ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414471

AISPEECH-SJTU ASR SYSTEM FOR THE ACCENTED ENGLISH SPEECH RECOGNITION CHALLENGE

Tian Tan[1,2], *Yizhou Lu*[2], *Rao Ma*[2], *Sen Zhu*[1], *Jiaqi Guo*[1], *Yanmin Qian*[2]

[1] AISpeech Ltd, Suzhou China
[2] MoE Key Lab of Artificial Intelligence, AI Institute,
SpeechLab, Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China
{tian.tan, sen.zhu, jiaqi.guo}@aispeech.com, {luyizhou4, rm1031, yanminqian}@sjtu.edu.cn

ABSTRACT

This paper describes the AISpeech-SJTU ASR system for the Interspeech-2020 Accented English Speech Recognition Challenge (AESRC). This task is challenging due to the diversity of pronunciation accuracy, intonation speed and pronunciation of some syllables. All participants were restricted to develop their systems based on the speech and text corpora provided by the organizer. To work around the data-scarcity problem, data augmentation was first explored including noise simulation, SpecAugment, speed perturbation and TTS simulation. Moreover, SOTA CNN-transformer-based joint CTC-attention system was built and accent adaptation was proposed to train an accent robust system. Finally, the first-pass recognition hypotheses generated from CTC head were rescored by forward, backward LSTM-LM and the attention head. Our system with the best configuration achieves second place in the challenge, resulting in a word error rate (WER) of 4.00% on dev set and 4.47% WER on test set, while WER on test set of the top-performing, second runner-up and official baseline systems are 4.06%, 4.52%, 8.29%, respectively.

Index Terms— accent speech recognition, accent adaptation, data augmentation, RNNLM

1. INTRODUCTION

After decades of development, speech techniques have improved significantly, ASR has achieved human parity in conversational speech recognition [1, 2]. Recently, end-to-end (E2E) ASR [3, 4, 5, 6, 7, 8] has made promising progress. It provides better results by directly optimizing the probability of output sequences given input speech observations with a single network. Joint CTC-attention model [9] takes the advantages from both CTC and sequence-to-sequence models by multi-task learning and obtained better performance and robustness. More recently, transformer network [10] first proposed for Neural Machine Translation was applied for ASR tasks and outperformed RNN-based end-to-end models [11]. However, there are still problems that degrade ASR performance a lot, such as recognising English with accents. The difficulty of recognising accented English includes the diversity of pronunciation accuracy, intonation speed and pronunciation of some syllables. Moreover, collecting adequate training data is challenging for accented English recognition. There has been a lot of research on solving this problem. In [12, 13], models were boosted using data from other dialects. Another usual approach is to define a common set of universal phone models [14, 15] and adapt the model on data from the language of interest [16, 17]. Model adaptation was proposed in [17, 18, 19, 20] to get a language dependent acoustic model.

The Interspeech-2020 AESRC challenge [21] focus on recognising accented English with limited training data, eight sets of accented English data from different countries were provided to the participants, covering various pronunciation characteristics and accents. The training corpus was restricted, only Librispeech corpus (960 hours) and accented English data (160 hours) were allowed to be used for training models. Organizers had developed the baseline system using state-of-the-art techniques including transformer based sequence to sequence ASR and SpecAugment.

This paper describes the AISpeech-SJTU ASR system for the AESRC challenge. To deal with the under resourced issue. Data augmentation technology is first investigated, including noise simulation [22, 23], speed perturbation [24], SpecAugment [5] and TTS simulation. Furthermore, our ASR system is built on the basis of the SOTA end-to-end ASR system, transformer and CNN are used as encoder network to extract more robust hidden representation. The whole model is trained by joint CTC-attention multi-task training. Meanwhile, accent adaptation is proposed, predicted accent label and accent embedding are investigated to adapt the hidden activation of the encoder network. Apart from the above, we propose a rescore scheme, first-pass recognition hypotheses are generated by CTC-head and then rescored by advanced NN language model and attention-head. After using all proposed methods, a significant improvement is obtained compared to the baseline system announced by the organizer. Figure 1 outlines the main contributions of our system.

The remainder of this paper is organized as follows: Section 2 shows the neural network model used in our system. Section 3 presents the details of the data augmentation. Section 4 describes the proposed accent adaptation method. Section 5 presents the NN language model used in this work and the proposed rescore scheme. After that, the experimental setup, results and discussion are given in Section 6. We conclude the paper Section 7.

2. ACOUSTIC MODEL

A GMM-HMM system was first trained for generating frame-level alignment. It was a MFCC-LDA-MLLT system with 8196 senones trained by standard Kaldi [25] recipe. A forced-alignment was performed to get the frame-level monophone labels for CE training.

