wavCSE: Learning Fixed-size Unified Speech Embeddings via Feature-based Multi-Task Learning

Braveenan Sritharan and Uthayasanker Thayasivam
Dept. of Computer Science & Engineering
University of Moratuwa
Sri Lanka
{braveenans.22, rtuthaya}@cse.mrt.ac.lk

Abstract

Modern speech applications require compact embeddings that generalize across both linguistic and paralinguistic tasks. However, most existing embeddings are task-specific and fail to transfer effectively across domains. We propose wavCSE, a feature-based multi-task learning model that produces a fixed-size unified speech embedding suitable for both linguistic and paralinguistic tasks. wavCSE is jointly trained on keyword spotting, speaker identification, and emotion recognition, achieving state-of-the-art performance on all three tasks. The resulting unified embedding is then evaluated on twelve downstream tasks spanning both linguistic and paralinguistic domains. Experimental results show that it outperforms strong baselines on nine of the twelve tasks, indicating effective generalization across domains. To streamline embedding generation, we introduce a recursive layer selection strategy that identifies the most informative hidden layer outputs from the upstream model and refine how these selected outputs are aggregated in the downstream model. These enhancements reduce memory usage and computational cost while improving task performance, making them broadly applicable to self-supervised learning-based speech processing models.

1 Introduction

Speech is a time-varying signal that conveys multiple layers of information, including linguistic content, speaker identity, emotional state, and other paralinguistic attributes (Yang et al., 2021). To represent raw speech effectively, prior work has explored two main strategies: feature engineering and representation learning (Latif et al., 2023). Feature engineering relies on domain expertise to manually design features such as Mel-frequency cepstral coefficients (MFCCs), which aim to extract relevant acoustic properties from the signal. In contrast, representation learning enables models to automatically learn informative features from data, which typically leads to better generalization across a variety of speech processing tasks.

Speech representation learning has evolved through successive methodological advances. Early approaches relied on clustering and statistical models such as Gaussian Mixture Models (GMMs) (Gauvain and Lee, 1994) and Hidden Markov Models (HMMs) (Bahl et al., 1986) to capture low-level acoustic patterns. These were followed by supervised deep neural networks, which enabled more expressive representations but required large amounts of labeled data. More recently, self-supervised learning (SSL) has become the dominant paradigm, with models such as wav2vec (Baevski et al., 2020), HuBERT (Hsu et al., 2021), WavLM (Chen et al., 2022), and Whisper (Radford et al., 2023) pre-trained on large-scale unlabeled speech corpora. Representations extracted from these SSL models have achieved state-of-the-art (SOTA) performance on a wide range of downstream tasks (Yang et al., 2021; Chen et al., 2022), demonstrating strong generalization and the ability to capture diverse speech characteristics.

The representations discussed so far are typically variable-length sequences of vectors that scale with the duration of the speech signal (Baevski et al., 2020). Each vector corresponds to a short, fixed-duration time window, commonly referred to as a frame, and captures low-level acoustic features specific to that frame. In contrast, a speech embedding is a higher-level representation derived by aggregating frame-level representations using neural networks, resulting in a single fixed-size vector that summarizes the entire speech signal, regardless of its duration (Shi et al., 2020). This compact format enables efficient storage on edge devices and real-time transmission of speech of any length. However, converting variable-length sequences into fixed-size vectors often leads to information loss (Porjazovski et al., 2024), posing

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 879–887

a key challenge in designing embeddings that preserve the full richness of the original speech signal.

Most existing speech embeddings are optimized for specific tasks and do not generalize well across different types of downstream tasks. For example, speaker embeddings such as i-vector (Dehak et al., 2011), d-vector (Variani et al., 2014), and x-vector (Snyder et al., 2018) are primarily designed for speaker verification (SV). Similarly, task-specific embeddings have been proposed for linguistic content (Haque et al., 2019). However, the development of a fixed-size unified speech embedding that supports both linguistic and paralinguistic tasks remains relatively underexplored. This limitation is increasingly problematic for modern speech applications such as virtual assistants, which demand models capable of performing multiple tasks simultaneously. For instance, keyword spotting (KS) enables wake-word detection, speaker identification (SID) enables personalization, and emotion recognition (ER) enhances user interaction. These use cases highlight the need for a compact, fixed-size speech embedding that generalizes well across diverse downstream tasks.

