Speech Major Exam

m23csa001

Code Repository

The complete codebase, datasets, and results for the denoising pipeline are available at the following GitHub repository:

`https://github.com/bahetiaditi/Speech_Major`

Question 1: Transcription and TTS in a Low-Resource Language

1. Introduction

This work focuses on building a speech processing pipeline that transcribes a lecture delivered in a code-switched format (English-Hindi), translates it into a low-resource language (Marathi), and generates speech in the my own voice using text-to-speech (TTS) synthesis. The overall objective is to simulate multilingual speech understanding and generation by integrating automatic speech recognition (ASR), translation, voice cloning, and TTS.

We selected the third lecture from the Speech Understanding course as our input. The lecture was first manually transcribed to serve as a ground truth reference. We then applied an automatic speech-to-text model to generate a machine transcription, which was subsequently translated into Marathi using a translation model. To produce natural-sounding output, we recorded a custom dataset in the my voice and used it to synthesize the Marathi translation via a cloned TTS pipeline. The final output is a complete Marathi audio version of the original bilingual lecture, spoken in the user's voice.

This showcases a realistic application of multilingual speech technology, particularly for low-resource languages, and evaluates the quality of both transcription and synthesized speech through objective metrics like Word Error Rate (WER), Character Error Rate (CER), and Perceptual Evaluation of Speech Quality (PESQ).

2. Dataset Description

For this project, we selected the third lecture from the Speech Understanding course, which was approximately 45 minutes in duration. The lecture was primarily in English, with occasional switches to Hindi, reflecting a natural code-switched discourse. The entire lecture was manually transcribed, resulting in a corpus of 6,880 tokens. A language distribution analysis showed that around 3.1% of the transcript tokens were in Hindi, while the remaining were in English. Figure 1 illustrates the language switching timeline across the lecture duration, and Figure 2 presents a word cloud highlighting the key concepts covered.

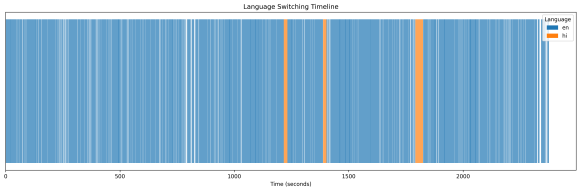

Figure 1: Language Switching Timeline in the Lecture

The transcription served as the input for our multilingual speech processing pipeline. English-Hindi code-switched text was automatically segmented and processed, followed by translation into Marathi, which served as the target low-resource language. To enable text-to-speech generation in the my own

Figure 2: Word Cloud of Lecture Content

voice, we prepared a custom Marathi voice dataset comprising 100 short samples (each approximately 3 seconds long) and 2 long-form recordings (each approximately 3 minutes), along with their corresponding text transcriptions. This voice dataset enabled speaker cloning and the generation of natural-sounding speech in Marathi using the translated transcript.

3. System Architecture

The complete system is organized into three core components: transcription, translation, and Marathi voice synthesis. These components interact through a modular architecture designed to handle code-switched input audio and produce a natural-sounding speech output in the user's voice.

The transcription pipeline begins with segmenting the input English-Hindi lecture using PyDub's silence detection. These segments are then transcribed using OpenAI's Whisper large-v3 model, which handles multilingual speech-to-text tasks efficiently. Post-transcription, the text is cleaned using NLTK to remove filler words, and word-level statistics are generated. Comparisons with manually transcribed references are carried out using BLEU, Word Error Rate (WER), and Character Error Rate (CER) metrics. Visualizations such as filler word trends, language switching timelines, and segment distributions are created using Matplotlib and Seaborn.

The next stage is translation. The cleaned transcript is translated from English and Hindi into Marathi using Meta's NLLB-200 model with 600 million parameters.The translation module pre-

