Interspeech 2024
1-5 September 2024, Kos, Greece

Speaker Change Detection with Weighted-sum Knowledge Distillation based on Self-supervised Pre-trained Models

Hang Su[1,2], *Yuxiang Kong*[1], *Lichun Fan*[1], *Peng Gao*[1], *Yujun Wang*[1], *Zhiyong Wu*[2]

[1]Xiaomi Inc., Beijing, China
[2]Shenzhen International Graduate School, Tsinghua University, China
{suhang10,kongyuxiang1,fanlichun1,gaopeng11,wangyujun}@xiaomi.com,
zywu@sz.tsinghua.edu.cn

Abstract

Speaker Change Detection (SCD) is an essential problem in speech processing and has various applications in many fields. The self-supervised models have shown impressive performance on many downstream tasks in the pre-training and fine-tuning paradigm. However, it has limitations to apply a fine-tuned self-supervised pre-trained model to frame-level SCD task in real industry because it typically requires a smaller model that consumes fewer computational resources. To tackle this issue, we propose using Knowledge Distillation (KD) to leverage the capabilities of the self-supervised model. First, a basic KD method based on the pre-trained model is proposed. Then, a weighted-sum KD method is proposed to selectively extract information from the pre-trained model. Experimental results demonstrate the effectiveness of the basic KD method as well as a further improvement for the weighted-sum KD method. The proposed method is more suitable for industrial applications compared with fine-tuning.

Index Terms: Speaker Change Detection, Self-supervised Pre-trained model, Knowledge Distillation, Weighted-sum

1. Introduction

Speaker Change Detection (SCD) is the task of identifying the points in an audio stream where the speaker changes. It is a fundamental problem in speech processing and has applications in various fields such as speaker diarization [1], Automatic Speech Recognition (ASR) [2] and multi-talker audio transcribing [3]. Conventional methods for SCD computed the distance between two adjacent sliding windows, and then determined the speaker change point based on a fine-tuned threshold [4, 5, 6]. Recently, many deep learning approaches have been proposed for the SCD task. Some works detected speaker change points between words, which were usually built with an ASR system and applied together with ASR in real applications [2, 7, 8]. Other works detected speaker change points in frame-level instead of word-level, which predict speaker change points directly on a frame basis using different architecture of neural network [9, 10, 11]. Although there are some limitations of frame-level speaker change detection such as being sensitive to silence and noise [12], it still deserves to be further investigated due to its wider application and higher decision resolution.

In recent years, self-supervised pre-trained models have gained a lot of attention due to their powerful modeling capabilities and effective use of large amounts of unlabeled data. Many models have exhibited strong abilities of audio representation by being trained to predict masked parts as well as employing sophisticated loss designs, such as HuBERT [13], MAE [14], BEST-RQ [15], data2vec2 [16], etc. Many downstream tasks have achieved very good results with the support of self-supervised pre-trained models [17, 18, 19]. The question of how to appropriately apply self-supervised pre-trained models to downstream tasks remains a hot topic.

Fine-tuning a large self-supervised pre-trained model on the SCD task is a way to improve the performance of SCD [9]. However, frame-level SCD methods usually serve as a front-end audio segmentation module in practical industrial applications, which require a small model that consumes fewer computational resources and has faster inference speeds. Therefore, directly fine-tuning a self-supervised pre-trained model on the SCD task still has limitations in real industrial applications due to the large resource consumption. Consequently, we propose to do knowledge distillation based on self-supervised pre-trained models in order to leverage the capabilities of self-supervised pre-trained models without increasing the model size during inference. Furthermore, self-supervised pre-trained models encode various information in different layers, and each layer has varying degrees of suitability for different tasks [20]. To fully leverage the information in each layer and automatically select the most task-relevant information, we propose a weighted-sum knowledge distillation method. This method employs learnable weights on different layers of self-supervised pre-trained models to achieve automatic information selection. Experimental results demonstrate the effectiveness of doing knowledge distillation based on self-supervised pre-trained models for the SCD task, as well as a further improvement for the proposed weighted-sum knowledge distillation method. The best result of our proposed method shows an absolute improvement of 1.32% compared to the method without knowledge distillation in terms of F1 score on the SCD task.

