A SPEECH-TO-TEXT ENABLED CHATBOT USING TRANSFORMER MODEL

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY

MINOR PROJECT

CEN-792

Supervisor :

Dr Faiyaz Ahmad

Presented By:

Faizan Ahamad 21BCS047

Mantasha Firdous 21BCS049

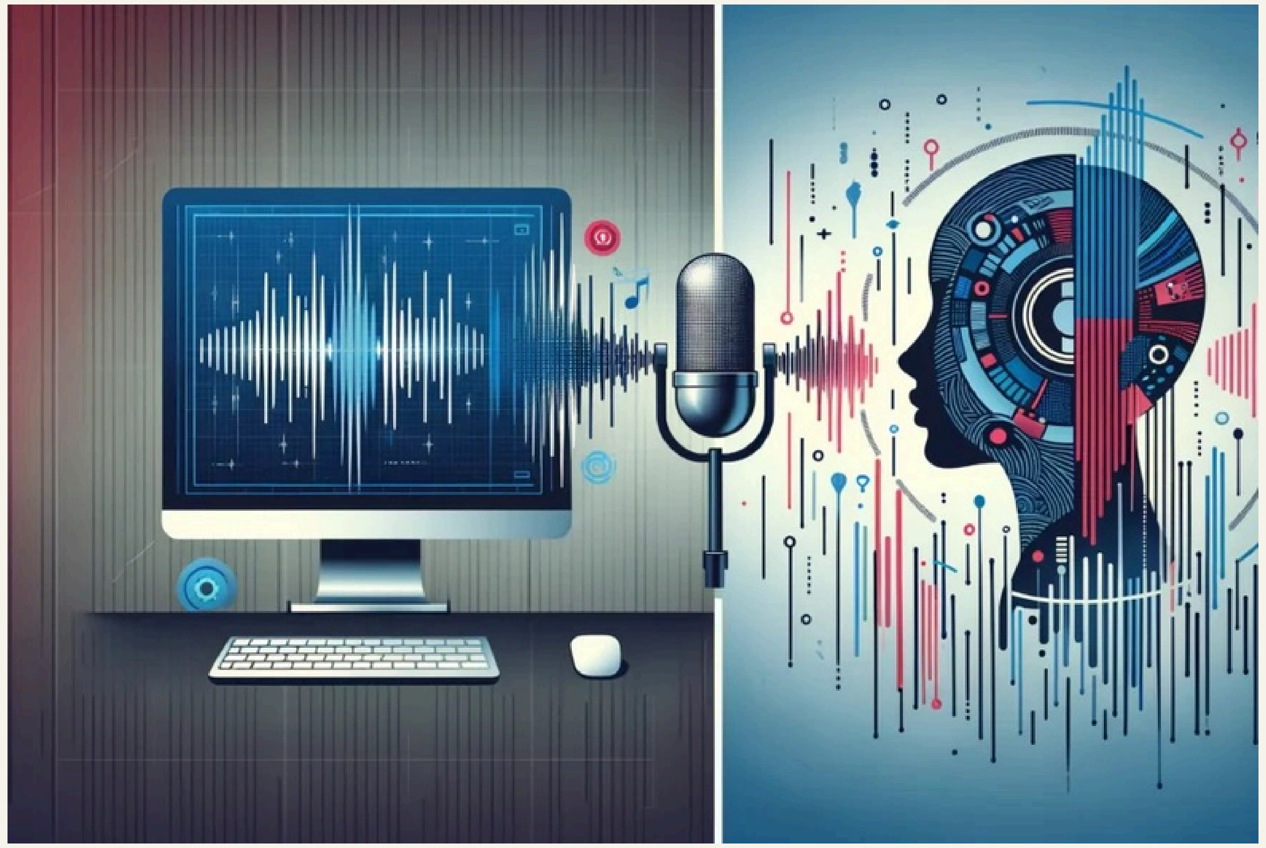
Jamia MIllia Islamia| 2024

CONTENT

- Introduction
- Problem Statement
- Literary Review
- Research Gap
- Methodology
- Programming environment
- Work Flow
- Architecture
- Dataset
- Evaluation Metrics
- Results
- What's next
- References

INTRODUCTION

- With the development of machine learning and deep learning algorithms, automated voice recognition has become a major study area.
- The use of online communication has resulted in a rapid rise in audio and visual information.
- It has been beneficial for the majority of people, those with special needs, such as the deaf, have few resources at their disposal.
- A speech-to-text Conversion programme is written.



PROBLEM STATEMENT

- By integrating the transformer architecture into STT systems, the project aims to develop a more responsive and reliable solution, paving the way for enhanced voice-driven applications and more natural human-computer interactions.
- These systems are vital for applications like virtual assistants and transcription services. They enhance user experiences by accurately converting spoken words into text, enabling natural communication with technology.

Literary review

Authors	Paper Name & Publisher	Model/ Architecture	Datasets	Remarks
Yunpeng Liu, Xukui Yang, Dan Qu	Exploration of Whisper Fine-tuning Strategies for Low-resource ASR Springer Nature (2024)	Whisper (OpenAI ASR model), Fine-tuned using various strategies	**Fleurs dataset (languages: Afrikaans, Belarusian, Icelandic, Kazakh, Marathi, Nepali, Swahili)**	This study explores fine-tuning strategies for Whisper in low-resource ASR tasks, including vanilla fine-tuning, specific parameter tuning, and additional modules. Fine-tuning improves performance, but different strategies have trade-offs.

Literary review

Authors	Paper Name & Publisher	Model/ Architecture	Datasets	Remarks
Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, Ilya Sutskever	Robust Speech Recognition via Large-Scale Weak Supervision. Published by OpenAI (2022)	Whisper Model: Uses large-scale weak supervision	**Diverse internet-based dataset covering 680,000 hours across multiple languages and environments**	Highlights the effectiveness of scaling weakly supervised pre-training for ASR, demonstrating robustness in multilingual and multitask settings without fine-tuning

Literary review

Authors	Paper Name & Publisher	Model/ Architecture	Datasets	Remarks
Ashish Vaswani, Noam Shazeer, Niki Parmar, etol.	Attention is All You Need IEEE (2017)	Transformer Architecture: Introduced self-attention and multi-head attention mechanisms, eliminating the need for recurrent networks in sequence tasks	WMT 2014 English-to-German, WMT 2014 English-to-French datasets	This groundbreaking paper introduced the Transformer model, revolutionizing NLP by significantly improving training efficiency and parallelization for sequence tasks. It forms the foundation for many modern models like BERT, GPT, and BERT

RESEARCH GAP

1. Current transformer-based models are often trained on general-purpose datasets, which may not perform optimally in specific domains. Developing domain-specific STT systems using transformers that are fine-tuned on specialized datasets could significantly improve accuracy and usability in these fields.