In this paper, we propose a feature-based multi-task learning (MTL) model called wavCSE, designed to generate a fixed-size speech embedding that generalizes across diverse tasks. Our pipeline is organized into two phases, each with independent training and testing. In Phase 1, wavCSE is jointly trained on three classification tasks: keyword spotting (KS), speaker identification (SID), and emotion recognition (ER), to determine the optimal unified embedding that captures linguistic, speaker-related, and emotional information. In Phase 2, the trained wavCSE model is frozen, and fixed embeddings extracted from it are used as input features to train and evaluate task-specific models on twelve downstream tasks, including KS, SID, and ER on new datasets, as well as additional tasks spanning both linguistic and paralinguistic domains. Experimental results show that in Phase 1, wavCSE achieves strong performance across the three training tasks, while in Phase 2, it outperforms strong task-specific baselines on nine of the twelve downstream tasks, demonstrating its effectiveness as a general-purpose speech embedding.

Beyond deriving a unified speech embedding, we introduce two architectural improvements in wavCSE that enhance embedding generation and are broadly applicable to any SSL-based speech processing pipeline. First, we propose a recursive layer selection strategy to identify the most informative transformer encoder layers from the pre-trained WavLM Large model. Unlike prior approaches that utilize all 25 layers (Chen et al., 2022), our method selects only 16, reducing upstream model memory usage by 24% while improving downstream task performance. Second, we replace the commonly used weighted average pooling (Yang et al., 2021) with learned-norm pooling to aggregate the selected transformer encoder layer outputs in the downstream model. This pooling mechanism dynamically adjusts each layer's output contribution based on its norm, enabling better capture of task-relevant information. Together, these enhancements reduce computational cost and improve accuracy, enhancing both the efficiency and scalability of SSL-based speech models.

2 Methodology

We propose wavCSE, a model designed to derive a unified speech embedding. As shown in Figure 1, its architecture builds on the SUPERB benchmark (Yang et al., 2021), which consists of two components: an upstream model and a downstream model. The upstream model is a self-supervised learning (SSL) model that extracts representations from raw speech signal, while the downstream model performs task-specific learning based on these representations. wavCSE adopts this structure and employs the pre-trained WavLM Large (Chen et al., 2022) as the upstream model, selected for its strong performance and ability to capture both linguistic and paralinguistic information. In contrast to SUPERB, which optimizes for task-specific outputs, wavCSE is designed to produce a single embedding that generalizes across tasks. To this end, we introduce three key modifications to the original SUPERB architecture.

2.1 Recursive Layer Selection

The first architectural modification alters how transformer encoder layer outputs from the upstream model are used in the downstream model. In the SUPERB architecture, all transformer encoder layer outputs, along with the input to the first transformer encoder layer, are used as speech representations for downstream tasks. Since wavCSE employs WavLM Large, which generates 25 hidden layer outputs, using all of them results in high-dimensional speech representations and increases computational complexity in the multi-task learn-