In this paper, we first propose a basic knowledge distillation method based on self-supervised pre-trained models in section 2, and then propose the weighted-sum knowledge distillation method in section 3. Section 4 shows the details of experiments and section 5 draws the conclusion.

2. Basic knowledge distillation method based on self-supervised pre-trained models

Figure 1 depicts the proposed basic knowledge distillation method for SCD, which is based on self-supervised pre-trained models. The structure of the method mainly comprises a self-supervised pre-trained model and a main SCD model. In this section, we will first introduce the main SCD model, and then introduce the three self-supervised pre-trained models used in this work. Finally, the details of the training process will be described.

10.21437/Interspeech.2024-1885

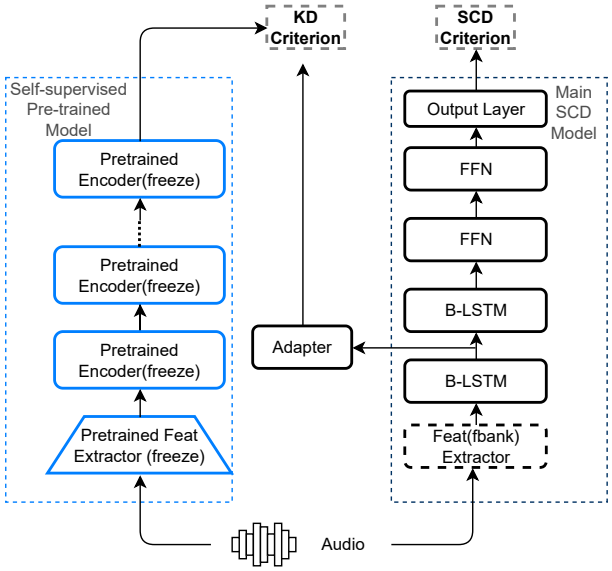

Figure 1: *Basic knowledge distillation method based on the self-supervised pre-trained model for SCD*

2.1. The main SCD model

The structure of the SCD model is the same as the model architecture proposed in [11], which consists of two Bi-LSTM layers, two fully connected feed-forward layers with tanh activation function, and one linear output layer. The input feature of the model is an 80-dimensional filter bank with a frame length of 25 ms and a frame shift of 10 ms. The output layer is followed by a softmax function to output the probabilities of two classes (No Speaker Change / Speaker Change) for each frame.

2.2. Self-supervised pre-trained models

HuBERT, BEST-RQ, and data2vec2 have been demonstrated to have strong feature representation capabilities [16, 18, 21], which are used in this work. These three models obtain powerful contextual modeling capabilities by being trained to predict masked regions. HuBERT uses multiple transformer encoder layers [22], with the input being the raw audio samples, which are then processed by a feature extractor to obtain audio encoder features. HuBERT obtains discrete pseudo-labels for the masked parts by performing k-means clustering on the hidden representations. BEST-RQ uses multiple conformer encoder layers [23], with an input of 80-dimensional filter bank features that are processed by multiple CNN layers to obtain hidden layer representation. BEST-RQ obtains discrete pseudo-labels of masked parts by mapping the features through a random-projection quantizer. Data2vec2 is structurally similar to HuBERT, but its labels of the masked region are obtained from the corresponding layer outputs of unmasked data processed by a teacher model, which is obtained by the student through the exponentially moving average algorithm. The high-dimensional representations extracted by these models will be used in the form of knowledge distillation to assist in the training of the SCD model.

