled at 16KHz.
- Context network converts the latent feature into contextual representation.
- Classification head converts contextualized representation into the logits.
- CTC converts the logits into the corresponding Nepali text spoken in audio.

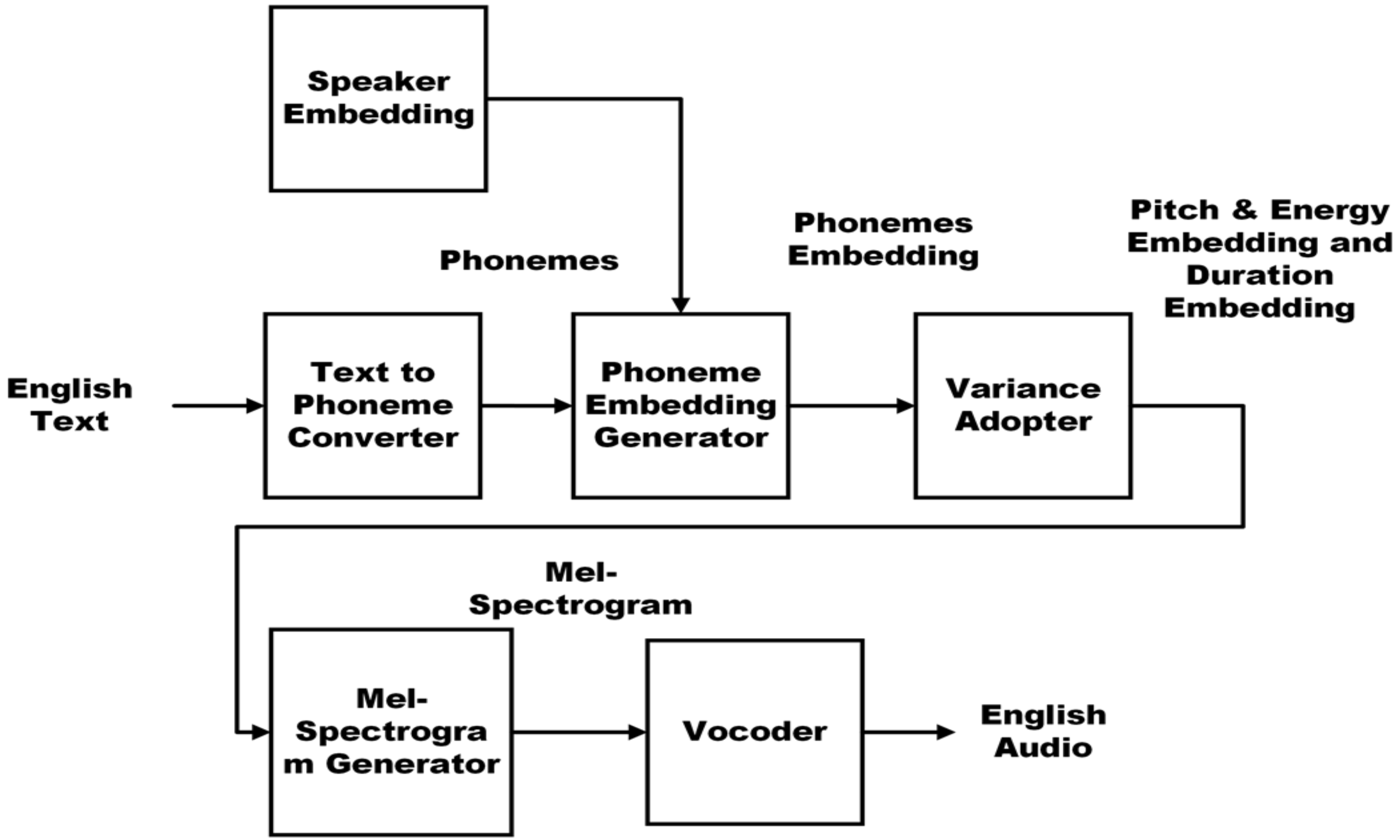
# Methodology - [5] (mBART)



# Methodology - [6] (mBART Description)

- Tokenizer converts the Nepali text into the tokens.
- Bidirectional encoder produces the hidden representation of the data.
- Encoded hidden representation is passed to the autoregressive decoder.
- Decoder produces the  transcribed English text.
- The transcribed English text is passed to a TTS model.

