GE2E-AC: Generalized End-to-End Loss Training for Accent Classification

Chihiro Watanabe* and Hirokazu Kameoka†
* NTT Communication Science Laboratories, Japan
E-mail: ch.watanabe@ntt.com Tel: +81-7049347331
† NTT Communication Science Laboratories, Japan
E-mail: hirokazu.kameoka@ntt.com Tel: +81-462403645

Abstract—**Accent classification or AC is a task to predict the accent type of an input utterance, and it can be used as a preliminary step toward accented speech recognition and accent conversion. Existing studies have often achieved such classification by training a neural network model to minimize the classification error of the predicted accent label, which can be obtained as a model output. Since we optimize the entire model only from the perspective of classification loss during training time in this approach, the model might learn to predict the accent type from irrelevant features, such as individual speaker identity, which are not informative during test time. To address this problem, we propose a GE2E-AC, in which we train a model to extract accent embedding or AE of an input utterance such that the AEs of the same accent class get closer, instead of directly minimizing the classification loss. We experimentally show the effectiveness of the proposed GE2E-AC, compared to the baseline model trained with the conventional cross-entropy-based loss[1].**

I. Introduction

Accent is one of the major non-linguistic factors of speech. To alleviate the performance degradation in speech recognition caused by different accents (e.g., English, American, and Scottish), a potential solution is to first classify the accent type of an input utterance and then use an accent-specific speech recognition model that has been trained for the predicted accent [1]–[3]. Such a task to classify an accent type of a given utterance is called accent classification (AC), and various data sets have been publicly available for AC which contain the accent labels of the utterances as well as the speaker ones [4]–[6].

Existing AC models are typically trained to minimize a loss function (e.g., cross-entropy or CE loss) between the target and predicted accent labels [7]–[9]. We will call this approach **CE-AC** (Fig. 1 (a)). While this approach directly optimizes the model's classification accuracy during training, it may lead to predictions that capture irrelevant information (e.g., speaker identity) from the input utterance to reduce the training loss, ultimately degrading test performance. This has also been pointed out in [10]. Moreover, CE-based approach is based on the implicit assumption that the set of potential accent types of input utterances is fixed. To utilize an additional data set with the accent types that have not been in the previous data set, a different model with a new output size must be retrained from scratch.

To address these limitations, we propose adapting the concept of generalized end-to-end (GE2E) loss training [11], which has proven highly effective in speaker verification tasks, for use in the AC task. Specifically, instead of training a model to directly minimize the error in the predicted accent class, we train it to extract accent embeddings (AEs), i.e., feature vectors representing accent information, from input utterances so that the AEs of the same accent class get closer, while those of different accent classes move farther apart. We call this approach **GE2E-AC** (Fig. 1 (b)).

This approach has the advantage that even when a test sample with an unknown accent comes in, it can determine which training sample is most similar. Additionally, it is easy to retrain without needing to modify the model structure. Our experiments show that this method is more effective for the AC task compared to the conventional CE-based baseline method. In the experiment, we also compared the basic GE2E-AC with **GE2E-AC-A**, which incorporates the additional adversarial speaker classification loss to further eliminate the speaker identity information from extracted AEs. Furthermore, since the optimal input for the AC task is not obvious, we also evaluated the performance of using phoneme-related bottleneck features (BNFs) [12] extracted from an encoder-decoder type automatic speech recognition (ASR) system and Hidden-Unit BERT (HuBERT) [13] features as inputs.

II. Baseline method: CE-based accent classification

Before we explain the proposed GE2E-based AC, we first describe a baseline CE-AC. In this approach, we use accent classifier $f_{\mathrm{CE}} : \mathbb{R}^{T\times F} \mapsto \mathbb{R}^C$ which maps input feature $X \in \mathbb{R}^{T\times F}$ (e.g., BNF [12] and HuBERT [13]) of an utterance to vector $\hat{\boldsymbol{y}} = f_{\mathrm{CE}}(X) \in [0,1]^C$, where T, F, and C indicate the numbers of time and frequency bins of an input feature and accent types, respectively. Throughout this paper, we assume that classifier f_{CE} is modeled as a neural network. By assuming that output vector $\hat{\boldsymbol{y}}$ represents the probabilities of categorical distribution on the accent types, we train the entire network f_{CE} by minimizing the CE loss $\mathcal{L}_{\mathrm{CE}}$ between the model output $\hat{\boldsymbol{y}}$ and the target accent label in one-hot

[1] This work was supported by JST CREST Grant Number JPMJCR19A3, Japan.