2.3. Training and inference of knowledge distillation method

During training, the long audio data is first split into many 1.5-second audio segments. Then random noise from MUSAN [24] is added to audio segments for data augmentation. Then, as shown in Figure 1, audio segments are fed into both the self-supervised pre-trained model and the main SCD model. The pre-trained model serves as the teacher model whose parameters do not update during the training process. The output of the pre-trained model can be regarded as a high-level hidden representation that encodes rich audio information including the information applicable to the SCD task. The SCD model needs to learn the knowledge both from the teacher model and from the SCD task. To learn the knowledge from the teacher model, Kullback–Leibler(KL) divergence loss [25] is used to make the distribution of a certain layer in the SCD model closer to the distribution of the teacher model output. In order to calculate the KL divergence loss, the output of a certain layer in the student model with the output of the teacher model are required to be aligned in both the time and feature dimensions. Therefore, we apply a single-layer CNN followed by a tanh activation function as an adapter. The output of the adapter has the same shape as the output of the self-supervised pre-trained model. The knowledge distillation loss is defined as equation (1),

$$L_{bkd} = \sum_{i \in X} KL(t(x_i), s(x_i)) \tag{1}$$

where $t(\cdot)$ is the softmax output of the teacher model, $s(\cdot)$ is the softmax output of the adapter layer, X is the set of training samples, $KL(\cdot)$ is KL divergence between distribution p and q defined by equation (2).

$$KL(p, q) = \sum_{i} p_i log(\frac{p_i}{q_i}) \tag{2}$$

Meanwhile, cross entropy loss is applied after getting the output of the SCD model to learn the SCD task. Same as [10, 11], 200 ms on both sides of the exact change point are all labeled as speaker change, while others are labeled as no speaker change. Knowledge distillation loss and cross entropy loss are added to form the total loss.

During inference, audio segments are only required to be fed into the small SCD model. The entire inference process is identical to [10, 11], except that the length of the split audio segment is changed from 2 seconds to 1.5 seconds.

3. Weighted-sum knowledge distillation for information selection from self-supervised pre-trained model

3.1. Overview

Information represented by different layers of the self-supervised pre-trained model varies, and their adaptability to different tasks also varies. Therefore, based on the basic knowledge distillation method, we propose a weighted-sum knowledge distillation approach to fully utilize information from each layer of the pre-trained model and automatically select task-adaptive information. As shown in Figure 2, the output of each encoder layer of the pre-trained model is multiplied by a learnable weight correspondingly. All weights are mapped to a value between zero and one by a sigmoid function. The weighted sum of all encoder layers is used as the output of the teacher model. The adapter and main SCD model remain unchanged compared to the basic knowledge distillation method mentioned in section 2. SCD criterion also remains unchanged, while the criterion of the knowledge distillation changes, which will be introduced in section 3.2.

3.2. Criterion of knowledge distillation

Some previous works have mentioned the weighted-sum fine-tuning method, which added weighted representations from var-

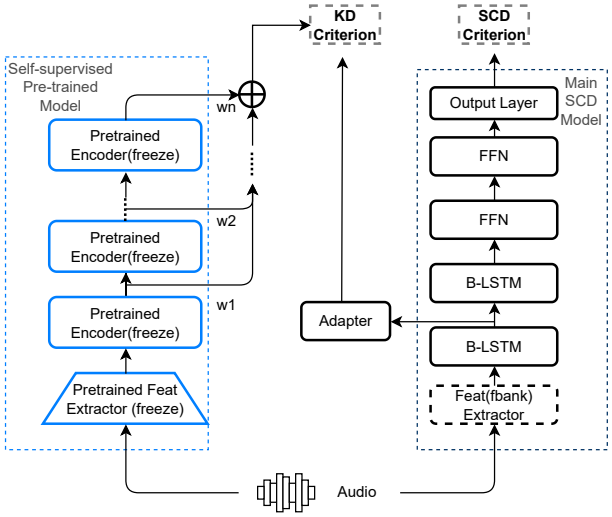

Figure 2: *Weighted-sum knowledge distillation method based on the self-supervised pre-trained model for SCD*

ious layers of the pre-trained model and fine-tuned the trainable weights along with the downstream tasks for better adaption to downstream tasks [20, 26]. Compared to the weighted-sum fine-tuning method, weighted-sum knowledge distillation is more complex because both the target and the main SCD model are trainable. We hope that the main SCD model can learn from the pre-trained model, while weights applied to the pre-trained model can be automatically adjusted based on the information learned by the main model through the SCD task for selecting more task-adaptive information. This is similar to the situation of training Vector Quantized Variational Autoencoder(VQ-VAE) model [27], where they hope to optimize the codebook based on the output of the encoder while updating the encoder based on the codebook. Therefore, inspired by VQ-VAE, the loss of knowledge distillation is designed as equation (3),

$$L_{wskd} = \sum_{i \in X} KL(sg[t(x_i)], s(x_i)) + \beta KL(t(x_i), sg[s(x_i)]) \tag{3}$$

where $t(\cdot)$, $s(\cdot)$, X and $KL(\cdot)$ have the same definition as those in equation (1), sg is the stop-gradient operator which is defined as an identity during forward computation time and has zero partial derivatives.

