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

Extending Pronunciation Dictionary with Automatically Detected Word Mispronunciations to Improve PAII's System for Interspeech 2021 Non-Native Child English Close Track ASR Challenge

Wei Chu, Peng Chang, and Jing Xiao

PAII Inc., USA

{chuwei129, changpeng805, xiaojing661}@pingan.com.cn

Abstract

This paper proposed to automatically detect mispronounced words over the regions that have low Goodness-of-Pronunciation scores through a constrained phone decoder, then add these word mispronunciations into the orthodox lexicon without colliding with existing pronunciations, finally use the expanded lexicon for decoding non-native speech. The constrained phone decoder is compiled by using a phone-level automatically generated one-edit-distance network to eliminate the need of extended recognition networks designed by phonologists. Results and analysis have shown that the pronunciation dictionary extension is effective in improving WER performance for non-native speech recognition. This paper also described the details of PAII's single-pass fusion-free hybrid system for this Interspeech 2021 non-native children English close track ASR challenge, especially showed the effective use of non-speech segments in the training set as noise sources to perform noise augmentation on the training data, and also conducted a comparision of acoustic models with different neural network architectures with analysis. Final WERs of 12.10%/28.25% are obtained compared to a well-optimized baseline with WERs of 13.37%/33.51% on development/evaluation set, respectively.

Index Terms: children speech recognition, pronunciation dictionary, non-native speech recognition, noise augmentation, mispronunciation detection

1. Introduction

Non-native speech recognition, especially for children, has been drawing attentions of research community recently [1] [2] [3] [4] [5] [6]. In fundamental, the task is facing two challenging acoustic issues which are speaker and channel variability [7] [8] [9], and encountering another two challenging linguistic issues which are disfluency and grammatical errors [10] [11] [12]. The speaker variability issue includes speaking style deviation from adult, and accent and pronunciation deviation from canonical lexicon [13]. The channel variability issue refers to the prevailing existence of different levels of additive and convolutional noise. The disfluency issue includes hesitation, repetition, and irregular prosody including abnormal intonation and long pause. The grammatical errors commonly exist among English as a second language learners (ESLs).

To handle speaker variability, a common adopted practice which has shown to be effective is to augment the samples in training set through volume, speed, and vocal tract length perturbations [14], when the amount of training data is not enough. TTS-based data augmentation approach is also emerging [15], however, the requirement to have training data with good quality for TTS model training may limit the usage of the approach when only a relatively small and noisy training set is available.

In a hybrid ASR system, the pronunciation model, or the pronunciation dictionary, or the lexicon, is developed by linguists and used to compose a finite state transducer with a language model to transduce a phonetic sequence into a word sequence. For ESLs, it is common to phonetically pronounce a word wrongly, e.g., "decade" as 'D IY K EY D'. Experience ESL teachers or human ASR transcribers are usually capable of identifying the intended but misspoken words. Chu et al [16] have shown that without the help of phonologists, an automatically generated One-Edit-Distance FSA can be used to detect phone-level mispronunciation errors of substitutions/insertions/deletions over the regions that are low in Goodness-of-Pronounciation (GOP) score, refine the phone alignments according to detected mispronunciations, and retrain an acoustic model with refined alignments to achieve a lower WER. In the perspective of machine learning, this approach made a way for the model to fit the data better by leveraging the expert knowledge that ESLs could have mispronounced words.

In this work, further efforts have been spent to explore the possible means of using the mispronunciation information learned at the training stage at the decoding stage. A non-collision mispronunciation addition (NCMA) algorithm is designed to selectively add word mispronunciations into the orthodox lexicon, i.e., CMU dictionary, and use this expanded lexicon in decoding. It is interesting to find this approach is effective in improving WER performance by only using non-native training data. We also analyze the results of the NCMA algorithm on the challenge data to understand the possible reasons why it works.

The rest of this work illustrates the details of PAII's system for Interspeech 2021 non-native child English close track ASR challenge, which includes a comparison of AMs with different architectures, finetuning with not-in-domain data, handling filler words and half words, training RNNLMs, and effectively using the noise augmentation to alleviate the mismatch between the training and testing.

