sensors

MDPI

Article

Accent Recognition with Hybrid Phonetic Features

Zhan Zhang, Yuehai Wang * and Jianyi Yang

Department of Information and Electronic Engineering, Zhejiang University, Hangzhou 310007, China;
zhan_zhang@zju.edu.cn (Z.Z.); yangjy@zju.edu.cn (J.Y.)
* Correspondence: wyuehai@zju.edu.cn

Abstract: The performance of voice-controlled systems is usually influenced by accented speech. To make these systems more robust, frontend accent recognition (AR) technologies have received increased attention in recent years. As accent is a high-level abstract feature that has a profound relationship with language knowledge, AR is more challenging than other language-agnostic audio classification tasks. In this paper, we use an auxiliary automatic speech recognition (ASR) task to extract language-related phonetic features. Furthermore, we propose a hybrid structure that incorporates the embeddings of both a fixed acoustic model and a trainable acoustic model, making the language-related acoustic feature more robust. We conduct several experiments on the AESRC dataset. The results demonstrate that our approach can obtain an 8.02% relative improvement compared with the Transformer baseline, showing the merits of the proposed method.

Keywords: accent recognition; audio classification; accented English speech recognition

check for updates

Citation: Zhang, Z.; Wang, Y.; Yang, J. Accent Recognition with Hybrid Phonetic Features. *Sensors* **2021**, *21*, 6258. https://doi.org/10.3390/s21186258

Academic Editors: Steven Guan and Yinong Chen

Received: 8 August 2021
Accepted: 14 September 2021
Published: 18 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

With the quick growth of voice-controlled systems, speech-related technologies are becoming part of our daily life. However, the variability of speech poses a serious challenge to these technologies. Among the many factors that can influence the speech variability, accent is a typical one that will cause degradation in recognition accuracy [1–3].

Accent is a diverse pronouncing behavior under certain languages, which can be influenced by social environment, education, residential zone, and so on. As analyzed in [4], English speakers are constructed by not only about 380 million natives, but also by close to 740 million non-native speakers. Influenced by their native language, the speakers may have a very wide variety of accents.

To analyze the accent attribute in the collected speech and make the whole voice-controlled system more generalized, accent recognition (AR) or accent classification technologies can be applied to custom the downstream subsystems. Thus, AR technologies have received increased attention in recent years.

From the point of deep learning, as accent is an utterance-level attribute, AR is also a classification task that converts an audio sequence into a certain class. In this respect, the audio classification tasks, including audio scene classification, speaker recognition, and AR, can share similar ideas on network structures. However, AR is a more challenging task.

Generally, acoustic scene classification or speaker recognition task can be finished using certain low-level discriminative features. For example, speaker recognition can be completed by recognizing the unique timbre (such as the frequency) of the speaker, which is unrelated to the language they speak. Thus, both acoustic scene classification and speaker recognition can be language-agnostic tasks.

In contrast, we generally realize someone has a certain accent when hearing that a specific pronunciation is different from the standard one. Therefore, for the AR task, to judge whether someone has a different accent, knowledge of that language is needed. The discriminative feature of the accented pronunciation is also more subtle. As a result, we think that AR differs from acoustic scene classification or speaker recognition because

AR requires a more fine-grained and language-related feature, which is more difficult compared to the language-agnostic tasks. For our method, we use a CNN-based ASR frontend to encode the language-related information. Then we append extra self-attention layers and an aggregation layer to capture the time sequence relevance and conduct classification. We apply the ASR loss as an auxiliary multitask learning (MTL) [5] target to extract language-related acoustic features. To make the acoustic features more robust, we fuse the embedding of a trainable acoustic model that is trained on the accented dataset and a fixed acoustic model trained on the standard dataset for a hybrid aggregation.

In this paper, first, we review the related work in Section 2. Next, we give the proposed method a detailed description in Section 3. We conduct dense experiments to compare the model architecture and analyze different training methods in Section 4. Finally, we give our conclusion in Section 5.

The contributions of this paper are summarized as follows.

- We propose a novel hybrid structure (Codes available at https://github.com/zju-zhan-zhang/Hybrid-AR, accessed on 7 September 2021) to solve the AR task. Our method adopts an ASR backbone and fuses the language-related phonetic features to perform the AR task along with ASR MTL.
- We investigate the relationship between ASR and AR. Specifically, we find that without the auxiliary ASR task, AR may easily overfit to the speaker label in the trainset. The proposed ASR MTL alleviates such a phenomenon.
- We conduct dense experiments on the AESRC dataset [6]. The results show that the recognition accuracy of the proposed method is 8.02% better than the Transformer-based baseline relatively. Moreover, we test the robustness by corrupting the text annotations. Compared with the baseline, the phonetic acoustic features extracted by the hybrid acoustic models can make the AR model more robust.

2. Related Works

For the audio classification task, on the one hand, the input features are of vital importance. Ref. [7] ensembles different channel features for a more robust classification. Ref. [8] explores how the classification knowledge is perceived in CNN and proposes to boost the performance using enhanced log-Mel input features. Moreover, Ref. [9] fuses different types of spectrogram to make use of their different characteristics on the time-frequency domain. Ref. [10] integrates both the speech attribute features and the acoustic cues to better capture foreign accents.

On the other hand, for the network structures, conventional methods including [11,12] adopt Hidden Markov Model (HMM) to model the time features. Ref. [13] applies Fisher Vector (FV) and Vector of Locally Aggregated Descriptors (VLAD) methods for the language identification task. [14] gives a thorough study about the encoding structures and loss functions for the language recognition task.

Besides the aforementioned works that focus on the recognition task alone, there are other works that apply AR as an auxiliary task to boost the performance of the other main tasks. Refs. [15,16] explore the relationship between accent and the ASR task, and show that the accent recognition task can lead to a more robust performance of the main ASR task.

Although research in this area has shown that AR can be an auxiliary task to boost the ASR performance, when ASR performs as the auxiliary task for AR, the effects have not been closely examined. How to combine AR with the ASR task for a better language-related feature remains to be investigated. This paucity inspires us to design our novel hybrid AR model combined with ASR MTL, which will be discussed in the next section.

