Wise@DravidianLangTech 2026: Dialect-Aware Tamil Speech Classification and Recognition via Cross-Pipeline Embedding Transfer

Ganesh Sundhar S[1], Hari Krishnan N[1], Gnanasabesan G[1], Suriya KP[1], Jyothish Lal G[1]

[1]Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India

{cb.en.u4aie22017, cb.en.u4aie22020, cb.en.u4aie22018, cb.en.u4aie22164}@cb.students.amrita.edu

g_jyothishlal@cb.amrita.edu

Abstract

This paper presents the **Wise** system for the shared task on dialect-based speech processing in Tamil, addressing two subtasks: **(1) four-way dialect region classification** (Northern, Southern, Western, Central), and **(2) dialectal Tamil ASR**. All audio is preprocessed using loudness normalization followed by neural denoising to ensure consistent audio quality for downstream models. For classification, we experminet different model variants combining multilingual and Tamil-pretrained **Wav2Vec2** backbones with five temporal pooling strategies under frozen and partial fine-tuning settings. Our best configuration i.e. learned attentive pooling with partial fine-tuning and a differentially-trained MLP head achieves a macro F1 of **0.79**, securing **1st place** (0.26-point margin). For ASR, we propose two novel **dialect-conditioned Whisper** architectures (residual injection and cross-attention) that inject dialect embeddings from the trained classifier into the ASR pipeline, and additionally evaluate a vanilla Whisper-Tamil fine-tuned baseline. The best model achieved a WER of **0.90**, securing **8th place**.

Keywords: Dialect Embeddings, Temporal Pooling, Frozen Backbone, Residual & Cross Attention Injection, Wav2Vec2, Whisper

1 Introduction

Tamil is one of the oldest classical languages in the world, spoken by millions across South India, Sri Lanka, and Singapore. Its wide geographic spread has produced rich regional dialects that differ in pronunciation, rhythm, and vocabulary, making it difficult for automatic speech recognition and dialect identification systems to perform well.

The shared task on dialect-based speech recognition and classification in Tamil (Bharathi et al., 2026) curated a dataset to address this limitation and facilitate the training of dialect-aware models (Bharathi et al., 2025). This dataset was used to train the models. The system presented in this paper addresses two subtasks—**(1) dialect identification** and **(2) speech transcription**—and makes three contributions: a preprocessing pipeline combining LUFS normalization and neural denoising, a study of 12 dialect classification variants showing learned attentive pooling performs best, and two dialect-conditioned Whisper architectures that inject dialect embeddings into the ASR encoder.

2 Related Work

Early dialect and accent identification systems relied on acoustic features such as MFCCs combined with statistical models like GMMs or HMMs (Levinson, 1986; Gorin et al., 2014). Deep learning approaches, including LSTMs, CNNs, and RNNs, later achieved significant improvements in speech modeling (Jain et al., 2020; Weninger et al., 2015). More recently, self-supervised models such as Wav2Vec 2.0 (Baevski et al., 2020) and Whisper (Radford et al., 2023) have established new benchmarks for speech processing tasks (S et al., 2025b). Research on hate speech in Dravidian languages (S et al., 2025a) has also progressed considerably using both classical and neural approaches (Sreelakshmi et al., 2024; B et al., 2023). For Tamil specifically, fine-tuned Whisper models have demonstrated strong performance on speech and language tasks (Jairam et al., 2024). Building on these advances, our work studies pooling strategies for dialect classification and integrates dialect embeddings into the ASR pipeline.

3 Dataset

The dataset comprises spontaneous and read speech from native speakers across four regional dialect groups. All recordings are sampled at 16 kHz and captured in natural acoustic environments. The training partition contains 9.22 hours of transcribed speech. Table 1 summarizes the train distribution.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 447–452

Dialect	Samples	Duration
Southern	1,427	2h 44m
Northern	1,696	3h 29m
Western	1,126	1h 59m
Central	885	1h 08m
Total	**5,134**	**9h 22m**

Table 1: Train Dataset distribution across dialect groups.