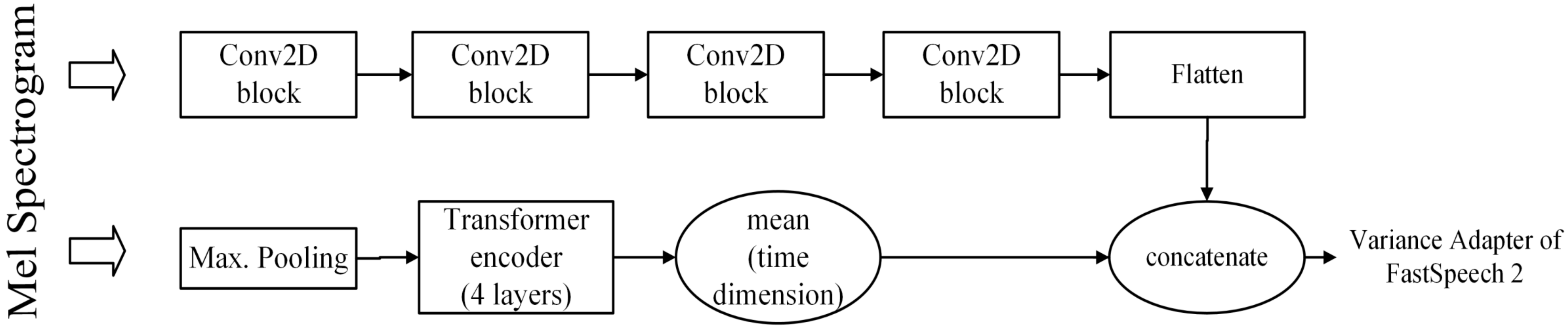
# Methodology - [7] (FastSpeech 2)



# Methodology - [8] (FastSpeech 2 Description)

- English text is converted to the respective phonemes.
- Phonemes encoder that produces the phoneme embedding.
- Variance adapter predicts parameters for the generation of the speech.
- Speaker Embedding is concatenated with the output of the variance.
- Mel-Spectrogram generator generates the spectrogram for the speech to be synthesized.

# Methodology - [9]
# (Speaker Encoder)



# Methodology - [10]
# (Speaker Encoder Description)

- 2D convolution blocks captures features from the input data
- The output of the final Conv2D block is flattened into a 1D vector.
- Max pooling is applied to reduce the dimensionality of
  the input features from Mel spectrogram.
- These pooled features are fed to Transformer encoder to capture
  dependencies in the data.