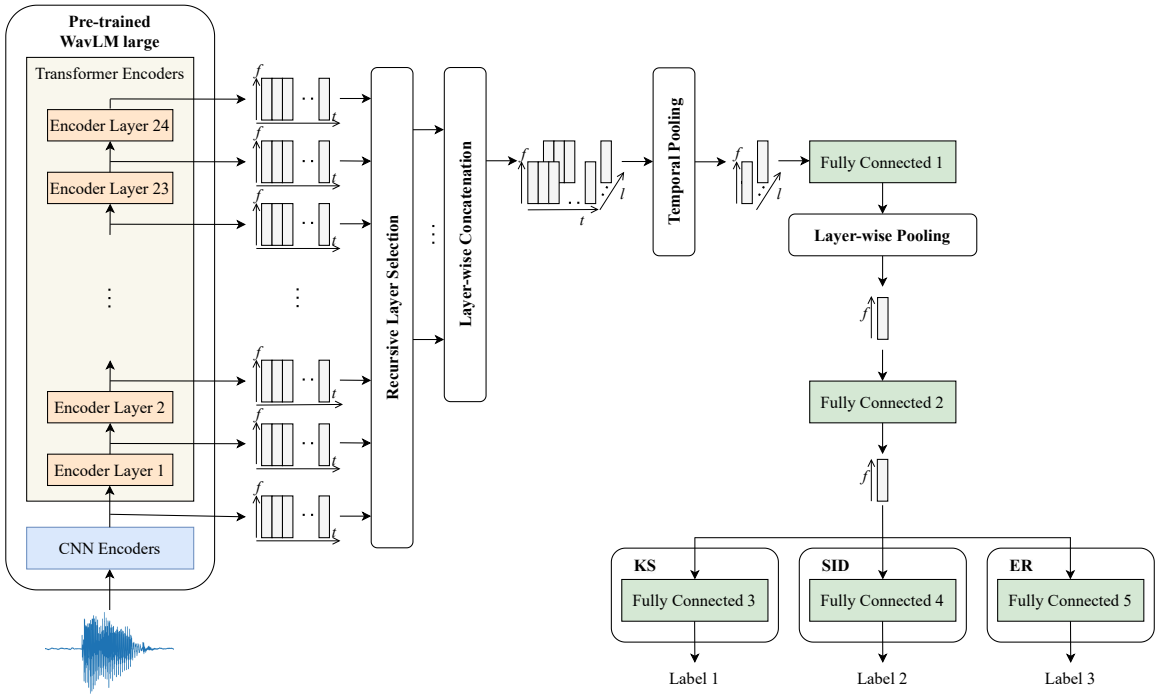

Figure 1: Overview of the proposed wavCSE architecture for deriving a unified speech embedding. The process begins by feeding input audio into the pre-trained WavLM-Large model (Chen et al., 2022), which outputs 25 frame-level hidden layer outputs. A subset of informative layer outputs from these 25 is selected using the proposed layer selection strategy. The selected layer outputs are then concatenated along the layer axis and aggregated using temporal pooling. The pooled output is passed through a fully connected layer, followed by layer-wise pooling, and then another fully connected layer to produce the final unified speech embedding. During wavCSE training, the unified embedding is optimized for three tasks: keyword spotting (KS), speaker identification (SID), and emotion recognition (ER). After training, the resulting embedding can be used as input to any downstream task model.

ing (MTL) setup in the downstream model.

To address this, wavCSE introduces a strategy called recursive layer selection, inspired by Recursive Feature Elimination (RFE) (Zhang and Liu, 2007). We begin by applying weighted average pooling (WAP) (Kalantidis et al., 2016) over all 25 layer outputs, using the learned weights to assess the relative importance of each output. The least informative layer output, as determined by its weight, is removed, and the model is retrained. This process continues recursively, removing one layer output at a time, until only a single output remains. Among all intermediate subsets of layer outputs generated during this process, we select the one that achieves the highest average accuracy across the three training tasks. The detailed evaluation and performance trend of this recursive process are presented in Section 4.

2.2 Refined Layer-wise Pooling Strategy

The second architectural modification addresses how the selected layer outputs are aggregated in the downstream model. While the SUPERB framework applies mean pooling over time and weighted average pooling across layers, wavCSE retains mean pooling for temporal aggregation, as the temporal structure of the outputs remains unchanged after selection. However, we re-evaluate the layer-wise pooling strategy to better accommodate the reduced number of selected layers. Specifically, we compare ten layer-wise pooling methods described in SUPERB-EP (Sritharan et al., 2025) and adopt the one that achieves the highest average accuracy across the three training tasks as the final pooling mechanism for layer-wise aggregation.