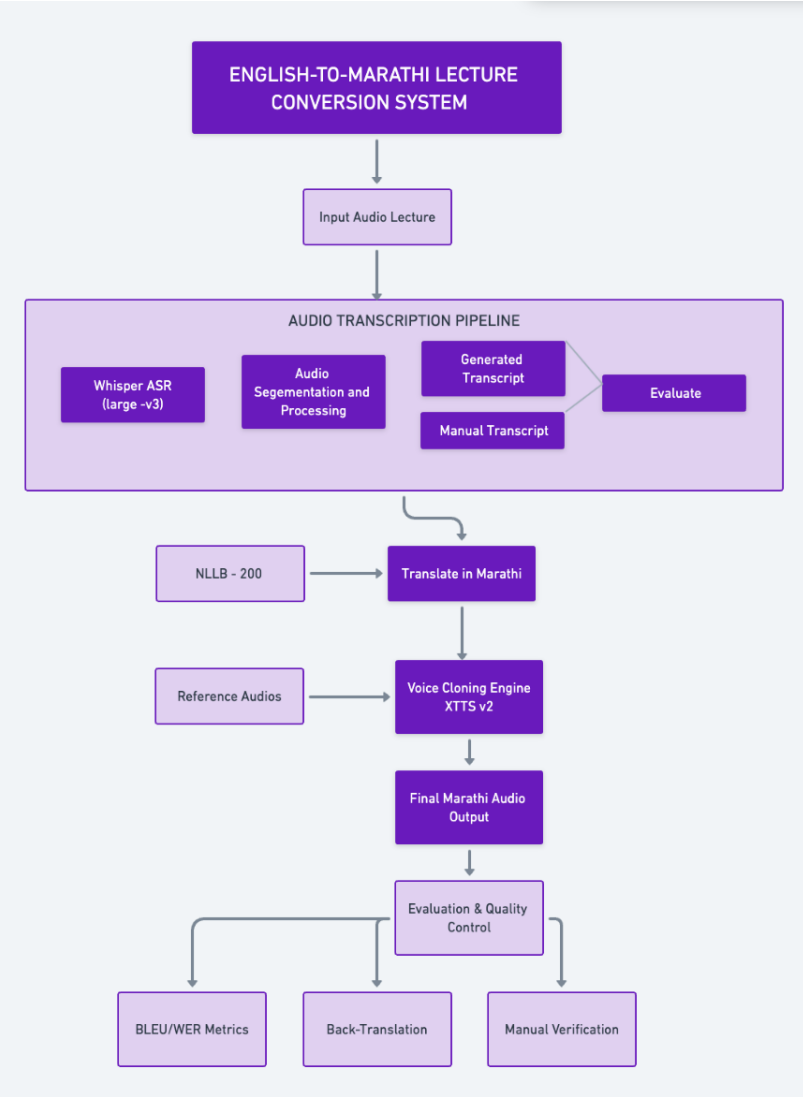

Figure 3: System Architecture Overview: From Audio to Personalized Marathi Voice Output

serves domain-specific technical terms and performs segment-wise translation, followed by grammatical adjustments. Quality of translation is assessed through back-translation using the same model and evaluated with metrics including BLEU, ChrF, METEOR, semantic similarity (via Sentence Transformers), and Jaccard similarity.

Finally, the Marathi voice synthesis is performed using Coqui's XTTS v2 model. A set of reference recordings from the user's voice—comprising both short (3-second) and long (3-minute) clips—is used to condition the model. The translated text is divided into naturally spoken chunks and synthesized into speech using the reference voice embeddings. The individual chunks are concatenated into a final continuous audio output. Audio quality is evaluated using the Perceptual Evaluation of Speech Quality (PESQ) metric and MOS-like estimations via SpeechBrain.

The complete architecture is shown in Figure 3, illustrating how each module flows into the next—from initial transcription to final personalized TTS generation with integrated quality evaluations.

4. Methodology and Tools Used

4.1 Preprocessing

The transcription output was cleaned to improve clarity and translation quality. We used a curated list of filler words (e.g., *uh, um, ahh, hmm*) and removed them using NLTK-based tokenization. Additionally, we implemented an n-gram based repetition filter to eliminate consecutive repeated phrases up to 3 words long (e.g., *"okay okay"* → *"okay"*, *"are you are you"* → *"are you"*). Punctuation was normalized and redundant whitespace removed. In total, **143 filler words** were removed, accounting for **2.14%** of the overall word count.

4.2 Transcription

We use the Whisper `large-v3` model for multilingual ASR, capable of handling Hindi-English code-switched content. Audio is segmented using silence detection (via PyDub), maintaining 15–30 second chunks. Each segment is transcribed and the detected language is recorded. Transcriptions are stored with timestamps, cleaned versions, and language metadata. Performance is evaluated against a manually created reference using BLEU, WER, and CER metrics. Visualizations (e.g., filler trends, language timeline) are generated using Matplotlib and Seaborn.

4.2 Translation

Translation to Marathi was done using the **NLLB-200 (600M)** model. Code-mixed segments were first language-identified and split sentence-wise. Technical terms like *"Machine Learning"*, *"Amplitude"*, etc., were preserved using placeholder substitution and restored post-translation. Back-translation and semantic similarity scoring (BLEU, ChrF, METEOR, Jaccard) were used to evaluate translation quality.

To understand content trends, a frequency-based Word Cloud was generated from the translated lecture.

Figure 4: Word Cloud of the Translated Marathi Lecture

4.4 Voice Cloning and TTS

To synthesize the translated Marathi text in the my voice, we implemented a personalized Text-to-Speech (TTS) pipeline using the **XTTS v2** model from Coqui. We curated a voice dataset consisting of a mixture of 12 reference recordings, including both long and short speech clips from my dataset of 100 files, to capture varied speech characteristics and ensure expressiveness.

The translated text was first cleaned to normalise punctuation and whitespace. It was then chunked into smaller segments of approximately 80–100 words for efficient GPU-based inference. Each chunk was synthesized using the XTTS model, conditioned on the user's reference voice samples. The generated chunks were concatenated with silence buffers to produce the final audio.