Yanmin Qian is the corresponding author.

This work is supported by the China NSFC project (No. 62071288)

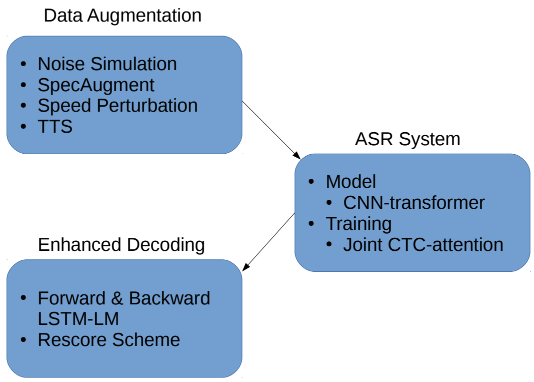

Fig. 1. Highlight of the AISpeech-SJTU ASR system

Joint CTC-attention model [9] was used in this work. The encoder contains four convolution layers; each layer follows a batch norm layer. After each two convolution layers, max-pooling was applied to half the feature map on both time and frequency dimensions. Then twenty transformer layers were stacked after the last max-pooling layer. The decoder contains six transformer layers. 40 dimensions and 80 dimensions FBANK features were compared as the frame-level acoustic feature vectors to be fed to the ASR system. Utterance level cepstral mean normalization (CMN) operation was conducted.

2.1. CE Initialization

Pre-training was adopted to get a better-initialized encoder model. The encoder was first trained to predict frame-level monophone labels generated by the GMM-HMM system. This system was further used in challenge Track 1 (Accent Classification Task). The predicted monophone posterior was used as phone posteriorgram (PPG) feature for accent classification.

2.2. Joint CTC-attention multi-task training

After CE initialization, the output layer of the encoder was replaced to predict BPE [26], and multi-task training was used to train the encoder and decoder simultaneously. The encoder network is shared with CTC and attention models. The output from the encoder predicts frame-level BPE posterior, and the output from the decoder predicts token-level BPE sequence. The final criteria is CTC loss on frame-level posterior plus CE loss on token-level BPE sequence, as shown in Equation 1.

$$\mathcal{L}_{\text{mtl}} = \mathcal{L}_{\text{ctc}} + \mathcal{L}_{\text{attention}} \tag{1}$$

Unlike in [9], CTC was used as an auxiliary task. In this work, CTC acted as important as attention; a lattice was first generated using WFST-based CTC decoding with word-level 4-gram. Then top-20 hypotheses were selected and re-ranked by the attention model. This method got better performance than decoding use only CTC or attention, and the decoding speed was faster than joint CTC-attention decoding in [9].

3. DATA AUGMENTATION

Only official 160 hours data (20 hours for each accent) and Librispeech were allowed to train the acoustic model in this challenge; it is too limited to train a robust acoustic model. Thus, data augmentation, which has been successfully applied in industry, was adopted to solve the data-scarcity problem. Four different methods were investigated in this work, including Noise Simulation, SpecAugment, Speed Perturbation, and TTS simulation.

3.1. Noise Simulation

Noise simulation is a popular technique that can generate noise data and has been successfully applied to HMM-based systems and E2E models. In this work, fifteen room impulse responses and fifteen additional noise were used. For each utterance in training data, a combination of impulse responses and additional noise was first chosen randomly; then, three simulated noise samples were generated by simulating reverberation, adding additional noise, or applying both. Thus, the training corpus was expanded 3-times. Repeating this process could generate more simulated data. However, the experiments showed that more is not always better; detailed comparison will be shown in section 6.2.

3.2. SpecAugment

Recently, SpecAugment[5] has shown its powerful generalization ability in many speech tasks, especially for the E2E ASR system. In this work, frequency masking and time masking were adopted; it was done online so that one utterance could generate different training samples in different epochs. Noise simulated data was also applied SpecAugment in this work.

3.3. Speed Perturbation

In this work, utterance-level speed perturbation was performed [24]. The speaking rate of a speech utterance was modified by re-sampling its waveform signal. An additional copy of the original speech training data was created by randomly choosing speaking rate 0.9 or 1.1 for each training utterance. Thus, the training data had been doubled.

3.4. TTS

Based on recent development in speech synthesis (text-to-speech, or TTS), a TTS system was built on the ASR training data. In this work, eight individual TTS models were trained for each accent. A FastSpeech [27] based multi-speaker speech synthesizer and a LPCNet [28] vocoder were adopted. Both networks were also conditioned on a 256-dimensional x-vector from a pre-trained speaker encoder. Our implementation of FastSpeech was based on the ESPnet toolkit [29].