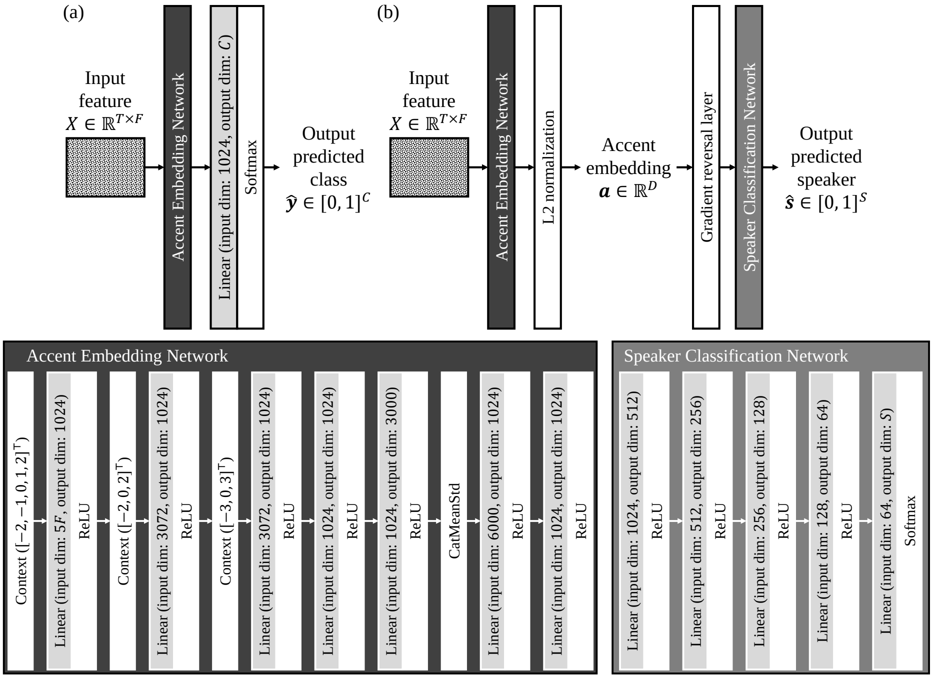
(a)
Input feature $X \in \mathbb{R}^{T\times F}$
Accent Embedding Network
Linear (input dim: 1024, output dim: C)
Softmax
Output predicted class $\hat{\boldsymbol{y}} \in [0,1]^C$
(b)
Input feature $X \in \mathbb{R}^{T\times F}$
Accent Embedding Network
L2 normalization
Accent embedding $\boldsymbol{a} \in \mathbb{R}^D$
Gradient reversal layer
Speaker Classification Network
Output predicted speaker $\hat{\boldsymbol{s}} \in [0,1]^S$
Accent Embedding Network
Context ($[-2,-1,0,1,2]^\mathsf{T}$)
Linear (input dim: $5F$, output dim: 1024)
ReLU
Context ($[-2,0,2]^\mathsf{T}$)
Linear (input dim: 3072, output dim: 1024)
ReLU
Context ($[-3,0,3]^\mathsf{T}$)
Linear (input dim: 3072, output dim: 1024)
ReLU
Linear (input dim: 1024, output dim: 1024)
ReLU
Linear (input dim: 1024, output dim: 3000)
ReLU
CatMeanStd
Linear (input dim: 6000, output dim: 1024)
ReLU
Linear (input dim: 1024, output dim: 1024)
ReLU
Speaker Classification Network
Linear (input dim: 1024, output dim: 512)
ReLU
Linear (input dim: 512, output dim: 256)
ReLU
Linear (input dim: 256, output dim: 128)
ReLU
Linear (input dim: 128, output dim: 64)
ReLU
Linear (input dim: 64, output dim: S)
Softmax