3. Proposed Method

In this section, we first start from a baseline that directly models the relationship between the input spectrum and the accent label. This baseline is constructed by the acoustic model and the attention-based aggregation model. However, such a baseline does not make full use of the spoken words. In the next subsection, we further analyze the relationship between the accent label and the speaker information. We propose our ASR MTL method which can incorporate the extra text information to prevent the model from overfitting to the speaker label. Finally, we analyze the different learning target between AR and ASR and propose to use hybrid acoustic models for a more robust prediction. The whole model structure is shown in Figure 1.

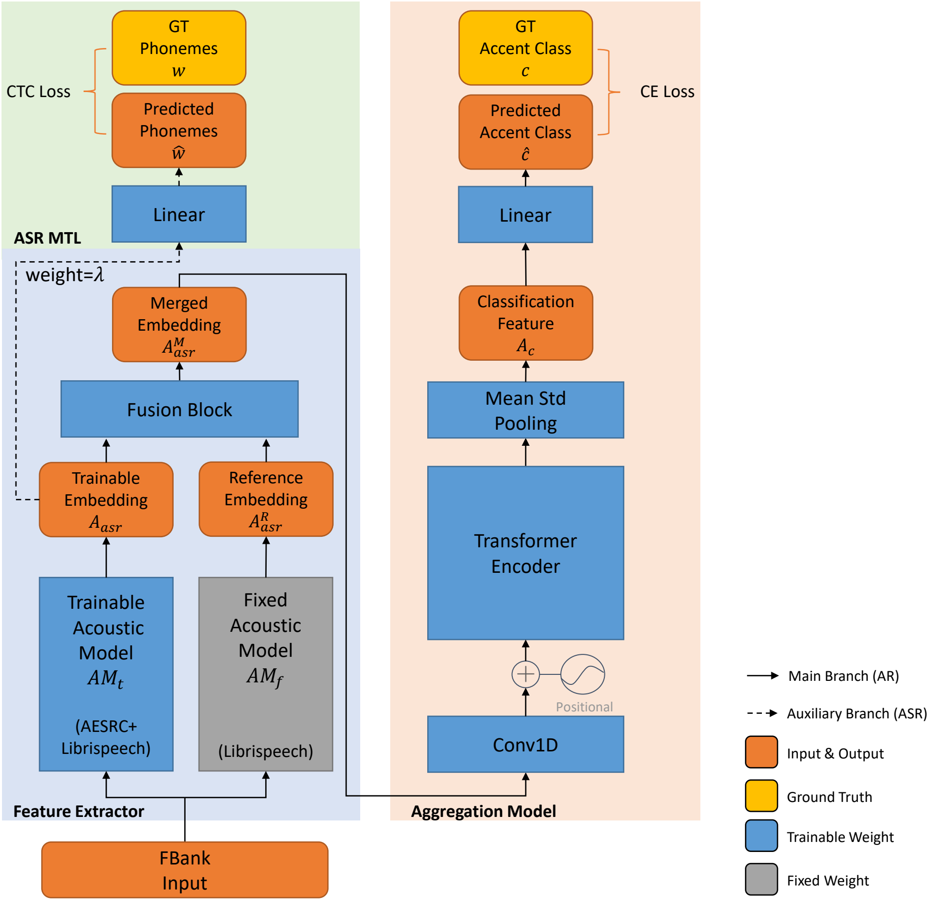


Figure 1. Proposed hybrid structure for accent recognition. We list the dataset used for the acoustic model (AM) in the brackets and use the gray color to indicate the fixed acoustic model does not participant in the AR training process. The auxiliary branch plotted in dash line (the green block) is used only during training.

3.1. Network Structure

To summarize, acoustic scene classification, speaker recognition, and AR tasks discussed in Section 1 all process a sequence into a certain label. As the AR task needs more fine-grained features, we choose to adopt an ASR-based backbone to extract the high-level features for each frame, and then aggregate them for the final accent prediction.

In this paper, we use a CNN-based acoustic model called Jasper [17] to extract deep phonetic features from the input 40-dim FBank spectrum. Jasper is constructed by multi-Jasper blocks, and each Jasper block is further constructed by multi-CNN-based subblocks (1D Conv, BatchNorm, ReLU, dropout) with residual connections. We use the Jasper 5×3 version (5 Jasper blocks, each Jasper block has 3 subblocks) for our experiments.

The attention mechanism can be a good choice to aggregate the extracted acoustic features, and there are also attention-based aggregation methods successfully applied to the accent classification task, including [13] and the self-attention pooling layer [14]. However, a single self-attention pooling layer appended to the acoustic model may not be enough. As we also perform an auxiliary ASR task (discussed in the next subsection) for the acoustic model, the acoustic model still needs to adapt from the ASR task to the AR task. In other words, a single pooling layer may be too shallow to transfer the task.

For our method, we tend to keep the aggregation model independent from the acoustic model so that the ASR loss applied to the acoustic model can be more efficient. Thus, we choose to stack the multi-head self-attention layers for a high-level extraction of the accent attribute for each frame. This extractor is, in fact, the encoder part of Transformer [18]. The attention mechanism finds the similarity between the query matrix and the key matrix, and use the similarity score to obtain the output from the value matrix. In our experiments, the attention mechanism is applied in the self-attention form, where the query, key, and value matrix is the same (the input of the self-attention layer). The self-attention layer finds the relationship between acoustic features across different time steps, and outputs the processed ones.

Formally, the attention mechanism can be described as,

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$ is the dot-product similarity of the query matrix Q and key matrix K^T (scaled by $\frac{1}{\sqrt{d_k}}$), constructing the weight of the value matrix V. The multi-head attention (MHA) is described as,

$$MHA(Q, K, V) = concat(hd_1, \dots, hd_h)W^O, \tag{2}$$

where $hd_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$. h is the head number, and W^Q, W^K, W^V, W^O is the projections matrices for query, key, value, and output, correspondingly. For the self-attention case, $Q = K = V = x$, where x is the input features for each attention layer.

Compared with the self-attention pooling layer, multiple attention heads help the model to capture different representation subspaces at different positions of the accent.

After the stacked multi-head self-attention layers, we use a statistic pooling layer [19] to compute the mean and standard deviation across the time dimension of the acoustic sequence. Finally, we use a fully connected layer to project the aggregated features into the final accent prediction.

For the aggregation model, we set the attention dim $d_{attn} = 256$, the feed-forward dim $d_{ff} = 1024$, the head number $h = 4$, and the number of stacked multi-head attention layers $n_{layers} = 3$.