2.3 Feature-based Multi-task Learning

The third architectural modification redesigns the downstream model to support the learning of a unified speech embedding. While SUPERB adopts separate single-task models, wavCSE employs a feature-based MTL framework (Zhang and Yang, 2022) to jointly train multiple tasks using shared features. The architecture includes shared layers

Downstream Task	Dataset	Language
Keyword Spotting (KS)	Football Keyword (Rostami et al., 2022)	fa
Language Identification (SLI)	VoxForge (MacLean, 2018)	de, en, es, fr it, ru
Speaker Identification (SID)	Kathbath (Javed et al., 2023)	bn, gu, hi, kn ml, mr, or, pa sa, ta, te, ur
Speaker Verification (SV)	CNCeleb v1 (Fan et al., 2020)	zh
Gender Recognition (SGR)	TIMIT (Garofolo et al., 1993)	en
Age Recognition (SAR)	TIMIT (Garofolo et al., 1993)	en
Dialect Recognition (SDR)	TIMIT (Garofolo et al., 1993)	en
Emotion Recognition (ER)	AESDD (Vryzas et al., 2018)	el
Valence Recognition (VR)	IEMOCAP (Busso et al., 2008)	en
Activation Recognition (AR)	IEMOCAP (Busso et al., 2008)	en
Dominance Recognition (DR)	IEMOCAP (Busso et al., 2008)	en
Intent Classification (IC)	Fluent Speech Commands (Lugosch et al., 2019)	en

Table 1: Downstream tasks, datasets, and corresponding languages used in Phase 2 experiments.

followed by task-specific output layers, allowing the model to learn generalizable features while preserving task-specific distinctions. During training, we compute individual losses for each task and combine them using the equal-weighting loss balancing strategy (Lin and Zhang, 2023), where all task losses contribute equally to the total loss. This approach is simple, effective, and commonly used in feature-based MTL models.

3 Experimental Setup

We conduct our experiments in two distinct phases. In Phase 1, the focus is on optimizing the wavCSE architecture and training it jointly on multiple tasks to obtain a robust unified embedding. In Phase 2, the trained wavCSE model is frozen, and the learned embedding is used to train and evaluate task-specific models on unseen downstream tasks and datasets in multiple languages. All datasets are used with their standard training and test splits in both phases to ensure fair and consistent evaluation. All experiments are implemented in PyTorch and executed on an NVIDIA Quadro RTX 6000 GPU with 30 GB of memory. For optimization,[1] we employ grid search to tune the batch size and learning rate, and apply Bayesian optimization (Wu et al., 2019) to determine the optimal layer dimensions and regularization parameters.

Multi-task learning (MTL) models are typically trained on datasets jointly annotated for all target tasks (Zhang and Yang, 2022). However, to the best of our knowledge, no single dataset exists that satisfies this requirement for our work. Following the approach of Tang et al. (2017), we construct a composite MTL dataset in Phase 1 by merging task-specific datasets. Specifically, we use Google Speech Commands v1.0 (Warden, 2018) for KS, VoxCeleb v1 (Nagrani et al., 2017) for SID, and IEMOCAP (Busso et al., 2008) for ER. As all three tasks are classification problems, we train wavCSE using cross-entropy loss for each task and report accuracy as the evaluation metric.

In Phase 2, we evaluate the generalizability of the learned speech embedding across 12 downstream tasks listed in Table 1. These include seven classification tasks (KS, SLI, SID, SGR, SDR, ER, and IC), four regression tasks (SAR, VR, AR, and DR), and one verification task (SV). Each classification task is modeled using a single-layer neural network, and performance is measured by accuracy. For regression, the affective dimensions (VR, AR, and DR) are jointly modeled using a single-layer neural network and evaluated with the Concordance Correlation Coefficient (CCC), whereas SAR is evaluated separately using Mean Absolute

[1]Experimental hyperparameters are as follows. For wavCSE, the two fully connected layers had output dimensions of 512 (FC1) and 2000 (FC2). We used a batch size of 2048 during Phase 1 and 64 during Phase 2. Regularization was applied in both phases with L1 $\lambda = 1 \times 10^{-7}$ and L2 $\lambda = 1 \times 10^{-5}$.

