INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Exploring wav2vec 2.0 on speaker verification and language identification

Zhiyun Fan[1,2], *Meng Li*[1], *Shiyu Zhou*[1], *Bo Xu*[1,2]

[1]Institute of Automation, Chinese Academy of Sciences, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
{fanzhiyun2017, limeng, zhoushiyu2013, xubo}@ia.ac.cn

Abstract

Wav2vec 2.0 is a recently proposed self-supervised framework for speech representation learning. It follows a two-stage training process of pre-training and fine-tuning, and performs well in speech recognition tasks especially ultra-low resource cases. In this work, we attempt to extend the self-supervised framework to speaker verification and language identification. First, we use some preliminary experiments to indicate that wav2vec 2.0 can capture the information about the speaker and language. Then we demonstrate the effectiveness of wav2vec 2.0 on the two tasks respectively. For speaker verification, we obtain a competitive result with the Equal Error Rate (EER) of 3.61% on the VoxCeleb1 dataset. For language identification, we obtain an EER of 12.02% on the 1 second condition and an EER of 3.47% on the full-length condition of the AP17-OLR dataset. Finally, we utilize one model to achieve the unified modeling by the multi-task learning for the two tasks.

Index Terms: Self-supervised, speaker verification, language identification, multi-task learning, wav2vec 2.0

1. Introduction

Recently, neural networks trained with a large amount of labeled data can meet most industrial needs in the field of speech processing [1, 2, 3, 4, 5, 6]. However, purely supervised learning seems to be inconsistent with the mechanism of human learning. Early on in their lives, human infants learn language by watching and listening to adults around them, which resembles an unsupervised learning process. Later, they learn reading and writing, which seems to be a supervised learning process. To simulate the two-stage learning process, a lot of self-supervised frameworks are proposed [7, 8, 9, 10, 11, 12].

In the field of speech processing, most self-supervised methods can be divided into two categories. One kind of method is conducted by the reconstruction loss, such as autoregressive predictive coding (APC) [13], masked predictive coding (MPC) [14] and so on. The other kind of method is conducted by contrastive predictive loss. The most representative work is the contrastive predictive coding (CPC) [15] and wav2vec [16]. The wav2vec 2.0 [17] used in this paper belongs to the latter category. Most of these self-supervised pre-training methods are applied to speech recognition. However, there is almost no work on whether pre-training methods could work on the speaker verification (SV) or the language identification (LID). In this paper, we use the framework of wav2vec 2.0 [17] to explore this feasibility.

We denote the model structure used in wav2vec 2.0 as w2v-encoder in this paper. It is illustrated in the dashed box of Fig. 1. It mainly consists of a convolutional neural network (CNN) encoder and a Transformer [18]. The CNN transfers raw waveform inputs to latent speech representations. They are fed to the Transformer after being masked and converted to context representations. A quantization module converts the latent speech representations to a discrete version which is used as the target. The whole model is trained to solve a contrastive task, which requires identifying the true quantized latent speech representations for a masked time step within a set of distractors [17]. Baevski et al. applied the pre-trained model to ultra-low resource speech recognition. Using only ten minutes of labeled data, their approach achieved word error rate (WER) of 5.7/10.1% on the clean/noisy test sets of Librispeech. The results demonstrate that the phoneme-related information is preserved during the pre-training of w2v-encoder and the downstream task such as speech recognition can benefit a lot from it. Audio is a series of complex signals that contain not only phoneme-related information but also factors about speaker, language, emotion, etc. Therefore, whether such a pre-training is effective for the SV and the LID tasks remains to be explored.