3.2. Multitask Learning

As discussed in the aforementioned Section 1, AR requires a more fine-grained feature compared with acoustic scene classification and speaker recognition. Meanwhile, acoustic scene classification and speaker recognition can be a language-agnostic task, which is easier compared to the language-related AR task. Consequently, the AR task may degrade to the speaker recognition task if we do not set extra constrains to the learning targets.

Formally, if we denote the model parameters as θ, the acoustic features for classification as A_c, the accent label as c, the predicted accent class $\hat{c}$ can be described as,

$$\hat{c} = argmax_c P(c|A_c; \theta), c \in \{c_i, \dots, c_n\}, \tag{3}$$

where n is the number of accents in the dataset.

We use the cross-entropy (CE) loss as the classification loss function between the predicted accent class $\hat{c}$ and the ground truth label c,

$$l_c = CE(c, \hat{c}). \tag{4}$$

However, we should note that for the training dataset, the accent attribute is labelled for each speaker. In other words, each accent label c_j can be inferred from the speaker label s_i.

As a result, if we only apply the cross-entropy loss for the accent classification, a possible solution for the network is to simply memorize each speaker s in the training dataset and learn to predict the speaker label $\hat{s}'$

$$\hat{s}' = argmax_s P(s|A_c;\theta), s \in \{s_i, \ldots, s_m\} \tag{5}$$

and map it to the accent label $\hat{c}'$, where m is the total number of speakers in the training dataset.

In other words, the network will giving up learning the accent label and overfit to the speaker label on the trainset as discriminating the speaker is easier than accent recognition. For unseen speakers, the AR performance will be severely degraded. We demonstrate this phenomenon in Section 4.3.

We assume that accent recognition using speaker-invariant features is more accurate than the method via speaker recognition, and we force the network to learn the language-related information. On the one hand, the text transcription is independent of the speaker information, and ASR MTL is suitable for this task. On the other hand, ASR MTL can offer complementary information for the main AR task. The ASR task forces the model to learn the language-related information, and the fine-grained phonetic feature can contribute to accent recognition. We build another branch of the proposed model in Figure 1 to perform the ASR task during training. The auxiliary ASR branch tries to predict the pronounced phonemes $\hat{w}$ based on the ASR acoustic features A_{asr} and the parameters of the ASR acoustic model θ_{asr},

$$\hat{w} = argmax_w P(w|A_{asr};\theta_{asr}). \tag{6}$$

We convert the text from word-level to phoneme-level using a grapheme to phoneme (G2P) tool (https://github.com/Kyubyong/g2p, accessed on 30 July 2021) and use CTC [20] as the ASR loss function,

$$l_{asr} = CTC(w, \hat{w}). \tag{7}$$

For the proposed model, the feature for accent classification, A_c, is aggregated from the feature for ASR, A_{asr},

$$A_c = AggregationModel(A_{asr}). \tag{8}$$

We use the parameter λ to balance the weight between the ASR task and the AR task,

$$l = l_c + \lambda l_{asr}. \tag{9}$$

3.3. Hybrid Phonetic Features

Although the AR performance can be boosted by the auxiliary ASR task, we should note that there is a difference between the ASR target and the AR target.

Given a certain speech, ASR allows mispronunciations caused by accents and corrects them to the target text. For example, non-native speakers with different accents may pronounce the target word "AE P AH L" (APPLE) to "AE P OW L" or "AE P AO L". We denote this pronounced phoneme as AH_{OW} or AH_{AO}. There is no need for ASR models to distinguish from different accents, and ASR models will ignore different accents and still map both AH_{OW} and AH_{AO} accented speech into the non-accented transcription of this dataset, AH. In other words, ASR models will not work hard to explore the differences between different accents.

On the contrary, AR needs to find the differences between the AH_{OW} and AH_{AO} referring to the original AH. This conflict inspired us to use another acoustic model trained on the non-accented dataset for a fixed reference.

For the proposed method, two Jasper acoustic models are used. The first one is trained on the non-accented dataset (Librispeech). The second one is first trained on the non-accented dataset and then trained on the accented dataset (AESRC, these two datasets will be introduced later). We pretrain these two acoustic models with the CTC loss and keep the model with the lowest validation phone error rate (PER) for further experiments. We freeze the weight parameters of the non-accented acoustic model (AM) and call it the "fixed AM" (denoted as AM_f). The accented one is further fine-tuned together with the aggregation model using Equation (9), and we call it the "trainable AM" (denoted as AM_t). As illustrated in Figure 1, we merge the reference phonetic embedding of the fixed AM into the trainable one with a fusion block.

In this paper, we consider three different fusion blocks as illustrated in Figure 2.

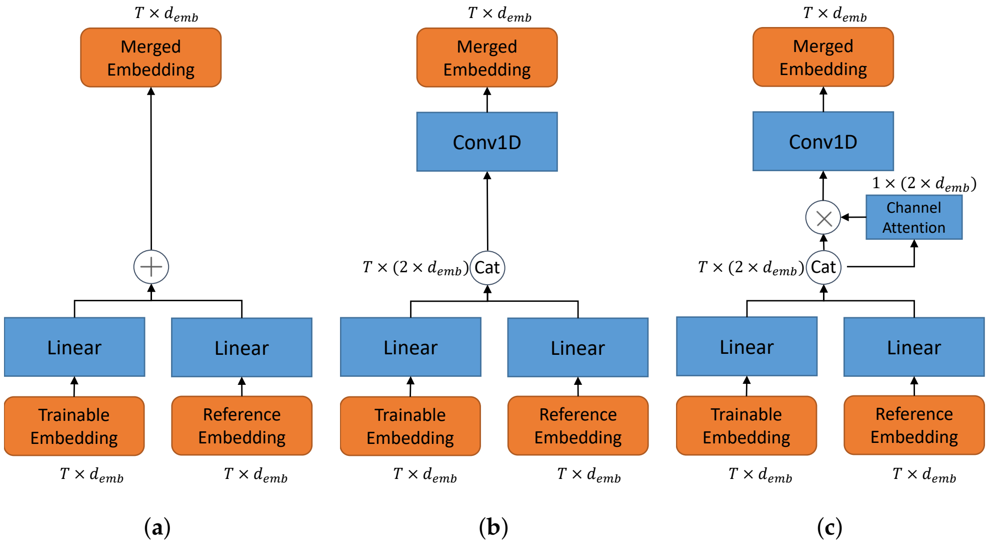


Figure 2. Fusion block with different structures. (**a**) Adding-based fusion. (**b**) Concatenation-based fusion. (**c**) Concatenation–ChannelAttention-based fusion. In (**b**,**c**), a Conv1D layer with kernelsize=1 is adopted to cast the concatenated channels $2 * d_{emb}$ back to d_{emb}.