This process resulted in a complete Marathi audio version of the lecture, rendered in a voice that closely mimics the speaker. The same reference dataset was reused for consistent voice conditioning across all segments.

5. Evaluation and Results

5.2 Transcription Quality Evaluation

To evaluate the quality of the automatic transcription, we compared it against a manually prepared ground truth transcript using standard metrics: BLEU, Word Error Rate (WER), and Character Error Rate (CER). The results are as follows:

- **BLEU Score**: 0.7091
- **Word Error Rate (WER)**: 26.96%
- **Character Error Rate (CER)**: 13.43%
- **Auto Transcript Word Count**: 6502
- **Manual Transcript Word Count**: 6880

The **BLEU score** measures the n-gram overlap between the generated and reference text. A score of 0.7091 indicates strong similarity, suggesting that the automatic transcription preserved the structure and content of the original fairly well.

The **Word Error Rate (WER)** represents the proportion of words that were substituted, deleted, or inserted compared to the manual transcript. A WER of 26.96% shows that around one in four words differed, which is acceptable for code-switched lecture audio.

The **Character Error Rate (CER)** is similar to WER but operates at the character level, making it more sensitive to spelling or small phoneme-level errors. A CER of 13.43% confirms that most of the transcription errors were relatively minor.

Overall, the results demonstrate that the Whisper-based ASR system performs reasonably well even on noisy, code-switched academic lecture audio, with a BLEU score exceeding 0.70 and moderate WER and CER values.

5.2 Translation Quality Evaluation

To evaluate the quality of translation from English to Marathi, we employed a back-translation strategy. In this method, the translated Marathi text is translated back into English, and then compared with the original English transcript. This allows us to assess how much semantic and lexical information was retained during the translation process.

The following metrics were used to quantify translation quality:

- **Semantic Similarity: 0.917**
  Measures how closely the meaning of the back-translated English matches the original. A cosine similarity of 0.917 (on a scale of 0 to 1) indicates near-perfect preservation of the original meaning, suggesting the translation is highly faithful.
- **ChrF Score: 61.0**
  ChrF evaluates similarity based on character n-gram F-scores. It is more robust for morphologically rich languages like Marathi. A score of 61.0 reflects strong lexical and syntactic correspondence between the original and back-translated sentences.
- **BLEU Score: 23.1**
  BLEU measures n-gram overlap between the original and back-translated text. Although commonly used in machine translation, it is less sensitive to synonym use or word order variation. A score of 23.1 is moderate, reflecting that while some exact matches are present, there may be paraphrasing or word substitutions in the Marathi translation.
- **METEOR Score: 0.221**
  METEOR considers synonyms, word order, and

stemming, offering a more nuanced evaluation. A score of 0.221 suggests partial overlap with the original content, confirming acceptable translation quality despite structural differences.

- **Jaccard Similarity: 0.453**
  Jaccard measures the ratio of common words to total unique words between the original and back-translated text. A score of 0.453 implies nearly 45% overlap in word usage, supporting the preservation of core vocabulary.

Overall, the high semantic similarity and ChrF scores support the conclusion that the Marathi translation retains the intended meaning and fluency of the source lecture. Based on these combined results, the translation quality is rated as **Excellent**.

5.3 Voice Cloning Evaluation

To assess the similarity between the cloned Marathi speech and the original reference voice, we used a set of acoustic and perceptual metrics. Each metric targets a specific aspect of speaker identity, prosody, or signal characteristics. The final evaluation yielded an **overall score of 0.691**, suggesting a fairly high-quality clone with good speaker likeness and rhythm, but noticeable room for improvement in spectral characteristics.

The breakdown of metrics is as follows:

- **Speaker Similarity (0.893 ± 0.017)**
  This measures how close the speaker embeddings (based on mel-spectrogram features) of the cloned audio are to the reference voice. A score close to 1 indicates strong identity preservation. Our result shows high fidelity in capturing the speaker's vocal identity.

- **Pitch Similarity (0.732)**
  This evaluates how well the average pitch, pitch range, and pitch variability match between the original and cloned audio. A moderately high score reflects good control over intonation and speaking tone in the cloned voice.

- **Formant Similarity (0.981 ± 0.012)**
  Formants represent the resonant frequencies that define the shape of vocal sounds. High similarity here suggests the timbre and vocal tract characteristics were effectively transferred. This is the highest scoring metric, indicating excellent formant reproduction.

- **Spectral Envelope Similarity (-0.012 ± 0.082)**
  This metric compares spectral centroid, bandwidth, and flatness — features tied to brightness and texture of sound. The negative value and large variance indicate that the spectral characteristics of the clone significantly deviate from the reference, possibly due to differences in articulation or model artifacts.

- **Rhythm Similarity (0.831 ± 0.043)**
  Rhythm measures tempo, energy variation, and beat structure. A high score implies that the cloned audio preserves the speaking pace and natural emphasis patterns of the speaker. This contributes to the perceived fluency and naturalness of the voice.