The TTS model was trained as following:

- First, a general synthesizer was trained on the mixture of 160 hours accent set and the auxiliary 960 hours Librispeech data.
- Then, eight accent-specific synthesizers were generated by finetune the general synthesizer on accent data, respectively.
- Two vocoders were trained separately on a clean subset of the speech data for males and females.

The augmentation process was applied as following:

- To increase the speaker variability, all utterances of a speaker were randomly grouping. Each group contained at most 30 utterances, and it was treated as an individual speaker. The mean, variance of acoustic features and x-vector were then calculated for different groups.

Table 1. WER (%) of baseline system on dev set

System	WER								
	Avg.	**US**	**UK**	**CHN**	**IND**	**JPN**	**KR**	**PT**	**RU**
Official	6.92	7.42	7.64	9.87	7.85	5.71	6.40	5.90	4.60
Baseline	6.97	6.91	6.27	10.69	7.65	6.24	7.57	5.85	4.27

- For each new speaker and each accent, 40 texts were selected from the training reference, and synthesized speech was generated, which retains the speaker's speaking style while adopting a new accent.

4. ACCENT ADAPTATION

In this challenge, accent labels were not provided in the evaluation stage; thus, an accent classifier [30] was trained to supply the accent information for adaptation. A time-delay neural network (TDNN) based accent classifier was trained using official 8-accents speech data and the augmented TTS data. It accepted PPG features as inputs and was trained to predict the eight accent categories. More details can be found in our accent identification system description for this challenge [30].

An accent-specific transformation was applied to the output of the last pooling layer. Let $\mathbf{h}_t$ denote the output of the last pooling layer, scaling and shifting were applied to it as shown in Equation 2.

$$\mathbf{h}_t^a = \boldsymbol{\gamma}(a) \odot \mathbf{h}_t + \boldsymbol{\beta}(a) \tag{2}$$

where $\mathbf{h}_t^a$ is the adapted output for accent a, $\boldsymbol{\gamma}(a)$ and $\boldsymbol{\beta}(a)$ are learnable parameters for accent a, $\odot$ denotes element-wise multiplication.

During the evaluation, the accent label a was predicted by the above classifier. However, using hard labels may cause trouble when classifier makes mistakes which is especially serious for unseen accents in evaluation. To address this problem, we used accent embedding, extracted from the penultimate outputs of accent classifier, as auxiliary feature. We hypothesize that this accent embedding may contain richer information for adaptation, which shall be helpful for improving the robustness of our multi-accent system. Let $\mathbf{z}$ be the accent embedding for an utterance. The scaling vector $\boldsymbol{\gamma}(\mathbf{z})$ and shifting vector $\boldsymbol{\beta}(\mathbf{z})$ were generated by a non-linear transformation as in Equation 3.

$$\boldsymbol{\gamma}(\mathbf{z}) = f(\mathbf{W}_\gamma \mathbf{z} + \mathbf{b}_\gamma), \quad \boldsymbol{\beta}(\mathbf{z}) = g(\mathbf{W}_\beta \mathbf{z} + \mathbf{b}_\beta) \tag{3}$$

where $\mathbf{W}_\gamma, \mathbf{b}_\gamma, \mathbf{W}_\beta, \mathbf{b}_\beta$ are learnable paramters, $f(\cdot) = \mathbf{1} + tanh(\cdot)$ and $g(\cdot) = tanh(\cdot)$.

5. LANGUAGE MODEL AND RESCORE SCHEME

Recurrent neural network language model has been widely used for speech recognition. In this work, inspired by [2, 31, 32], not only forward-predicting LSTM-LMs but backward LSTM-LMs were trained for rescoring hypotheses generated by CTC system using WFST. Backward LSTM-LM is an RNNLM that predicts words sequence in a reverse temporal order. Both forward and backward LSTM-LM were 2-layer LSTM with 1024 hidden cells. Words were used as the model unit, and the vocabulary size is 94169. Cross-entropy was used as the training loss. Both models were first trained using text from Librispeech and accent data. Then they were finetuned using only text from accent data.