2. Transformer models tend to struggle in noisy environments or with speech that includes significant background noise. Developing robust STT systems that incorporate noise reduction techniques or that are trained on noisy datasets can improve performance in real-world conditions.

WORKFLOW

Setup and Dependencies

- Mount Google Drive
- Download the dataset from a URL
- Install necessary libraries

Data Loading and Preprocessing

- Load the unzipped dataset and prepare it for fine-tuning.

Model Setup

- Load the Whisper model architecture and define the configuration for fine-tuning.
- Set up Seq2Seq TrainingArguments

Training the Model:

- Initialize a trainer using Seq2SeqTrainer, passing in the model, dataset, and training arguments.
- Start the training loop

Evaluation and Logging:

- Use metrics such as Word Error Rate (WER) to evaluate the model's performance during training.
- Log results to Weights & Biases (wandb) for monitoring.

Saving and Exporting the Model:

- After training, load the model and save it to a specified directory for later use.

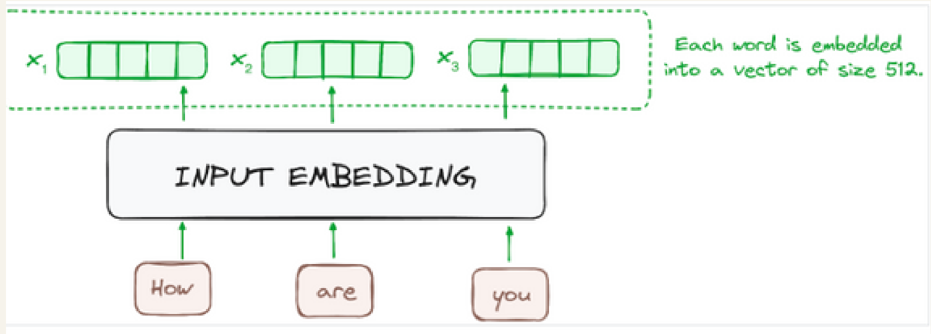
STAGES OF AUDIO PROCESSING

- **Raw Audio Input:** Audio input starts as raw waveform data (.wav/.mp3). Resampling to 16 kHz for consistency.
- **Feature Extraction:** Converts audio to log-mel spectrograms representing frequency distribution
- **Input Encoding:** Tokenizes, adds position embeddings, applies padding and masking.
- **Model Processing:** Passes through transformer encoder layers to capture temporal dependencies
- **Decoding:** The decoder predicts the text sequence in an autoregressive manner, generating one token at a time based on previously generated tokens and the encoder's output.
- **Post-processing:** Final text is cleaned by removing unnecessary tokens or padding.

TRANSFORMER ARCHITECTURE

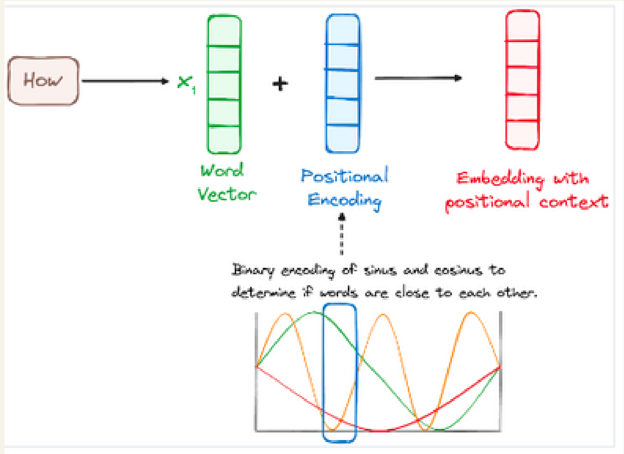
The Encoder workflow:

STEP 1:



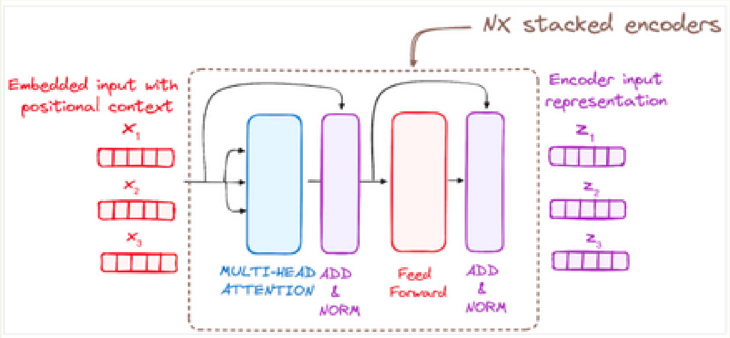
INPUT EMBEDDINGS

STEP 2:



POSITIONAL ENCODINGS

STEP 3:



STACK OF ENCODED LAYERS

STEP 4:

a set of vectors

OUTPUT OF ENCODER

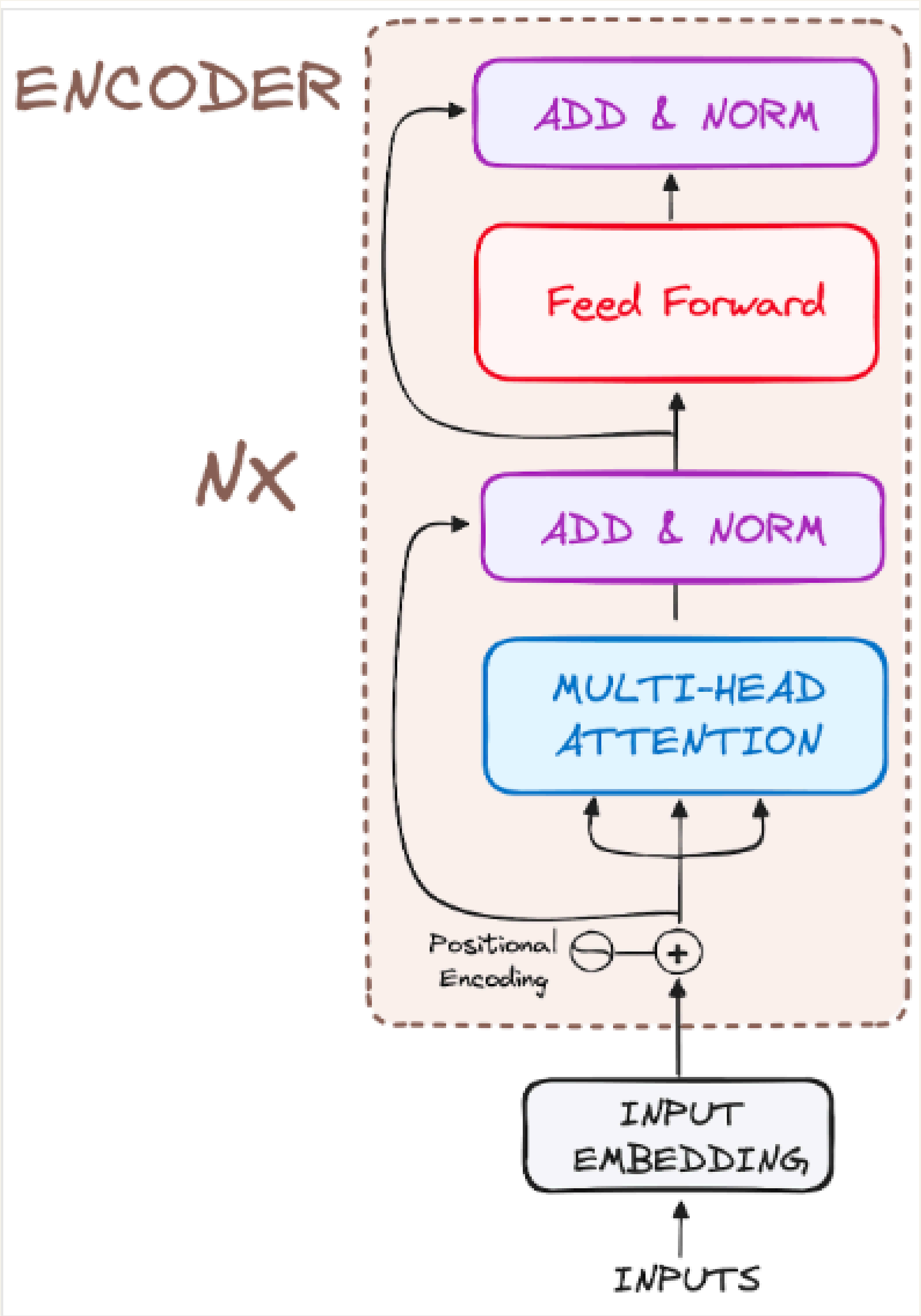


Fig: Encoder architecture

TRANSFORMER ARCHITECTURE

The Decoder workflow:

STEP 1: OUTPUT EMBEDDINGS

↓

STEP 2: POSITIONAL ENCODING

↓

STEP 3: STACK OF DECODED LAYERS

↓

STEP 4: OUTPUT OF DECODER

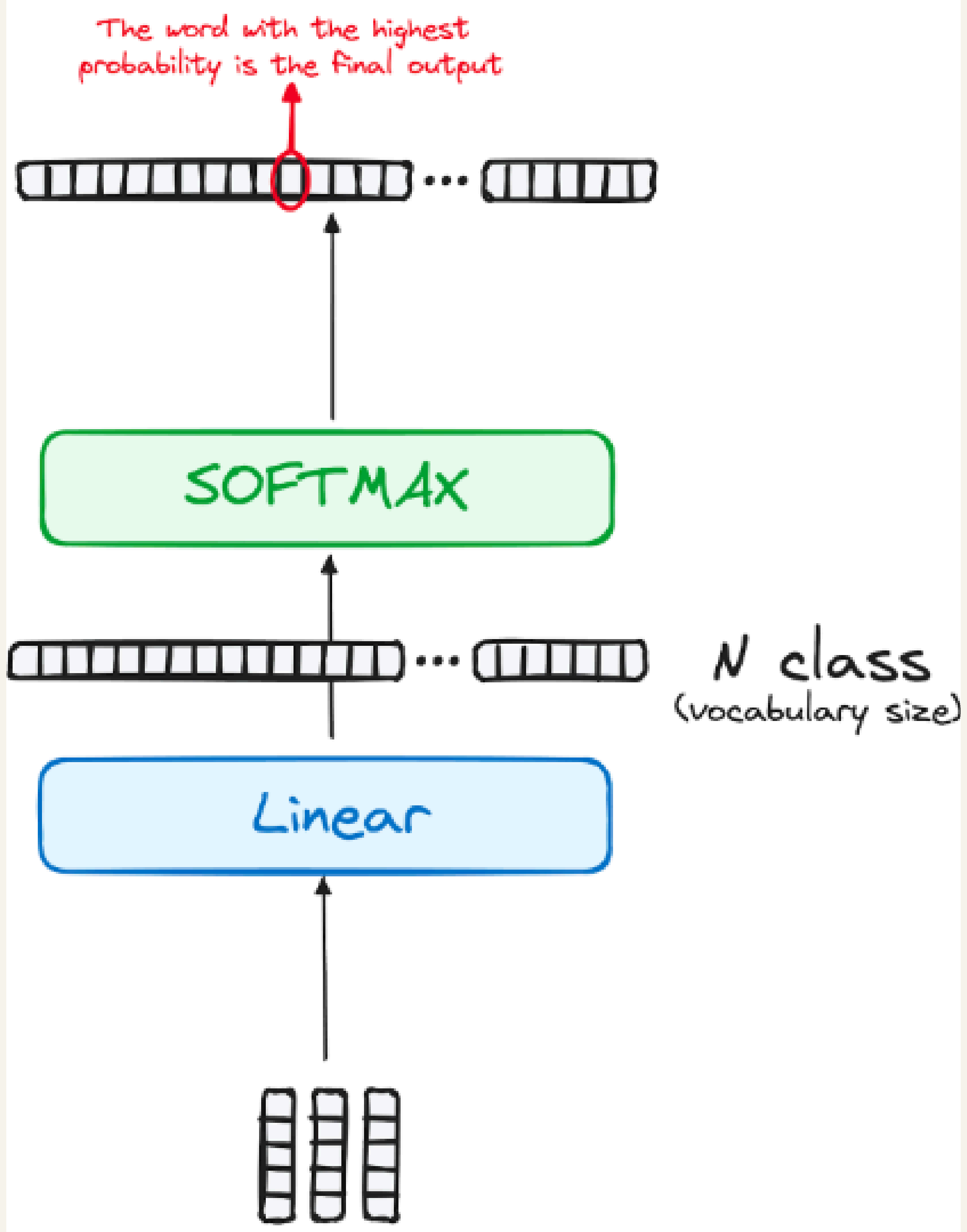


Fig: Transformer's Final Output

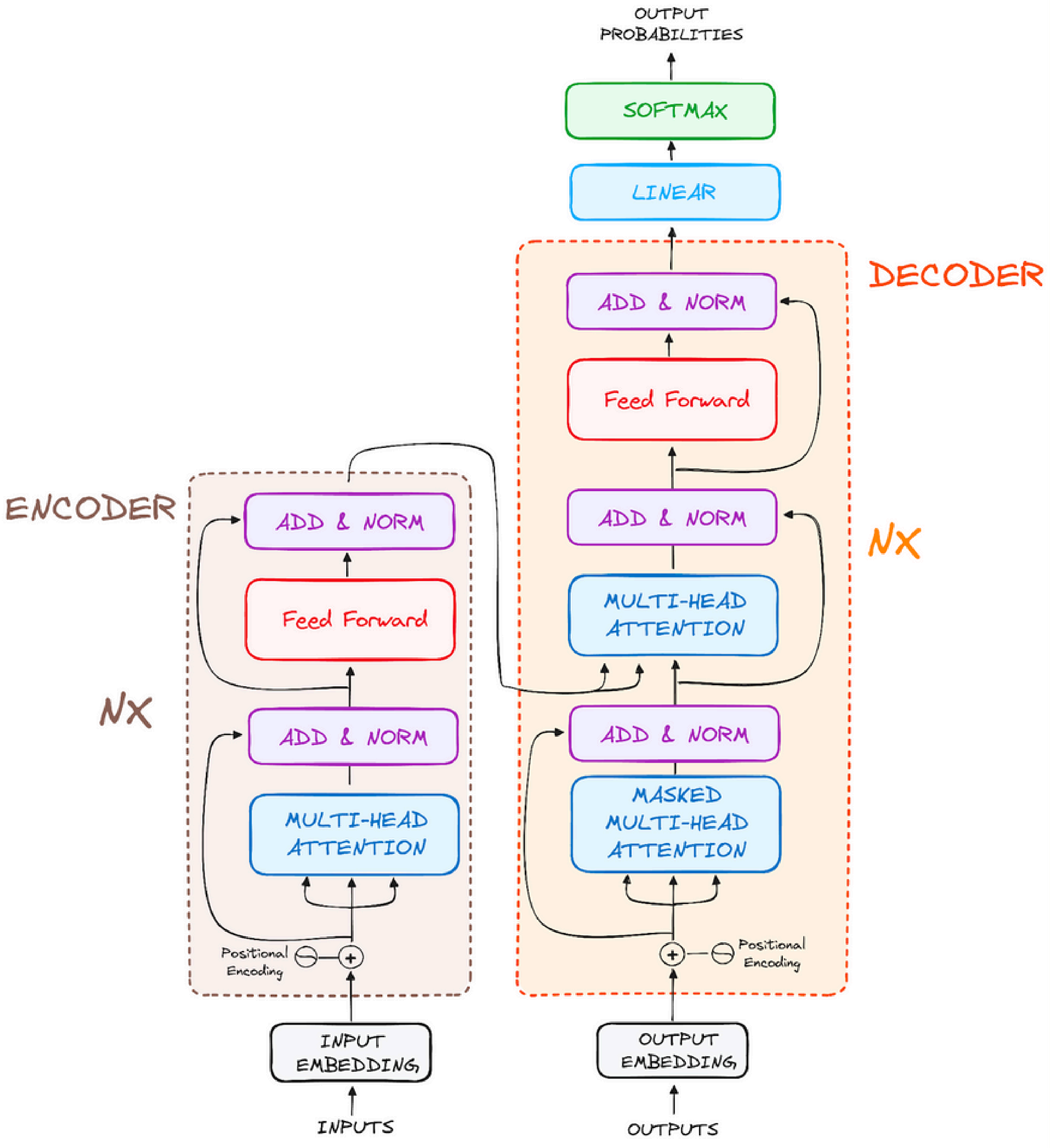


Fig: Transformer architecture

PROGRAMMING ENVIRONMENT

- **Platform:** Google Colab (Python 3.11) with GPU support
- **Dataset Source:** Kaggle
- **Libraries:** PyTorch, torchaudio, Hugging Face Transformers, and Datasets
- **Experiment Tracking:** wandb, integrated with Google Colab
- **Visualization:** matplotlib, seaborn

DATASET USED

- **Medical Speech, Transcription, and Intent[5]**
  - Total Audio Clips: 6,661
  - Total Duration: Approximately 8.5 hours of audio
  - Symptom Categories: 25 distinct medical symptom types, covering a wide variety of medical scenarios
  - Audio Format: All files are in WAV format
  - Transcriptions: Each audio clip has an associated transcription for accurate speech-to-text training
  - Intent Annotations: In addition to transcriptions, each audio clip is labeled with intent, making this dataset suitable for intent classification tasks in medical speech applications

