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

Improving Accent Identification and Accented Speech Recognition Under a Framework of Self-supervised Learning

Keqi Deng[1,2,*], *Songjun Cao*[1,*], *Long Ma*[1]

[1]Tencent Cloud Xiaowei, Beijing, China
[2]University of Chinese Academy of Sciences, China
dengkeqi20@mails.ucas.ac.cn, {songjuncao, malonema}@tencent.com

Abstract

Recently, self-supervised pre-training has gained success in automatic speech recognition (ASR). However, considering the difference between speech accents in real scenarios, how to identify accents and use accent features to improve ASR is still challenging. In this paper, we employ the self-supervised pre-training method for both accent identification and accented speech recognition tasks. For the former task, a standard deviation constraint loss (SDC-loss) based end-to-end (E2E) architecture is proposed to identify accents under the same language. As for accented speech recognition task, we design an accent-dependent ASR system, which can utilize additional accent input features. Furthermore, we propose a frame-level accent feature, which is extracted based on the proposed accent identification model and can be dynamically adjusted. We pre-train our models using 960 hours unlabeled LibriSpeech dataset and fine-tune them on AESRC2020 speech dataset. The experimental results show that our proposed accent-dependent ASR system is significantly ahead of the AESRC2020 baseline and achieves 6.5% relative word error rate (WER) reduction compared with our accent-independent ASR system.

Index Terms: self-supervised pre-training, accent identification, accented speech recognition

1. Introduction

In real scenarios, accent is one of the main and common sources of speech variability, which poses a huge challenge to automatic speech recognition (ASR). People coming from different countries or regions have their own distinctive accents and pronunciations. The differences between accents are mainly reflected in three aspects: stress, tone and length, which are challenging for ASR modeling [1]. Although different accents may share some similarities, there are obvious differences at the phonological level. As a result, the ASR system that trained on several kinds of accented speech simultaneously may fail to generalize well for each individual accent. Therefore, it is beneficial to leverage the accent feature for the ASR model when recognizing accented speech.

However, since the accent category of the speech is not always provided in real scenarios, we still need an accent identification model to provide accent-related information and features for ASR systems. Compared with the individual-level identification task like speaker identification, accent identification throws a more challenging issue in extracting compact group-level features. Without a good discriminative accent feature space, over-fitting phenomenon always happens to accent identification. In addition, the phonological differences caused by accents are inconsistent in different words. Therefore, dynamically adjusting the accent-related information for the ASR system according to this inconsistency is still a valuable challenge. Furthermore, for the accent identification and accented speech recognition tasks, labeled data is much harder to get than unlabeled data. So it is meaningful to explore the self-supervised pre-training methods [2, 3, 4, 5] to alleviate this problem in real scenarios.

In this paper, we propose a novel architecture for accent identification. Different from x-vector [6, 7], we directly identify the accent based on each frame rather than sentence-level vector, and then calculate the mean value of all frame-level outputs as the model's final prediction. In addition, we propose the standard deviation constraint loss (SDC-loss), which is based on the standard deviation of the frame-level outputs. The SDC-loss requires the predictions of each frame to be consistent, thereby curbing overfitting. As for accented speech recognition, we design an accent-dependent ASR system, which can utilize additional accent input features. Furthermore, for situations where the ground truth of accent category is not provided, we propose a frame-level accent feature, which is extracted based on the proposed accent identification model and can be dynamically adjusted according to frame-level information. We pre-train our models using 960 hours unlabeled LibriSpeech dataset following the wav2vec 2.0 [8] and fine-tune them on AESRC2020 [1] speech dataset. The experimental results show that our proposed accent-dependent ASR system can outperform all the previous baseline after fine-tuning.

The rest of this paper is organized as follows. In Section 2, we introduce the related works. In Section 3, we describe the proposed architectures. The experiments and conclusions are presented in Sections 4 and 5, respectively.