4 Methodology

Our system comprises three tightly integrated stages: audio preprocessing, dialect classification (Subtask 1), and automatic speech recognition (Subtask 2). A key design principle is that the dialect classifier trained in Subtask 1 directly feeds into the ASR system of Subtask 2 through dialect embedding transfer. We describe each component in detail below.

4.1 Audio Preprocessing

The raw audio recordings exhibit considerable variation in recording conditions—including ambient noise, varying volume levels, multi-channel recordings, and DC bias—which can degrade downstream model performance. We design a two-stage preprocessing pipeline to address these issues.

Stage 1 — Audio Standardization. We apply three sequential transformations to each audio file:

1. **Channel Selection**: For multi-channel recordings, we estimate the Signal-to-Noise Ratio (SNR) for each channel. The one with the highest SNR is retained.
2. **DC Offset Removal**: The mean of the waveform is subtracted ($x'(t) = x(t) - \bar{x}$) to eliminate any residual DC bias introduced by recording hardware.
3. **LUFS Normalization**: We normalize loudness to -23 LUFS using the ITU-R BS.1770 standard via the `pyloudnorm` library (Johnson, 2006). Unlike simple peak normalization, LUFS normalization accounts for perceptual loudness, ensuring consistent apparent volume across utterances regardless of their dynamic range.

Stage 2 — Neural Denoising. We apply Facebook's Denoiser DNS64 model (Defossez et al., 2020) for speech enhancement. This model is based on a causal U-Net encoder-decoder architecture that operates directly in the time domain, enabling it to preserve phase information.

4.2 Subtask 1: Dialect Classification

For dialect classification, rather than committing to a single architecture, we conduct a systematic ablation study of 12 model variants to identify the optimal configuration. All variants share a common three-component architecture: (1) a self-supervised speech backbone for feature extraction, (2) a temporal pooling layer to aggregate frame-level representations into a fixed-dimensional utterance embedding, and (3) a multi-layer perceptron (MLP) (Rumelhart et al., 1986) classification head.

4.2.1 Speech Backbone.

We experiment with the original multi-lingual (*facebook/wav2vec2-large-xlsr-53*) model and the Tamil finetuned version (*Harveenchadha/vakyansh-wav2vec2-tamil-tam-250*) as our speech feature extractor. We investigate two fine-tuning strategies:

- **Frozen**: The entire backbone is frozen, and only the pooling and classification layers are trained.
- **Partial Fine-tuning**: The CNN feature encoder and the bottom transformer layers are frozen, while the top N transformer layers are unfrozen and trained jointly with the pooling and classification layers. This allows the model to adapt its higher-level contextual representations to the dialect classification task while preventing catastrophic forgetting in the lower layers.

Through experimentation we find that unfreezing the top 4 (out of 12) transformer layers provides the best trade-off between model capacity and overfitting risk on this relatively small 9.22-hour dataset.

4.2.2 Temporal Pooling Strategies.

The encoder produces a sequence of hidden states, where a temporal pooling function is needed to produce a fixed-dimensional utterance embedding. We systematically evaluate five pooling strategies:

1. **Mean Pooling**: The masked average over all valid frames (Arora et al., 2017).
2. **Attentive Pooling**: A learnable linear projection computes scalar attention weights over frames, followed by softmax-weighted aggregation.
3. **Learned Attentive Pooling**: An enhanced nonlinear attention mechanism that replaces the single linear projection with a two-layer bottleneck network (Okabe et al., 2018).
4. **Mean + Attentive**: Concatenation of mean-pooled and attentive-pooled vectors.
5. **Learned Mean + Attentive**: Concat of mean-pooled and learned-attentive-pooled vectors.