First, a linear projection is applied to the embeddings,

$$A'_{asr} = linear(A_{asr}), \tag{10}$$

$$A^{R'}_{asr} = linear\left(A^{R}_{asr}\right). \tag{11}$$

Adding-based fusion. As illustrate in Figure 2a, an add function is applied to merge the processed embeddings.

$$A^{M}_{asr} = A'_{asr} + A^{R'}_{asr}. \tag{12}$$

However, simply adding them may lose the information of the raw data.

Concatenation-based fusion. Instead, we stack the embeddings on the channel domain and use a Conv1D(kernelsize=1) to half the channels of the concatenated embedding A^{C}_{asr}. This structure is illustrated in Figure 2b. This process can be described as,

$$A^{C}_{asr} = ChannelCat\left(A'_{asr}, A^{R'}_{asr}\right), \tag{13}$$

$$A^{M}_{asr} = Conv\left(A^{C}_{asr}\right), \tag{14}$$

where $A^{C}_{asr} \in \mathbb{R}^{T\times 2d_{emb}}$ and $A^{M}_{asr} \in \mathbb{R}^{T\times d_{emb}}$.

Concatenation-ChannelAttention-based fusion. Furthermore, we adopt channel-attention to better control the importance of the reference embedding as illustrated in Figure 2c. The channel-attention is obtained by the squeeze-excite block [21,22]. Based on Equation (13), the concatenated feature A_{asr}^{C} goes through a global MaxPooling and a global AveragePooling on the time domain,

$$C = TimeMax(A_{asr}^{C}) + TimeAve(A_{asr}^{C}), \tag{15}$$

where $C \in \mathbb{R}^{1\times 2d_{emb}}$. C is further squeezed and excited,

$$C_{squeeze} = ReLU(linear(C)), \tag{16}$$

$$C_{excite} = ReLU(linear(C_{squeeze})), \tag{17}$$

where $C_{squeeze} \in \mathbb{R}^{1\times \frac{2d_{emb}}{r}}$ and $C_{excite} \in \mathbb{R}^{1\times 2d_{emb}}$. We choose $r = 16$.

The channel-attention CA is obtained by the sigmoid activation of C_{excite},

$$CA = Sigmoid(C_{excite}), \tag{18}$$

and used as the scaling factor of A_{asr}^{C} on the channel domain,

$$A_{asr}^{M} = Conv\left(CA \cdot A_{asr}^{C}\right). \tag{19}$$

Finally, for these three fusion methods, we aggregate the merged ASR feature A_{asr}^{M} instead of the original A_{asr} in Equation (8) and optimize the whole model with the aforementioned MTL loss (Equation (9)) for AR classification. We use 39 English phonemes plus ⟨BLANK⟩ for CTC classification, and there are 8 accents included in the AESRC dataset. The detailed model description is summarized in Table 1.

Table 1. Output summary of the proposed model. T_{in} is the length of the input sequence. Please note that the acoustic model will downsample the feature sequence by 2 on the time domain, i.e., $T_{in} = 2T$.

Module	Output Shape	Output Description
FBank	$T_{in} \times d_{fbank}$	Input feature, $d_{fbank} = 40$
AM_t	$T \times d_{emb}$	Trainable embedding, A_{asr}, $d_{emb} = 1024$
Linear	$T \times d_{phoneme}$	Phoneme prediction, $\hat{w}$, $d_{phoneme} = 40$
AM_f	$T \times d_{emb}$	Reference embedding, A_{asr}^{R}
Fusion Block	$T \times d_{emb}$	Merged embedding, A_{asr}^{M}
Conv1D (kernelsize = 1)	$T \times d_{attn}$	Reshaped embedding , $d_{attn} = 256$
Self-Attention Aggregation	$1 \times d_{attn}$	Aggregated feature, A_c
Linear	$1 \times d_{accent}$	Predicted accent class, $\hat{c}$, $d_{accent} = 8$

4. Experiments

In this section, we first give a detailed description of the dataset and the experiment environment. We show the AR results in the following subsection. Next, we conduct a speaker recognition test to demonstrate that the auxiliary ASR task can be of vital importance to keep the AR task from overfitting of the speaker recognition task. Finally, we explore the relationship between the given transcription for the ASR task and the performance of the AR task to test the robustness.