The log probabilities from both models were interpolated by weights 0.5 and 0.5 as following:

$$\mathcal{S}_{\text{rnnlm}}(\mathbf{w}) = 0.5 \times \mathcal{S}_{\text{forward}}(\mathbf{w}) + 0.5 \times \mathcal{S}_{\text{backword}}(\mathbf{w}) \tag{4}$$

Where $\mathbf{w}$ is the words sequence. The rescore scheme of our final system is shown as following:

- Generate the n-best hypotheses by decoding CTC head with WFST using a word-level 4-gram LM.
- Generate BPE sequence from each hypothesis and use attention head to calculate the log probability of each sequence.
- Calculate the interpolated word-level log probability using forward and backward LSTM-LM.
- The final score is the sum of above three parts

$$\mathcal{S}_{\text{final}} = \mathcal{S}_{\text{ctc}} + \mathcal{S}_{\text{attention}} + \mathcal{S}_{\text{rnnlm}} \tag{5}$$

6. EXPERIMENTS

In this work, all NN models were trained using PyTorch. BPE was used for CTC and E2E training, 500 BPEs was generated using SentencePiece toolkit [33].

6.1. Baseline System

The baseline system was trained using CTC from scratch with SpecAugment. SpecAugment was applied with 2 frequency masks with maximum frequency mask (F = 15), and 2 time masks with maximum time mask (T = 30) for 40-dim FBANK. Maximum frequency mask (F=27) was used for 80-dim FBANK, other configurations were the same as 40-dim FBANK. Baseline was trained with both Librispeech data and official accented data. A 4-gram was trained using all training text for decoding. The performance of our baseline system and official baseline system [21] are summarised in Table 1.

6.2. Data augmentation

Data augmentation was first evaluated. We first compared using different amounts of noise simulation data. All systems were trained using CTC from scratch with SpecAugment. Firstly, each utterance in the train set was randomly adjusted its speed by 0.9 or 1.1. Then, 1.1K hours noise data was added. As shown in Table 2, significant improvement was obtained by using both noise simulation data and speed perturb data. Using 2 more times noise data obtained further improvement. However, when continually using more noise data, the token accuracy on develop set became worse. Thus, the final submitted system used 3.3K noise data.

Then, the TTS system described in Sec 3.4 was built for generating more accented data. As shown in Table 2, using as much TTS data as the train set can obtain additional improvement compared to the system adding noise simulation data and speed perturbation.

Table 2. WER (%) of adding different amount of noise simulation data and TTS data on dev set

Methods	Hours	WER	
		Libri	Accent
Baseline	1.1K	6.74	6.97
+ Noise & Speed (B2)	3.3K	5.73	6.26
++ 2×Noise	5.5K	5.37	5.99
++ TTS	4.4K	5.75	5.90

6.3. Accent Adaptation

Accent adaptation systems were then constructed. We started with using hard accent ID, oracle accent ID and predicted accent ID were compared. The predicted ID was the class with the highest score in the accent classifier [30]. As shown in Table 3, using predicted ID only degraded a little performance on accent set since it was matched with train data. However, significant degradation was observed on Librispeech data; using hard labels is not a suitable way to do adaptation on unseen accents.

Table 3. WER (%) of accent adaptation on dev set

Methods	WER	
	Libri	Accent
B2	5.73	6.26
+ Oracle ID	5.46	5.61
+ Predicted ID	6.20	5.71
+ Embedding	5.57	5.82

Thus, accent embedding extracted from the accent classifier was used instead of hard labels. As shown in Table 3, using embedding was more stable than using a hard label, significant improvement was obtained on both Librispeech and accent set.

6.4. Final System

Before we combined all proposed techniques, we further explored 80-dim FBANK v.s. 40-dim FBANK, initialize network with CE training and different amount of TTS data. As shown in Table 4 and 5, using 80-dim FBANK feature obtained slightly gain than 40-dim FBANK, significant improvement was obtained with CE initialization. However, using more TTS data didn't show further improvement.

Table 4. WER (%) of using different feature and w/o CE-init on dev set

Methods	WER	
	Libri	Accent
fbank40	5.73	6.26
+ CE-init	5.64	5.78
fbank80	5.76	6.09
+ CE-init (B3)	5.40	5.54

Then, based on the best configuration, the performance of our final system is shown in Table 6, using data augmentation obtained 24% to 26% relative improvement; Accent adaptation and finetune the network with accented data obtained more 13% relative improvement on accent set. Finally, rescoring the 20-best hypotheses with the attention model and RNNLM obtained a further 11% relative improvement. Compare to the baseline system, 42.6% relative improvement was obtained on the accented English set.