Fig. 1. The network architectures of (a) the baseline CE-based and (b) the proposed GE2E-based accent classification. For arbitrary vector $\boldsymbol{\phi} \in \mathbb{R}^K$, $\mathrm{Context}(\boldsymbol{\phi})$ indicates a function $f_{\boldsymbol{\phi}} : \mathbb{R}^{T\times F'} \mapsto \mathbb{R}^{T\times KF'}$, which maps matrix $X \in \mathbb{R}^{T\times F'}$ to $f_{\boldsymbol{\phi}}(X) = \left[M^{(1)}X, \cdots, M^{(K)}X\right]$, where $M^{(k)} \in \mathbb{R}^{T\times T}$ is a matrix each of whose entries $M^{(k)}_{ij}$ is one if $j = i + \phi_k$ and zero otherwise for all $k \in \{1, \ldots, K\}$. $\mathrm{CatMeanStd}$ is a function $f : \mathbb{R}^{T\times F'} \mapsto \mathbb{R}^{2F'}$, which maps matrix $X \in \mathbb{R}^{T\times F'}$ to $\left[\boldsymbol{\mu}^\mathsf{T}, \boldsymbol{\sigma}^\mathsf{T}\right]^\mathsf{T}$, where $\boldsymbol{\mu} \in \mathbb{R}^{F'}$ and $\boldsymbol{\sigma} \in \mathbb{R}^{F'}$ are the vectors each of whose entries μ_k and σ_k are the mean and the standard deviation of vector $[X_{1k}, \ldots, X_{Tk}]^\mathsf{T}$, respectively.

representation $\boldsymbol{y} \in \left\{\boldsymbol{y}' \in \{0,1\}^C \middle| \sum_{j=1}^{C} y'_j = 1\right\}$. For a mini-batch with size B, the CE loss is given by

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{B}\sum_{l=1}^{B}\sum_{j=1}^{C} y_{lj} \log \hat{y}_{lj}, \qquad (1)$$

where $\boldsymbol{y}_l = (y_{lj})_{1\le j\le C}$ and $\hat{\boldsymbol{y}}_l = (\hat{y}_{lj})_{1\le j\le C}$ are the target and predicted accent labels of the lth sample in a mini-batch, respectively.

In the experiment in Section IV, we used a classifier based on the architecture in [14], as shown in Fig. 1 (a). This architecture is also used as a sub-model in [15], which achieved the highest accuracy in track 1 (i.e., English accent recognition task) of the accented English speech recognition challenge 2020 (AESRC2020) [10].

III. Proposed method: GE2E-based accent classification

A. GE2E-AC

During training, the baseline method described in Section II focuses on directly minimizing the error in the predicted accent label. However, this training approach may cause the model to rely on irrelevant information, such as speaker identity, to reduce training loss, leading to poor test performance. To prevent this, we propose using a metric learning approach that trains the model to extract AEs from input utterances, ensuring that the AEs of utterances with the same accent class are closer together while those of different accent classes are farther apart. This approach also offers significant advantages in potential applications beyond improving the model's generalization ability. Specifically, the ability to extract AEs allows it to be used for tasks like one-shot accent conversion, where input speech is converted to match the accent of a reference speech sample, even if the reference accent is unknown.

GE2E loss training is a type of metric learning approach originally used for speaker verification [11]. It has proven highly effective in this task by extracting speaker embeddings from input utterances so that embeddings from the same speaker are closer together, while those from different speakers are farther apart. In our proposed method, we adapt this approach for the AC task, treating accent classes similar to how speaker identities are handled in speaker verification. In the proposed GE2E-AC, by pulling the AEs from the utterances of the same accent type closer together and pushing those from differently accented utterances away, the model would learn to extract an abstract representation of accent, thereby avoiding overfitting to individual speakers. Although there are other possible methods of deep metric learning such as Siamese network [16], we employ GE2E loss here, since its effectiveness has already been shown in another speech processing task [11].