4.1. Dataset and Environment

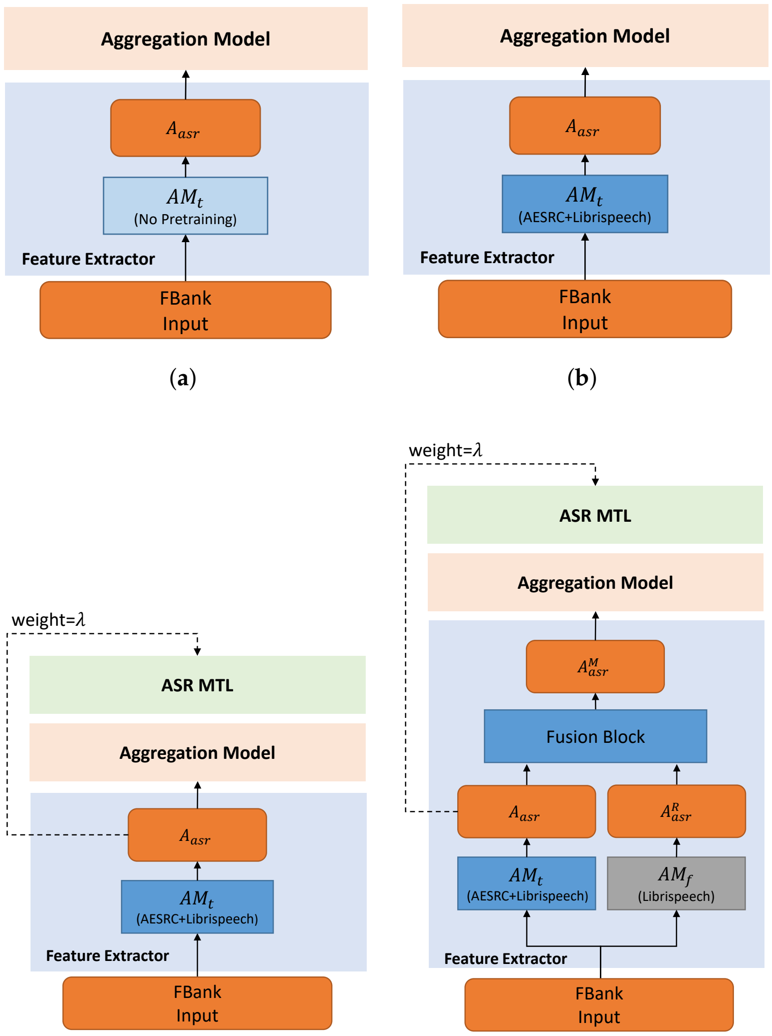
We use two datasets for training in our experiments. The first one is Librispeech [23], which does not contain accent labels. This dataset is constructed by approximately 1000 h of 16 kHz read English speech. The second one is AESRC [6], which is composed of 8 different accents, including Chinese (CN), Indian (IN), Japanese (JP), Korean (KR), American (US), British (UK), Portuguese (PT), Russian (RU). Each accent is made up of 20 h of speech data collected from around 60 speakers on average. The speech recordings are collected in relatively quiet environments with cellphones and provided in Microsoft wav format (16 kHz, 16 bit and mono).

Utterances read by certain specific speakers are kept for the test set.

We use Pytorch [24] to build our systems. We use the 40-dim FBank spectrum features extracted by Kaldi toolkit [25] for the input. Furthermore, we apply SpecAug [26] to perform the data augmentation. We conduct the experiments on a server with Intel Xeon E5-2680 CPU, and 4 NVIDIA P100 GPUs. We use the Adam optimizer ($lr = 10^{-4}$) for both the ASR pretraining and the AR task. The ASR pretraining task on Librispeech takes about 3.5 days to converge, while adapting to the merged Librispeech plus AESRC takes about another 2 days. For the accent recognition task, we train each version for about one day. All experiments are conducted using the same hardware.

4.2. Accent Recognition Test

To better demonstrate the training process, we show different training methods in Figure 3.



(**a**) (**b**)

(**c**) (**d**)

Figure 3. Different training methods. (**a**) Directly train the whole model for the AR task. (**b**) Pretrain the acoustic model for ASR and then train the whole model for AR. (**c**) Pretrain the acoustic model for ASR and then train the whole model for AR and ASR MTL (weight = λ). (**d**) The proposed hybrid method. Based on (**c**), the non-accented embedding is merged for reference.

We show the accent recognition results on the validation set of different models in Table 2. The baseline systems use the encoder of Transformer and a statical pooling to perform the AR task. In Table 2, they are denoted as Transformer-XL, where X represents the number of the encoder layers.

Table 2. Accuracy comparison. The vanilla versions (Id 0–2, 4) are demonstrated in Figure 3a. ASR-initialized versions (Id 3,5) in Figure 3b. ASR MTL versions (Id 6–8) in Figure 3c. The fusion-based versions (Id 9–11) in Figure 3d (Figure 2a–c, correspondingly.)

Id	Model	US	UK	CN	IN	JP	KR	PT	RU	Ave
	Baseline									
0	Trans 3L	45.7	70.0	56.2	83.5	48.5	45.0	57.2	30.0	54.1
1	Trans 6L	30.6	74.5	50.9	75.2	44.0	43.7	65.7	34.0	52.2
2	Trans 12L	21.2	85.0	38.2	66.1	42.7	26.0	51.8	49.6	47.8
3	Trans 12L(ASR)	60.2	93.9	67.0	97.0	73.2	55.6	85.5	75.7	76.1
	Proposed									
4	Vanilla	45.7	81.0	40.1	79.1	45.7	37.8	84.5	63.6	60.0
5	ASR	40.3	93.7	75.0	97.3	76.3	52.1	88.3	76.0	75.1
6	MTL, $\lambda = 0.1$	68.6	91.8	86.9	99.1	71.2	56.6	89.5	76.0	79.9
7	MTL, $\lambda = 0.2$	57.8	92.0	79.5	98.5	70.6	71.3	84.7	66.2	77.5
8	MTL, $\lambda = 0.3$	60.9	85.3	85.1	98.9	66.1	69.6	80.8	63.2	76.0
9	Add, $\lambda = 0.1$	50.4	95.0	83.3	99.4	72.1	73.5	92.9	77.0	80.6
10	Cat, $\lambda = 0.1$	61.8	88.9	89.6	98.9	71.9	66.0	95.1	74.5	80.8
11	CatCA, $\lambda = 0.1$	63.1	93.3	88.9	98.3	73.9	66.3	95.3	73.7	82.2

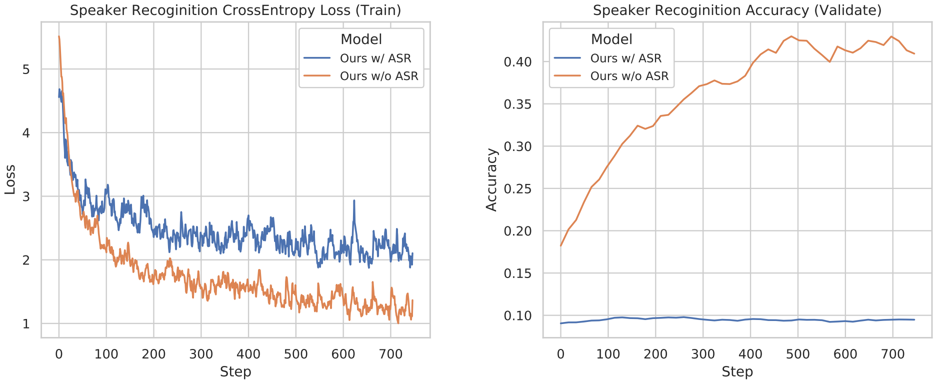
As we can see from Table 2, a bigger model (from Id 1 to Id 3) will lead to overfitting and a degraded classification accuracy. However, as we can observe from both the baseline models and our models, this overfitting phenomenon can be alleviated by the ASR task. ASR pretraining on Librispeech and AESRC (Id 3, Id 5) can greatly improve the AR accuracy compared with the model without pretraining (Id 2, Id 4, correspondingly). Furthermore, the proposed MTL version (Id 6–8) is better than ASR initiation only. We obtain the best result of the MTL-based models by setting the ASR weight to $\lambda = 0.1$. By merging the outputs of the fixed AM and the trainable AM, the proposed hybrid methods (Id 9–11) show a better performance compared with the MTL version. The Concatenation-ChannelAttention-based fusion method (Id 11) shows the best performance among these models, which results in an 8.02% relative improvement over the best ASR-initialized baseline (Id 3). Please note that the performance of US data is relatively low. We think this is because the accented dataset supposes that the data collected in each country belongs to one type of accent. Thus, the accent label is decided by which country the speaker belongs to, rather than their native language. As speakers in the US are various, the performance for US can be downgraded.