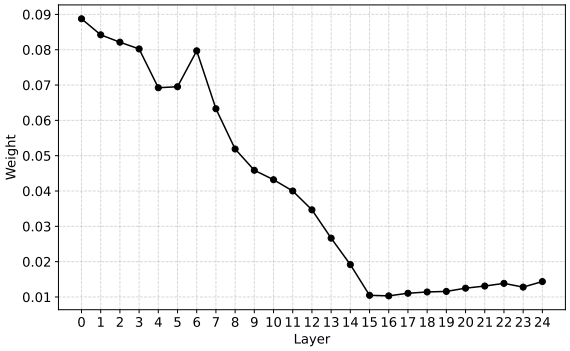

Figure 2: Layer-wise importance weights assigned by weighted average pooling in the initial wavCSE model. The x-axis denotes encoder layers (0 to 24) of WavLM-Large, and the y-axis shows the learned weight for each layer. Layer 0 represents the input to the first transformer encoder, while the others correspond to the outputs of the respective encoder layers.

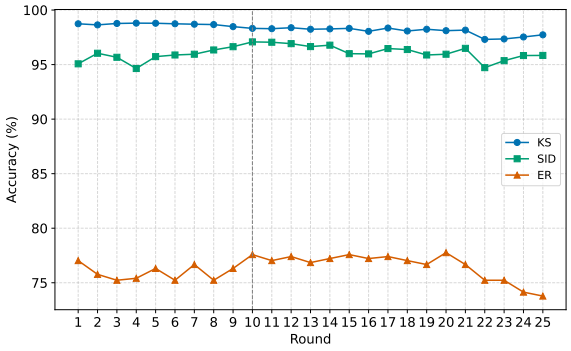

Figure 3: Performance of the initial wavCSE model across recursive elimination rounds. The x-axis shows the number of elimination rounds (1–25), and the y-axis presents task accuracy for keyword spotting (KS), speaker identification (SID), and emotion recognition (ER). Round 1 corresponds to using all 25 layer outputs.

Error (MAE). SV is performed using Probabilistic Linear Discriminant Analysis (PLDA), with performance measured by Equal Error Rate (EER).

4 Results and Discussion

In the SUPERB architecture (Yang et al., 2021), all 25 hidden layer outputs from the upstream model are aggregated using weighted average pooling (WAP). We adopt the same approach in our initial wavCSE setup and examine the distribution of learned importance weights across these 25 outputs from WavLM Large. As shown in Figure 2, lower-layer outputs consistently receive higher weights than upper layers. This suggests that lower layers capture general acoustic information transferable across tasks, whereas higher layers encode more specialized or task-specific features that contribute less to overall generalization. Motivated by this observation, we aim to eliminate less informative layers. However, defining a fixed threshold for removal is nontrivial due to potential interdependencies among layers.

To address this, we apply the recursive representation selection strategy introduced in Section 2.1. As shown in Figure 3, the highest average accuracy across KS, SID, and ER is achieved in the 10^{th} round, by which point nine layers have been eliminated. The selected subset at this round includes layers 0–14 and layer 17. These results support the earlier observation that upper layers contribute less and demonstrate that only 16 of the original 25 hidden layer outputs are sufficient to construct effective representations for downstream tasks. This also implies that loading up to the 17^{th} transformer encoder layer is sufficient when using WavLM Large as the upstream model, reducing its parameter count from 315M to 240M and memory usage from 1.175 GB to 0.894 GB.

Pooling	KS	SID	ER
Weighted Average	98.32	97.08	77.58
Max	98.43	96.59	76.85
Mean	98.23	97.10	75.95
Mixed	98.62	96.91	77.94
Gated	98.55	97.44	77.03
Learned-Norm	**98.81**	97.59	**79.39**
Log-Sum-Exp	98.52	**97.89**	77.22
Smooth-Maximum	98.36	97.18	77.94
Auto	98.45	97.41	78.12
Self-Attention	98.55	96.99	76.13

Table 2: Comparison of different layer-wise pooling strategies in wavCSE, with measured performance on keyword spotting (KS), speaker identification (SID), and emotion recognition (ER).