2. Non-Collision Mispronunciation Addition

In this section, the 4 steps of the proposed non-collision mispronunciation addition algorithm is shown as follows:

• **Step 1**: *Detect all mispronounced words*

This step is the same as the combined Step 1, 2, and 3 in the Section 'The Proposed AM Training Procedure' in [16], except for the following settings: the GMM trained from non-native speech is used for Goodness-of-Pronunciation (GOP) [17] calculation since no native speech is available, i.e., searching for mispronounced words used a non-native GMM. The assump-

http://dx.doi.org/10.21437/Interspeech.2021-2053

tion is that although most words spoken in the training set have accent, they are still correctly pronounced. Therefore, the GMM trained on non-native data could still be capable of capturing those significant pronunciation outliers, a.k.a., mispronunciations.

In this paper, two measures are adopted to further constrain the phone decoder used in [16]. First, it only searches for substitution errors, and does not search for deletion and insertion errors. Second, suppose the original phone is a vowel, only vowels are added as competing phone arcs in the finite state acceptor in [16]. The consonant case is similar to the vowel case.

After attaching the searched one or multiple phone substitutions in each utterance to their corresponding words, we have cases of mispronounced words.

• **Step 2**: *Non-Collision Mispronunciation Addition*

For each mispronunciation of word detected in Step 1, it can be added to the lexicon if the following 3 conditions are met:

Constraint 1: Suppose the mispronounced word is found in a speech segment,

$$\begin{aligned} \text{S-GOP}(y_n) &> 0 \\ \text{S-GOP}(y_n) - \text{S-GOP}(y_o) &> \hat{\mu} \end{aligned} \tag{1}$$

where S-GOP(y_o) and S-GOP(y_n) denote the new and old segmental GOP scores after the mispronunciation search,

$$\hat{\mu} = \frac{1}{N_{det}} \sum_i (\text{S-GOP}_i(y_n) - \text{S-GOP}_i(y_o)) \tag{2}$$

where N_{det} denotes the number of all detected mispronunciation cases. A segmental GOP (S-GOP) score is defined as an criterion for when to stop searching for the mispronunciations:

$$\text{S-GOP}(y) = \frac{\sum_j N_j \cdot \text{GOP}(y_j)}{\sum_j N_j} \tag{3}$$

where j denotes the index of non-silence phones in y, GOP(y) denotes the GOP score of phone y.

Constraint 2: The number of this detected mispronunciation occurrences in the training set is greater than N_a.

Note that one word can have multiple mispronunciations because ESLs have different spoken error patterns. And each mispronunciation of a word can have multiple occurrences because ESLs share the same spoken error pattern.

Constraint 3: This detected mispronunciation does not collide with any phonetic sequence in the orthodox dictionary. Take the mispronounced word 'decade' as 'D EY K EY D' for example, the mispronunciation will not be added into the dictionary since it is the same as the orthodox pronunciation word 'decayed'.

Note that no effort has been spent on learning pronunciation error patterns to add more mispronunciations for the words which are not mispronounced in the training set.

• **Step 3**: *Subsequent AM training*

Update the pronunciation probabilities of the extended dictionary by retrain the GMM. Then, the updated alignments obtained from the new GMM is used for training neural network AM.

• **Step 4**: *Decode with the extended dictionary*

Compile a decoder with the AM, the extended dictionary, and an LM, and then perform recognition.

3. Experiments

3.1. Data

The ETS data used in the children English ASR challenge is shown in Tab. 1. The training data contains 1.8% filler words (um, laughter,...), 1.1% of unknown words which include non-English words and unintelligible words, and 2.2% of half words (refri-, be-, ...), and 0.5% of word and phrase repetitions.