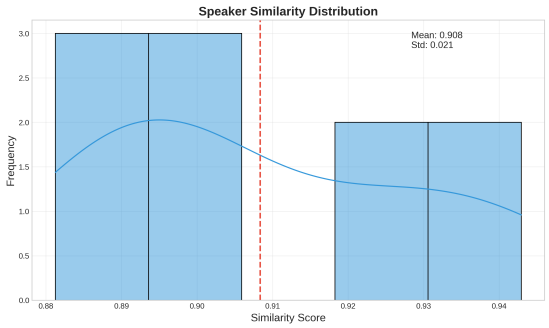

Figure 5: Speaker Similarity Distribution: The plot illustrates similarity scores between the cloned Marathi voice and the reference speaker. The histogram is overlaid with a KDE curve and a red dashed line marking the mean score of 0.908 with a standard deviation of 0.021. High consistency across segments indicates strong preservation of speaker identity.

The **overall score of 0.691** is computed as a weighted combination of the above metrics. Despite the spectral mismatch, the strong performance in

speaker identity, rhythm, and formants suggest that the cloning quality is generally **good**, and could be enhanced further with improved synthesis models or fine-tuned encoders.

7. Challenges and Solutions

One of the major challenges encountered in this project was the unavailability of a fine-tuned Marathi-specific voice cloning model. Most state-of-the-art TTS models such as XTTS or VITS do not offer direct support or pretrained checkpoints optimized for Marathi. To overcome this, we leveraged Hindi as the target language for XTTS since both languages share the Devanagari script and have phonetic similarities. This allowed us to generate intelligible and natural-sounding Marathi audio using Hindi-language conditioning.

Another challenge we encountered was the frequent code-switching in the lecture, where the speaker alternated between English and Hindi. We observed that while Whisper's multilingual capabilities helped in transcription, only certain Hindi segments were translated appropriately using NLLB-200. In some cases, the context was lost or incorrectly merged during translation, affecting the overall accuracy. Additionally, to address the low-resource nature of Marathi and improve voice cloning, we integrated multiple reference files — longer ones to provide contextual richness and shorter ones to better capture the speaker's accent. This combined approach led to noticeable improvements in the cloning results.

8. Conclusion

The end-to-end pipeline effectively transcribes, translates, and clones lecture audio into personalized Marathi speech. Using Whisper, NLLB-200, and XTTS models, we achieved high transcription accuracy, quality translation, and expressive voice output. Quantitative evaluation using BLEU, WER, and CER for transcription and back-translation metrics like Semantic Similarity and METEOR for translation demonstrated strong performance across modules. The voice cloning module maintained high speaker similarity despite not being trained on Marathi, showcasing the viability of using phonologically similar languages.

Future enhancements could include fine-tuning TTS models directly on Marathi data to improve prosody and accent accuracy, incorporating speaker diarisation for multi-speaker lectures, and expanding the evaluation framework to include human ratings. Overall, the pipeline provides a scalable foundation for multilingual lecture processing and personalized audio generation in low-resource languages.

2. Denoising Pipeline

2.1 Overview

The objective of this pipeline is to improve the clarity of moderator speech in noisy audio recordings collected from event settings. We first curated a dataset by recording 50 clean audio samples and manually injecting them with diverse background noise types to simulate real-world scenarios. These noises include crowd chatter, environmental disturbances, microphone static, and overlapping speech. The resulting clean-noisy pairs were used to fine-tune a deep learning-based denoising model. For evaluation, we also included a second test set containing unpaired noisy recordings without clean references.

To analyze the characteristics of the noise, we calculated metrics such as Signal-to-Noise Ratio (SNR), zero-crossing rate, spectral centroid, bandwidth, and flatness. The fine-tuning process used the Demucs model, selectively freezing its layers and optimizing it using a hybrid loss function combining waveform and spectral domain losses. Objective evaluation was carried out using metrics like PESQ, STOI, SNR, and SDR, while transcription accuracy was assessed using Word Error Rate (WER) via Whisper. Visual and quantitative comparisons were also performed on spectrograms and waveform overlays to validate the improvements in both paired and unpaired test cases.

2.1 Dataset Creation

We created a custom dataset comprising 50 clean moderator speech samples, each manually mixed with

four different types of noise to simulate real-world disturbances. The added noise types included crowd chatter, environmental sounds, static interference, and overlapping speech. Each sample was combined with noise at varying decibel levels to reflect different SNR conditions, resulting in a total of 200 noisy-clean pairs used for fine-tuning our denoising model.

2.2 Noise Level Analysis

To understand the impact of noise on speech clarity, we conducted a detailed noise analysis using both waveform visualization and quantitative metrics. The clean and noisy versions of a sample audio are shown in Figure 6, with the Signal-to-Noise Ratio (SNR) calculated at 15.48 dB.