2. Related works

To distinguish different English accents, Teixeira et al. [9] propose contextual HMM units and Deshpande et al. [10] employ format frequency features into GMM. More recently, Shi et al. [1] propose an end-to-end (E2E) architecture for accent identification, but the phenomenon of over-fitting is still hard to avoid. Transfer learning and multi-task [11, 12] like using ASR downstream task to initialize the accent identification model is effective [1]. But this puts higher requirements on labeled data and self-supervised learning is more meaningful in real scenarios.

In general, accented ASR related research mainly falls into two ways: "multi-model" and "single-model" [13]. In multi-model approaches, an individual acoustic model is trained for each dialect accent when each accent's data is enough [14, 15]. In single-model approaches, a single acoustic model is trained to deal with all dialect accents [16, 17, 18]. And Li et al. [19] incorporate accent information into a single ASR model by a

[*] Equal contribution.

http://dx.doi.org/10.21437/Interspeech.2021-1186

1-hot representation. But the representation they proposed is sentence-level and cannot be used in scenarios that do not provide true accent categories. In this paper, we propose a frame-level accent feature and design a single accent-dependent ASR system to recognize all accented speech.

In machine learning, self-supervised learning has gained success in many fields [20, 21]. Wav2vec [22] learns the representations of raw audio by solving a self-supervised context-prediction task. In addition, Baevski et al. propose vq-wec2vec [23] that learns discrete speech representations through a context prediction task instead of reconstructing the input. Furthermore, Baevski et al. [8] propose wav2vec 2.0, which learns the discrete speech units and contextual representations end-to-end. In this paper, we employ the wav2vec 2.0 pre-training method.

3. Proposed Method

Under the framework of wav2vec 2.0 [8], our main focus is on fine-tuning stage. We first propose a SDC-loss based accent identification architecture. For accented speech recognition, we mainly explore how to utilize additional accent input features to improve accent-dependent ASR system. Both architectures contain a CNN based feature encoder, a Transformer based context network and a fully connected layer.

3.1. Accent identification

The proposed accent identification model is shown in Fig. 1, where CNN and $\mathrm{Transformer}$ denote CNN based feature encoder and Transformer based context network, respectively. $\mathbf{h}_t$ is the output of the Transformer [24] at t step. FC is the fully connected layer and converts the $\mathbf{h}_t$ to $\mathbf{a}_t$ whose dimension is equal to the number of accent categories. Mean and Std denote calculating the mean and standard deviation.

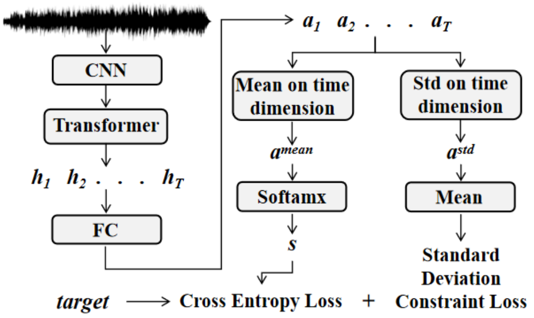

Figure 1: *Illustration of the proposed accent identification architecture.*

The proposed accent identification model directly predicts the accent category for each frame instead of sentence-level vector, and the final prediction is got by averaging. Suppose $\mathbf{a} = (\mathbf{a}_1 \cdots \mathbf{a}_T)$ is the output of FC and C denotes the number of total accent categories. We first calculate the mean and standard deviation of $\mathbf{a}$ and denote them as $\mathbf{a}^{mean}$ and $\mathbf{a}^{std}$, respectively. Then the SDC-loss is defined as:

$$\mathcal{L}_{SDC} = \frac{1}{C}\sum_{j=1}^{C} a_j^{std} \tag{1}$$

We set $\mathbf{a}^{mean}$ as the final prediction, and compare it with the target after softmax to obtain the cross entropy loss (CE-loss).