In this paper, we explore the effectiveness of self-supervised pre-training on the SV and the LID tasks. We utilize the pre-trained w2v-encoder to extract context representations, and use t-SNE [19] tools to visualize them. We find that they have distinguishability among different speakers and languages even if pre-training of wav2vec 2.0 is problem-agnostic. Moreover, we find the lower layer has the stronger distinguishability. This distinguishability is exactly what the SV and the LID tasks need. It also verifies the feasibility of applying the self-supervised pre-training to the two tasks. Thus, we attempt to fine-tune the pre-trained model on these two downstream tasks respectively. For the SV task, we obtain an EER of 3.61% on the test set of the VoxCeleb1 dataset. For the LID task, we obtain an EER of 12.02% on the 1 second condition and 3.47% on the full-length condition of the AP17-OLR dataset. Furthermore, in order to simplify the fine-tuning process and reduce model parameters, we utilize the multi-task learning to conduct the fine-tuning on the two tasks simultaneously.

2. Method

In this section, we first review the pre-training of the wav2vec 2.0 [17]. Then we introduce how to apply the pre-trained model to downstream tasks. Fig. 1 illustrates the pre-training and fine-tuning.

2.1. Pre-training of wav2vec 2.0

The left side of Fig. 1 gives an illustration of the w2v-encoder and its pre-training. The main body of the model consists of a CNN-based feature encoder, a Transformer-based context network and a quantization module. The CNN encoder stacks seven blocks, and in each block the temporal convolutions followed by a GELU activation function [20] have 512 channels with strides $(5, 2, 2, 2, 2, 2, 2)$ and kernel widths $(10, 3, 3, 3, 3, 2, 2)$. The CNN encoder maps the raw audio X into latent speech representations Z.

http://dx.doi.org/10.21437/Interspeech.2021-1280

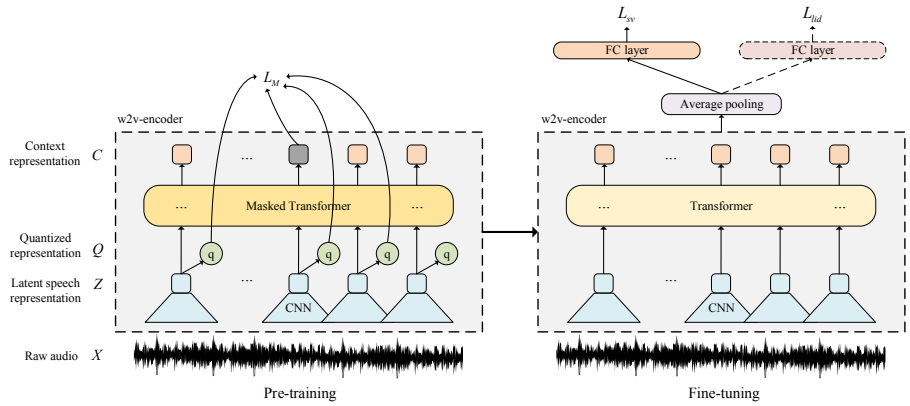

Figure 1: *An overview of the pre-training and fine-tuning. The model architectures used in pre-training stage and fine-tuning stage are identical, except the quantization modules and extra output layers.*

The context network stacks 12 Transformer blocks with model dimension 768, inner dimension $3,072$, and 8 attention heads. Before sending Z into the context network, all time steps of Z are randomly sampled as starting indices with probability $p = 0.065$, and consecutive ten time steps from every sampled index are masked. Then the relative positional embedding is added to the masked representations. The Transformer contextualizes the masked representations and generates context representations C.

The quantization module is used to discretize latent speech representations Z into Q. There are $G = 2$ codebooks in the quantization module. Each of them contains $V = 320$ entries with a size of 128. The quantization module firstly maps the Z to logits $l \in \mathbb{R}^{G \times V}$. Then the gumbel softmax [21] is used to choose one entry from each codebook in a fully differentiable way. All the entries selected are concatenated to the resulting vector $[e_1; e_2; ...; e_G]$, which is linearly mapped to q. The loss function is as follows:

$$L = L_M + \alpha L_D + \beta L_F \tag{1}$$

$$L_M = -\log \frac{\exp(sim(c_t, q_t))/k}{\sum_{\tilde{q} \sim Q_t} \exp(sim(c_t, \tilde{q}))/k} \tag{2}$$