# Methodology - [11]
# (Speaker Encoder Description)

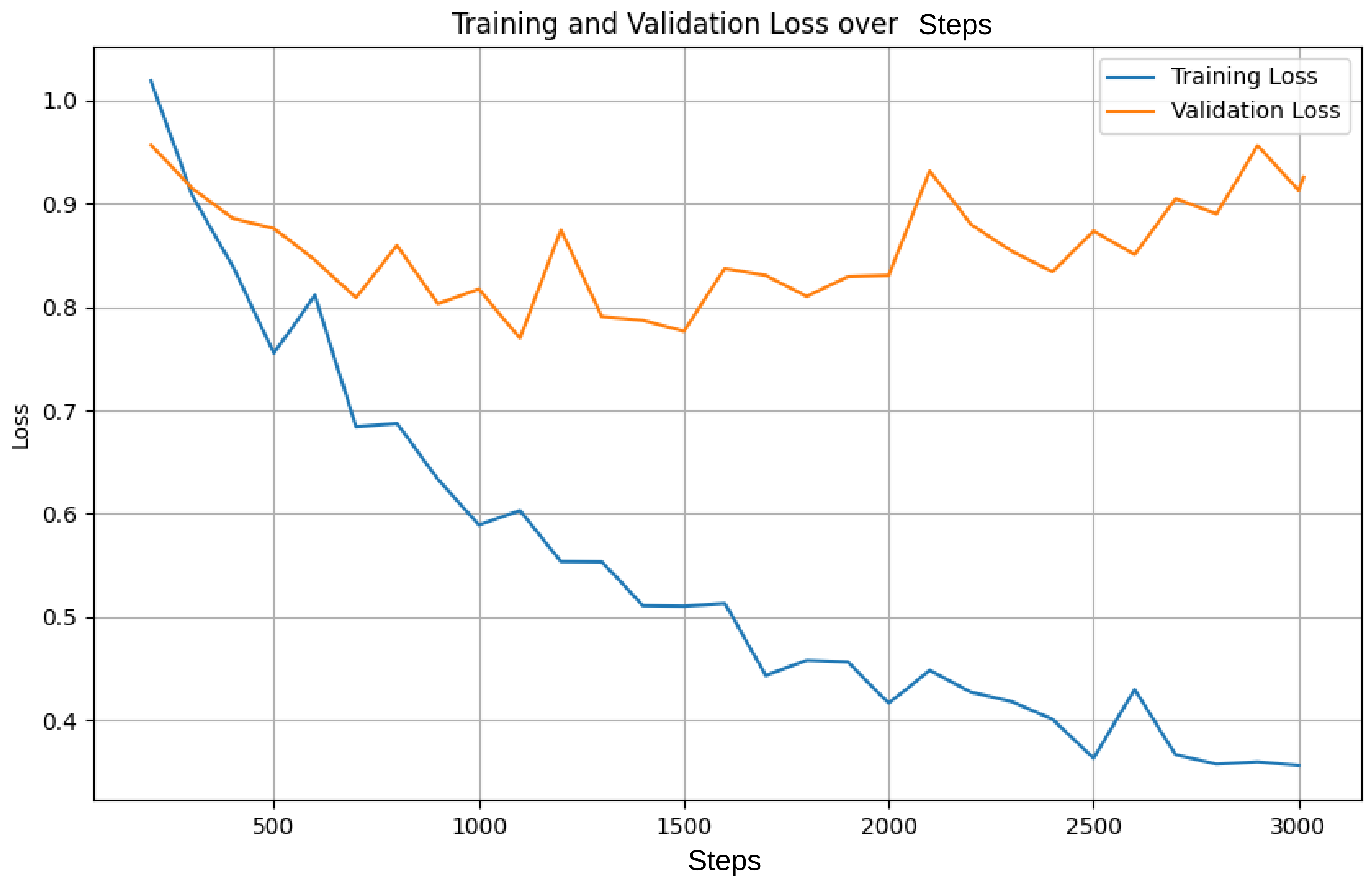
- Output of the transformer encoder is averaged which Summarizes the temporal features.
- Output of Conv2D and Transformer are concatenated forming single feature vector.
- Concatenation combines the local features captured by Conv2D blocks with the global features captured by Transformer encoder.
- The combined feature vector is fed into the Variance Adapter of FastSpeech 2.