4.3. Speaker Recognition Test

As discussed above, the model with ASR MTL is better than the one without the ASR pretraining (Id 6, Id 4 in Table 2). To validate our assumption in Section 3.2 that directly training the network for AR may overfit to the speaker label on the trainset, and ASR MTL can help the model to learn a speaker-invariant feature thus alleviating this phenomenon, we compare the speaker recognition performances of these two models.

To probe whether the output of these two models contains speaker information, we freeze the weight parameters of these two models and replace the final linear layer of the aggregation model to perform the speaker recognition task. We only train this linear layer to see how the original feature extracted by each model correlates with the speaker information. We train the linear layer with the cross-entropy loss and Adam optimizer ($lr = 10^{-4}$). We should note that for the original AR experiments, the training and validation dataset is split by different speakers. To perform the speaker recognition task, we did another splitting by utterances, and a certain speaker appears in both the training set and the validation set.

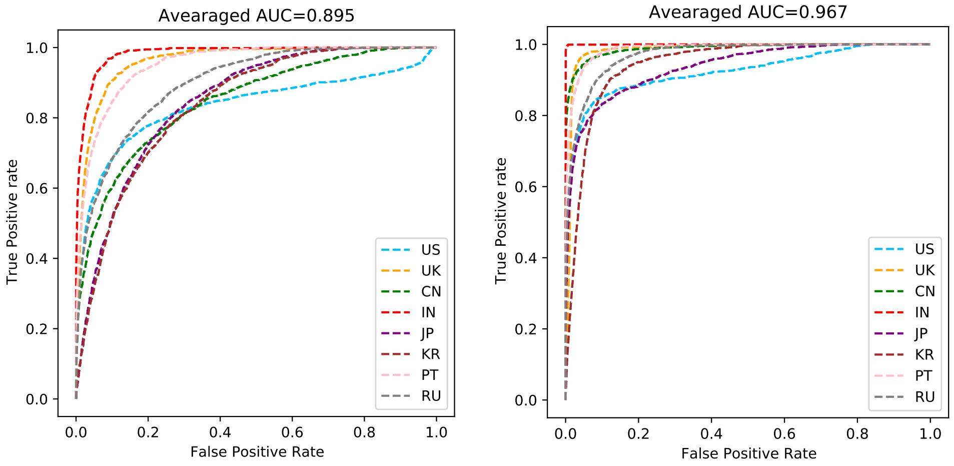
We show the training process in Figure 4a and the validation process in Figure 4b.



(**a**) (**b**)

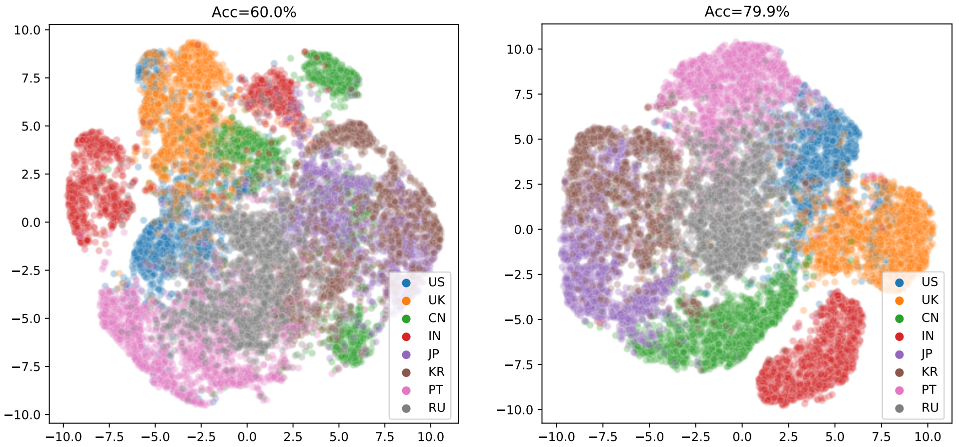
Figure 4. Speaker recognition test. (**a**) Cross-entropy loss curve for the speaker label while training the models with/without the ASR task. (**b**) Speaker recognition accuracy curve on the validation dataset while training the models with/without the ASR task.

The model without ASR pretraining has a lower training cross-entropy loss for the speaker label and a much higher speaker recognition accuracy compared to the ASR MTL version. Such a phenomenon suggests that adding the ASR task indeed helps the model to learn a speaker-invariant feature. We plot ROC curves of the compared two models in Figure 5. As shown in Figure 5b, the speaker-invariant feature improves the Area Under Curve (AUC) score (the averaged AUC score is 0.967) greatly compared to Figure 5a (the averaged AUC score is 0.895), suggesting that this feature is more powerful to solve the AR task. We also plot the embeddings of the predictions on the validation dataset in Figure 6 using t-SNE [27]. The embeddings learned by the MTL version are in a more reasonable subspace.



(**a**) (**b**)

Figure 5. ROC curve of each accent. (**a**) The baseline version without the auxiliary ASR task. The averaged Area Under Curve (AUC) score is 0.895. (**b**) The proposed MTL version. The averaged Area Under Curve (AUC) score is 0.967.



(**a**) (**b**)

Figure 6. 2D embeddings of the accent features. (**a**) The baseline version without the auxiliary ASR task. (**b**) The proposed MTL version.