$$\mathcal{L}_{CE} = \text{Cross-entropy}(\mathbf{a}^{mean}, \mathbf{g}^{true}) \tag{2}$$

where $\mathbf{g}^{true}$ represents the target and $\mathcal{L}_{CE}$ denotes the CE-loss. The final loss function is obtained by adding the SDC-loss and CE-loss together.

$$\mathcal{L}_{final} = \mathcal{L}_{CE} + \mathcal{L}_{SDC}, \tag{3}$$

where $\mathcal{L}_{final}$ denotes the final loss function. The $\mathcal{L}_{final}$ requires not only that the final prediction matches the target, but also that the difference between the predictions of each frame to be small. This can not only suppress over-fitting, but also has further significance for extracting frame-level accent features for ASR system, which will be explained in next section.

3.2. Accented speech recognition

The proposed accented speech recognition model is based on CTC and shown in Fig. 2, where $\mathbf{a_i}$ is the output of accent identification model as shown in Fig. 1. Expand means broadcasting the data to expand the size same as the dot multiplied object. $\odot$, $\otimes$ and $\oplus$ denote dot product, scalar multiplication and point-wise plus respectively. $\mathrm{FC}_{\mathrm{CNN}}$ and $\mathrm{FC}_{\mathrm{Trans}}$ are the fully connected layer for CNN and $\mathrm{Transformer}$ respectively. $\mathbf{g}^{true}$ is a one-hot vector, which corresponds to the ground truth of the utterance's accent category, and we expand its size as frame level.

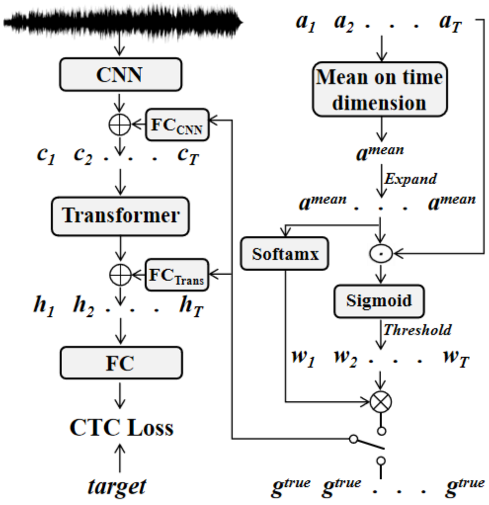

Figure 2: *Illustration of the proposed accented speech recognition architecture.*

But in the real scenarios, the true accent category is not always provided. Therefore, we need to use the accent identification model to generate accent-related information to improve the accent-dependent ASR system. Next, we will separately describe the accented speech recognition in these two conditions.

3.2.1. Accented speech recognition with true accent category

Supposed the outputs of CNN and Transformer are $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{h}}_i$, where $i \in (1, T)$. When the true accent category is provided, the $\mathbf{g}^{true}$ is linearly transformed by the $\mathrm{FC}_{\mathrm{CNN}}$ and added to $\hat{\mathbf{c}}_i$. Similarly, we also add it to the $\hat{\mathbf{h}}_\mathbf{i}$ through the $\mathrm{FC}_{\mathrm{Trans}}$.

$$\begin{aligned} \mathbf{c}_i &= \hat{\mathbf{c}}_i + \mathrm{FC}_{\mathrm{CNN}}(\mathbf{g}^{true}), & (4) \\ \mathbf{h}_i &= \hat{\mathbf{h}}_i + \mathrm{FC}_{\mathrm{Trans}}(\mathbf{g}^{true}), & (5) \end{aligned}$$

This effectively enables the accent-dependent ASR system to learn accent-related biases.