# Methodology - [11] (Hyperparameters Table for ASR)

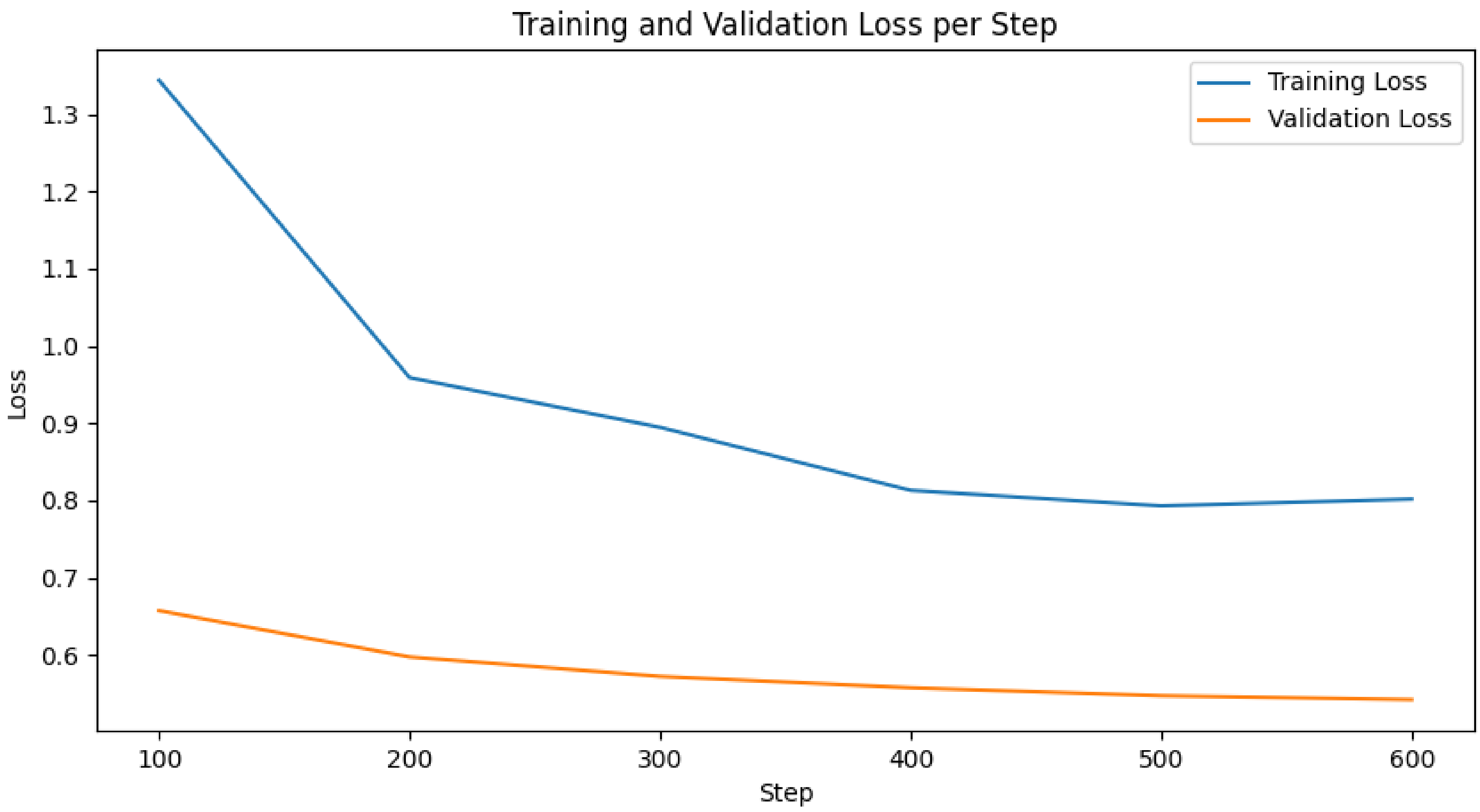
| | |
|---|---|
| Batch Size | 4 |
| Evaluation Strategy | steps |
| fp16 | TRUE |
| Training Epochs | 15 |
| Steps to Model Saving | 100 |
| Steps to Model Evaluation | 100 |

| | |
|---|---|
| Gradient Accumulation Steps | 4 |
| Steps to Log Results | 100 |
| Learning Rates | 3.00E-05 |
| Load Best Model | TRUE |
| Metrics | WER |