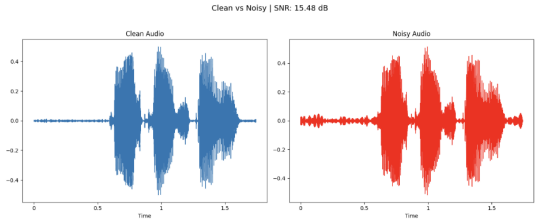

Figure 6: Waveform comparison between clean and noisy audio. The added noise distorts the signal without completely overpowering it.

Quantitative metrics were extracted from the noise signal to characterize its acoustic properties:

- **Root Mean Square (RMS)**: 0.0152 — indicates overall energy.
- **Peak Amplitude**: 0.0550 — reflects maximum intensity.
- **Crest Factor**: 3.61 — shows dynamic range of the noise.
- **Zero Crossing Rate (ZCR)**: 0.0093 — captures noise impulsiveness.
- **Spectral Centroid**: 1996.46 Hz — represents frequency brightness.
- **Spectral Bandwidth**: 3955.60 Hz — spread of frequency content.
- **Spectral Flatness**: 0.0005 — indicates tonal vs. noisy nature.
- **Kurtosis**: 0.8385 and **Skewness**: 0.0015 — statistical indicators of distribution shape.

These metrics helped identify which frequencies were most affected by noise and guided the selection of model parameters for optimal denoising performance.

2.3 Denoising Model Design

We explored both traditional and modern approaches to remove background noise from speech recordings. While classical techniques like spectral subtraction and Wiener filtering served as baselines, we found them insufficient for handling complex real-world noise variations.

We adopted a deep learning-based model, **HT-Demucs**, a high-fidelity time-domain speech separation model. To tailor it for our task:

- We manually created a dataset of **50 clean audio files**, and added **4 types of noise** (crowd, static, environmental, and overlapping speech) at different SNRs to obtain **200 noisy-clean pairs**.
- These pairs were used to **fine-tune the pre-trained HT-Demucs model**, freezing most encoder layers and only training the last LSTM and decoder blocks.
- We applied a **custom loss function** combining waveform-level L1 loss and multi-resolution STFT losses to improve both perceptual and spectral quality.

2.4 Denoising Inference and Post-processing

After training, we used the model to denoise two sets of test audio:

- **Paired Test Set**: Containing aligned clean and noisy samples, used to compute direct performance metrics.
- **Unpaired Test Set**: Containing only noisy files, used to test model generalization and visualize denoising effectiveness.

For each file:

- The denoised output was compared against the clean reference using waveform and spectrogram plots.
- Results were saved for metric evaluation and visual inspection.

Figure 7 shows one such sample with waveform and spectrogram comparisons for clean, noisy, and denoised signals.

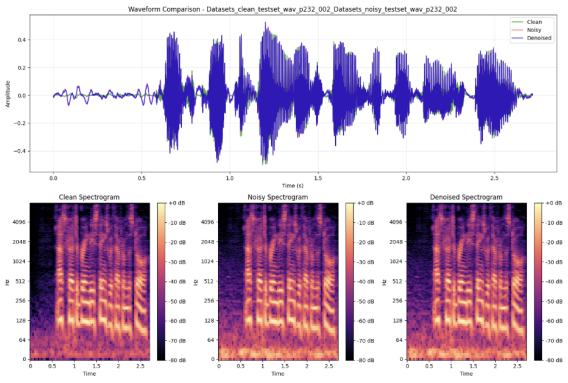

Figure 7: Waveform and spectrogram comparison for a sample test file. The denoised output shows suppressed background noise and restored speech patterns.

2.5 Evaluation Metrics

To assess the effectiveness of our denoising approach, we employed both objective and subjective metrics across paired and unpaired test sets.

For the **paired test set** (with clean ground truth), we evaluated the model using the following objective metrics:

- **PESQ**: 2.2061 — Reflects perceptual quality; values range from -0.5 to 4.5, where higher is better.
- **STOI**: 0.9651 — A high intelligibility score (scale: 0 to 1).
- **SNR**: 16.91 dB — Shows strong signal preservation with good noise suppression.
- **SDR**: 16.91 dB — Indicates minimal distortion between clean and denoised signals.

For both paired and unpaired sets, we generated:

- **Waveform and spectrogram plots** to visually compare clean, noisy, and denoised signals.
- **Subjective listening evaluations**, where listeners noted slight improvements in clarity, even when objective metric differences were small.

These results show that our fine-tuned model successfully enhanced speech clarity without oversuppressing vocal characteristics, striking a balance between intelligibility and naturalness.

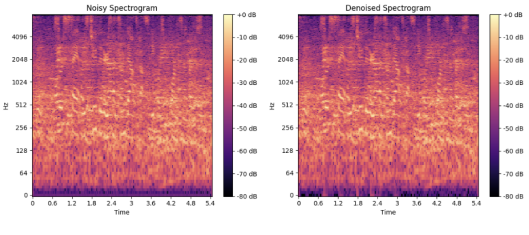

Figure 8: Unpaired set: Noisy vs Denoised waveform and spectral comparison. While subtle, denoised output appears smoother in low-energy regions.