$$L_D = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v} \tag{3}$$

$$\bar{p} = \text{GumbelSoftmax}(\bar{l}) \tag{4}$$

The loss is the weighted sum of three terms. In the Eq. 1, L_F is a $L2$ penalty. The weight β is set to 10. The L_M is the contrastive loss to make the model distinguish the true discrete representations from the latent distractors $\tilde{q}$. The distractors are uniformly sampled from other masked time steps of the same utterance. In Eq. 2, the sim represents cosine similarity, and the Q_t includes q_t and $K = 100$ distractors, and the temperature k is set to 0.1. The L_D is the diversity loss designed to increase the use of the quantized codebook representations. The α in Eq. 1 is set to 0.1. The $\bar{l}$ in Eq. 4 represents the average of logits l across utterances in a batch.

The pre-training process is optimized with Adam [22]. During the first 8% of the updates, the learning rate warms up to a peak of 5×10^{-3}, and then it decays linearly. For more details about the pre-training of wav2vec 2.0, we refer readers to [17].

2.2. Fine-tuning

Before the post-training, we add an average pooling layer and a fully connected layer on the top of w2v-encoder. The average pooling layer converts the frame-level context representations given by w2v-encoder into the sentence-level representations, and the fully connected layer classifies each sentence into some speaker or some language.

The newly added fully connected layer is randomly initialized, and w2v-encoder is initialized with the base model released by Baevski et al [1]. The cross-entropy criteria is employed as the loss function for the classification of speakers or languages. Specially, for the training of speaker classification, AM-softmax [23] is used to increase the discrimination of the learned embedding to the speaker.

In the multi-task fine-tuning, we add a pooling layer and two parallel fully connected layers to predict the speaker and language respectively. The training loss is obtained by the weighted sum of the losses of these two tasks. The L_{sv} and the L_{lid} in Eq. 5 represent the CE loss of the SV and the LID tasks respectively.

$$L_{mul} = \lambda L_{sv} + (1 - \lambda) L_{lid} \tag{5}$$

Due to the problem of unbalanced data volume in the datasets of the speakers and languages, the batch is generated by sampling from two datasets with equal probability to ensure the data used in the training process is balanced. In addition, the two tasks also have the problem of inconsistent convergence speed. We mitigate this issue by adjusting the weight of the loss of the two tasks through the development set.

3. Experiments

Various informative factors are mixed in speech signals, including semantics, speaker, emotion, etc. Baevski et al. have shown that the representations underlying pre-trained w2v-encoder can capture the linguistic factors. It remains unclear whether the problem-agnostic pre-training of wav2vec 2.0 can learn about any other factors. In the experiment part, we take speaker and language factors as examples to explore this question, and try to apply wav2vec 2.0 to the SV and the LID tasks.

[1] https://github.com/pytorch/fairseq/blob/master/examples/wav2vec/

3.1. Datasets

VoxCeleb1 [24] and AP17-OLR [25] datasets are used in our experiments for the SV and the LID respectively.

Speaker verification dataset: VoxCeleb1 [24] contains over 100,000 utterances from 1,251 celebrities. It can be used for both speaker identification and verification. We use the VoxCeleb1 to conduct the SV task. And the consine distance is used to calculate the similarity score. The data split of the VoxCeleb1 dataset for verification is listed in Table 1.

Table 1: *Data split of the VoxCeleb1 dataset for verification.*

	Train	**Valid**	**Test**
#speakers	1211	1145	40
#Utterances	143642	5000	4874
Dur(hrs.)	329.06	11.34	11.20

Language identification dataset: AP17-OLR [25] consists of 10 different languages (Mandarin, Cantonese, Indonesian, Japanese, Russian, Korean, Vietnamese, Kazakh, Tibetan and Uyghur). The duration of training data for each language is about 10 hours with the speech sampled at 16 kHz. The test set contains three subsets with different durations (1 second, 3 second, and full length). These subsets respectively contain 17964, 16404 and 17964 utterances.