We further investigated whether WAP remained the most effective method for aggregating the selected layer outputs or if alternative pooling strategies could offer improved performance. To this end, we evaluated ten pooling techniques, including WAP, as described as layer-wise pooling methods in SUPERB-EP (Sritharan et al., 2025), and measured accuracy on KS, SID, and ER. As shown in Table 2, learned-norm pooling (LNP) achieved the highest average accuracy across the three tasks, outperforming all other methods on KS and ER, and ranking second on SID. Unlike WAP, which performs a linear combination of the selected layer

Model	KS	SID	ER
Vygon et al. (2021)	98.55	–	–
Hu et al. (2023)	–	95.65	–
Peng et al. (2021)	–	–	79.10
wav2vec 2.0 Large	96.66	86.14	65.64
HuBERT Large	95.29	90.33	67.62
WavLM Large	97.86	95.49	70.62
wavCSE	**98.81**	**97.59**	**79.39**

Table 3: Performance comparison of the proposed wavCSE model against task-specific models and SSL-based baselines on keyword spotting (KS), speaker identification (SID), and emotion recognition (ER).

outputs, LNP applies a non-linear transformation that adapts to their statistical distribution. These results suggest that wavCSE benefits from non-linear pooling strategies when aggregating information across layers.

Based on the experiments discussed thus far, we finalize the wavCSE architecture and now evaluate the finalized model against state-of-the-art (SOTA) baselines on the three tasks used for model development. These baselines include top-performing individual models for KS, SID, and ER (Vygon and Mikhaylovskiy, 2021; Hu et al., 2023; Peng et al., 2021), as well as self-supervised learning (SSL) models such as wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and WavLM (Chen et al., 2022). As shown in Table 3, wavCSE outperforms all baselines across the three tasks, surpassing both task-specific models and SSL-based models. These results validate the effectiveness of the proposed modifications and confirm that the finalized wavCSE model is competitive with SOTA approaches.

We next evaluate the unified speech embedding derived from the trained wavCSE model by extracting the output of the final shared layer, following Shi et al. (2020). In this second phase, the wavCSE model is kept frozen, and the extracted fixed-size embeddings are used as input features to train and evaluate lightweight task-specific models across twelve downstream tasks, as described in Section 3. Each task is benchmarked against its corresponding SOTA baseline: KS (Rostami et al., 2022), SLU (Sarthak et al., 2019), SID (Sritharan and Thayasivam, 2025), SV (Fan et al., 2020), SGR, SAR, SDR (Wang and Sun, 2024), ER (Ma et al., 2024), VR, AR, DR (Messaoudi et al., 2024), and IC (Chen et al., 2022).

Table 4 presents the performance of downstream models trained using wavCSE-based embeddings. The results show that wavCSE-based embeddings outperform the SOTA baselines on nine out of twelve tasks, demonstrating strong generalizability across a diverse range of downstream settings. The largest gains are observed on SDR and VR, likely due to the diversity of accents and emotional expressiveness captured during wavCSE development. For linguistic tasks, the embeddings improve over the best baselines on KS and SLI, while among paralinguistic tasks, consistent gains are observed on SGR, SAR, ER, AR, and DR. These findings confirm that the fixed-size embedding learned in Phase 1 transfers effectively to both linguistic and paralinguistic tasks in Phase 2.

Among the three tasks where wavCSE-based embeddings do not achieve the top performance, SID falls marginally short, differing only in the first decimal place from the best baseline. The baseline employs an upstream model trained on a multilingual corpus including low-resource Indian languages, whereas wavCSE builds on WavLM Large, pre-trained solely on English data. For SV, the results indicate a lack of fine-grained speaker-discriminative cues, despite strong performance on related speaker profiling tasks such as SGR, SAR, and SDR. The largest gap appears in IC, where the embedding achieves about 70% of the SOTA score. While linguistic tasks such as KS and SLI are handled well, IC likely requires deeper semantic abstraction not yet captured by the current embedding, motivating the inclusion of semantically oriented tasks in future wavCSE development.

5 Conclusion and Future Work

This paper presented wavCSE, a feature-based multi-task learning model built on WavLM Large, designed to learn fixed-size unified speech embeddings that support both linguistic and paralinguistic tasks. In the first phase, wavCSE was optimized through joint training on keyword spotting, speaker identification, and emotion recognition to learn a robust unified embedding that captures linguistic, speaker-related, and emotional information. In the second phase, the trained model was frozen, and the learned embedding was extracted and evaluated across twelve downstream tasks using datasets from twenty-one languages covering both high-resource and low-resource conditions. The embedding outperformed strong task-specific baselines on nine tasks and demonstrated consis-