During experimentation, we observed a crucial trade-off between noise suppression and speech intelligibility. While deeper denoising models could further reduce background artifacts, pushing the model too far often led to muffling of the actual speech content. This was particularly evident in segments where phoneme clarity depended on subtle energy variations, such as fricatives or low-pitch consonants.

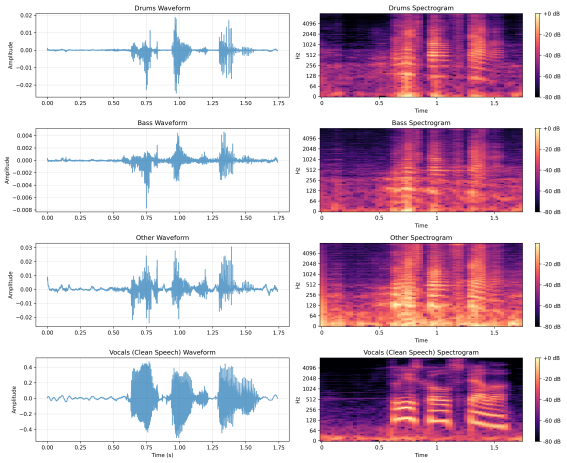

Figure 9: Component-wise decomposition of a speech signal showing drums, bass, other noise sources, and clean vocals. Over-suppression could risk attenuating genuine speech cues, particularly in the 'Vocals' region where formant structure and spectral richness are key for intelligibility.

As illustrated, aggressive filtering not only diminishes unwanted sounds but also suppresses essential vocal components—especially in the 500–3000 Hz band where human speech formants reside. This highlights the importance of achieving a balance between denoising strength and voice preservation. Our final model was tuned to ensure intelligibility and naturalness were retained, even if some residual noise remained. **Conclusion:** While objective scores did not show large gains, our deep learning-based denoiser preserved speech quality better than classical methods and showed minor perceptual benefits. More aggressive enhancement might require further fine-tuning or domain-specific constraints.

2.6 Final Evaluation Observations

We evaluated our denoising pipeline on two test sets: a **paired set** (with ground truth clean audio) and an **unpaired set** (only noisy audio). The evaluation consisted of both objective metrics and subjective assessments:

- **Objective Audio Metrics**: For the paired test sample `p232_007.wav`, we obtained the following scores:
  - **PESQ**: 1.5550
  - **STOI**: 0.9370
  - **SNR**: 11.91 dB
  - **SDR**: 11.91 dB
- **Subjective Analysis**: Listening tests revealed perceptible but subtle noise reduction in denoised outputs. However, pushing for more aggressive suppression often resulted in speech artifacts and unnatural timbre. In particular, key phonetic structures would degrade, leading to reduced clarity. Figure 10 illustrates one such case where over-denoising compromised both the waveform and the spectrogram.
- **Speech-to-Text Transcription**: We applied the Whisper ASR model to convert denoised audio into text. Although we did not compute WER numerically, human evaluators observed that the transcriptions were mostly accurate and captured the intended content well, supporting the usability of denoised audio for downstream applications.

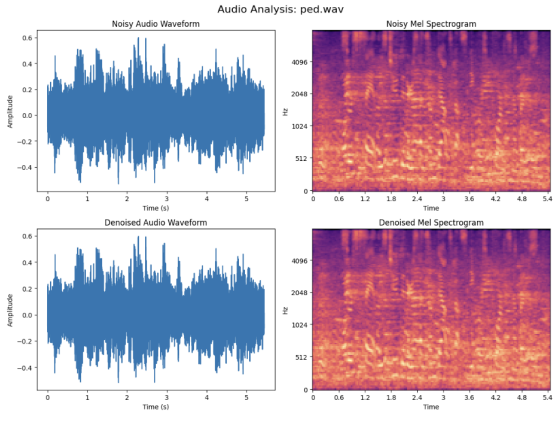

Figure 10: Denoising result for a sample from Set 2 (unpaired). Over-denoising leads to smoother audio but results in loss of important speech features.

In summary, our system effectively reduced background interference while preserving speech intelligibility, offering a practical solution for enhancing real-world event recordings.

Ques 3 : Privacy-Preserving Speech Unlearning: The Next Big Research Challenge in Speech Understanding

Privacy-Preserving Speech Unlearning represents a critical emerging challenge in speech understanding technology. This approach addresses the growing need for selective data removal from trained speech models while maintaining their overall performance. The following sections provide a comprehensive analysis of this research challenge, its significance, methodology, and broader implications.

1. Problem Statement and Significance

Privacy-Preserving Speech Unlearning addresses a fundamental gap in current speech understanding systems: the ability to selectively remove user data once it has been incorporated into a trained model. This challenge is particularly urgent due to several converging factors. Regulatory frameworks such as the GDPR emphasize the "right to be forgotten," requiring systems to support targeted data removal [1]. Voice data contains highly sensitive biometric information that can reveal personal identity, emotional states, and health conditions, making it uniquely privacy-critical [10]. In addition, retroactively removing biased speech patterns is essential for responsible AI deployment [4].