3.2.2. Accented speech recognition without true accent category

In this part, we describe our proposed accent-dependent ASR system when the true accent category is not provided. In traditional method, the accent-related features obtained from the ground truth of accent category are sentence-level [18, 19], which have no specific requirement for frame-level vectors before pooling. However, since the accent identification model we proposed is based on frame-level vector and the SDC-loss requires the prediction of each frame match the final prediction, we can get frame-level information. We believe that in accented speech recognition, it is not necessary to add accent-related bias for every frame. Instead, different weights should be given according to the importance of each frame for accent identification. In this paper, we propose frame-level accent feature, which utilizes the frame-level information to adjust the weight of the accent bias for each frame. The details are as follows:

$$w_i = \mathrm{Sigmoid}(\mathbf{a}_i \cdot \mathbf{a}^{mean}), \tag{6}$$

where $\cdot$ is the dot product and w_i is the weight of the i-th accent feature. We also set a threshold k for w_i:

$$w_i = \begin{cases} w_i, & w_i > k \\ 0, & w_i < k \end{cases} \tag{7}$$

Finally, the accented-related bias is added to the output of the CNN and Transformer:

$$\mathbf{c}_i = \hat{\mathbf{c}}_\mathbf{i} + \mathrm{FC}_{\mathrm{CNN}}(w_i \cdot \mathrm{Softmax}(\mathbf{a}^{mean})), \tag{8}$$

$$\mathbf{h}_i = \hat{\mathbf{h}}_\mathbf{i} + \mathrm{FC}_{\mathrm{Trans}}(w_i \cdot \mathrm{Softmax}(\mathbf{a}^{mean})), \tag{9}$$

4. Experiments

4.1. Corpus

We pre-train both the accent identification model and the accented speech recognition model on the Librispeech corpus [25] without transcriptions containing 960 hours of audio (LS-960). And we fine-tune the models on the AESRC2020 speech corpus [1]. The AESRC2020 speech corpus contains 160 hours speech data and includes 8 accents: English, America, China, Japan, Russia, India, Portugal, Korea. The training set contains around 120000 utterances and the testing set we used consists of around 12000 utterances.

4.2. Model descriptions

We use the Fairseq toolkit [26] to build the models. For the acoustic input, we employ the waveform following wav2vec 2.0 [8]. For the text output, we used a 28 vocabulary set, including 26 English characters and 2 punctuations.

CNN based feature encoder consists of seven convolution layers (i.e., 512 channels with kernel size 10,3,3,3,3,2 and 2 and strides 5,2,2,2,2,2 and 2). Transformer based context network contains 12 Transformer layers with 768 model dimensions, 3072 inner dimensions (FFN) and eight attention heads. The dimension of FC in accent identification model is 8. In accented ASR system, the dimensions of the $\mathrm{FC}_{\mathrm{CNN}}$ and $\mathrm{FC}_{\mathrm{Trans}}$ are 512 and 768, respectively. And the threshold k in Eq. 7 is set to 0.4.

During pre-training, we follow by Fairseq recipe [26]. During fine-tuning, we use a batch size of 3.2M samples per GPU. For accent identification tasks, we fine-tune the model by the Adam [27] optimizer (learning rate 2e-5) with 1600 warm steps on 4 GPUs. The max update number is 3000 and for the first 2000 updates only the FC is optimized, after which the Transformer based context network is also updated. The CNN based feature encoder is fixed during fine-tuning. As for accented speech recognition task, we also employ the Adam optimizer (learning rate 2e-5) with 8000 warm steps and we train the model on 8 GPUs. The max update number is 40000 and all the networks can be updated except the CNN based feature encoder. In order to prevent overfitting, we used dropout [28] (dropout rate=0.1) after each sub-layer of the Transformer based context network and the output of the CNN based feature encoder. We also employ the layer dropout method (dropout rate=0.1). During decoding, the beam size is fixed as 500 and the word insertion penalty [29] is -0.52.

