ICASSP 2026 Cadenza Challenge Technical Report

Team ID: T024, Authors: Austin Wagenknecht, Edward Wersocki

December 15, 2025

Abstract

The Whisper baseline published by the Candeza Team correlates well with lyrics intelligibility but can be refined by combining it with additional multimodal features. We employ a lightweight regression model that incorporates 1d, 2d, and scalar features. The feature extraction for our system requires the unprocessed signal, the processed signal, and the reference text. The system takes as input Whisper encoder embeddings, MFCCs, baseline STOI and Whisper scalar scores, and vocal-to-accompaniment ratio, summarizes each mode, and uses a small MLP to predict lyrics intelligibility from the feature projections. This combination of features improves the validation correlation by approximately 6.7% compared to the Whisper baseline.

1 Pre-processing and Feature Extraction

We utilized the scores from the STOI baseline and Whisper baseline, which were extracted without modification according to the details on the Cadenza Challenge website[3]. The STOI baseline uses the unprocessed signal for source separation with Hybrid Demucs[1] trained on the MUSDB-HQ[4] dataset. The Whisper baseline uses the processed signal and reference text. In addition to the global Whisper score, we use the Whisper encoder embeddings of the processed signal, which are vectors of size 512 per frame.

We computed a vocal-to-accompaniment ratio (VAR). The vocals and accompaniment are estimated from the processed audio signals using source separation with Hybrid Demucs and then RMS-normalized. The VAR feature is the ratio in decibels of the total energy in the estimated vocals to the total energy in the estimated accompaniment.

For the remaining features, we apply additional pre-processing. First, we applied a band-pass filter to each processed audio sample to restrict the signal to the frequency range we deemed relevant for the task. The filter passes content between 50 Hz and 15,000 Hz, removing energy below and above this band, which in our dataset does not contribute meaningful information for lyric intelligibility. A 4th-order Butterworth filter was used to obtain smooth band transitions. We then applied RMS normalization to ensure consistent loudness across audio samples. Since raw material varied in dynamic range and amplitude, unnormalized input could introduce unnecessary variance, making the network sensitive to loudness differences rather than acoustic-linguistic features relevant to intelligibility. RMS normalization enforces a stable reference level so the model learns from spectral content, not volume fluctuations. To further stabilize input levels, we applied dynamic range compression followed by peak normalization. While RMS normalization sets an overall loudness target, large transient peaks can still disproportionately influence feature extraction and model behavior. This compression stage prevents those peaks from dominating the signal and helps maintain a more uniform representation of audio intensity.

Following pre-processing, we extracted Mel-Frequency Cepstral Coefficients (MFCCs) with 13 coefficients per frame that provide a compact representation of the short-term power spectrum using a perceptually-spaced Mel scale. Rather than encoding raw frequency bins, MFCCs compress spectral information into coefficients aligned more closely with human hearing. Cepstral Mean and Variance Normalization (CMVN) was applied to reduce channel and loudness variance, ensuring more robust model generalization.

Spectral Centroids were extracted to provide a quantitative descriptor of brightness and high-frequency content, both of which are strongly linked to consonant detection. As a complement to the Spectral centroids, we also extracted spectral Rolloff, which represents how much of the spectral energy is concentrated at lower vs. higher frequencies. Ultimately, we did not see improvement by incorporating spectral centroid and rolloff in the model and chose to exclude them from the final submission.

2 Model Architecture

Left and right channel Whisper encoder embeddings, $x_{1d} \in \mathbf{R}^{512 \times t_{1d}}$, are processed independently. Sequences are padded to the maximum temporal length. We then apply masked mean pooling over time, linear projection to size 64 summary vector for each channel, and ReLU activation.

Cepstral mean and variance-normalized MFCCs, $x_{2d} \in \mathbf{R}^{2\times 13\times t_{2d}}$, are processed in stereo and padded to the maximum temporal length. We then apply mean pooling over frequency, masked mean pooling over time, linear projection to a size 32 summary vector, and ReLU activation.

The scalar inputs (STOI baseline score, Whisper baseline score, and z-score normalized VAR) are processed through a small MLP (Linear $\rightarrow$ ReLU) to obtain a size 32 embedding.

The projections from each branch are concatenated to a size 192 vector and passed through a final MLP (192 $\rightarrow$ 64 $\rightarrow$ 32 $\rightarrow$ 1). ReLU activation and dropout with probability 1% are applied between hidden layers. Finally, a sigmoid function outputs an intelligibility prediction in the range [0, 1].

We trained the model with a batch size of 8 and the SmoothL1Loss loss function with beta value 0.1. We found SmoothL1Loss helpful to improve training stability versus MSE loss. We used the Adam optimizer with learning rate 2e-5 and weight decay 2e-3. We monitored both SmoothL1Loss and Pearson correlation as a criterion for early stopping and found that the best score was typically achieved around epoch 25-30.

We employ several data augmentation strategies. We swap left/right channels on Whisper embeddings and MFCCs with probability of 50%. We add Gaussian noise to Whisper embeddings and MFCCs with magnitude 1% of the calculated per-sample standard deviation. We add 1% jitter to the scalar features and to the target scalars to account for noise in the human-annotated intelligibility labels.

The majority of compute costs for our system are in feature extraction, especially source separation with Demucs and Whisper ASR. The multimodal regression model contains approx. 80,000 trainable parameters, which is small compared to the Whisper base.en model (74M parameters)[2] used to produce the input to the model. Training for 50 epochs requires about 4 hours on a single NVIDIA GTX 1650. Inference requires about 12ms per audio sample.

3 Results

Our system achieved a validation RMSE of 28.68 and Correlation of 0.63 and an evaluation RMSE of 28.69 and Correlation of 0.61. This represents an improvement over the Whisper baseline of about 2.2% validation RMSE, 6.7% validation Correlation, 1.3% evaluation RMSE, and 5.2% evaluation Correlation. The addition of MFCC, STOI, and VAR features leads to higher correlation with lyrics intelligibility than the Whisper score alone, and the inclusion of Whisper embeddings allows the model to learn more deeply. We attempted adding complexity to the model, including adding spectral centroid/rolloff and temporal attention pooling. However, adding these features did not improve the model validation performance as shown in Table 1.

Features Used	**Validation RMSE**	**Validation Correlation**
Whisper baseline, STOI baseline, VAR, MFCC	29.25	0.60
Whisper baseline, STOI baseline, VAR, MFCC, Whisper embeddings [Final System]	28.68	0.63
Whisper baseline, STOI baseline, VAR, MFCC, Whisper embeddings, Spectral Centroid/Rolloff	29.20	0.62

Table 1: Ablation Results

References

[1] Alexandre Défossez. Hybrid spectrogram and waveform source separation. *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[2] OpenAI. Model card: Whisper. `https://github.com/openai/whisper/blob/main/model-card.md`. Accessed: 2025-12-01.

[3] The Cadenza Team. Baseline system. `https://cadenzachallenge.org/docs/clip1/baseline`. Accessed: 2025-12-01.

[4] Fabian-Robert Stöter Stylianos Ioannis Mimilakis Zafar Rafii, Antoine Liutkus and Rachel Bittner. Musdb18-hq - an uncompressed version of musdb18, December 2019.