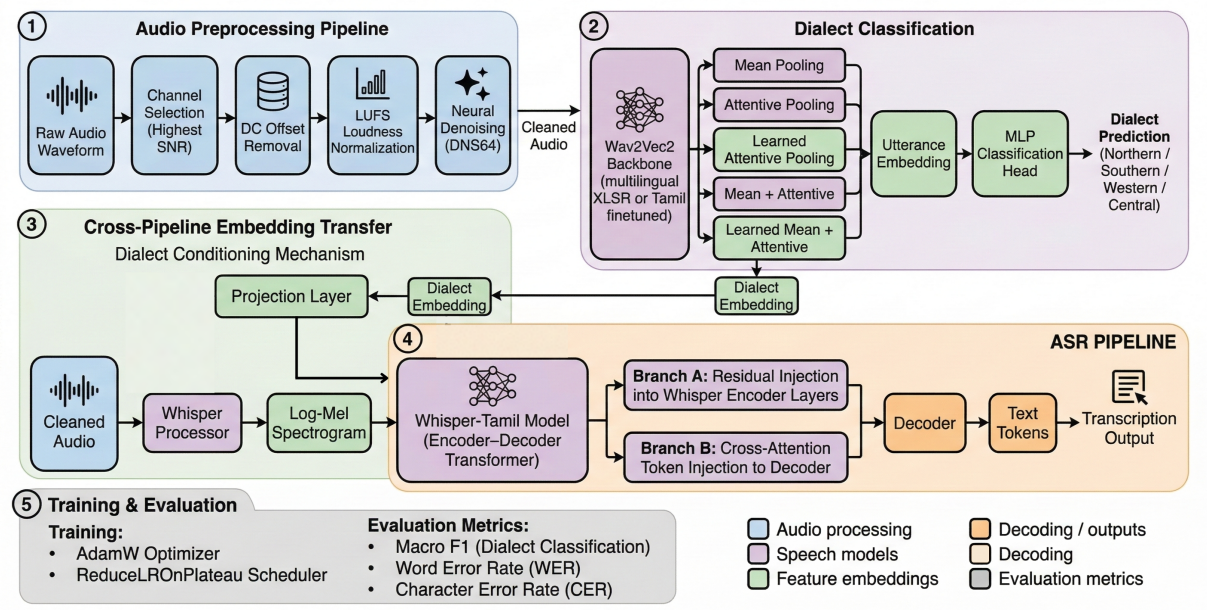

Figure 1: Overview of the proposed dialect-aware Tamil speech processing architecture. Raw audio is first standardized through channel selection, DC offset removal, LUFS normalization, and neural denoising. A Wav2Vec2-based dialect classifier extracts utterance-level dialect embeddings using temporal pooling strategies and an MLP head. These embeddings are transferred across pipelines to condition the Whisper-Tamil ASR model through residual or cross-attention injection, enabling dialect-aware transcription.

4.2.3 Classification Head.

The pooled utterance embedding is passed through a three-layer MLP with progressively decreasing hidden dimensions. The dropout rate of 0.3 is applied after each ReLU activation to regularize the classifier and prevent overfitting.

4.3 Subtask 2: Automatic Speech Recognition

For ASR, we explore both dialect-conditioned and baseline approaches using the Whisper-Tamil models (*vasista22/whisper-tamil-small* and *vasista22/whisper-tamil-medium*), which are fine-tuned variants of OpenAI's Whisper. We compare their performance through experiments to analyze the impact of model scale on dialect-aware ASR.

4.3.1 Dialect Embedding Extraction.

A central component of our dialect-conditioned ASR approach is the extraction of utterance-level dialect embeddings from the trained dialect classifier. Specifically, for each audio sample, we perform a forward pass through the best dialect classifier and extract the output of the learned attentive pooling layer before the MLP classification head. These embeddings capture rich dialect-discriminative information including phonetic quality, prosodic patterns, and speaking rate characteristics of each dialect region in a continuous vector space. The embeddings are pre-computed for all training samples and stored as a NumPy archive for efficient loading during ASR training.