The actual flow of GE2E-AC is as follows (Fig. 1 (b)). We first extract a D-dimensional AE $\boldsymbol{a} = f_{\mathrm{AE}}(X) \in \mathbb{R}^D$ from input feature X by using an accent embedding network $f_{\mathrm{AE}} : \mathbb{R}^{T\times F} \mapsto \mathbb{R}^{D}$. During training, we define a mini-batch of input features as a set of M utterances for each of C accent types. Let $\boldsymbol{a}_{ji} \in \mathbb{R}^{D}$ be the AE of the ith utterance of the jth accent for $i = 1,\ldots,M$, and $j = 1,\ldots,C$. As shown in Fig. 1 (b), we define an AE such that it has a unit L2 norm (i.e., $\|\boldsymbol{a}_{ji}\|_2 = 1$). Based on these AEs, we can compute two types of centroids for each accent class j, which are given by

$$\boldsymbol{c}_j = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{a}_{ji}, \quad \boldsymbol{c}_j^{(-i)} = \frac{1}{M-1}\sum_{\substack{m=1\\ m\neq i}}^{M}\boldsymbol{a}_{ji}. \tag{2}$$

We also define the L2 normalization of the centroids by

$$\bar{\boldsymbol{c}}_j = \frac{\boldsymbol{c}_j}{\|\boldsymbol{c}_j\|_2}, \quad \bar{\boldsymbol{c}}_j^{(-i)} = \frac{\boldsymbol{c}_j^{(-i)}}{\|\boldsymbol{c}_j^{(-i)}\|_2}. \tag{3}$$

By using these L2 normalized centroids, we define $MC \times C$ similarity matrix $S = (S_{lk})_{1\leq l\leq MC, 1\leq k\leq C}$, whose (l,k)th entry is given by

$$S_{lk} = \begin{cases} w\boldsymbol{a}_{ji}\cdot\bar{\boldsymbol{c}}_k^{(-i)} + b & j = k,\\ w\boldsymbol{a}_{ji}\cdot\bar{\boldsymbol{c}}_k + b & \text{otherwise,}\end{cases} \tag{4}$$

where w and b are learnable weight and bias parameters for the similarity metric and $l = (j-1)M + i$ for $i = 1,\ldots,M$ and $j = 1,\ldots,C$. As in [11], we define each entry S_{lk} differently in the cases of $j = k$ and $j \neq k$ for training stability.

Based on similarity matrix S in (4), we train the entire network to minimize the GE2E loss, which is given by

$$\mathcal{L}_{\mathrm{GE2E}} = -\frac{1}{MC}\sum_{l=1}^{MC}\sum_{k=1}^{C} y_{lk}\log\hat{y}_{lk}, \tag{5}$$

where $\hat{y}_{lk} = \exp(S_{lk})/\{\sum_{k'=1}^{C}\exp(S_{lk'})\}$. In regard to the final formulation, the GE2E loss in (5) looks similar to the CE loss in (1). However, unlike in the CE loss, the predicted accent labels $\{\hat{y}_{lk}\}$ in the GE2E loss are computed based on the similarity $\{S_{lk}\}$ between the AEs and their centroids. To minimize the GE2E loss, the accent embedding network should learn to pull the AEs closer to the centroid of its accent type and to push them away from the other centroids. This facilitates the AEs of the same accent type getting closer together, even if they are uttered by mutually different speakers, which would improve the generalization performance in AC.

In the experiment in Section IV, to evaluate the effect of introducing the GE2E loss instead of the CE one, we set the shallower part of the model (i.e., "Accent Embedding Network" in Fig. 1) to be the same as that of CE-AC. In the standard GE2E-AC, we used only the shallower part than the gradient reverse layer.

B. GE2E-AC-A

Although introduction of GE2E loss alone would improve the performance of AC during test time, AEs of GE2E-AC might still contain some information related to speaker identity, since there are no samples of different accents from the same speaker. To further eliminate speaker-related information from AEs and improve the generalization performance, we also propose GE2E-AC-A, in which we incorporate an adversarial speaker classification or SC loss $\mathcal{L}_{\mathrm{SC}}$ into the training criterion. We consider such a loss to avoid mixing speaker information in AEs. It must be noted that a similar technique has also been used in [17], although our main contribution lies in introduction of GE2E loss for AEs, rather than in that of SC loss.