Table 5. WER (%) of using different amount of TTS on dev set

Methods	Hours	WER	
		Libri	Accent
B3	5.5K	5.40	5.54
+ TTS	6.6K	4.97	5.26
+ 3×TTS	8.8K	4.94	5.34

Table 6. WER (%) of final system on dev set

Methods	WER	
	Libri	Accent
Baseline	6.74	6.97
+ B3 & Data Augment	4.97	5.26
++ Adaptation	7.12	4.53
+++ Rescore	6.75	4.00

At last, we compared our submitted system (named 'S2') with other participants' systems on official test set. Our system achieved the second place. As shown in Table 7, 46% relative improvement was obtained on average compared to the baseline system.

Table 7. WER (%) of final system on test set

Ranking	Team	WER
1	Q2	4.06
2	**S2**	**4.47**
3	E2	4.52
4	A2	4.71
5	T2	4.72
6	F	4.95
17	Official	8.29

7. CONCLUSIONS

In this paper, we presented the AISpeech-SJTU system for the AESRC. Data augmentation, advanced ASR model, and enhanced rescore scheme were explored for improving the robustness for accented English ASR. We found that simulating suitable noise data and speed perturb are effective for solving the data-scarcity problem. SpecAugment and TTS simulation can further improve performance. By using data augmentation, 24% to 26% relative improvement were obtained. Accent adaptation obtained further 13% relative improvement, and enhanced rescore scheme gave us a consistent 11% relative improvement. Overall, we achieved WER improvements of absolute 2.92% and 3.82% over the baseline of 6.92% and 8.29% released by the challenge organizers.

8. REFERENCES

[1] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al., "English conversational telephone speech recognition by humans and machines," in *Interspeech*, 2017.

[2] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke, "The microsoft 2017 conversational speech recognition system," in *ICASSP*, 2018.

[3] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015.

[4] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[5] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[6] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual speech recognition with a single end-to-end model," in *ICASSP*, 2018.

[7] Bo Li, Yu Zhang, Tara Sainath, Yonghui Wu, and William Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *ICASSP*, 2019.

[8] Xuankai Chang, Wangyou Zhang, Yanmin Qian, Jonathan Le Roux, and Shinji Watanabe, "Mimo-speech: End-to-end multi-channel multi-speaker speech recognition," *arXiv preprint arXiv:1910.06522*, 2019.

[9] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*, 2017.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[11] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on transformer vs rnn in speech applications," *arXiv preprint arXiv:1909.06317*, 2019.

[12] Mohamed Elfeky, Meysam Bastani, Xavier Velez, Pedro Moreno, and Austin Waters, "Towards acoustic model unification across dialects," in *SLT*, 2016.

[13] Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee, "A study on multilingual acoustic modeling for large vocabulary asr," in *ICASSP*, 2009.

[14] Hui Lin, Li Deng, Jasha Droppo, Dong Yu, and Alex Acero, "Learning methods in multilingual speech recognition," *NIPS*, 2008.

[15] Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz, and Hervé Bourlard, "Multilingual deep neural network based acoustic modeling for rapid language adaptation," in *ICASSP*, 2014.

[16] Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013.

[17] Kanishka Rao and Haşim Sak, "Multi-accent speech recognition with hierarchical grapheme based models," in *ICASSP*, 2017.

[18] Bo Li, Tara N Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Zhifeng Chen, Yanghui Wu, and Kanishka Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *ICASSP*, 2018.

[19] Han Zhu, Li Wang, Pengyuan Zhang, and Yonghong Yan, "Multi-accent adaptation based on gate mechanism," 2019.

[20] Sanghyun Yoo, Inchul Song, and Yoshua Bengio, "A highly adaptive acoustic model for accurate multi-dialect speech recognition," in *ICASSP*, 2019.

[21] Xian Shi, Fan Yu, Yizhou Lu, Qiangze Feng, Daliang Wang, Yanmin Qian, and Lei Xie, "The accented english speech recognition challenge 2020: open datasets, tracks, baselines, results and methods," 2020.

[22] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *ICASSP*, 2013.

[23] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017.

[24] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi.," in *Interspeech*, 2016.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *ASRU*, 2011.

[26] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[27] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," in *NIPS*, 2019.

[28] Jean-Marc Valin and Jan Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP*, 2019.

[29] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[30] Houjun Huang, Xu Xiang, Yexin Yang, Rao Ma, and Yanmin Qian, "Aispeech-sjtu accent identification system for the accented english speech recognition challenge," 2020.

[31] Yanmin Qian, Tian Tan, Hu Hu, and Qi Liu, "Noise robust speech recognition on aurora4 by humans and machines," in *ICASSP*, 2018.

[32] Qi Liu, Yanmin Qian, and Kai Yu, "Future vector enhanced lstm language model for lvcsr," in *ASRU*, 2017.

[33] Taku Kudo and John Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.