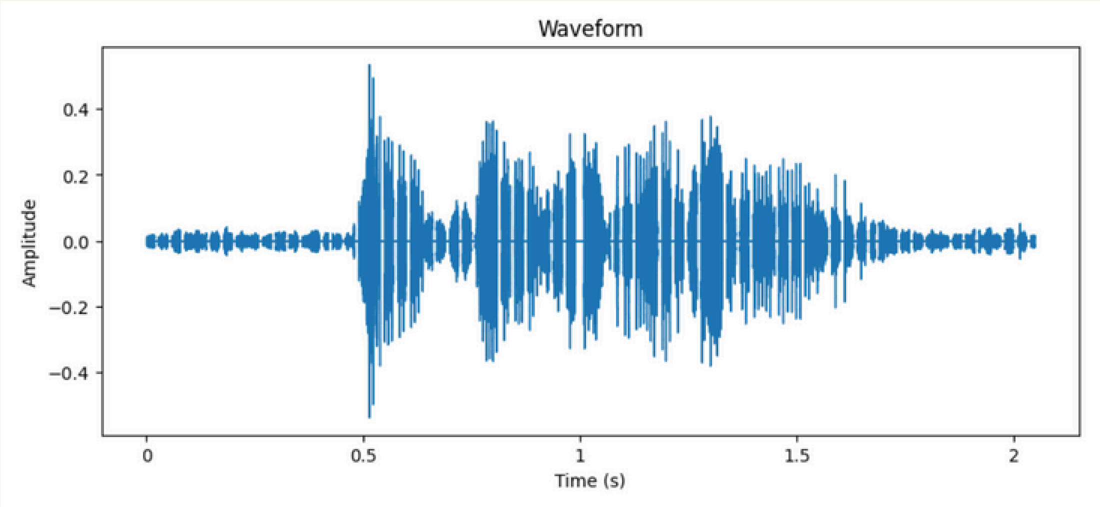
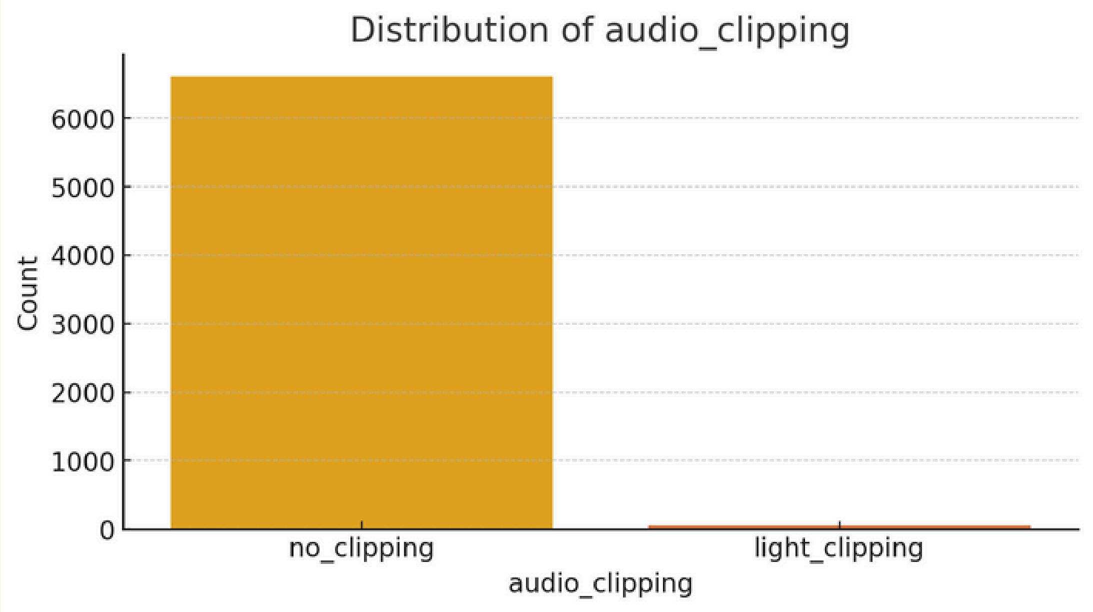
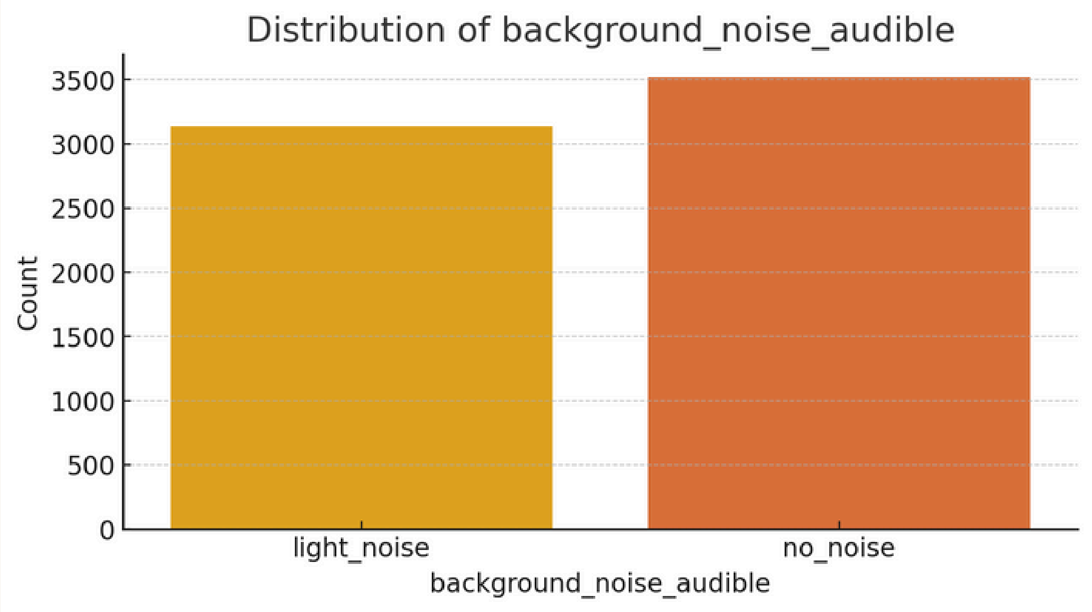
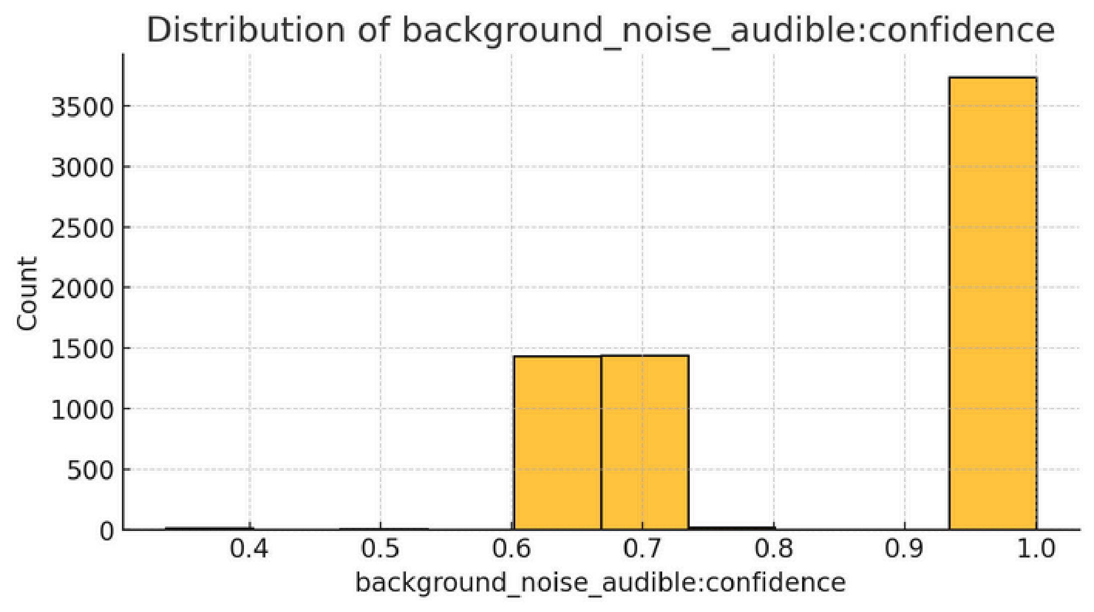