4.3. Accent identification results

First, we extract the i-vector [30] and x-vector [6] using ASV-Subtools [31], based on which we train a logistic regression model for accent identification. Second, based on the pre-trained model, a mean + std pooling layer is applied after $\mathrm{Transformer}$ to pool the $\mathbf{h}_t$, which is the same as the method [1]. We denote this model as $\mathbf{AI}_1$ and train it end-to-end. At last, we denote the proposed SDC-loss based accent identification model as $\mathbf{AI}_2$. As for ablation study, we do not introduce the SDC-loss of $\mathbf{AI}_2$ and mark it as $\mathbf{AI}_3$. The accent identification results are as shown in Table 1, where B, A, C, J, R, I, P, K represent the accent of British, America, China, Japan, Russia, India, Portugal, Korea. AESRC-12L and AESRC-3L denote the results of 12-layer and 3-layer Transformer in AESRC2020 [1]. And the ”+ASR init” employs the ASR initialization method [1] based on the ”AESRC-12L”.

Table 1: *The accuracy (%) of the accent identification on AESRC2020.*

Method	B	A	C	J	R	I	P	K	All
AESRC-12L	85.0	21.2	38.2	42.7	49.6	66.1	51.8	26.0	47.8
AESRC-3L	70.0	45.7	56.2	48.5	30.0	83.5	57.2	45.0	54.1
+ASR init	93.9	60.2	67.0	73.2	75.7	97.0	85.5	55.6	76.1
x-vector [6]	66.0	46.4	62.2	53.5	59.9	77.4	54.7	56.1	59.1
i-vector [30]	72.1	48.8	70.9	45.9	64.3	87.4	58.0	58.1	62.6
AI_1	67.7	87.2	61.5	94.8	51.4	63.0	82.0	74.7	72.7
AI_2	83.2	89.6	54.0	95.8	52.6	65.2	82.7	70.2	**73.9**
AI_3	76.3	87.2	61.5	94.8	51.41	63.0	82.1	74.7	72.9

The results in Table 1 show that, based on the self-supervised pre-trained model, end-to-end fine-tuning outperforms the vanilla x-vector and i-vector methods. In addition, the $\mathbf{AI}_1$ are sentence-level accent identification model, while the $\mathbf{AI}_2$ and $\mathbf{AI}_3$ identify the accent based on frame-level vector. From the results we can see that the proposed method $\mathbf{AI}_2$ outperforms the sentence-level model $\mathbf{AI}_1$. Furthermore, through the ablation study, the proposed SDC-loss that encourages the prediction of each frame to be consistent does improve the model's identification performance. It puts forward higher requirements on the model's prediction, thereby curbing overfitting. At last, although AESRC-12L has more parameters than AESRC-3L, due to over-fitting, its results are worse. Our proposed $\mathbf{AI}_2$ also contains 12-layer Transformer, but due to the self-supervised pre-training method and the proposed SDC-loss, $\mathbf{AI}_2$ greatly outperforms AESRC-3L and AESRC-12L. After employing the ASR initialization method [1], AESRC-

12L's performance has been greatly improved. But we think that this method requires more supervised labeled information and the self-supervised pre-training method is more meaningful in real scenarios. And under the framework of self-supervised learning, we get close results.

4.4. Accented speech recognition results

As for accented speech recognition, based on the pre-trained model, we apply a FC and fine-tune it based on CTC loss. We mainly investigate utilizing the additional accent input features obtained from ground truth or the proposed accent identification model for the accent-dependent ASR system.

4.4.1. With true accent category

With the true accent category, we do not need an accent identification model. First, we train an accent-independent ASR model, which ignores the accent category and is denoted as $\mathbf{AR}_0$. And we denote the proposed architecture as $\mathbf{AR}_1$, which chooses $\mathbf{g}^{true}$ as accent vector and is illustrated in Fig. 2. As for ablation study, 1. adding the accent-related bias only after the Transformer based context network ($\mathbf{AR}_2$); 2. concatenating the output of the FC_{Trans} with the $\hat{\mathbf{h_i}}$ as $\mathbf{h}_i$ in Eq. 9 ($\mathbf{AR}_3$); 3. taking the $\mathbf{AR}_0$ as starting point and only updating accent-dependent output layers (eight in total and one for each accent) for extra 3500 update numbers ($\mathbf{AR}_4$).