Since the priority of updating the main SCD model should be higher than that of updating the weights, the value of β should be less than 1. In this study, we set β to 0.25, which is also the value used in the original paper of VQ-VAE. In addition, we use different learning rates for updating the weights and the main SCD model, with the learning rate for the weights being 0.1 times that of the main network.

3.3. Training and inference

The training process is very similar to the process mentioned in section 2.3. Split audio with random noise is fed into both the self-supervised pre-trained model and the main SCD model. The output of all encoder layers of the self-supervised pre-trained model are weighted summed to be the output of the teacher model. Knowledge distillation loss mentioned in section 3.2 is calculated between the softmax output of the teacher model and the softmax output of the adapter. This loss is added to the main network cross entropy loss to obtain the total loss. The inference process is exactly the same as mentioned in section 2.3.

4. Experiments

4.1. Datasets

The AMI corpus [28] is utilized for conducting experiments on the SCD task. It is an open-source dataset comprising 100 hours of English conversational recordings. In this study, the training set, validation set, and test set were configured in the same manner as [10, 11], with approximately 70 hours, 15 hours, and 15 hours of audio data, respectively. The libri-heavy-large dataset [29] without the text label is used for the pre-training of self-supervised models, which contains about 50000 hours speech data in English.

4.2. Implementation details

In this work, we train three self-supervised models using the libri-speech-large dataset. Both HuBERT and data2vec2 implementations are based on the fairseq platform [1], using the HuBERT-base and data2vec2-base configurations, respectively. The main network architecture of HuBERT and data2vec2 consisted of 12 transformer layers with 768-dimensional hidden states. The BEST-RQ model we trained is similar in size to the HuBERT-base, consisting of 12 conformer layers with 512-dimensional hidden states, and other configurations are the same as in the paper [15]. All three models are trained using 8 A100 GPUs with 400000 iterations. The number of parameters of HuBERT, data2vec2 and BEST-RQ are 94M, 93M, and 98M respectively. The hidden size of the Bi-LSTM layers in the main SCD model is 64, and the hidden size of the linear layer is 128. The total number of parameters of the main SCD model is 0.18M, which is quite a small model. Adam [30] is applied as an optimizer with a learning rate of 0.001 for the main SCD model and 0.0001 for the learnable weights. Both learning rates are linearly decayed by a factor of 0.9 every 15 epochs.

4.3. Results and discussions

The F1 score is used as the evaluation metric in this study, which is the harmonic average of coverage and purity [6]. The model and the threshold used for speaker change decisions are selected based on the maximal F1 score on the validation set. The selected model and threshold are then applied to the test set to obtain the final test results.

The results of training the main SCD model without knowledge distillation, as well as the results of using three self-supervised pre-trained models to constrain the first layer output of the SCD model based on two proposed knowledge distillation methods are shown in Table 1. All results based on the basic knowledge distillation methods outperform the result without knowledge distillation in terms of F1 scores, regardless of which pre-trained model is used. This demonstrates the effectiveness of using self-supervised pre-trained models for knowledge distillation in the SCD task, which can improve the SCD performance without increasing model size. Moreover, all results based on weighted-sum knowledge distillation outperform those based on basic knowledge distillation in terms of F1 scores, indicating the effectiveness of our proposed weighted-sum knowledge distillation method, which is able to utilize the information from each layer and automatically select task-relevant information.

