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

Keqi Deng\textsuperscript{1,2,*}, Songjun Cao\textsuperscript{1,*}, Long Ma\textsuperscript{1}

\textsuperscript{1}Tencent Cloud Xiaowei, Beijing, China
\textsuperscript{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 accent discrimination task, our 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 using and Daren et al. [10] employ format frequency features into GMM. More recently, Shi et al. [11] 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
We set a target after softmax to obtain the cross entropy loss (CE-loss). In this paper, we employ the wav2vec 2.0 pre-training method to learn 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 accent prediction, 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 Transformer denote CNN based feature encoder and Transformer based context network, respectively. $h_i$ is the output of the Transformer [24] at $t$ step. FC is the fully connected layer and converts the $h_i$ to $a_i$ whose dimension is equal to the number of accent categories. Mean and Std denote calculating the mean and standard deviation.

$$L_{SDC} = \frac{1}{C} \sum_{j=1}^{C} a_{j}^{std}$$

where $a^{mean}$ represents the target and $L_{CE}$ denotes the CE-loss. The final loss function is obtained by adding the SDC-loss and CE-loss together.

$$L_{final} = L_{CE} + L_{SDC}$$

where $L_{final}$ denotes the final loss function. The $L_{final}$ 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 $a_i$ 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 $+\odot$ denote dot product, scalar multiplication and point-wise plus respectively. $F_{CNN}$ and $F_{Transformer}$ are the fully connected layer for CNN and Transformer respectively. $g^{true}$ 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.

$$e_i = \tilde{e}_i + F_{CNN}(g^{true}), \quad h_i = \tilde{h}_i + F_{Transformer}(g^{true}),$$

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

![Figure 1: Illustration of the proposed accent identification architecture.](image1)

![Figure 2: Illustration of the proposed accented speech recognition architecture.](image2)
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:

\[ w_i = \text{Sigmoid}(a_i \cdot a_i^{\text{mean}}), \]  

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 \( w_i \):

\[ w_i = \begin{cases} 
  w_i, & w_i > k \\
  0, & w_i < k
\end{cases} \]  

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

\[ c_i = c_i + FC_{\text{CNN}}(w_i \cdot \text{Softmax}(a_i^{\text{mean}})), \]  

\[ h_i = h_i + FC_{\text{Trans}}(w_i \cdot \text{Softmax}(a_i^{\text{mean}})), \]

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 FC\(_{\text{CNN}}\) and FC\(_{\text{Trans}}\) 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 Transformer to pool the h\(_t\), which is the same as the method [1]. We denote this model as AI\(_1\) and train it end-to-end. At last, we denote the proposed SDC-loss based accent identification model as AI\(_2\). As for ablation study, we do not introduce the SDC-loss of AI\(_2\) and mark it as AI\(_1\). 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>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>A</th>
<th>C</th>
<th>J</th>
<th>R</th>
<th>I</th>
<th>P</th>
<th>K</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESRC-12L</td>
<td>85.0</td>
<td>21.2</td>
<td>38.2</td>
<td>42.7</td>
<td>49.6</td>
<td>66.1</td>
<td>51.8</td>
<td>26.0</td>
<td>47.8</td>
</tr>
<tr>
<td>AESRC-3L</td>
<td>70.0</td>
<td>45.7</td>
<td>56.2</td>
<td>48.5</td>
<td>30.0</td>
<td>83.5</td>
<td>57.2</td>
<td>45.0</td>
<td>54.1</td>
</tr>
<tr>
<td>+ASR init</td>
<td>93.9</td>
<td>60.2</td>
<td>67.0</td>
<td>73.2</td>
<td>75.7</td>
<td>97.0</td>
<td>85.5</td>
<td>55.6</td>
<td>76.1</td>
</tr>
<tr>
<td>x-vector [6]</td>
<td>66.0</td>
<td>46.4</td>
<td>62.2</td>
<td>53.5</td>
<td>59.9</td>
<td>77.4</td>
<td>54.7</td>
<td>56.1</td>
<td>59.1</td>
</tr>
<tr>
<td>i-vector [30]</td>
<td>72.1</td>
<td>48.8</td>
<td>70.9</td>
<td>45.9</td>
<td>64.3</td>
<td>87.4</td>
<td>58.0</td>
<td>58.1</td>
<td>62.6</td>
</tr>
<tr>
<td>AI(_1)</td>
<td>67.7</td>
<td>87.2</td>
<td>61.5</td>
<td>94.8</td>
<td>51.4</td>
<td>63.0</td>
<td>82.0</td>
<td>74.7</td>
<td>72.7</td>
</tr>
<tr>
<td>AI(_2)</td>
<td>83.2</td>
<td>89.6</td>
<td>54.0</td>
<td>95.8</td>
<td>52.6</td>
<td>65.2</td>
<td>82.7</td>
<td>70.2</td>
<td>73.9</td>
</tr>
<tr>
<td>AI(_3)</td>
<td>76.3</td>
<td>87.2</td>
<td>61.5</td>
<td>94.8</td>
<td>51.4</td>
<td>63.0</td>
<td>82.1</td>
<td>74.7</td>
<td>72.9</td>
</tr>
</tbody>
</table>