4.4. Robustness Test

As noted in Section 3.3, if the ASR dataset is constructed by non-native speakers with accents, the transcription of certain phonemes may be labelled as the text that the speakers are asked to read, rather than the pronounced one. In other words, pronunciations with different accents are merged and all labelled as the same text. If the model is dominated by the auxiliary ASR task, it will be hard for the model to find the uniqueness for different accents. As a result, the performance may be degraded for the AR task. In this subsection, we set experiments to validate the performance of the proposed hybrid model under different ASR situations, as shown in Table 3.

To simulate the transcription confusion introduced by accents, we randomly map a certain phoneme w to the upper-level group it belongs to (denoted as $G(w)$), with the probability p sampled from a uniform distribution $U(0,1)$. The hierarchy of phonemes in English language can be seen in Figure 7, and is commonly used by Ref. [28–30].

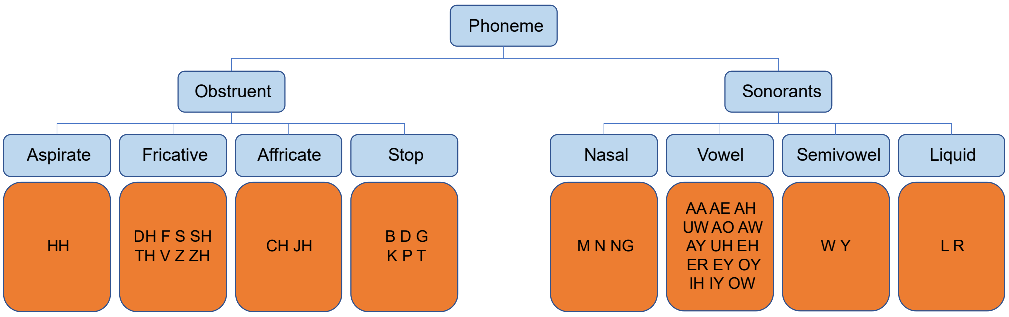


Figure 7. The hierarchy of phonemes in English language.

Formally, this process can be denoted as follows,

$$w' = \begin{cases} G(w) & \text{if } p < \theta \\ w & \text{otherwise} \end{cases}, p \sim U(0,1). \tag{20}$$

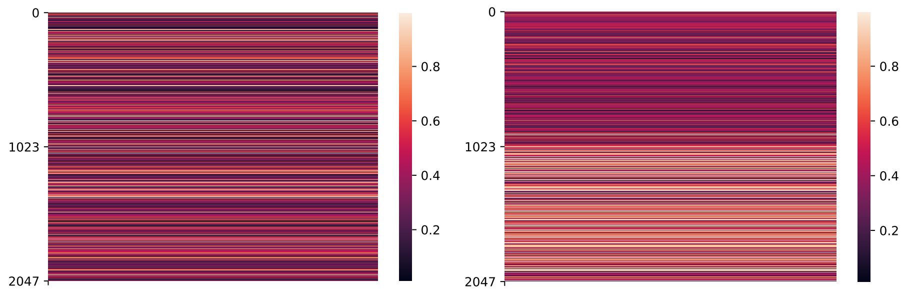
As θ increases, there will be more pronunciations that are not labelled as the original phonemes, but are messed into the upper group. We use the group index for classification instead of the original phoneme index.

Meanwhile, to test the effect of the phonetic information for AR, we also test an extreme situation that the text transcriptions are randomly generated. We still use the MTL version (Figure 3c) for experiment, but the trainable acoustic model AM_t is not pretrained on AESRC or Librispeech. Under this situation, the whole model learns the wrong phonetic information of the accented speech. We also test the proposed hybrid version (Figure 3d, AM_t is also not pretrained) for comparison.

Table 3. Accuracy for the robustness test under different transcription situations. For example, "50% ↓" suggests that $\theta = 0.5$ (i.e., 50% of the transcriptions are messed). All models use $\lambda = 0.1$. For the hybrid version, we use the channel-attention-based fusion (Figure 2c). Please note that Id 12, 13 is in fact the same as Id 6, 11 in Table 2, correspondingly.

Trans	Id	Model	US	UK	CN	IN	JP	KR	PT	RU	Ave
0% ↓	12	MTL	68.6	91.8	86.9	99.1	71.2	56.6	89.5	76.0	79.9
	13	Hybrid	63.1	93.3	88.9	98.3	73.9	66.3	95.3	73.7	82.2
50% ↓	14	MTL	54.5	93.7	91.9	99.6	62.6	67.2	79.6	66.2	76.8
	15	Hybrid	50.1	91.8	88.3	98.6	72.9	63.7	94.6	73.3	79.3
100% ↓	16	MTL	49.0	93.6	86.6	99.5	56.6	58.4	76.7	77.1	74.7
	17	Hybrid	52.6	90.9	78.6	98.3	72.7	59.6	97.5	68.9	77.5
Random	18	MTL	28.8	81.9	33.6	73.3	39.1	34.9	73.3	45.1	51.5
	19	Hybrid	46.6	73.5	61.8	88.6	49.7	43.0	86.6	68.1	64.8

As the random transcription is not helpful for the AR task, the channel-attention mechanism learns to pay more attention to the reference embedding in Figure 8b compared to the normal situation in Figure 8a.



(**a**) (**b**)

Figure 8. Channel-attention weight ($CA \in \mathbb{R}^{1 \times 2d_{emb}}$) under different training conditions. When the transcription becomes unreliable, the channel-attention block pays more attention to the reference embedding. For the model trained with normal transcription (**a**), summed attention weight for the reference embedding dividing summed attention weight for the trainable embedding is $(\sum_{c=d_{emb}}^{2d_{emb}} CA_c)/(\sum_{c=0}^{d_{emb}-1} CA_c) = 1.02$. In contrast, for the model trained with randomly generated transcription (**b**), $(\sum_{c=d_{emb}}^{2d_{emb}} CA_c)/(\sum_{c=0}^{d_{emb}-1} CA_c) = 1.63$.

5. Conclusions

In this paper, we propose that the AR model may overfit to the speaker recognition task as using the speaker information in the trainset to distinguish the accent label is easier compared with using language-related information. To validate this assumption, we probe the speaker-related information of the baseline and the proposed MTL version by freezing the model and replacing the last linear layer for another speaker recognition training. The results show that the auxiliary ASR task can force the model to extract speaker-invariant and language-related features, and this auxiliary task can lead to a better AR performance.