4.3.2 Variant 1: Residual Injection

The pre-extracted dialect embedding is first projected to the Whisper encoder's hidden dimension using a linear layer followed by layer normalization. The projected embedding is then injected into the encoder's residual stream at every transformer layer. Specifically, at each layer ℓ, the dialect embedding is added to the hidden states before the self-attention computation, analogous to the injection of conditional embeddings used in diffusion models. This repeated injection introduces a persistent dialect bias in the encoder representations, guiding the model toward dialect-aware acoustic processing across multiple levels of abstraction.

4.3.3 Variant 2: Cross-Attention Injection.

In this approach, the dialect embedding is projected into a single token and concatenated to the front of the encoder output sequence. The decoder's cross-attention mechanism then attends to this augmented sequence, allowing it to condition the transcription on dialect identity through its attention queries. A staged training schedule is employed: during warmup, the entire Whisper model is frozen and only the dialect projection layer is trained, al-

lowing it to learn a meaningful mapping. After warmup, the decoder and projection output layer are unfrozen for joint fine-tuning.

4.3.4 Variant 3: Vanilla Whisper Baseline.

A standard fine-tuning without any dialect conditioning, serving as a controlled baseline. All model parameters are trainable during this fine-tuning.

5 Experimental Setup

Experiments were conducted on an NVIDIA RTX 4080 Laptop GPU locally and on cloud instances equipped with NVIDIA RTX A5000 GPUs for computationally intensive workloads. Wav2Vec2 normalization was disabled since recordings were already LUFS-normalized. A fixed random seed of 17 was used to ensure reproducibility.

All models were optimized using AdamW (Loshchilov and Hutter, 2019). For dialect classification, differential learning rates were applied: 10^{-5} for unfrozen Wav2Vec2 encoder layers and 10^{-4} for the pooling and MLP layers. For ASR models, a learning rate of 10^{-5} with weight decay 10^{-2} was used. Training employed ReduceLROnPlateau scheduling with a factor of 0.5 and patience of 5 based on validation performance. Dialect classifiers were evaluated on a held-out 20% development split using macro F1 (Van Rijsbergen, 1979) `torchmetrics`), and the best-performing model was used for the final submission. ASR performance was evaluated using Word Error Rate (WER) (Rabiner and Juang, 1993).

6 Results and Analysis

6.1 Subtask 1: Dialect Classification

Our best-performing dialect classification variant used learned attentive pooling with partial fine-tuning of the top four Wav2Vec2 transformer layers. This configuration achieved a macro F1 of **0.79**, placing **1st** with a margin of 0.26 over the next team (Table 2). Learned attentive pooling was particularly effective as it captures temporally localized dialect cues such as formant transitions and intonation rather than averaging representations across all frames.

6.2 Subtask 2: Automatic Speech Recognition

Our submission used the dialect-conditioned Whisper architecture with residual dialect embedding injection and achieved a WER of **0.90**, ranking 8th (Table 3). Compared to the vanilla Whisper baseline, the dialect-conditioned model showed improved robustness to regional pronunciation variations. Although the model demonstrated stable validation behavior during experimentation, it was only partially trained due to time and computational constraints, which likely limited its performance in the shared task. We expect that fully training the dialect-conditioned models—particularly the cross-attention variant—would further reduce WER by guiding the decoder toward dialect-specific vocabulary and acoustic patterns.

Rank	Team	Macro F1
1	**Wise**	**0.79**
2	Wave2Word	0.53
3	IIITK_SpeechScape	0.48

Table 2: Top teams in Subtask 1 (Dialect Classification).

Rank	Team	WER
1	CHMOD_777	0.51
2	CUET_InferX	0.54
3	Wave2Word	0.55
8	**Wise**	**0.90**

Table 3: Top teams in Subtask 2 (Speech Recognition).