3.2. Model description

In the experiments, we utilize the base model released by Baevski et al. and three models fine-tuned by us. For simplicity, we use some symbols to represent them, and the explanations are as follows:

- **M-nofinetune**: the base model pre-trained on the Librispeech corpus [26].
- **M-sv**: We fine-tune M-nofinetune on the VoxCeleb1 dataset for speaker verification.
- **M-lid**: We fine-tune M-nofinetune on the AP17-OLR dataset for language identification.
- **M-multi**: We fine-tune M-nofinetune on the AP17-OLR and VoxCeleb1 dataset simultaneously in a multi-task form.

3.3. Feasibility analysis

In this section, we explore whether the speaker and language factors are retained during the pre-training of wav2vec 2.0. It determines whether the pre-training method can be used for these two tasks.

We directly extract context representations from the test set of AP17-OLR and VoxCeleb1 with the M-nofinetune model. Then we visualize the context representations by t-SNE [19], a nonlinear dimensionality reduction algorithm for visualizing high-dimensional data. The results are shown in Fig. 2. The left three images are the visualization results of the features from the three layers of the Transformer. Different colors represent different speakers. It is not difficult to find that all the three layers have certain speaker distinguishability, and this distinguishability is more obvious at the bottom of the Transformer. In the three images on the right, different colors represent different languages. It can also be found that these features are distinguished by languages, and the lower the layer, the stronger the distinction. The phenomena presented in Fig. 2 show that the model pre-trained by wav2vec 2.0 can effectively extract the characteristics of the speaker and language of the speech.

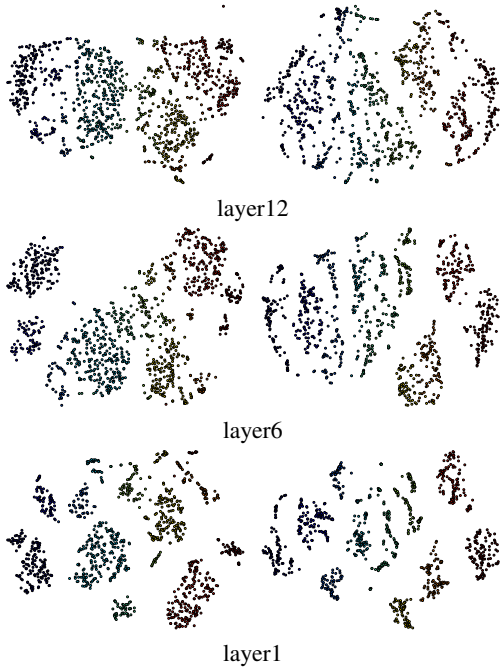

Figure 2: *2D t-SNE plot of representations extracted from the bottom layer (layer1), the middle layer (layer6), and the high layer (layer12) of the Transformer of M-nofinetune. The left column is the clustering of the representations of 10 speakers in the test set of VoxCeleb1 extracted by M-nofinetune, and each color represents a speaker. The right column is the clustering of the representations of 1000 samples in the test set of the AP17-OLR extracted by M-nofinetune, and each color represents a language.*

We further quantify this claim by performing the SV and the LID with a simple fully connected layer. The pre-trained w2v-encoder (M-nofinetune) acts as a feature extractor. The fully connected layer is optimized to distinguish the 10 languages or 1211 speakers for the two tasks respectively. The test results are listed in Table 2.

Table 2: *The EER (%) results of using context representations extracted by M-nofinetune to finish the SV and the LID.*

Model	**SV**	**LID**
random	47.93	50.05
M-nofinetuning	15.62	42.34

The $random$ results are evaluated on the randomly initialized w2v-encoder. The comparison of these two results in Table 2 further illustrates that the model pre-trained by wav2vec 2.0 can extract speaker and language-related characteristics, which provides a basis for the application of wav2vec 2.0 to the SV and the LID.