Model	KS
ResNet	95.88
EfficientNet	95.83
wavCSE	**96.46**

Model	SLI
1D ConvNet	93.70
2D ConvNet	94.30
wavCSE	**99.23**

Model	SID
IndicWav2Vec	79.26
Sritharan et al.	**97.96**
wavCSE	97.33

Model	SV
i-vector	15.00
x-vector	**11.99**
wavCSE	16.87

Model	SGR
MLP	98.00
LSTM	99.00
wavCSE	**99.84**

Model	SAR
MLP	6.66
LSTM	5.97
wavCSE	**3.79**

Model	SDR
MLP	16.00
LSTM	15.00
wavCSE	**51.27**

Model	ER
data2vec 2.0	83.07
emotion2vec	84.85
wavCSE	**89.26**

Model	VR
LSTM	0.32
CNN1D	0.35
wavCSE	**0.67**

Model	AR
LSTM	0.67
CNN1D	0.65
wavCSE	**0.68**

Model	DR
LSTM	0.53
CNN1D	0.53
wavCSE	**0.59**

Model	IC
HuBERT Large	98.76
WavLM Large	**99.31**
wavCSE	71.00

Table 4: Comparison of downstream models trained on wavCSE-based embeddings with task-specific baselines across twelve downstream tasks. Metrics and datasets are defined in Section 3.

tent generalization across linguistic and paralinguistic domains. Although performance was slightly lower on certain speaker-related and semantically demanding tasks, the results confirm the effectiveness and transferability of the embedding learned by the proposed model.

Architectural enhancements introduced in wavCSE for generating unified embeddings are broadly applicable to any self-supervised learning-based speech processing pipeline. First, we proposed a recursive layer selection strategy to reduce the number of transformer encoder outputs used from the pre-trained WavLM Large model, resulting in a more compact and efficient upstream configuration. Second, we replaced weighted average pooling with learned-norm pooling to aggregate the selected outputs, which consistently improved task performance across training objectives. For future work, we plan to enhance the embedding's ability to capture semantic content, aiming for improved results on tasks such as intent classification and slot filling. We also intend to extend its applicability beyond classification, regression, and verification to generative tasks such as speech synthesis and automatic speech recognition.

Limitations

This work focuses on developing a unified speech embedding that supports classification, regression, and verification tasks across both linguistic and paralinguistic domains. While the embedding demonstrates strong performance in these areas, it has not been extended to generative applications such as speech synthesis or automatic speech recognition, which we leave for future work. Additionally, we intentionally avoid data augmentation to ensure that the model learns embeddings directly from raw audio, consistent with our goal of generalizable learning without task-specific heuristics. Finally, we adopt WavLM Large as the upstream model, which was pre-trained solely on English. Despite this, our unified embedding demonstrates strong performance across twenty-one languages, including low-resource settings, as shown in Section 4.

Acknowledgments

We would like to thank the National Languages Processing (NLP) Center and the DataSEARCH Research Center at the University of Moratuwa for providing the GPU resources used in this research.

References

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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

L. Bahl, P. Brown, P. de Souza, and R. Mercer. 1986. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 49–52.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2011. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Y. Fan, J.W. Kang, L.T. Li, K.C. Li, H.L. Chen, S.T. Cheng, P.Y. Zhang, Z.Y. Zhou, Y.Q. Cai, and D. Wang. 2020. Cn-celeb: A challenging chinese speaker recognition dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7604–7608.

John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, and Victor Zue. 1993. Timit acoustic-phonetic continuous speech corpus. Web Download. LDC93S1.

J.-L. Gauvain and Chin-Hui Lee. 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298.

Albert Haque, Michelle Guo, Prateek Verma, and Li Fei-Fei. 2019. Audio-linguistic embeddings for spoken sentences. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7355–7359.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Zhangfang Hu, Caiyun Lv, and Changbo Wu. 2023. Speaker recognition algorithm based on Fca-Res2Net. *IAENG International Journal of Computer Science*, 50(4):1319–1329.