# Results and Analysis - [1]

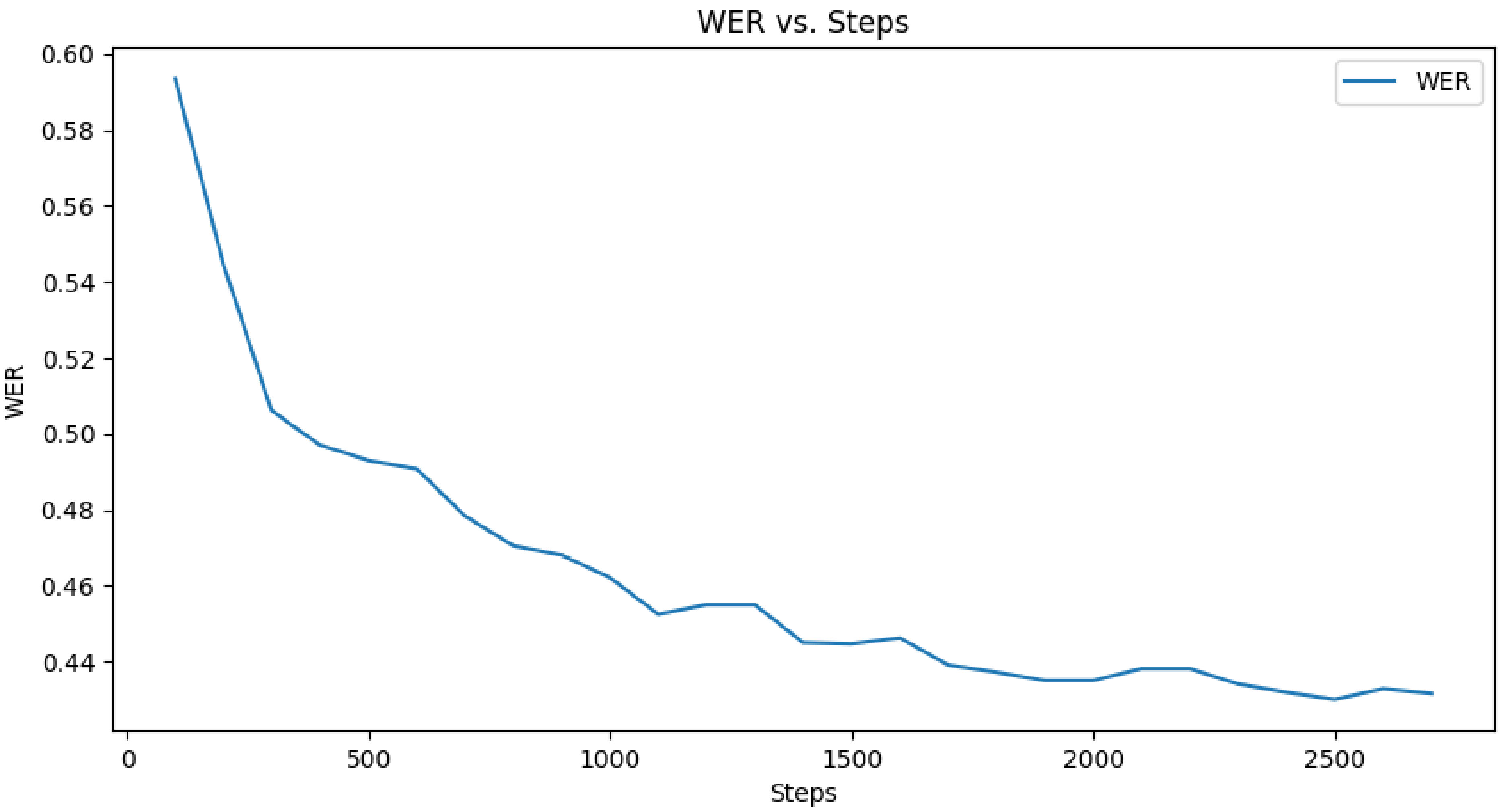


# Results and Analysis - [2]



# Results and Analysis - [3]

# Results and Analysis - [4]



# Result and Analysis - [5]
# (Top Error Rate on Validation Data)

| True Word | Predicted Word | Count |
|---|---|---|
| छ | र | 37 |
| र | छ | 31 |
| छ | पनि | 21 |
| छ | हो | 19 |
| हो | छ | 18 |
| छ | यो | 18 |
| हो | र | 18 |
| र | पनि | 17 |
| छ | रहेको | 16 |
| यो | र | 16 |

Validation Dataset = 10% of Training Set

= 264 Audio Samples

Average WER = 0.422

# Result - [6] (Top Error Rate on Test Dataset)

| Word | Confused With | Count |
|---|---|---|
| विधायन | विधान | 5 |
| र | <del> | 4 |
| संशोधन | संसोधन | 3 |
| <ins> | हाम्रो | 3 |
| अध्यक्षज्यू | अध्यक्ष | 3 |
| चाहन्छु | चाहन्छ | 3 |
| यो | य | 3 |
| हुनेमाननीय | हुने | 3 |
| सदस्यहरूले | माननीय | 3 |
| सम्माननीय | सम्माननीय | 3 |

Total Test Audios:

625

Average WER: 0.28

Substituted: 2414

Inserted: 47

Deleted: 59

# Discussion and Conclusion

- Achieved WER of 0.422 on Validation set and 0.2824 on Test set.
- Consistent decrease in WER suggests wav2vec 2.0 as a good model for ASR.
- On self-recorded audio, model could not produce satisfactory results.
- Model showed signs of overfitting, as validation loss showed fluctuations over number of steps.
- Results suggest on the need for adjusting learning rates and applying regularization techniques to reduce overfitting.

# Remaining Tasks

- Increasing Accuracy in wav2vec 2.0
- Finetuning mBART
- Speaker Encoder
- Finetuning FastSpeech 2
- Concatenation of TTS with Speaker Embedding from Speaker Encoder
- Creating a pipeline.

# References

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” *arXiv.org*, Jun. 20, 2020. https://arxiv.org/abs/2006.11477

[2] Y. Liu *et al.*, “Multilingual Denoising Pre-training for Neural Machine Translation,” *arXiv.org*, Jan. 22, 2020. https://arxiv.org/abs/2001.08210 (accessed Jul. 18, 2024).

[3] Y. Ren *et al.*, “FastSpeech 2: Fast and High-Quality End-to-End Text to Speech,” *arXiv.org*, Jun. 08, 2020. https://arxiv.org/abs/2006.04558