3.4. Speaker verification

From the experiments in the previous section, we can see that the pre-trained w2v-encoder, M-nofinetune, can extract features that contain a certain speaker distinguishability. This kind of

Table 3: *Comparison with the previously published EER (%) results on Voxceleb1 dataset.*

Model	EER
I-vectors + PLDA [24]	8.8
CNN + Embedding [24]	7.8
LDE-ASoftmax [27]	4.41
attentive statistics [28]	3.85
siamese capsule [29]	**3.14**
no pre-training	24.28
M-sv	3.61

speaker distinguishing learning is exactly required in the SV task. Then we attempt to fine-tune the pre-trained w2v-encoder, M-nofinetune, to finish the SV task. We initialize the w2v-encoder with M-nofinetune, and add a randomly initialized fully connected layer on the top of it to predict speakers. The fine-tuning is conducted on the VoxCeleb1 dataset [24]. All parameters are adjustable during fine-tuning. However, at the first 10000 steps, the w2v-encoder is frozen. We optimize the model with Adam, the learning rate warms up to 5×10^{-3} during the first 6000 steps, and then it decays linearly during the remaining 7000 steps.

The $no\ pre\text{-}training$ in Table 3 represents training the w2v-encoder from scratch. Our fine-tuning model, M-sv, outperforms the $no\ pre\text{-}training$ result by a significant margin (EER of 3.61% vs 24.28%). The gap between them illustrates the benefits of pre-training. Moreover, our model outperforms most of the baselines in Table 3, and obtains competitive results on the VoxCeleb1 dataset. It means the pre-training of wav2vec 2.0 is useful to the SV task and can work well without any task-specific adjustment of model structure.

3.5. Language identification

Although Baevski et al. only used English data during the pre-training of M-nofinetune, it can be seen from the visualization results in section 3.3 that the features extracted by M-nofinetune still retain the distinction of language. It means that the model obtained by this pre-training method may be useful to the language identification system. Similarly, we add a fully connected layer on top of the w2v-encoder to predict language. We initialize w2v-encoder with M-nofinetune and randomly initialize the extra fully connected layer. Then the whole model is fine-tuned on the AP17-OLR dataset [25]. We optimize the model with Adam, the learning rate warms up to 5×10^{-3} during the first 5000 steps, and then it decays linearly during the remaining 8000 steps. The parameters of the w2v-encoder part are frozen at the first 5000 steps. After training, we test on the model, which obtains the best performance on the development set.

Table 4: *Comparison with the previously published C_{avg} and EER (%) results on AP-17 dataset.*

Model	1 second		Full-Length	
	C_{avg}	EER	C_{avg}	EER
i-vector + PLDA[25]	0.1746	17.51	0.0596	5.86
TDNN [25]	0.1282	14.04	0.1034	11.31
TSM-DNN-BN-LSTM [30]	**0.067**	**6.95**	**0.007**	**0.86**
no pre-training	0.2813	29.25	0.1254	13.83
M-lid	0.1158	12.02	0.0310	3.47

The first two rows in Table 4 are the two baselines released by the organizer of the AP17-OLR challenge. The $TSM\text{-}DNN\text{-}BN\text{-}LSTM$ [30] is one of the best models on this benchmark. The $no\ pre\text{-}training$ in Table 4 means that the fine-tuning starts from scratch on the AP17-OLR dataset. The M-lid, which is fine-tuned from the pre-trained M-nofinetune, outperforms the $no\ pre\text{-}training$ result by a large margin on both the 1 second condition and the full-length condition. The gap between them illustrates the benefits brought by pre-training to the LID task. Compared with baselines released by the organizer, M-lid shows a clear performance advantage on the two test conditions. However, it is far from the best results. It means that wav2vec 2.0 is useful to the LID task. However, its effectiveness on the LID task is not good as the SV task. We consider that the use of multiple languages during pre-training (not just English) can mitigate this issue. In addition, we find that the performance of the $no\ pre\text{-}training$ is influenced by overfitting seriously. This problem is obviously alleviated during the fine-tuning of M-lid, which benefits from pre-training.