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 AI\(_1\) are sentence-level accent identification model, while the AI\(_2\) and AI\(_3\) identify the accent based on frame-level vector. From the results we can see that the proposed method AI\(_2\) outperforms the sentence-level model 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 AI\(_2\) also contains 12-layer Transformer, but due to the self-supervised pre-training method and the proposed SDC-loss, 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 AR₀. And we denote the proposed architecture as AR₁, which chooses g^\text{vec} 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 (AR₂); 2. concatenating the output of the FC\text{Trans} with the \( \hat{h}_k \) as \( \hat{h}_k \), in Eq. (9) (AR₃); 3. taking the AR₀ as starting point and only updating accent-dependent output layers (eight in total and one for each accent) for extra 3500 update numbers (AR₄).

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

<table>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>A</th>
<th>C</th>
<th>J</th>
<th>R</th>
<th>I</th>
<th>P</th>
<th>K</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR₀</td>
<td>7.50</td>
<td>6.29</td>
<td>10.64</td>
<td>8.21</td>
<td>7.46</td>
<td>7.86</td>
<td>6.22</td>
<td>5.00</td>
<td>7.37</td>
</tr>
<tr>
<td>AR₁</td>
<td>7.44</td>
<td>5.99</td>
<td>9.67</td>
<td>7.48</td>
<td>7.13</td>
<td>7.07</td>
<td>5.86</td>
<td>4.72</td>
<td>6.89</td>
</tr>
<tr>
<td>AR₂</td>
<td>7.15</td>
<td>5.91</td>
<td>10.81</td>
<td>7.74</td>
<td>7.11</td>
<td>7.71</td>
<td>5.87</td>
<td>4.68</td>
<td>7.09</td>
</tr>
<tr>
<td>AR₃</td>
<td>7.48</td>
<td>6.05</td>
<td>10.69</td>
<td>7.96</td>
<td>7.12</td>
<td>7.64</td>
<td>5.91</td>
<td>4.78</td>
<td>7.17</td>
</tr>
<tr>
<td>AR₄</td>
<td>7.21</td>
<td>6.18</td>
<td>10.64</td>
<td>7.91</td>
<td>7.14</td>
<td>7.58</td>
<td>6.04</td>
<td>4.95</td>
<td>7.18</td>
</tr>
</tbody>
</table>

The results in Table 1 show that the proposed AR₁ achieves 6.4% relative word error rate (WER) reduction compared with the AR₀, 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 \( \hat{h}_k \) 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 AR₂. As for ablation, 1. we do not introduce the threshold \( k \) on the basis of AR₃ and denote it as AR₄; 2. we do not introduce the weight of accent-related bias \( w_i \) in Eq. 6 and denote it as AR₅. In this ways, AR₆ can be regarded as all \( w_i \) equals to 1 and thus can not utilize the frame-level information.

From the results from 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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>A</th>
<th>C</th>
<th>J</th>
<th>R</th>
<th>I</th>
<th>P</th>
<th>K</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR₁</td>
<td>7.44</td>
<td>5.99</td>
<td>9.67</td>
<td>7.48</td>
<td>7.13</td>
<td>7.07</td>
<td>5.86</td>
<td>4.72</td>
<td>6.89</td>
</tr>
<tr>
<td>AR₂</td>
<td>6.89</td>
<td>5.85</td>
<td>10.52</td>
<td>7.84</td>
<td>6.92</td>
<td>7.80</td>
<td>5.82</td>
<td>4.72</td>
<td>7.02</td>
</tr>
<tr>
<td>AR₃</td>
<td>7.22</td>
<td>6.01</td>
<td>10.44</td>
<td>7.92</td>
<td>6.99</td>
<td>7.76</td>
<td>6.01</td>
<td>4.69</td>
<td>7.09</td>
</tr>
<tr>
<td>AR₄</td>
<td>7.18</td>
<td>5.98</td>
<td>10.46</td>
<td>8.12</td>
<td>7.19</td>
<td>7.67</td>
<td>6.14</td>
<td>4.84</td>
<td>7.17</td>
</tr>
</tbody>
</table>

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