7 Discussion

Our core contribution is cross-pipeline dialect embedding transfer: embeddings from the classifier's attentive pooling layer are injected into Whisper via residual injection (at every encoder layer) or cross-attention conditioning (as a decoder context token), with minimal added parameters. The superiority of non-linear learned attentive pooling over mean pooling confirms that dialect cues are sparse and localized, making a trainable frame-scoring network more effective than simple averaging.

8 Conclusion

The **Wise** system achieved **1st place** in Tamil dialect classification (macro F1: 0.79) using learned attentive pooling with partial Wav2Vec2 fine-tuning. We also proposed two dialect-conditioned Whisper architectures (residual injection and cross-attention) for dialect-aware ASR. Future work will complete training of these ASR variants and also address the class imbalance.

Code: `https://github.com/Ganesh2609/DialectBasedSpeechProcessing`

Limitations

1. The training dataset is relatively small (9.22 hours of speech), which limits the ability of large speech models to fully adapt to dialectal variations.
2. The dataset is class-imbalanced (Central: 885 vs. Northern: 1,696 samples), which may reduce recall for underrepresented dialect groups.
3. Some transcriptions in the dataset contain minor inaccuracies or partially incorrect words, which can introduce noise during ASR training and evaluation.

Acknowledgments

We thank the organizers of the DravidianLangTech@ACL 2026 shared task for curating the Tamil dialect speech corpus and providing the evaluation infrastructure.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Premjith B, Jyothish Lal G, Sowmya V, Bharathi Raja Chakravarthi, Rajeswari Natarajan, Nandhini K, Abirami Murugappan, Bharathi B, Kaushik M, Prasanth Sn, Aswin Raj R, and Vijai Simmon S. 2023. Findings of the shared task on multimodal abusive language detection and sentiment analysis in tamil and malayalam. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. 2020. Real time speech enhancement in the waveform domain. *arXiv preprint arXiv:2006.12847*.

Allen Gorin and 1 others. 2014. Investigating multidialectal Arabic acoustic modeling. In *Proceedings of Interspeech*.

Akshit Jain and 1 others. 2020. Contextual semi-supervised learning: An approach to leverage air-surveillance and general data in speech recognition. In *Proceedings of Interspeech*.

R Jairam and 1 others. 2024. CEN_Amrita at DravidianLangTech-2024: Whisper-based dialect identification for Tamil. In *Proceedings of the Fifth Workshop on Speech and Language Technologies for Dravidian Languages*.

Don Johnson. 2006. Signal loudness measurement using the ITU-R BS.1770 standard. *AES Journal*.

Stephen E Levinson. 1986. Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech & Language*, 1(1):29–45.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. 2018. Attentive statistics pooling for deep speaker embedding. In *Interspeech*.

Lawrence R Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Prentice Hall.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning*.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323:533–536.

Ganesh Sundhar S, Durai Singh K, Gnanasabesan G, Hari Krishnan N, and Mc Dhanush. 2025a. Wise@LT-EDI-2025: Combining classical and neural representations with multi-scale ensemble learning for code-mixed hate speech detection. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 54–62, Naples, Italy. Unior Press.

Ganesh Sundhar S, Hari Krishnan N, Arun Prasad T D, Shruthikaa V, and Jyothish Lal G. 2025b. CrewX@LT-EDI-2025: Transformer-based Tamil ASR fine-tuning with AVMD denoising and GRU-VAD for enhanced transcription accuracy. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 11–16, Naples, Italy. Unior Press.

K Sreelakshmi, B Premjith, Bharathi Raja Chakravarthi, and K P Soman. 2024. Detection of hate speech and offensive language codemix text in dravidian languages using cost-sensitive learning approach. *IEEE Access*.

C. J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworth-Heinemann.

Felix Weninger, Hakan Erdogan, Shinji Watanabe, and 1 others. 2015. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *International Conference on Latent Variable Analysis and Signal Separation*.