Specifically, the SC loss can be computed by using another network $f_{\mathrm{SC}} : \mathbb{R}^{D} \mapsto [0,1]^{S}$ (i.e., "Speaker Classification Network" in Fig. 1), which predicts a speaker index from an AE. Let $\boldsymbol{s} \in \left\{\boldsymbol{s}' \in \{0,1\}^S \,|\, \sum_{j=1}^{S} s'_j = 1\right\}$ be the target speaker label in one-hot representation and let $\hat{\boldsymbol{s}} = f_{\mathrm{SC}}(\boldsymbol{a}) \in [0,1]^S$ be the predicted one. The SC loss is given by

$$\mathcal{L}_{\mathrm{SC}} = -\frac{1}{B}\sum_{l=1}^{B}\sum_{j=1}^{S} s_{lj}\log\hat{s}_{lj}, \tag{6}$$

where $\boldsymbol{s}_l = (s_{lj})_{1\leq j\leq S}$ and $\hat{\boldsymbol{s}}_l = (\hat{s}_{lj})_{1\leq j\leq S}$ are the target and predicted speaker labels of the lth sample in a mini-batch, respectively.

The training strategy of GE2E-AC-A is two-fold. The speaker classification network should be trained to minimize SC loss $\mathcal{L}_{\mathrm{SC}}$, while the accent embedding network should be trained to maximize it. We can implement such an adversarial learning by adding a gradient reversal layer (GRL) [18] before the speaker classification network. The GRL is an identity function in forward propagation, while it multiplies the gradient from the deeper layers by -1 in backpropagation. By introducing GRL, we can formulate the overall training criterion of GE2E-AC-A to be minimized by $\mathcal{L}_{\mathrm{GE2E}} + \lambda_{\mathrm{SC}}\mathcal{L}_{\mathrm{SC}}$, where λ_{SC} is a hyperparameter. As GE2E-AC, we used a network architecture in Fig. 1 (b) in the experiment in Section IV.

IV. Experiment

To confirm the effectiveness of the proposed GE2E-AC, we conducted an experiment using CSTR VCTK Corpus [19]. Among this data set, we only used the samples of accent types of "English," "American," "Scottish," "Irish," and "Canadian," which consist of 33, 22, 19, 9, and 8 speakers, respectively. For each accent type, we randomly chose five test speakers and used the other speakers for training. We trimmed leading and trailing silence of each sample and resampled it to 16 KHz.

As input features, we adopted BNF [12] and HuBERT [13]. We extracted these features from WAV samples by using the codes provided by [20], [21] and trained each model of CE-AC (baseline), GE2E-AC, and GE2E-AC-A with them.

The hyperparameters for training the models were as follows. For all the models, we used Adam [22] with the learning rate of 1.0×10^{-5} for optimization and set the minibatch size to 16, the number of epochs to 100, λ_{SC} to 1.0×10^{-5}, and the number M of utterances per accent type to 10. It must be noted that, since a mini-batch of GE2E-AC and GE2E-AC-A consists of the same number M of utterances for all the accents with generally different sample sizes $\{n_1, \ldots, n_C\}$, we could not determine the meaning of one epoch in the usual sense. Therefore, for these methods, we defined the nominal "sample size" for a sample which includes one input feature for each accent as the maximum sample size $\max_{j=1,\ldots,C} n_j$ in the experiment. We clipped the max L2 norm of the gradients of the model parameters to 1.0 for CE-AC and 3.0 for GE2E-AC and GE2E-AC-A. Finally, we computed confusion matrices by using the trained models. The (i, j)th entry of a confusion matrix indicates the number of training or test samples whose target accent index is i and whose predicted counterpart is j. Since outputs of the GE2E-based models are accent embeddings $\boldsymbol{a}$, not predicted accent classes $\hat{\boldsymbol{y}}$, we need to determine $\hat{\boldsymbol{y}}$ from $\boldsymbol{a}$ to compute a confusion matrix. Based on the accent embeddings of all the training samples, we first computed the centroid (i.e., mean of embeddings) of each accent class. We defined the predicted accent class of a given sample with accent embedding $\boldsymbol{a}'$ as the class whose centroid had the highest cosine similarity with $\boldsymbol{a}'$.