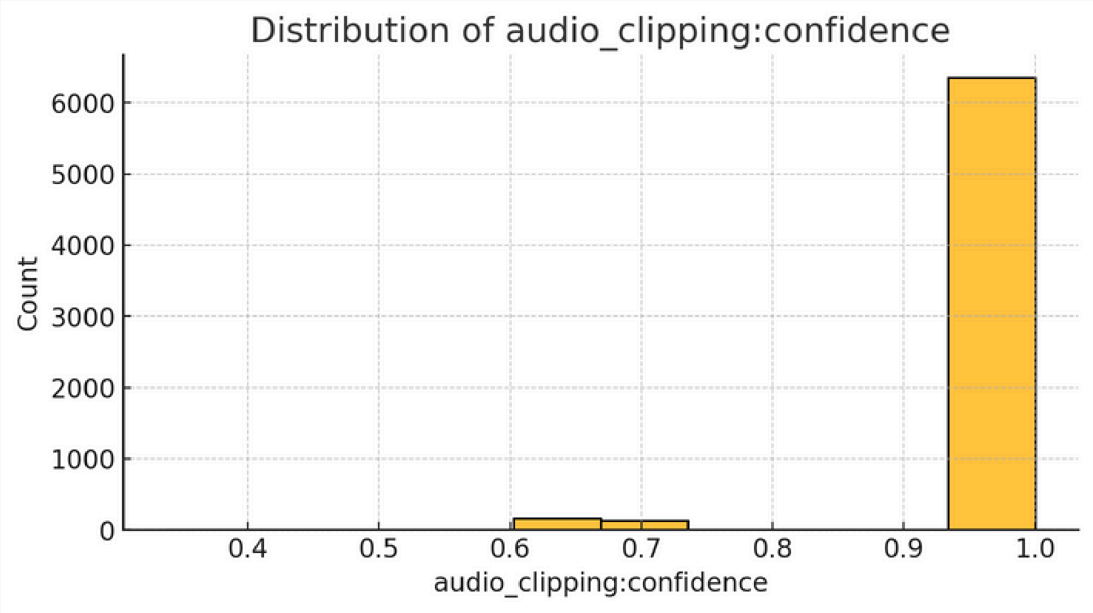
Fig: Sample of Audio from Test Data

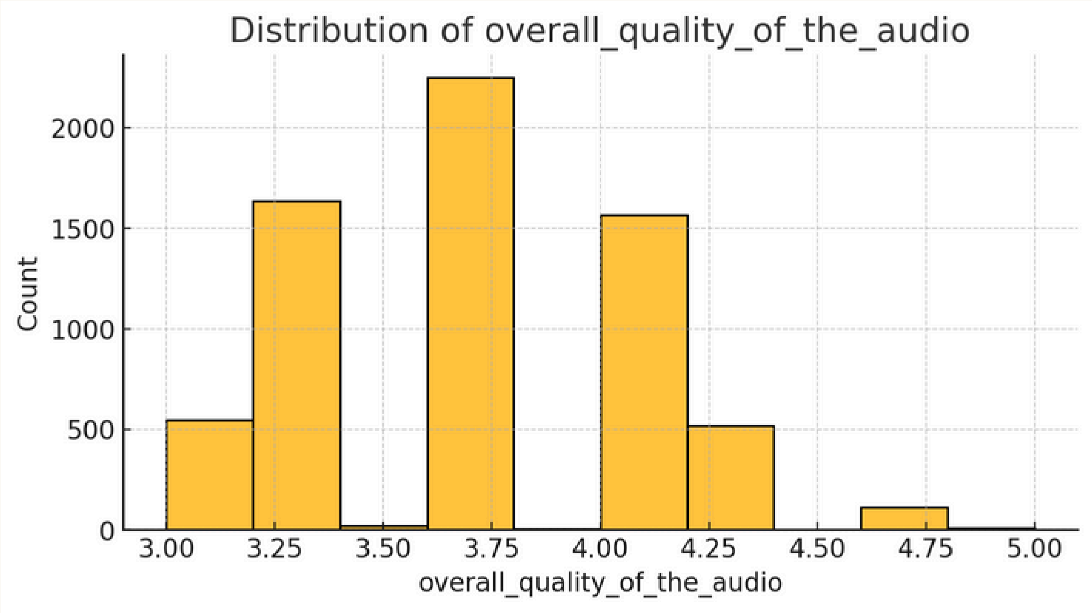
DATASET OVERVIEW

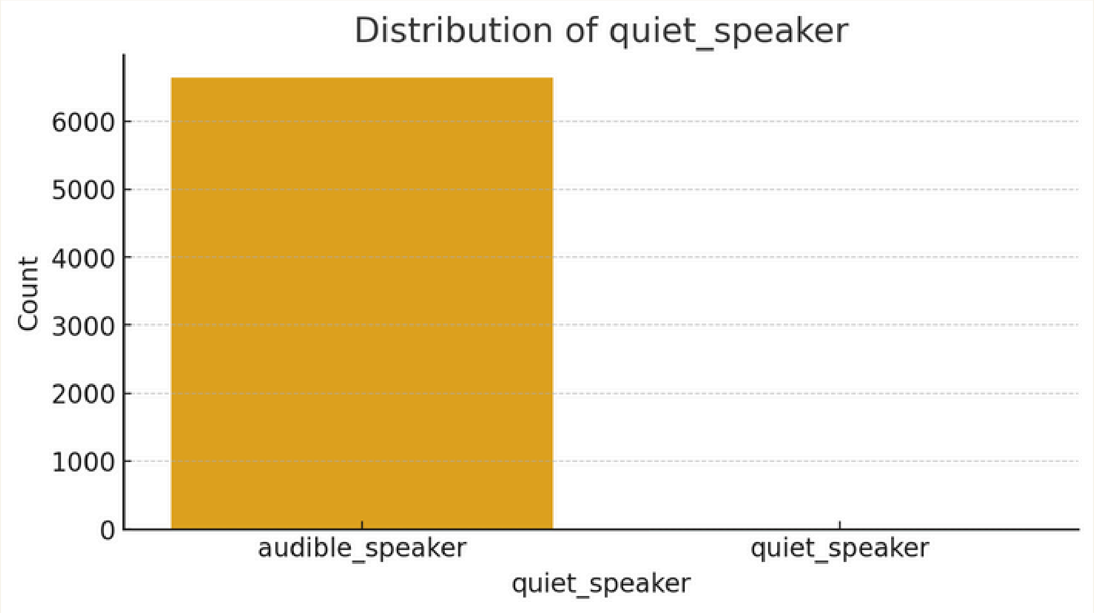












EVALUATION METRICES

WORD ERROR RATE (WER):

WER is calculated as the ratio of the minimum number of operations required to transform the model's predicted words into the reference words (ground truth) divided by the total number of words in the reference. The operations include insertions, deletions, and substitutions of words.
Lower WER indicate better ASR model performance.

The formula for WER is:

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

Where:

- **S**: Number of substitutions (incorrect words)
- **D**: Number of deletions (words in the reference that are missing in the prediction)