Table 1: *ETLT2021 English challenge ETS data. Dev and eval set are not allowed to be used for training of finetuning model. Dev2 set is an extra dataset released for finetuning. These recordings involve both read as well as spontaneous speech. The test takers come mainly from South America, Korea, Japan and Turkey. There are 174k/12k/12k words in ETS train/dev/eval set, respectively.*

	Duration	# of Recordings	# of Speakers
train	51.4 hr	3078	800
dev	3.3 hr	200	50
dev2	2.5 hr	150	50
eval	3.3 hr	200	50

Table 2: *A perplexity analysis.* ***3/4/5g-ppl****: the perplexity of 3/4/5-gram on a test set. Note that the filler words are excluded.*

training	test	3g-ppl	4g-ppl	5g-ppl
train	dev	9.3	8.8	9.2
train + dev2	dev	9.6	9.1	9.1
train + dev2	eval	10.2	9.6	9.5

No in-depth analysis is conducted on 50 hrs of FBK data which are from Italian children. One can find its description in the documentation released by the organizer.

3.2. Baseline AM and LM

The organizer shared a hybrid baseline system which only uses ETS training data. 13 dimensional MFCC features are used in GMM neural network training. The senone number of the GMM is 3848. CMU pronunciation dictionary is used. For out-of-vocabulary (OOV) words, a Sequitur Grapheme-to-Phoneme model [18] trained on CMU dictionary is used to generate their pronunciations. For AM, it uses 40 dimensional MFCC plus 100 dimensional i-vector as input to a CNN+TDNN model [19] [20]. The parameters of the bottom 6 layers of CNN are shown in Tab. 3, the time-offset and height-offset of CNNs are all set as {-1,0,1}. In upper 9 factorized-TDNN layers, the input/output dimension is set as 1024, the bottleneck dimension is set as 128, the time stride is set as 3 except for the first TDNN layer. The number of senones is 2120. L2 regularization for each layer is set to 0.01, except for the output layer in which it is set to 0.005. The bypass-scale of each TDNN layer is set to 0.66. In TDNN, dropout-proportion of 0 is used. The initial/final learning rates are set to 5e-4 and 5e-5. No online cepstral mean and variance normalization is used. A dropout schedule {0,0@0.20,0.5@0.50,0} of is used. The number of epochs is set to 10. Speed and volume perturbation, and SpecAugment [21] are used.

A 4-gram LM is used. It can be seen from Tab. 2 that the dev and eval set are not significantly mismatched in content.

3.3. A comparision of different AMs

In the following, only the differences to the organizer's baseline are mentioned.

Table 3: *The parameters of CNN layers*

layer	in	out	subsample-out	num-filter
CNN1	40	40	N/A	64
CNN2	40	40	N/A	64
CNN3	40	20	2	128
CNN4	20	20	N/A	128
CNN5	20	10	2	256
CNN6	10	10	N/A	256

Table 4: *A comparision of different AMs. A 4-gram LM is used. AM F1 is used in challenge. F2* added the recognized text into training which is just for analysis.*

	WER (%)
A: 6CNN+9TDNN w/ 4-gram (baseline)	13.37
B: A + probability dictionary (PD)	13.10
C1: B + remove filler words (RFW)	12.83
C2: B + remove half words (RHW)	12.98
D1: C1 + output L2-reg=2.5e-3	12.69
D2: 25TDNN w/ PD, RFW	12.93
D3: 4TDNN+3(2TDNN+1LSTM) w/ PD, RFW	13.58
E: D1 + auto-detected mispronunciations	12.43
F1: E + supervised-RNNLM rescore	12.10
F2*: E + semi-supervised-RNNLM rescore	11.79

The GMM in our AM is also used to learn a dictionary with probability which is used in subsequent training and decoding [22]. The senone number of the GMM is 3104 vs 3848 in baseline.

All the filler words are mapped to silence word, which enables us to train LMs without filler words. All the half words (refri- ...) are kept. And the pronunciations of half words and unseen words are obtained from a Tensor2Tensor G2P trained on CMU dictionary [23].

In GMM training, the standard librispeech [24] training recipe in Kaldi's repository [25] is adopted with changing the parameters in each step, and no data cleaning is performed. 6CNN+9TDNN model is also used, the output L2 regularization is set to 2.5e-3. The senone number is 2096.

A TDNN model with 25 TDNN layers is compared. Each TDNN layer has 1536 dimensions of input and 160 dimensions of bottleneck. The output L2 regularization is set to 2.5e-3.