Table 2: *The WER (%) of the accented speech recognition on AESRC2020 with true accent category provided.*

Method	B	A	C	J	R	I	P	K	All
AR_0	7.50	6.29	10.64	8.21	7.46	7.86	6.22	5.00	7.37
AR_1	7.44	5.99	9.67	7.48	7.13	7.07	5.86	4.72	**6.89**
AR_2	7.15	5.91	10.81	7.74	7.11	7.71	5.87	4.68	7.09
AR_3	7.48	6.05	10.69	7.96	7.12	7.64	5.91	4.78	7.17
AR_4	7.21	6.18	10.64	7.91	7.14	7.58	6.04	4.95	7.18

The results in Table 1 show that the proposed $\mathbf{AR}_1$ achieves 6.4% relative word error rate (WER) reduction compared with the $\mathbf{AR}_0$, which proves that the accent-related bias does improve the ASR system's performance on accented speech. Through ablation study, we found that 1. adding the accent-related bias after both CNN and Transformer achieves the best; 2. adding the accent-related bias to the $\hat{\mathbf{h_i}}$ outperforms concatenating them. 3. the improvement of employing accent-dependent output layers is limited.

4.4.2. Without true accent category

In real scenarios, the true accent category is not always provided. In this part, we utilize the accent identification model to produce the accent-related feature for the ASR. We denote our proposed architecture shown in Fig. 2 as $\mathbf{AR}_5$. As for ablation, 1. we do not introduce the threshold k on the basis of $\mathbf{AR}_5$ and denote it as $\mathbf{AR}_6$; 2. we do not introduce the weight of accent-related bias w_i in Eq. 6 and denote it as $\mathbf{AR}_7$. In this ways, $\mathbf{AR}_7$ can be regarded as all w_i equals to 1 and thus can not utilize the frame-level information.

From the results in Table 3, we can see that with the proposed frame-level accent features, the ASR system can get close to the result that is provided the ground truth of accent categories. In addition, through the ablation study, we can prove that providing frame-level information does help the accent-related bias better improve the performance of ASR on accented speech. And using the weight w_i to represent the importance of each frame for accent identification and dynamically adjusting the accent-related bias outperform adding the same accent-related bias for each frame.

Table 3: *The WER (%) of the accented speech recognition on AESRC2020 without true accent category provided.*

Method	B	A	C	J	R	I	P	K	All
AR_1	7.44	5.99	9.67	7.48	7.13	7.07	5.86	4.72	6.89
AR_5	6.95	5.85	10.52	7.84	6.92	7.80	5.82	4.72	**7.02**
AR_6	7.22	6.01	10.44	7.92	6.99	7.66	6.01	4.69	7.09
AR_7	7.18	5.98	10.46	8.12	7.19	7.67	6.14	4.84	7.17

4.4.3. Comparison with other methods

In this part, we compare our method with the results in AESRC [1]. The results are as follows, where the "AESRC" represents the result of only using the labeled data from AESRC2020 speech corpus with RNN language model (LM) and the "+LS-960" uses additional labeled data from the LS-960 dataset.

Table 4: *WER (%) comparison of different methods on AESRC.*

Method	B	A	C	J	R	I	P	K	All
AESRC	10.06	9.96	11.77	6.79	5.26	10.05	7.45	7.69	8.63
+LS-960	7.64	7.42	9.87	5.71	4.60	7.85	5.90	6.40	6.92
AR_1	7.44	5.99	9.67	7.48	7.13	7.07	5.86	4.72	**6.89**
+4-gram	4.81	4.06	7.09	4.51	4.44	4.22	3.73	2.55	**4.42**
AR_5	6.95	5.85	10.52	7.84	6.92	7.80	5.82	4.72	**7.02**
+4-gram	4.68	4.20	7.60	4.72	4.33	4.47	3.71	2.57	**4.52**

It can be seen from the results that self-supervised pre-training is effective for accented speech recognition. Except for AESRC2020 dataset, we only use the unlabeled data from LS-960 dataset additionally. Without using LM, we can still surpass the results of "AESRC". And we train a 4-gram LM [32] on the same data as RNN LM: transcription of LS-960 and AESRC2020 datasets. After employing the 4-gram LM (weight: 1.74), we achieve 36.1% relative WER reduction. This also shows that LM is of great help to the CTC-based ASR system, as CTC model has conditional independence assumption.