- **I**: Number of insertions (extra words in the prediction that are not in the reference)
- **N**: Total number of words in the reference sentence

EVALUATION METRICES

CHARACTER ERROR RATE (CER)

CER is similar to WER but works at the character level instead of the word level. It calculates the ratio of character-level edits required to transform the predicted transcription into the reference transcription, divided by the total number of characters in the reference.
Lower CER indicate better ASR model performance.

The formula for CER is:

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

Where:

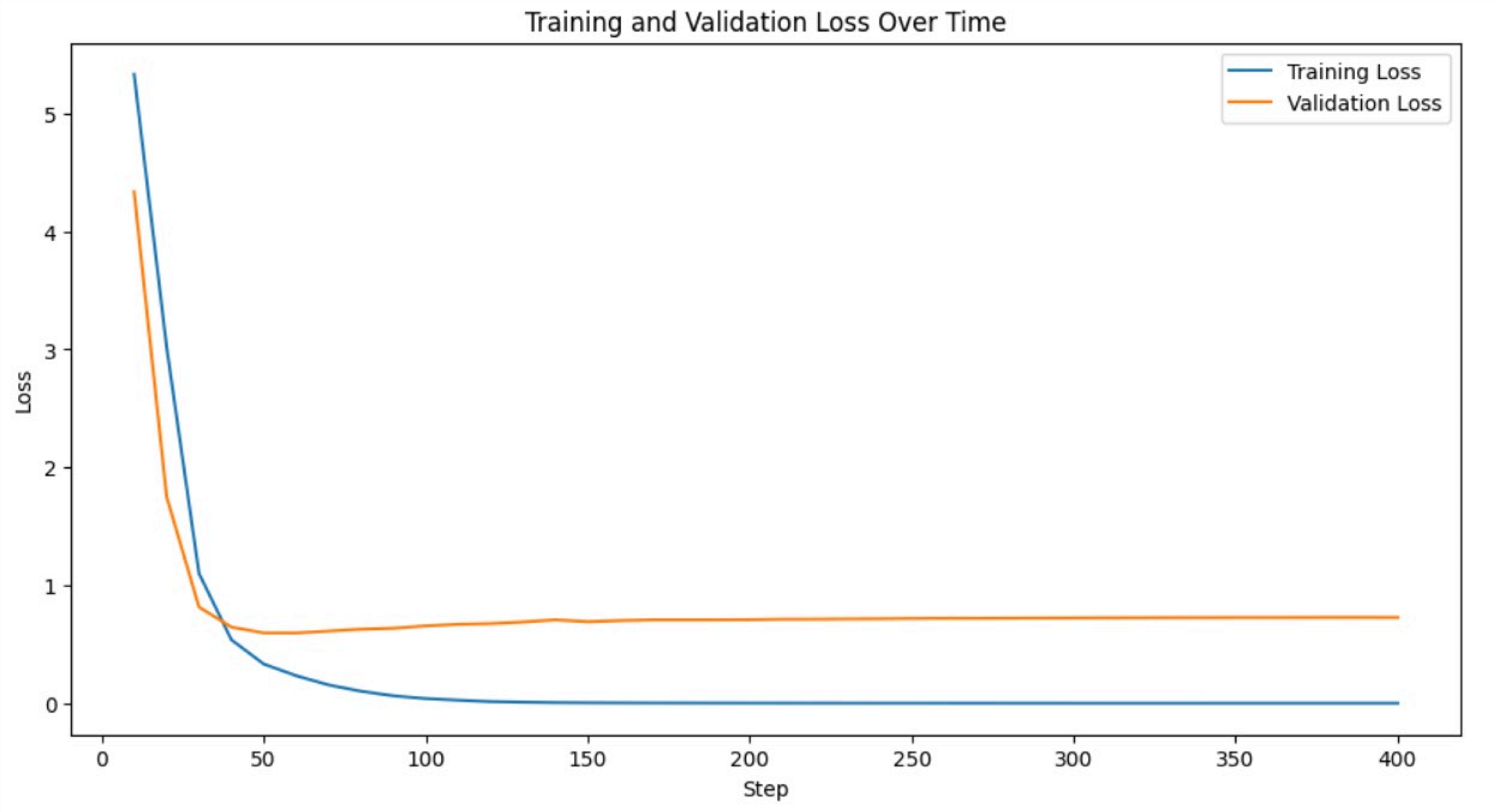
- S: Number of character substitutions
- D: Number of character deletions
- I: Number of character insertions
- N: Total number of characters in the reference sentence