Tahir Javed, Kaushal Bhogale, Abhigyan Raman, Pratyush Kumar, Anoop Kunchukuttan, and Mitesh M. Khapra. 2023. Indicsuperb: a speech processing universal performance benchmark for indian languages. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-dimensional weighting for aggregated deep convolutional features. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14*, volume 9913, pages 685–701.

Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Junaid Qadir, and Björn Schuller. 2023. Survey of deep representation learning for speech emotion recognition. *IEEE Transactions on Affective Computing*, 14(2):1634–1654.

Baijiong Lin and Yu Zhang. 2023. LibMTL: A python library for multi-task learning. *Journal of Machine Learning Research*, 24(209):1–7.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. In *Interspeech 2019*, pages 814–818.

Ziyang Ma, Zhisheng Zheng, Jiaxin Ye, Jinchao Li, Zhifu Gao, ShiLiang Zhang, and Xie Chen. 2024. emotion2vec: Self-supervised pre-training for speech emotion representation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15747–15760, Bangkok, Thailand. Association for Computational Linguistics.

Ken MacLean. 2018. Voxforge. *Ken MacLean. [Online]. Available:* `https://www.voxforge.org/`.

Awatef Messaoudi, Hayet Boughrara, and Zied Lachiri. 2024. Speech emotion recognition in continuous space using iemocap database. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–6.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: A large-scale speaker identification dataset. In *Interspeech 2017*, pages 2616–2620.

Zixuan Peng, Yu Lu, Shengfeng Pan, and Yunfeng Liu. 2021. Efficient speech emotion recognition using multi-scale cnn and attention. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3020–3024.

Dejan Porjazovski, Tamás Grósz, and Mikko Kurimo. 2024. From raw speech to fixed representations: A comprehensive evaluation of speech embedding techniques. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3546–3560.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, pages 28492–28518, Honolulu, Hawaii, USA. PMLR.

Amir Mohammad Rostami, Ali Karimi, and Mohammad Ali Akhaee. 2022. Keyword spotting in continuous speech using convolutional neural network. *Speech Communication*, 142:15–21.

Sarthak, Shikhar Shukla, and Govind Mittal. 2019. Spoken language identification using convnets. In *Proceedings of the International Conference on Ambient Intelligence*, pages 252–265. Springer.

Yanpei Shi, Qiang Huang, and Thomas Hain. 2020. H-vectors: Utterance-level speaker embedding using a hierarchical attention model. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7579–7583.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

Braveenan Sritharan and Uthayasanker Thayasivam. 2025. Advancing multilingual speaker identification and verification for Indo-Aryan and Dravidian languages. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 67–73, Abu Dhabi. Association for Computational Linguistics.

Braveenan Sritharan, Uthayasanker Thayasivam, and Supun Jayaminda Bandara. 2025. Superb-ep: Evaluating encoder pooling techniques in self-supervised learning models for speech classification. In *2025 IEEE Symposium on Computational Intelligence in Natural Language Processing and Social Media (CI-NLPSoMe)*, pages 1–7.

Zhiyuan Tang, Lantian Li, Dong Wang, and Ravichander Vipperla. 2017. Collaborative joint training with multitask recurrent model for speech and speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(3):493–504.

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. 2014. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056.

Nikolaos Vryzas, Rigas Kotsakis, Aikaterini Liatsou, Charalampos Dimoulas, and George Kalliris. 2018. Speech emotion recognition for performance interaction. *Journal of the Audio Engineering Society. Audio Engineering Society*, 66:457–467.

Roman Vygon and Nikolay Mikhaylovskiy. 2021. Learning efficient representations for keyword spotting with triplet loss. In *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings*, page 773–785, Berlin, Heidelberg. Springer-Verlag.

Rong Wang and Kun Sun. 2024. Timit speaker profiling: A comparison of multi-task learning and single-task learning approaches. *arXiv preprint arXiv:2404.12077*.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. 2019. Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 17(1):26–40.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. 2021. Superb: Speech processing universal performance benchmark. In *Interspeech 2021*, pages 1194–1198.

Wen Zhang and Juan Liu. 2007. Gene selection for cancer classification using relevance vector machine. In *2007 1st International Conference on Bioinformatics and Biomedical Engineering*, pages 184–187.

Yu Zhang and Qiang Yang. 2022. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.