5. Conclusion

In this paper, we explore the self-supervised pre-training methods to solve the accent identification and accented speech recognition tasks. Based on the pre-trained model following wav2vec 2.0, we propose a SDC-loss based E2E architecture to identify accents under the same language. As for accented speech recognition, we design an accent-dependent ASR system, which can utilize additional accent input features extracted from the ground truth of accent category or the proposed accent identification model. Furthermore, we propose a frame-level accent feature, which is extracted based on the proposed accent identification model and can leverage the frame-level information. We pre-train the networks using 960 hours unlabeled LibriSpeech dataset and fine-tune them on AESRC2020 speech dataset. The experimental results show that our proposed accent-dependent ASR system is significantly ahead of the AESRC2020 baseline and achieves 6.5% relative WER reduction compared with our accent-independent ASR system.

6. References

[1] X. Shi, F. Yu, Y. Lu, Y. Liang, Q. Feng, D. Wang, Y. Qian, and L. Xie, "The accented english speech recognition challenge 2020: Open datasets, tracks, baselines, results and methods," 2021.

[2] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for asr," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7694–7698.

[3] Y. A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.

[4] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Jun. 2018, pp. 2227–2237.

[5] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[7] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[9] C. Teixeira, I. Trancoso, and A. Serralheiro, "Accent identification," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, vol. 3, 1996, pp. 1784–1787 vol.3.

[10] S. Deshpande, S. Chikkerur, and V. Govindaraju, "Accent classification in speech," in *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, 2005, pp. 139–143.

[11] Z. Meng, H. Hu, J. Li, C. Liu, Y. Huang, Y. Gong, and C. Lee, "L-vector: Neural label embedding for domain adaptation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, p. 7389–7393.

[12] T. Viglino, P. Motlicek, and M. Cernak, "End-to-End Accented Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2140–2144.

[13] S. Cao, Y. Zhang, X. Feng, and M. Long, "Improving speech recognition accuracy of local poi using geographical models," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021.

[14] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the kld-regularized model adaptation," in *Interspeech 2014*, September 2014.

[15] M. Chen, Z. Yang, J. Liang, Y. Li, and W. Liu, "Improving deep neural networks based multi-accent mandarin speech recognition using i-vectors and accent-specific top layer," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 3620–3624.

[16] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning," in *Proc. Interspeech 2018*, 2018, pp. 2454–2458.

[17] K. Rao and H. Sak, "Multi-accent speech recognition with hierarchical grapheme based models," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4815–4819.

[18] S. Yoo, I. Song, and Y. Bengio, "A highly adaptive acoustic model for accurate multi-dialect speech recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5716–5720.

[19] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4749–4753.

[20] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[22] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[23] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. USA: Curran Associates Inc., 2017, pp. 6000–6010.

[25] 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)*, 2015, pp. 5206–5210.

[26] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[27] 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.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[29] T. Valenta and L. Šmídl, "On the impact of sentence length on recognition accuracy," in *2014 12th International Conference on Signal Processing (ICSP)*, 2014, pp. 500–504.

[30] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[31] M. Zhao, J. Zhou, Z. Li, H. Lu, and F. Tong, "Asv-subtools: An open source tools for speaker recognition," https://github.com/Snowdar/asv-subtools, 2021, gitHub repository.

[32] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, pp. 403–434, 2001.