RESULTS

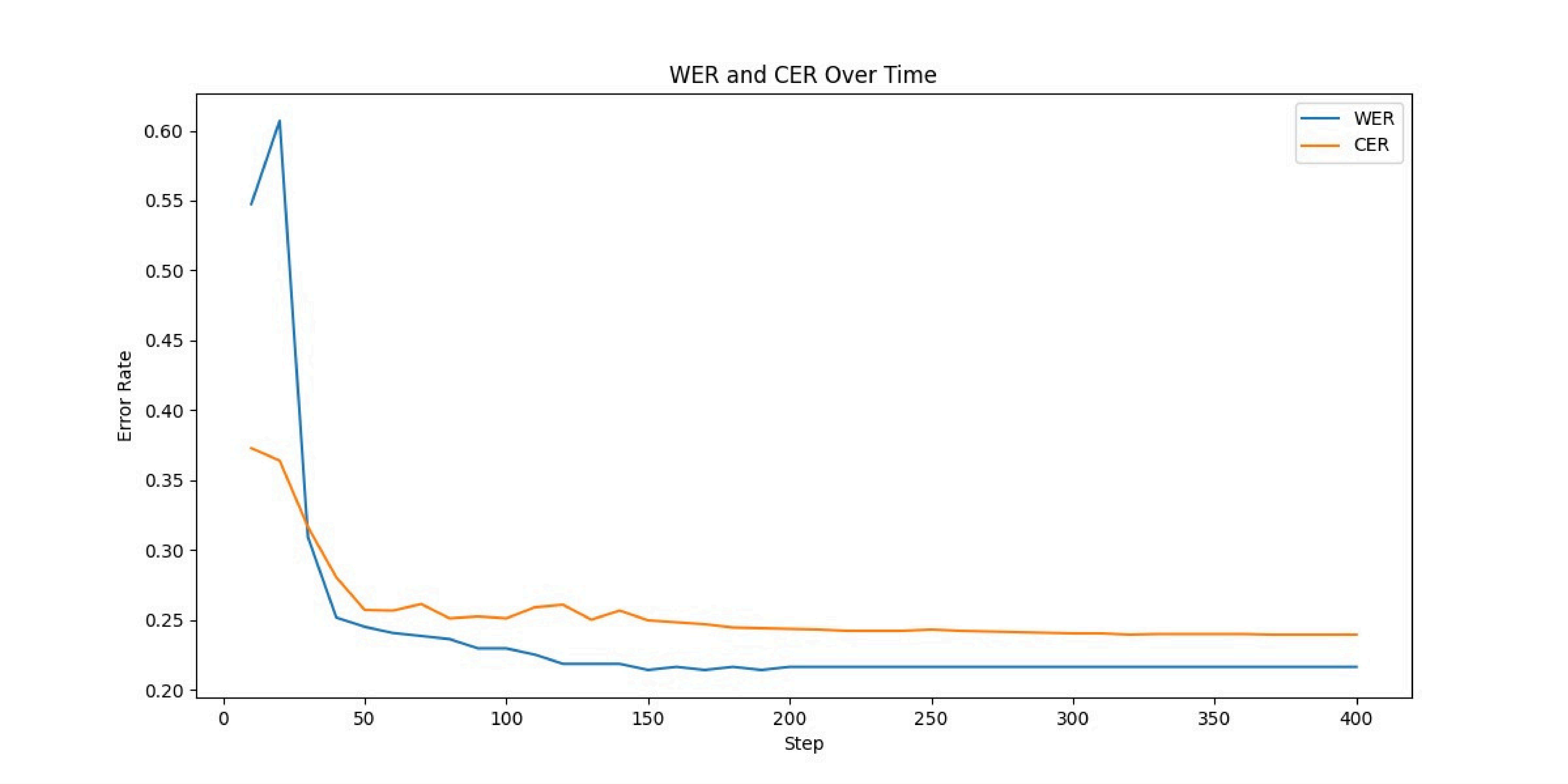
Step	Training Loss	Validation Loss	Wer	Cer	Accuracy
10	5.331300	4.335788	0.689038	0.606005	0.651361
20	3.020900	1.751867	0.626398	0.554734	0.715986
30	1.101200	0.815523	0.387025	0.522864	0.823129
40	0.538600	0.646439	0.295302	0.502079	0.858844
50	0.332400	0.596712	0.275168	0.484527	0.867347
60	0.233900	0.596296	0.272931	0.478522	0.863946
70	0.156200	0.612399	0.270694	0.474365	0.863946
80	0.102500	0.627885	0.261745	0.488222	0.867347
90	0.063600	0.636856	0.268456	0.501155	0.858844
100	0.040000	0.655986	0.270694	0.495612	0.853741

"Fig: Training and Validation Loss with WER, CER, and Accuracy over Fine-tuning Steps"

RESULTS


Training and Validation Loss Over Time
Training Loss
Validation Loss
Loss
Step
0
1
2
3
4
5
0
50
100
150
200
250
300
350
400

RESULTS



WER and CER Over Time
WER
CER
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Error Rate
0
50
100
150
200
250
300
350
400
Step

RESULTS


Accuracy Over Training Steps
Accuracy
0.85
0.80
0.75
0.70
0.65
0
50
100
150
200
250
300
350
400
Step
Accuracy

WHAT'S NEXT

- **Chatbot Integration:** Integrate the ASR model with a chatbot to provide interactive, conversational experiences. This will enable users to engage with the model's transcriptions and functionalities through natural language interactions.
- **User Interface Development**: Create a user-friendly interface that allows users to access the ASR functionalities seamlessly. This includes designing a clean, intuitive dashboard and controls for uploading audio files, viewing transcriptions, and interacting with the chatbot.

REFERENCES

[1]Robust Speech Recognition via Large-Scale Weak Supervision.

- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. OpenAI. Retrieved from https://cdn.openai.com/papers/whisper.pdf

[2]Exploration of Whisper Fine-tuning Strategies for Low-resource ASR

- Liu, Y., Yang, X., & Qu, D. (2024). Exploration of Whisper fine-tuning strategies for low-resource ASR. EURASIP Journal on Audio, Speech, and Music Processing, 2024(29). https://doi.org/10.1186/s13636-024-00349-3

[3]Attention is All You Need

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30 (NeurIPS 2017).

REFERENCES

[4] https://www.datacamp.com/tutorial/how-transformers-work

[5] Dataset: https://www.kaggle.com/datasets/paultimothymooney/medical-speech-transcription-and-intent

Jamia MIllia Islamia| 2024

THANK YOU