A TDNN+LSTM model with 4 layers of TDNN in the bottom and 3 TDNN+LSTM blocks on top is compared. Each TDNN layer has 1024 dimensions of input and 128 dimensions of bottleneck. In each TDNN+LSTM block, the first TDNN layer's input is concatenated from the output from the previous layer at time 0 and -3, and the output of the second TDNN in the lower TDNN+LSTM block at time 0. In each LSTM layer, the cell dimension is set to 128, the recurrent are non-recurrent projection dimemsion are both set to 64. The TDNN/LSTM/output L2 regularization factor is set to 2e-3/5e-4/5e-4, respectively. No label delay is used. The initial/final learning rates are set to 1e-3 and 1e-4. The number of epochs is set to 20.

A fusion of lattices generated from D1, D2, and D3 shown in Tab. 4 which have close WERs has been tried, but the WER can not be further lowered.

Table 5: *Train with ETS and FBK data for ETS dev set*

	WER (%)
B: 6CNN+9TDNN w/ PD	13.10
G: B + FBK data	13.69
H: G + finetune on ETS data	14.57

3.4. Mixing in data with other accents and finetuning

To see if the FBK data can be leveraged in the AM training, we tried mixed-style training which uses all the ETS and FBK training data, then finetune on ETS training data. The best finetune results shown in Tab. 5 that we can get is when using initial/final learning rates of 5e-3/5e-4 for 4 epochs.

Due the resource limitation, no more effort is spent on further investigation.

3.5. RNNLM

Besides the 4-gram model given by the organizer, another LM of 2 stacked TDNN+LSTM blocks is used for rescoring [26]. Each block has 1 TDNN and 1 LSTM layer. The input, output, and the cell in LSTM dimensions are set to 128. The recurrent and non-recurrent projection dimensions all set to 32. The initial/final learning rates are set to 1e-4/1e-5, respectively. The number of epochs is set to 400. The training stops around 10 iterations before it overfits. For the ETS dev set, the validation set use 10% of the ETS training text. For the final ETS eval set, we use the ETS eval2 set as the validation set.

3.6. NCMA experiment and analysis

Here we show the results and analysis of the NCMA mentioned in Section 2.

452 words are detected with 2030 mispronunciation cases from the training data. This accounts for 1.5% of the word instances and affects 20% of the lexicon. Even more words instances could have pronunciation issues since the detection model is trained on the non-native speech per se. Therefore, it can be seen here that ESLs have difficulty in phonetically pronouncing words in an orthodox manner. It could be beneficial to explore approaches to address this issue, i.e., expand the orthodox pronunciation lexicon for non-native speaker.

Take the word 'decade' (D EH K EY D) for example, when an ESL mispronounces it as 'D IH K IH D', the mispronunciation does not exist in existing lexicon. During decoding, word 'decay' (D IH K EY) could have a chance to be wrongly recognized as the spoken word, since it has the same closest phonetic edit distance to 'D IH K IH D' as the intended spoken word 'decade'. So attaching the mispronunciation to the word 'decade' could reduce the lexicon-related ambiguity during decoding. One may argue that senone/phone-level AM and word-level LM could exert acoustic and linguistic constraints to prevent a word from being wrongly recognized, which suggests that fine phonetic modeling is not necessary. It can be seen from the data that non-native speakers have non-negligible accent and grammar issues compared to native speakers. Therefore, this expansion in lexicon can be regarded as a mean of exerting extra phonetic constraints when the aforementioned acoustic and linguistic constraints do not necessarily function well for ESLs.