[1] https://github.com/facebookresearch/fairseq

Table 1: *The results(%) of training the main SCD model without knowledge distillation, as well as the results of using three self-supervised pre-trained models to constrain the first layer output of the SCD model based on two proposed knowledge distillation methods. KD: Knowledge distillation*

	F1	Purity	Coverage
No KD	86.46	84.46	88.56
HuBERT			
Basic KD	87.00	84.33	89.84
Weighted-sum KD	87.65	85.83	89.55
data2vec2			
Basic KD	87.17	85.15	89.28
Weighted-sum KD	87.67	85.79	89.63
BEST-RQ			
Basic KD	87.11	85.16	89.15
Weighted-sum KD	87.51	85.80	89.29

Besides, we also conduct experiments that using three self-supervised pre-trained models to constrain the second and third layer output of the SCD model based on two proposed knowledge distillation methods, which are shown in Table 2. It can be observed that regardless of which layer of the SCD model is taught using a self-supervised pre-trained model, the weighted-sum knowledge distillation method outperforms the basic knowledge distillation method. Additionally, using a self-supervised pre-trained model to constrain the first layer of SCD yields better results than constraining the other two layers based on the weighted-sum knowledge distillation method. We also attempted to constrain the fourth layer of the SCD model using self-supervised pre-trained models, but the results were not good (F1 scores are around 85%). This meets our expectation since the output of the fourth layer is only followed by a linear output layer after learning the representation from the self-supervised pre-trained model, which is insufficient for classification.

Table 2: *The results(%) of using three self-supervised pre-trained models to constrain the second and third layer output of the SCD model based on two proposed knowledge distillation methods. WS-KD: Weighted-sum knowledge distillation*

	Second layer			Third layer		
	F1	Purity	Coverage	F1	Purity	Coverage
HuBERT						
Basic KD	86.98	85.55	88.45	87.11	85.17	89.13
WS-KD	87.24	85.41	89.16	87.56	85.50	89.72
data2vec2						
Basic KD	87.14	84.69	89.73	87.19	85.63	88.80
WS-KD	87.41	85.23	89.70	87.51	86.04	89.03
BEST-RQ						
Basic KD	86.81	85.18	88.51	86.85	85.07	88.69
WS-KD	87.10	85.29	89.00	87.13	85.48	88.84

Furthermore, an experiment is conducted by using three self-supervised pre-trained models together as teachers. More specifically, the total knowledge distillation loss is obtained by adding three knowledge distillation losses calculated between the output of each pre-trained model and the output of its corresponding adapter. The experiment results of using three pre-trained models together to constrain the first layer of the main SCD model are shown in Table 3. It can be observed that the result of the weighted-sum knowledge distillation method outperforms the basic knowledge distillation method, which keeps the consistency with other experiments. Moreover, although marginal improvement is achieved compared with the result of only using data2vec2 as a teacher, using three self-supervised pre-trained models together as teachers to teach the first layer of the main SCD model based on the weighted-sum knowledge distillation method reach the best performance, which shows an absolute improvement of 1.32% compared to the method without knowledge distillation in terms of F1 score.

Table 3: *The results(%) of using three pre-trained models together as teachers to constrain the first layer output of the SCD model based on two proposed knowledge distillation methods. KD: Knowledge distillation*

HuBERT + data2vec2+ BEST-RQ	F1	Purity	Coverage
Basic KD	87.44	85.38	89.61
Weighted-sum KD	87.78	85.36	90.33

Since the proposed weighted-sum knowledge distillation method does not increase the size of the model during inference, the performance of SCD can be improved without increasing the inference time cost as well as the memory and computational resources used. This makes it more feasible to apply the method in practical industrial scenarios. Furthermore, since we obtain consistent results across multiple self-supervised pre-trained models, we believe that the weighted-sum knowledge distillation method can be generalized to other self-supervised pre-trained models.