3.6. Multi-task system

The wav2vec 2.0 consists of a problem-agnostic pre-training and a task-related fine-tuning. Fine-tuning two models for the SV and the LID tasks independently will consume a lot of time and resources. Hence, we try to use one model to finish these two tasks simultaneously. On the top of the w2v-encoder we connect two fully connected layers in parallel to predict the speaker and language respectively. We follow the experiment settings described in section 2.2. The λ in Eq. 5 is set to 0.7.

Table 5: *Performance of single-task model and multi-task model on the VoxCeleb1 and the AP17-OLR dataset in terms of EER(%).*

Model	SV	LID
M-sv	3.61	-
M-lid	-	3.47
M-multi	4.18	4.88

Results in Table 5 show that compared with single-task training, although the performance of multi-task form is a bit reduced, it achieves good results with fewer parameters on the SV and the LID tasks. It shows that the pre-training of wav2vec 2.0 can be combined with multi-task learning to achieve unified modeling of the two tasks. This greatly simplifies the use of pre-trained model and can save a lot of time spent on fine-tuning to each task. In addition, it can reduce the demand for storage.

4. Conclusion

In this paper, we explore the application of wav2vec 2.0 on the SV and the LID tasks. First of all, through some preliminary experiments and visualization methods, we find that the features extracted by the pre-trained w2v-encoder have the distinction between speakers and languages, and this distinction is more obvious in lower layers. This illustrates the feasibility of using the pre-trained model for the two tasks. We further verify the effectiveness of the pre-trained model on the two tasks and obtain competitive results on the VoxCeleb1 and the AP17-OLR datasets. Finally, we use a multi-task learning mechanism to simplify the fine-tuning process on the SV and the LID, and realize the unified modeling for the two tasks.

5. References

[1] J. Li, X. Wang, Y. Li, and Y. Zhao, "The speech transformer for large-scale mandarin chinese speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7095–7099.

[2] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," in *Proc. ISCA Interspeech*, 2019, pp. 2130–2134.

[3] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5791–5795.

[4] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "**Voxceleb: La**rge-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[5] B. Padi, A. Mohan, and S. Ganapathy, "Attention based hybrid i-vector blstm model for language recognition." in *Proc. ISCA Interspeech*, 2019, pp. 1263–1267.

[6] P. Shen, X. Lu, S. Li, and H. Kawai, "Interactive learning of teacher-student model for short utterance spoken language identification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5981–5985.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019, pp. 4171–4186.

[8] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2018, pp. 2227–2237.

[9] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[10] O. J. Hénaff, "Data-efficient image recognition with contrastive predictive coding," in *Proc. International Conference on Machine Learning, ICML*, 2020, pp. 4182–4192.

[11] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 535–15 545.

[12] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6989–6993.

[13] Y. Chung, W. Hsu, H. Tang, and J. R. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. ISCA Interspeech*, 2019, pp. 146–150.

[14] D. Jiang, X. Lei, W. Li, N. Luo, Y. Hu, W. Zou, and X. Li, "Improving transformer-based speech recognition using unsupervised pre-training," *arXiv preprint arXiv:1910.09932*, 2019.

[15] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[16] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. ISCA Interspeech*, 2019, pp. 3465–3469.

[17] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[19] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[20] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," *arXiv preprint arXiv:1606.08415*, 2016.

[21] E. J. Gumbel, "Statistical theory of extreme values and some practical applications," *NBS Applied Mathematics Series*, vol. 33, 1954.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[24] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. ISCA Interspeech*, 2017, pp. 2616–2620.

[25] Z. Tang, D. Wang, and Q. Chen, "AP18-OLR challenge: Three tasks and their baselines," in *Asia-Pacific Signal and Information Processing Association APSIPA*, 2018, pp. 596–600.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[27] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[28] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. ISCA Interspeech*, 2018, pp. 2252–2256.

[29] A. Hajavi and A. Etemad, "Siamese capsule network for end-to-end speaker recognition in the wild," *arXiv preprint arXiv:2009.13480*, 2020.

[30] Z. Ma, H. Yu, W. Chen, and J. Guo, "Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features," *IEEE transactions on vehicular technology*, vol. 68, no. 1, pp. 121–128, 2018.