From Tab. 6, it can been seen that the Constraint 3 which avoid the collision with existing pronunciations results in a relatively 2% better WER compared with baseline, on the basis of

Table 6: *A comparison of AMs trained with different settings of non-collision mispronunciaiton addition algorithm. The number of words appeared in training text including half words is 2467. The number of total pronunciations in the orthodox dictionary, i.e., CMU dictionary, is 3565.* ***#AWM****: the number of added word mispronunciations into the orthodox lexicon.* ***Constraint 1,2,3 and*** N_a*: see the illustrations in Step 2 of Section 2. If* $N_a > 0$*, it indicates Constraint 2 is active.*

	WER (%)	#AWM
D1: 6CNN+9TDNN	12.69	-
E1: D1 + Constraint 1	13.18	1649
E2: E1 + N_a=1	12.94	342
E3: E1 + N_a=1 + Constraint 3	**12.43**	272
E4: E1 + N_a=2 + Constraint 3	12.57	93
E5: E1 + N_a=3 + Constraint 3	12.75	38

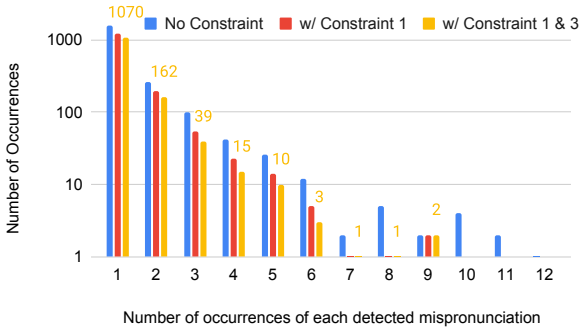

Figure 1: *Occurrences of mispronunciations before and after applying constraints.* ***Constraint 1,3****: see the illustrations in Step 2 of Section 2.*

applying Constraint 1 and 2. One possible explanation is that although a word that has its mispronunciations added could be better recognized, a word could also be wrongly recognized into another word whose mispronunciation collides with the orthodox pronunciation of the first word. Avoiding the pronunciation collision avoids this potential issue. And when demanding more mispronunciation occurrences in Constraint 2, the WER performance drops slightly. The reason could be that the few highly occurred mispronunciations in training set may not appear in test set.

From Fig. 1, it can be seen that more than 85% of the detected mispronunciations only occur once. There are very few mispronunciations which have more than 5 occurrences. Applying non-collision constraint indeed eliminates about 1/4 mispronunciations.

3.7. Noise augmentation and final result submission

When running on eval set, the parameters of the decoder are fixed after being tuned on the dev set. For 4-gram LM, the LM weight is fixed to 10.0, the penalty factor is fixed to 0.0. In lattice-based RNNLM rescoring, the interpolation weight is fixed to 0.40.

The WER of baseline on eval set (33.21%) increased significantly compared to dev set (13.37%). So effort is attempted on noise augmentation [27].

Table 7: *The acoustic model F1 which was referred in Tab. 4 shown here is trained on ETS train and dev2 data. RNNLM A is only trained with text in ETS train set. RNNLM B uses text in ETS train and dev2 set.* ***FG/BGNA****: foreground/background noise augmentation.* ***1/4 or all****: uses 1/4 or all noise recordings in the ETS train data. Numbers with * are the results submitted to the challenge. In all experiments, foreground noise is randomly added at 15, 10, 5, or 0 dB. The background noise is randomly added at 10, 5, or 0 dB.*

	WER (%)		
	4-gram	RNNLM A	RNNLM B
A	33.21	-	-
F1 on train+dev2 data	30.68	30.54*	30.36
G1: F1 + FGNA 1/4	29.79	29.74*	29.06
G2: F1 + FGNA all	28.58	28.63	28.55
G3: F1 + BGNA all	28.84	28.82	28.64
G4: G2 + 3 TDNNs	**28.45**	**28.41**	**28.25**

We did not use any noise data other than the 97 recordings in the ETS training data which contain no or only one unknown words in transcription. Another dev set (dev2) released by the organizer is added into the AM and LM training. Note that the dev set has never been used in AM and LM training.

Foreground and background noise augmentation are tested. No reverberate noise or room impulse response filter is applied, since this is a close track. Foreground noises are added sequentially, according to a specified interval. Background noises are added to the entire recording, and repeated as necessary to cover the full length. No overlapped background or foreground noises are added.

The foreground or background corrupted data are then combined with uncorrupted data to form a new training set. As the speed perturbation augments the data by 3 times. This noise augmentation eventually augments the original training data by 6 times. We could have combined the foreground and background corrupted data with the uncorrupted data, i.e. augment the original training data by 9 times, and further adding multiple overlapping foreground and background noises. However, that was not explored due to resource limitation although that could have further improved the WER performance.