5. Conclusions and future work

In this study, knowledge distillation methods based on self-supervised pre-trained models are proposed to improve the performance of the SCD task, which aims at leveraging the capabilities of self-supervised pre-trained models without increasing computational cost during inference. First, a basic knowledge distillation method is proposed to demonstrate the effectiveness of the knowledge distillation based on self-supervised pre-trained models for the SCD task. Experimental results demonstrate that the basic knowledge distillation method based on various self-supervised models outperforms the model trained without knowledge distillation. Then, a weighted-sum knowledge distillation method is proposed to leverage the information from multi-layers of the self-supervised pre-trained models and select the most task-relevant information, which utilizes learnable weights added to each layer of the self-supervised model with a well-designed knowledge distillation loss. Experimental results demonstrate that this method outperforms the basic knowledge distillation method on various self-supervised pre-trained models. Overall, our proposed weighted-sum knowledge distillation method is able to enhance the performance of the SCD task without increasing inference time or computational resources during inference, making it more practical for real-world applications. Additionally, this method can also be generalized to other self-supervised pre-trained models.

Our future work aims to validate the effectiveness of the proposed weighted-sum knowledge distillation method on more tasks. We believe that the proposed method should be applicable to more small model tasks.

6. References

[1] L. Bullock, H. Bredin, and L. P. Garcia-Perera, "Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7114–7118.

[2] J. Wu, Z. Chen, M. Hu, X. Xiao, and J. Li, "Speaker change detection for transformer transducer asr," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[3] H. Aronowitz and W. Zhu, "Context and uncertainty modeling for online speaker change detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8379–8383.

[4] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[5] L. V. Neri, H. N. Pinheiro, R. Tsang, G. D. d. C. Cavalcanti, and A. G. Adami, "Speaker segmentation using i-vector in meetings domain," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5455–5459.

[6] H. Bredin, "TristouNet: triplet loss for speaker turn embedding," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[7] W. Xia, H. Lu, Q. Wang, A. Tripathi, Y. Huang, I. L. Moreno, and H. Sak, "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8077–8081.

[8] Z. Fan, Z. Liang, L. Dong, Y. Liu, S. Zhou, M. Cai, J. Zhang, Z. Ma, and B. Xu, "Token-level Speaker Change Detection Using Speaker Difference and Speech Content via Continuous Integrate-and-fire," in *Proc. Interspeech 2022*, 2022, pp. 3749–3753.

[9] M. Kunešová and Z. Zajíc, "Multitask detection of speaker changes, overlapping speech and voice activity using wav2vec 2.0," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[10] H. Su, D. Zhao, L. Dang, M. Li, X. Wu, X. Liu, and H. Meng, "A multitask learning framework for speaker change detection with content information from unsupervised speech decomposition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8087–8091.

[11] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "Pyannote. audio: neural building blocks for speaker diarization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7124–7128.

[12] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4945–4949.

[13] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[15] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.

[16] A. Baevski, A. Babu, W.-N. Hsu, and M. Auli, "Efficient self-supervised learning with contextualized target representations for vision, speech and language," in *International Conference on Machine Learning*. PMLR, 2023, pp. 1416–1429.

[17] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang *et al.*, "Google usm: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[18] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *arXiv preprint arXiv:2111.02735*, 2021.

[19] L. Guo, X. Yang, Q. Wang, Y. Kong, Z. Yao, F. Cui, F. Kuang, W. Kang, L. Lin, M. Luo *et al.*, "Predicting multi-codebook vector quantization indexes for knowledge distillation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[20] S. W. Yang, P. H. Chi, Y. S. Chuang, C. I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," in *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*. International Speech Communication Association, 2021, pp. 3161–3165.

[21] B. Li, D. Hwang, Z. Huo, J. Bai, G. Prakash, T. N. Sainath, K. C. Sim, Y. Zhang, W. Han, T. Strohman *et al.*, "Efficient domain adaptation for speech foundation models," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[23] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech 2020*, 2020.

[24] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *stat*, vol. 1050, p. 9, 2015.

[26] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, Y.-H. Chen, S.-W. Li, and H.-y. Lee, "Audio albert: A lite bert for self-supervised learning of audio representation," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 344–350.

[27] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[28] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[29] W. Kang, X. Yang, Z. Yao, F. Kuang, Y. Yang, L. Guo, L. Lin, and D. Povey, "Libriheavy: a 50,000 hours asr corpus with punctuation casing and context," *arXiv preprint arXiv:2309.08105*, 2023.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980