Tables I, II, and III show the classification accuracy of each method in the cases of three, four, and five accent types, respectively. Aside from the case of using HuBERT for classifying four accent types, the proposed GE2E-AC or GE2E-AC-A achieved better test performance than that of the CE-based baseline method. In some cases (e.g., using HuBERT for classifying four accent types), the adversarial speaker classification loss in combination with the GE2E loss was effective, while in other cases it was not. In regard to the input feature type, HuBERT yielded better test performance than BNF did in most cases. As more detailed results, Figs. 2, 3, and 4 show the confusion matrices in the cases of three, four, and five accent types, respectively. Although there were slight differences in the values of the confusion matrices, they showed similar classification tendency to each other among all the settings. For instance, when the number of accent types was five, it was hard for all the models to correctly classify the Canadian speakers, and most of the utterances by Canadian speakers were classified as "American." This would be probably due to the bias in sample size, and to mitigate such a problem would be an important direction of a future study.

V. Conclusion

In this study, we proposed a new training method for AC based on the similarity between a pair of accent embeddings of the utterances so that the model can avoid overfitting to specific speakers. The effectiveness of the proposed GE2E-AC has been shown experimentally, compared to the existing CE-based method in which we directly minimize the classification loss for individual utterances. A possible future direction of this study would be to utilize the proposed method as a preliminary step toward accent conversion, where we aim to convert the accent of an input utterance to some different one. By extracting an AE from a target sample with a trained GE2E-AC model and using it for test-time accent conversion, it would be possible to convert accents in zero-shot manner.

TABLE I
ACCURACY OF ACCENT RECOGNITION FOR EACH SETTING WITH **THREE** ACCENT TYPES.

Method (Feature type)	Training accuracy	Test accuracy
CE-AC (BNF)	99.9%	78.8%
GE2E-AC (BNF)	100.0%	**81.2%**
GE2E-AC-A (BNF)	99.7%	77.8%
CE-AC (HuBERT)	98.7%	81.9%
GE2E-AC (HuBERT)	99.9%	85.0%
GE2E-AC-A (HuBERT)	100.0%	**85.2%**

TABLE II
ACCURACY OF ACCENT RECOGNITION FOR EACH SETTING WITH **FOUR** ACCENT TYPES.

Method (Feature type)	Training accuracy	Test accuracy
CE-AC (BNF)	99.8%	59.2%
GE2E-AC (BNF)	100.0%	**63.8%**
GE2E-AC-A (BNF)	100.0%	62.1%
CE-AC (HuBERT)	99.7%	**65.9%**
GE2E-AC (HuBERT)	100.0%	63.5%
GE2E-AC-A (HuBERT)	99.6%	64.9%

TABLE III
ACCURACY OF ACCENT RECOGNITION FOR EACH SETTING WITH **FIVE** ACCENT TYPES.

Method (Feature type)	Training accuracy	Test accuracy
CE-AC (BNF)	99.2%	46.6%
GE2E-AC (BNF)	100.0%	**49.7%**
GE2E-AC-A (BNF)	100.0%	49.6%
CE-AC (HuBERT)	99.4%	50.9%
GE2E-AC (HuBERT)	100.0%	52.8%
GE2E-AC-A (HuBERT)	100.0%	**53.4%**

References

[1] V. Gupta and P. Mermelstein, "Effects of speaker accent on the performance of a speaker-independent, isolated-word recognizer," *The Journal of the Acoustical Society of America*, vol. 71, no. 6, pp. 1581–1587, 1982.

[2] L. M. Arslan and J. H. L. Hansen, "Language accent classification in american english," *Speech Communication*, vol. 18, no. 4, pp. 353–367, 1996.

[3] A. Faria, "Accent classification for speech recognition," in *International Workshop on Machine Learning for Multimodal Interaction*, 2006, pp. 285–293.

[4] I. Demirsahin, O. Kjartansson, A. Gutkin, and C. Rivera, "Open-source multi-speaker corpora of the English accents in the British isles," in *Language Resources and Evaluation Conference (LREC)*, 2020, pp. 6532–6541.

