Whisper

Roberto Gonzales Matos, Julie Chung, Alison Zeng, Anant Saraf

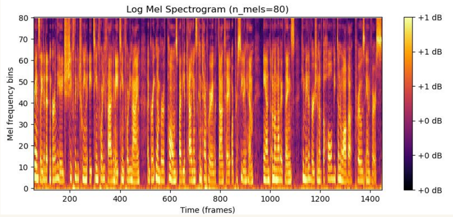
BROWN

Introduction

OpenAI's Whisper is an encoder-decoder based speech transcription model, trained for general-purpose transcription across a wide and diverse set of datasets. For our reimplementation, we aimed to reimplement a smaller, English-only version of the Whisper model in Tensorflow instead of Pytorch and optimize the system for training under limited data and computational resource constraints. We remapped the tokenizer to a reduced English vocabulary and fully reimplemented Whisper's encoder-decoder transformer architecture in TensorFlow, adapting it to suit smaller-scale training scenarios. Our work focused particularly on low-resource conditions and small-data performance, aiming to maintain transcription quality despite restricted input diversity and volume. This approach enables the development of lightweight speech recognition models suitable for environments with limited computational capacity. Furthermore, it makes speech-to-text technologies more accessible for smaller datasets, niche domains, and low-resource languages. Finally, by reducing the model complexity, this reimplementation creates new opportunities for custom fine-tuning and faster, more efficient deployment.

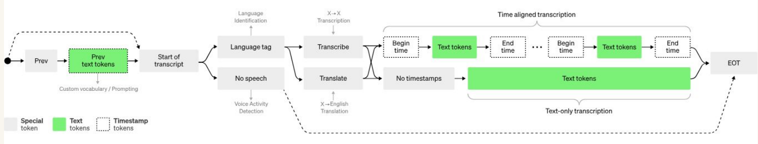
Data & Preprocessing

We used the LibriSpeech dev-clean subset for training and evaluation. This dataset contains approximately 5.4 hours of English audiobook recordings, featuring clean, high-quality speech with minimal background noise and clear articulation. It provides an ideal baseline for evaluating transcription accuracy under controlled conditions. Audio recordings were resampled to 16 kHz and converted into log-Mel spectrograms. Inputs were standardized to 30-second segments, allowing for compact, consistent feature representations while preserving key frequency and temporal information. This preprocessing simplified the input and enabled efficient training through convolutional operations on the spectrograms. However, we are aware that reliance on the dev-clean subset introduces bias. Because the recordings are scripted, noise-free, and drawn from a narrow speaker demographic, models trained on this data may struggle to generalize to real-world, spontaneous, or accented speech.



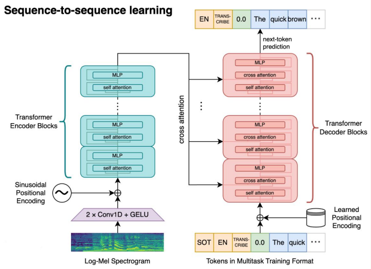
Architecture

The architecture of our Whisper reimplementation follows an encoder-decoder Transformer structure. The audio input is first converted into a log-Mel spectrogram, and the transcriptions are tokenized into small segments (tokens). Special tokens such as Start-of-Transcript (SOT) and End-of-Text (EOT) are used to manage the transcription process. These tokens are aligned with the audio frames and processed by the model to generate transcriptions.



The AudioEncoder takes the log-Mel spectrogram as input, applying two 1D convolutional layers with GELU activation. It then adds sinusoidal positional embeddings to the sequence and passes the data through a series of Residual Attention Blocks, which consist of self-attention and MLP layers. This results in the output of encoded audio features.

The TextDecoder then takes in these encoded audio features alongside input token sequences as inputs. It applies learned token embeddings and learned positional embeddings to the input. The decoder uses causal self-attention (masked) to process past tokens and applies cross-attention over the audio features to condition the text generation. Finally, the data is passed through another stack of Residual Attention Blocks, producing logits for the next token prediction.


Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
...
next-token prediction
MLP
cross attention
self attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
...
Tokens in Multitask Training Format

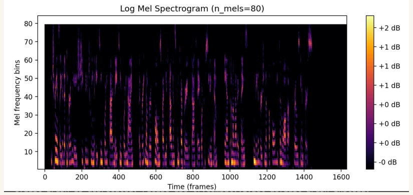
Challenges

A central challenge was the overfitting of the decoder and underfitting of the encoder. Early in training, the model often prioritized memorized transcript fragments over attending to the input spectrograms, limiting its ability to generalize. Decoder attention to encoder outputs was weak, which we partially mitigated through teacher forcing to guide better alignment between input features and predictions.

In the figure below, our original spectrograms were reducing low-energy regions, which resulted in a lot of dead space within the features. To improve spectrogram quality, we refined the normalization and scaling process to preserve these regions, allowing quieter sounds to remain visible in the log-Mel spectrograms, as seen in the figure. These adjustments enhanced feature extraction and contributed to more stable model training.

Training efficiency was also hindered by the original vocabulary size. Whisper's default ~100,000-token vocabulary increased model size, slowed training, and added unnecessary complexity. We reduced the vocabulary by nearly half through retokenization, resulting in faster convergence and improved transcription accuracy. Finally, the limited scale of our dataset posed a fundamental constraint. While OpenAI's Whisper was trained on approximately 680,000 hours of diverse audio, our model was trained on only 5.4 hours of clean, scripted audiobook recordings. This several-orders-of-magnitude difference restricted model robustness and generalization to broader speech conditions.

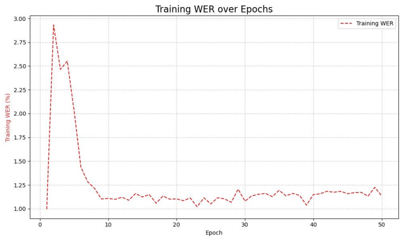
Ground Truth: MORREL SUFFERED AN EXCLAMATION OF HORROR AND SURPRISE TO ESCAPE HIM
Transcription: SOME TIME NOTHING WAS HEARD IN THAT CHAMBER BUT SOBS EXCL AMATIONS AND PRAYERS NOTATH CHAMBERBS EXBSBS I NOTAMAMAM BUT SOAMBS EXATION ANDAM SECAM THEBSREAMATIONSA MBS THEBS EXATIONSBSBSAMAMAM BUT THBSBSBSBSBS EXEDAMAMAMEDAMBS EXBSATIONSBSBSAMAMAMBSAM HANDEDAMAM THEBS BS ANDAMBSBSBSBSBS EXCLAMAMAMEDED THEBSBSBS EXATIONSBSBSBSBSBSBSAMBSBSBSBS

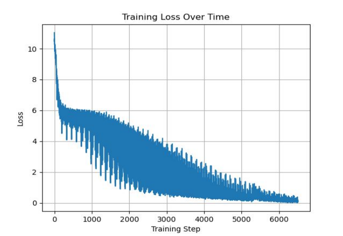


Results

We were able to realistically train from 50-100 epochs on the training data. As the model trained, the model learned rapidly in the first 10 epochs, with a steep decrease in our loss. After epoch 10, our loss continued to decrease, albeit more gradually as our model made more stable improvements. At the end of training, our loss was at 0.0586, effectively minimizing the loss. Our WER followed a similar trend, with WER being very high in the first few epochs, and then reaching a stable WER between 1 and 1.25.

Whereas OpenAI's Whisper was trained on 680,000 hours of large-scale data, we were limited to training on 5.4 hours of data due to computation and time limitations. However, we were able to successfully train our model on the dev-clean data, and after training, our model was able to accurately transcribe audio samples. We believe that training on more robust datasets such as TED-LIUM Release 3 or LibriSpeech ASR Corpus full training data (including noisy audio) would have improved the performance of the model and will continue to experiment.





Epoch 50/50
GT: THE OLD MAN'S EYES REMAINED FIXED ON THE DOOR<|endoftext|>
PR: <|startoftranscript|>THE OLD MAN'S EYES REMAINED FIXED ON THE DOOR<|endoftext|>
WER: 0.11