Furthermore, the hybrid structure is designed to fuse the embeddings of two acoustic models. With the proposed Concatenation-ChannelAttention-based fusion, the model can choose to pay more attention to the standard reference embedding if the accented transcription is not reliable. Thus, the extracted features can be more robust. The proposed method is 8.02% better than the best Transformer-based baseline relatively, showing the merits of our method.

However, despite the improved performance, the training process of the proposed method is quite complex as it requires two acoustic models to be trained separately. Meanwhile, as the accent recognition model is generally applied as a frontend to custom the downstream system, we expect the model to be smaller in its parameters. Therefore, it is worthy to explore a more efficient accent prediction structure in the future.

Author Contributions: Investigation, Y.W.; Methodology, Z.Z.; Resources, J.Y.; Software, Z.Z.; Supervision, Y.W. and J.Y.; Validation, Y.W.; Writing original draft, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chu, X.; Combs, E.; Wang, A.; Picheny, M. Accented Speech Recognition Inspired by Human Perception. 2021. Available online: https://arxiv.org/pdf/2104.04627.pdf (accessed on 30 July 2021) .
2. Huang, C.; Chen, T.; Li, S.; Chang, E.; Zhou, J. Analysis of speaker variability. In Proceedings of the 7th European Conference on Speech Communication and Technology, Aalborg, Denmark, 3–7 September 2001; pp. 1377–1380.
3. Levis, J.; Barriuso, T. Nonnative speakers' pronunciation errors in spoken and read English. In Proceedings of the 3rd Pronunciation in Second Language Learning and Teaching (PSLLTP), Ames, IA, USA, 16–17 September 2011; pp. 187–194.
4. Viglino, T.; Motlicek, P.; Cernak, M. End-to-end accented speech recognition. In Proceedings of the 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019 ; pp. 2140–2144. [CrossRef]
5. Crawshaw, M. Multi-Task Learning with Deep Neural Networks: A Survey. 2020. Available online: https://arxiv.org/abs/2009.09796 (accessed on 29 July 2021) .
6. Shi, X.; Yu, F.; Lu, Y.; Liang, Y.; Feng, Q.; Wang, D.; Qian, Y.; Xie, L. The accented english speech recognition challenge 2020: Open datasets, tracks, baselines, results and methods. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada, 6–11 June 2021; pp. 6918–6922.
7. Naranjo-Alcazar, J.; Perez-Castanos, S.; Zuccarello, P.; Cobos, M. CNN Depth Analysis with Different Channel Inputs for Acoustic Scene Classification. 2019. Available online: https://arxiv.org/abs/1906.04591 (accessed on 28 July 2021)
8. Wu, Y.; Lee, T. Enhancing Sound Texture in CNN-based Acoustic Scene Classification. In Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 12–17 May 2019; pp. 815–819. [CrossRef]
9. Zheng, W.; Mo, Z.; Xing, X.; Zhao, G. CNNs-Based Acoustic Scene Classification Using Multi-Spectrogram Fusion and Label Expansions. 2018. Available online: https://arxiv.org/abs/1809.01543 (accessed on 30 July 2021).
10. Behravan, H.; Hautama, V.; Siniscalchi, S.M.; Kinnunen, T.; Lee, C.H. Introducing attribute features to foreign accent recognition. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 5332–5336.
11. Mannepalli, K.; Sastry, P.N.; Suman, M. MFCC-GMM based accent recognition system for Telugu speech signals. *Int. J. Speech Technol.* **2016**, *19*, 87–93. [CrossRef]
12. Behravan, H.; Hautamaki, V.; Siniscalchi, S.M.; Kinnunen, T.; Lee, C.H. I-Vector Modeling of Speech Attributes for Automatic Foreign Accent Recognition. *Ieee/Acm Trans. Audio Speech Lang. Process.* **2016**, *24*, 29–41. [CrossRef]
13. Chen, J.; Cai, W.; Cai, D.; Cai, Z.; Zhong, H.; Li, M. End-to-end language identification using NetFV and NetVLAD. In Proceedings of the 2018 11th International Symposium on Chinese Spoken Language Processing, Taipei, Taiwan, 26–29 November 2018; pp. 319–323. [CrossRef]
14. Cai, W.; Chen, J.; Li, M. Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System. In Proceedings of the Odyssey 2018 The Speaker and Language Recognition Workshop, Les Sables d'Olonne, France, 26–29 June 2018; pp. 74–81.
15. Najafian, M.; Russell, M. Automatic accent identification as an analytical tool for accent robust automatic speech recognition. *Speech Commun.* **2020**, *122*, 44–55. [CrossRef]
16. Weninger, F.; Sun, Y.; Park, J.; Willett, D.; Zhan, P. Deep learning based Mandarin accent identification for accent robust ASR. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 510–514. [CrossRef]

17. Li, J.; Lavrukhin, V.; Ginsburg, B.; Leary, R.; Kuchaiev, O.; Cohen, J.M.; Nguyen, H.; Gadde, R.T. Jasper An End-to-End Convolutional Neural Acoustic Model. In Proceedings of the Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 71–75.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
19. Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. X-vectors: Robust dnn embeddings for speaker recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
20. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *Acm Int. Conf. Proc. Ser.* **2006**, *148*, 369–376. [CrossRef]
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ, USA, 18–22 June 2018; pp. 7132–7141. [CrossRef]
22. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
23. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5206–5210. [CrossRef]
24. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
25. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlíček, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Big Island, HI, USA, 11–15 December 2011.
26. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 2613–2617. [CrossRef]
27. Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
28. Hamooni, H.; Mueen, A. Dual-Domain Hierarchical Classification of Phonetic Time Series. In Proceedings of the 2014 IEEE International Conference on Data Mining, Shenzhen, China, 14–17 December 2014; pp. 160–169. [CrossRef]
29. Dekel, O.; Keshet, J.; Singer, Y. An Online Algorithm for Hierarchical Phoneme Classification. In Proceedings of the International Workshop Machine Learning for Multimodal Interaction, Edinburgh, UK, 11–13 July 2005; pp. 146–158. [CrossRef]
30. Lee, K.F.; Hon, H.W. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 1641–1648. [CrossRef]