[5] R. Sanabria, N. Bogoychev, N. Markl, A. Carmantini, and P. B. Ondrej Klejch and, "The Edinburgh international accents of English corpus: Towards the democratization of English ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[6] W. Wang, Y. Song, and S. Jha, "GLOBE: A high-quality English corpus with global accents for zero-shot speaker adaptive text-to-speech," arXiv:2406.14875, 2024.

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

[8] M. Lesnichaia, V. Mikhailava, N. Bogach, I. Lezhenin, J. Blake, and E. Pyshkin, "Classification of accented English using CNN model trained on amplitude mel-spectrograms," in *Interspeech*, 2022, pp. 3669–3673.

[9] V. Nechaev and S. Kosyakov, "Non-autoregressive real-time accent conversion model with voice cloning," arXiv:2405.13162, 2024.

[10] 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," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6918–6922.

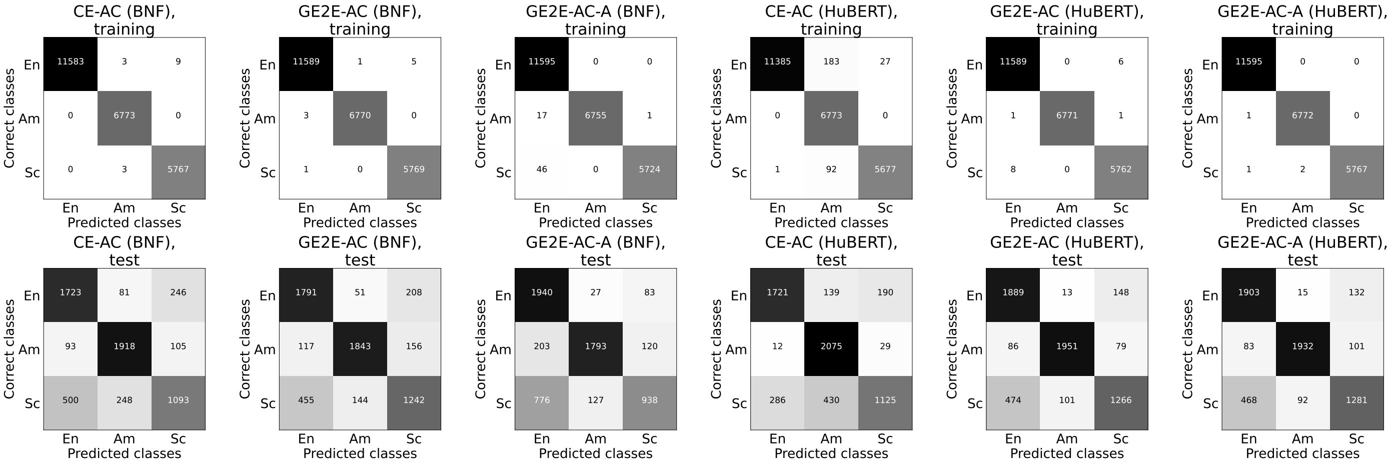

Fig. 2. Confusion matrices with **three** accent types. The value plotted in each entry indicates the number of corresponding samples. “En,” “Am,” and “Sc” stand for English, American, and Scottish, respectively.