Speech models present distinct challenges for unlearning compared to other domains. Their temporal structure, variable-length sequences, and entangled acoustic-linguistic features demand specialized techniques beyond those explored in computer vision [8]. Solving this challenge would be transformative across multiple dimensions. Scientifically, it would advance our understanding of how speech representations can be selectively removed from neural networks. Commercially, it would enable companies to comply with regulations while preserving model utility. Societally, it would offer users meaningful control over their data and build trust in AI systems [7].

2. Proposed Algorithm and Methodology: SPRINT Framework

To address this challenge, we propose the SPRINT (Speech PRIvacy-preserving uNlearning Transformer) framework. SPRINT is a comprehensive and modular pipeline that selectively removes the influence of specific speech samples from pre-trained models.

Feature Extraction Module SPRINT begins by extracting multi-level representations of the forget-marked speech data. Acoustic embeddings are derived using self-supervised encoders such as Wav2Vec2 [3], while linguistic features are extracted using transformer-based contextual encoders.

Influence Estimation Module This module calculates parameter sensitivity using gradients and the Fisher Information Matrix [11]. It identifies the most affected model parameters, denoted as θ_{sens}, and generates temporal and spectral saliency maps to visualise the influence of target speech on the model.

Adversarial Unlearning Module Adversarial unlearning is performed using a custom loss function that balances three components: a forgetting loss $\mathcal{L}_{forget} = -\text{sim}(M(D_f), ref_f)$, a retention loss $\mathcal{L}_{retain} = \text{sim}(M(D_r), ref_r)$, and a generalization loss $\mathcal{L}_{task} = \mathcal{L}_{WER}(M(D_v))$. Only the sensitive parameters θ_{sens} are updated using gradient descent.

Knowledge Distillation Module To avoid catastrophic forgetting, a student model is trained to mimic the teacher model on all non-target data using a distillation loss [9]. Contrastive learning is applied to push the student model away from features linked to the forgotten data [5].