It can been from Tab. 7 that noise augmentation can significantly alleviate the mismatch issue. Foreground noise augmentation is slightly better than background noise augmentation. Since the data is doubled after augmentation, we added 3 more TDNN layers and found it can give a further improvement in WER.

4. Conclusions

The paper proposed a non-collision mispronunciation addition method to expand the non-native pronunciation dictionary and used it in PAII's system for Interspeech 2021 non-native child English close track ASR challenge. PAII's system also effectively leveraged noise augmentation method. The single-pass fusion-free system obtained a relatively 9.5%/14.9% lower WER compared to the baseline system on the development/evaluation set, respectively.

5. References

[1] R. Gretter, M. Matassoni, D. Falavigna, K. Evanini, and C. W. Leong, "Overview of the interspeech tlt2020 shared task on asr for non-native children's speech," *Proc. Interspeech 2020*, pp. 245–249, 2020.

[2] K. M. Knill, L. Wang, Y. Wang, X. Wu, and M. J. Gales, "Non-native children's automatic speech recognition: the interspeech 2020 shared task alta systems," *Proc. of Interspeech*, pp. 255–259, 2020.

[3] M. Shahin, R. Lu, J. Epps, and B. Ahmed, "Unsw system description for the shared task on automatic speech recognition for non-native children's speech," *Proc. Interspeech 2020*, pp. 265–268, 2020.

[4] T.-H. Lo, F.-A. Chao, S.-Y. Weng, and B. Chen, "The ntnu system at the interspeech 2020 non-native children's speech asr challenge," *Proc. Interspeech 2020*, 2020.

[5] H. Kathania, M. Singh, T. Grósz, and M. Kurimo, "Data augmentation using prosody and false starts to recognize non-native children's speech," *Proc. Interspeech 2020*, 2020.

[6] R. Fan, A. Afshan, and A. Alwan, "Bi-apc: Bidirectional autoregressive predictive coding for unsupervised pre-training and its application to children's asr," *Proc. of ICASSP*, 2021.

[7] R. M. Stern, A. Acero, F.-H. Liu, and Y. Ohshima, "Signal processing for robust speech recognition," in *Automatic Speech and Speaker Recognition*. Springer, 1996, pp. 357–384.

[8] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris *et al.*, "Automatic speech recognition and speech variability: A review," *Speech communication*, vol. 49, no. 10-11, pp. 763–786, 2007.

[9] C. Huang, T. Chen, S. Li, E. Chang, and J. Zhou, "Analysis of speaker variability," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[10] S. Schachter, N. Christenfeld, B. Ravina, and F. Bilous, "Speech disfluency and the structure of knowledge." *Journal of personality and social psychology*, vol. 60, no. 3, p. 362, 1991.

[11] J. S. Yaruss, R. M. Newman, and T. Flora, "Language and disfluency in nonstuttering children's conversational speech," *Journal of Fluency Disorders*, vol. 24, no. 3, pp. 185–207, 1999.

[12] M. E. Smith, "Grammatical errors in the speech of pre-school children," *Child Development*, vol. 4, no. 2, pp. 183–190, 1933.

[13] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[14] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, "Speech recognition with augmented synthesized speech," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 996–1002.

[16] W. Chu, Y. Liu, and J. Zhou, "Recognize mispronunciations to improve non-native acoustic modeling through a phone decoder built from one edit distance finite state automaton," *Proc. of Interspeech*, pp. 3062–3066, 2020.

[17] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[18] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech communication*, vol. 50, no. 5, pp. 434–451, 2008.

[19] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[20] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Interspeech*, 2018, pp. 3743–3747.

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[22] G. Chen, H. Xu, M. Wu, D. Povey, and S. Khudanpur, "Pronunciation and silence probability modeling for asr," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[23] L. Kaiser, "Accelerating deep learning research with the tensor2tensor library," *Google Research Blog*, 2017.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[26] H. Xu, K. Li, Y. Wang, J. Wang, S. Kang, X. Chen, D. Povey, and S. Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 6109–6113.

[27] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.