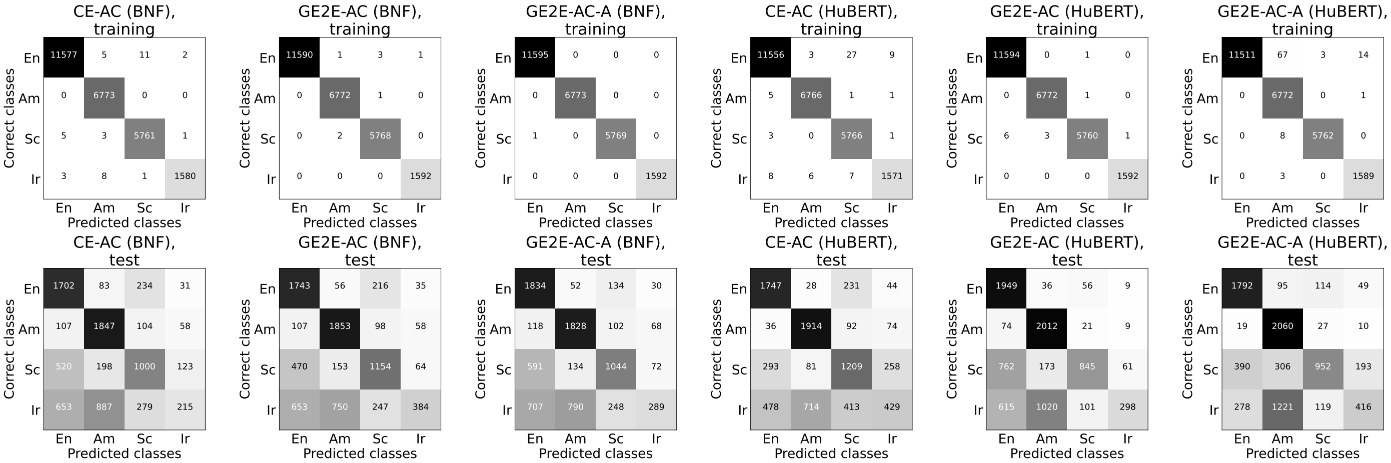

Fig. 3. Confusion matrices with **four** accent types. “Ir” stands for Irish.

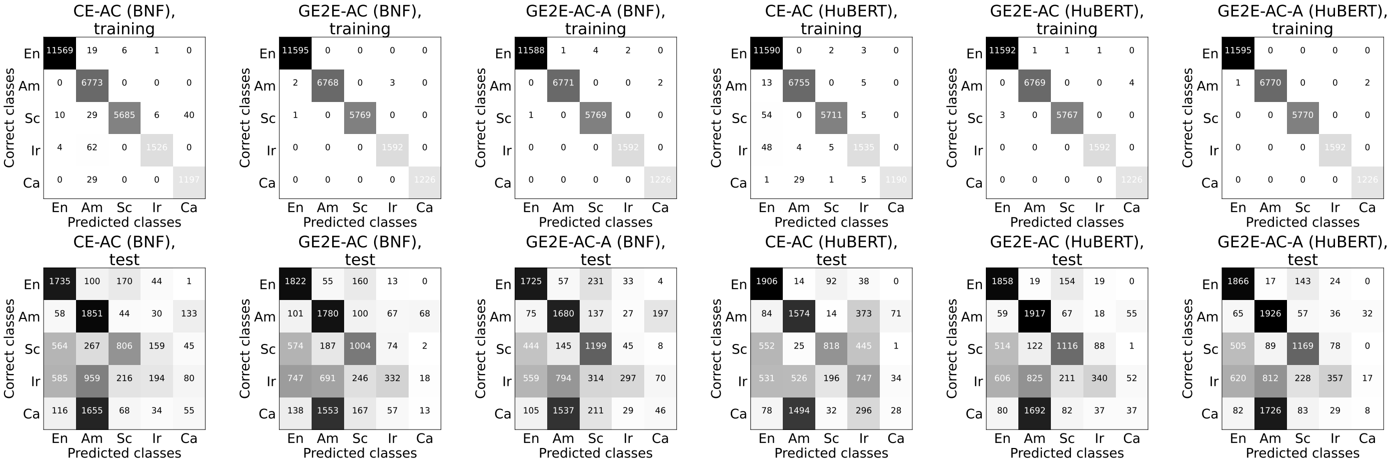

Fig. 4. Confusion matrices with **five** accent types. “Ca” stands for Canadian.

[11] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.

[12] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.

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

[15] H. Huang, X. Xiang, Y. Yang, R. Ma, and Y. Qian, "AISpeech-SJTU accent identification system for the accented English speech recognition challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6254–6258.

[16] A. Siddhant, P. Jyothi, and S. Ganapathy, "Leveraging native language speech for accent identification using deep Siamese networks," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 621–628.

[17] X. Zhou, M. Zhang, Y. Zhou, Z. Wu, and H. Li, "Multi-scale accent modeling with disentangling for multi-speaker multi-accent TTS synthesis," arXiv:2406.10844, 2024.

[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[19] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2017.

[20] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and H. Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1717–1728, 2021, https://github.com/liusongxiang/ppg-vc.

[21] "Hubert," Hugging Face, https://huggingface.co/docs/transformers/model_doc/hubert.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, 2015.