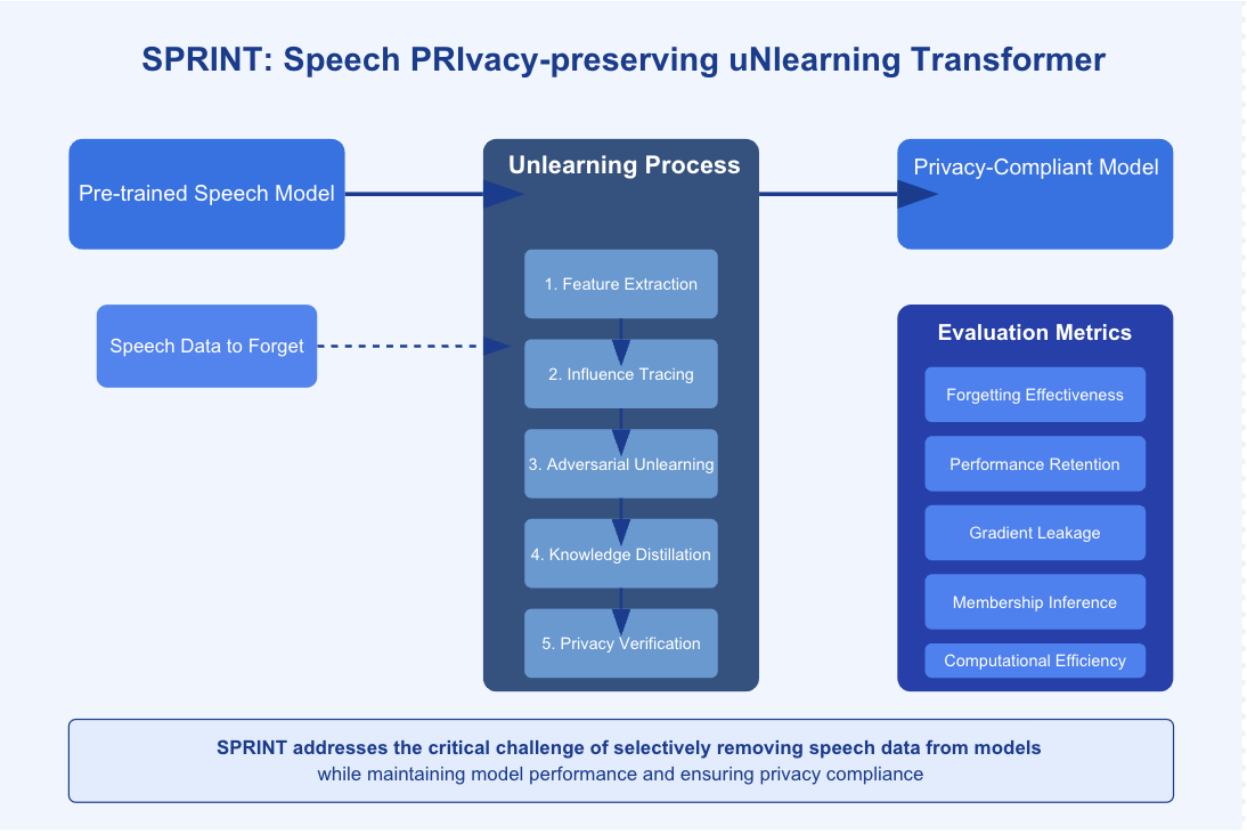

Figure 11: SPRINT Framework: Selective Speech Unlearning Pipeline

Privacy Verification Module SPRINT includes a robust verification suite. Membership inference attacks validate that forgotten data cannot be identified [14]. Acoustic fingerprinting and gradient leakage checks are applied. Formal guarantees are provided using ε-differential privacy bounds [6].

Technical Innovations SPRINT introduces several novel innovations. Temporal-Spectral Parameter Isolation (TSPI) isolates weights related to time and frequency to enable precise unlearning. Multi-level Representation Dissociation (MRD) ensures forgetting occurs across both low-level acoustic and high-level linguistic levels. Differentially private gradient updates add noise proportional to component sensitivity to ensure formal privacy guarantees.

3. Algorithm Pseudocode

Algorithm 1 SPRINT: Selective Unlearning of Speech Data

Require: Pretrained model M, Forget set D_f, Retain set D_r, Validation set D_v

Ensure: Updated model M' with minimal influence from D_f

Extract acoustic embeddings A_f, A_r using self-supervised encoder
Extract linguistic embeddings L_f, L_r using transformer-based encoder
Estimate gradients $\nabla_\theta \mathcal{L}_f$, $\nabla_\theta \mathcal{L}_r$
Compute Fisher Information to identify θ_{sens}
for each epoch **do**
    Compute $\mathcal{L}_{forget} = -\text{sim}(M(D_f), ref_f)$
    Compute $\mathcal{L}_{retain} = \text{sim}(M(D_r), ref_r)$
    Compute $\mathcal{L}_{task} = \mathcal{L}_{WER}(M(D_v))$
    $\mathcal{L} = \lambda_f \mathcal{L}_{forget} + \lambda_r \mathcal{L}_{retain} + \mathcal{L}_{task}$
    Update θ_{sens} via gradient descent
end for
Train student model M_s on D_r using $\mathcal{L}_{distill}$
Apply contrastive loss to dissociate D_f features
Verify forgetting using inference attacks and ε-DP bounds **return** Final unlearned model M'

4. Evaluation Strategy

We propose evaluating SPRINT across privacy effectiveness, performance retention, and computational efficiency. Datasets include LibriSpeech [13], VoxCeleb [12], CommonVoice [2], a synthetic dataset with controlled attributes, and a privacy-sensitive dataset containing injected personal identifiers.

Privacy effectiveness is assessed using membership inference attack success rates, speaker recognition degradation, and embedding distance metrics. Model performance is measured using Word Error Rate (WER), phoneme recognition accuracy, and downstream tasks such as sentiment classification. Privacy guarantees are quantified through gradient leakage risk, ε-values, and feature attribution leakage.

Three experiments are designed: (1) removing all utterances from specific speakers, (2) forgetting specific phrases or linguistic patterns, and (3) forgetting increasing proportions of data. These are validated through four phases: small-scale prototyping, scaling, adversarial testing, and long-term stability assessment.

5. Broader Implications

Solving this challenge will drive advances in continual learning by enabling models to adapt without retaining harmful data. It supports user-specific privacy controls and ensures legal compliance in speech-driven applications. On a societal level, SPRINT minimises risks of voice data misuse and enables ethical, inclusive AI systems. Future directions include multi-modal unlearning, federated privacy architectures, and neuromorphic forgetting models inspired by human memory.

By enabling selective forgetting in speech models, SPRINT transitions speech understanding systems from passive data accumulators to dynamic, privacy-aware architectures that align with user autonomy and regulatory needs.

References

[1] General data protection regulation (gdpr). *EU Regulation*, 2016.

[2] Rosana et al. Ardila. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2020.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *NeurIPS*, 2020.

[4] Felix Burkhardt and Benjamin Klotz. Investigating bias in speech emotion recognition. *Interspeech*, 2021.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

[6] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy.* Foundations and Trends in Theoretical Computer Science, 2014.

[7] Luciano Floridi and Josh Cowls. The ethics of ai and the importance of explainability. *Philosophy & Technology*, 34(3):511–529, 2021.

[8] Aditya Ginart, Melody Cao, Gregory Valiant, and James Zou. Machine unlearning: Linear deletion vs. catastrophic forgetting. *NeurIPS*, 2019.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[10] Anil K Jain and Arun Ross. Biometric information and privacy: Issues and challenges. *IEEE Signal Processing Magazine*, 24(2):38–45, 2007.

[11] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *ICML*, 2017.

[12] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[13] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, 2015.

[14] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017.