

DATA AUGMENTATION FOR END-TO-END CODE-SWITCHING SPEECH RECOGNITION

Chenpeng Du¹, Hao Li¹, Yizhou Lu¹, Lan Wang², Yanmin Qian¹

¹ MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China

² CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems,
Shenzhen Institutes of Advanced Technology

{duchenpeng, lh575526, luyizhou4, yanminqian}@sjtu.edu.cn, lan.wang@siat.ac.cn

ABSTRACT

Training a code-switching end-to-end automatic speech recognition (ASR) model normally requires a large amount of data, while code-switching data is often limited. In this paper, three novel approaches are proposed for code-switching data augmentation. Specifically, they are audio splicing with the existing code-switching data, and TTS with new code-switching texts generated by word translation or word insertion. Our experiments on 200 hours Mandarin-English code-switching dataset show that all the three proposed approaches yield significant improvements on code-switching ASR individually. Moreover, all the proposed approaches can be combined with recent popular SpecAugment, and an addition gain can be obtained. WER is significantly reduced by relative 24.0% compared to the system without any data augmentation, and still relative 13.0% gain compared to the system with only SpecAugment.

Index Terms: end-to-end speech recognition, code-switching, data augmentation, text-to-speech

1. INTRODUCTION

Code-switching is a common language phenomenon that uses two or more languages within one utterance. For example, Chinese sometimes use English words in Mandarin conversations, which presents challenges to automatic speech recognition (ASR). The work in [1] gives an effective method to detect the language boundary in code-switching utterances and use monolingual ASR systems to recognize corresponding segments. [2, 3] attempt to address the problem by constructing an appropriate pronunciation dictionary and modifying the language model. End-to-end ASR system for code-switching is also explored in recent studies [4, 5, 6, 7] that improves the performance of code-switching end-to-end ASR with language identity information.

Typically, sufficient amount of training data is required for ASR training while code-switching data is difficult to be collected. Several approaches have been proposed to alleviate

the problem. [8, 9] use semi-supervised training to leverage untranscribed data. [10] encourages the distributions of output token embeddings of monolingual languages to be similar and trains code-switching ASR system with only monolingual data. [11] first trains the model using monolingual data and then finetunes on code-switching data.

Another important method to deal with this problem is data augmentation. Traditional data augmentation methods in ASR include noise augmentation [12], speed perturbation [13] and vocal tract length perturbation [14]. Recently, SpecAugment [15] is proposed as a powerful data augmentation approach using time warping and frequency masking. However, all these previous data augmentation approaches above only provide variations on acoustic features. [16] concatenates monolingual utterances from language-dependent corpora to generate code-switching data. However, this type of generated data only contains the code-switching that occurs between two utterances, while another type of code-switching that occurs within a single utterance is what we are interested in here.

In this paper, we propose three data augmentation methods for code-switching ASR training to improve both the acoustic and linguistic diversity of code-switching data. The first method is audio splicing. We splices language-dependent segments from various utterances together according to the alignments between transcripts and acoustic features, and thus generate new utterances for ASR training. Furthermore, although code-switching data is limited, monolingual texts is often easy to be collected, so we propose another two data augmentation methods to firstly generate new code-switching texts by word translation and word insertion on the original monolingual texts, and then produce new speech data using a TTS system. Experimental results show that all the three approaches yield improvements in code-switching ASR. Additionally, ASR still benefits from our approaches when SpecAugment is further combined.

In the rest of this paper, we introduce end-to-end ASR system in Section 2 and the proposed code-switching data aug-

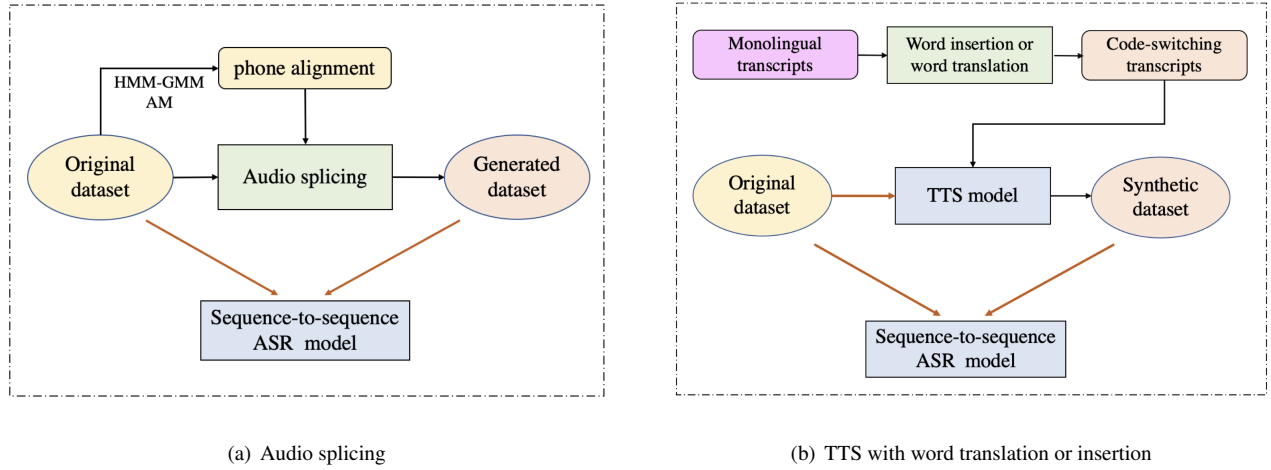


Fig. 1. Process of code-switching data augmentation methods proposed in this work

mentation methods in Section 3. Section 4 gives experiments comparison and results analysis. Finally, we concludes the paper in Section 5.

2. END-TO-END SPEECH RECOGNITION

The end-to-end ASR system is briefly reviewed in this section. The neural network in E2E ASR models the conditional probability $p(\mathbf{y}|\mathbf{x})$, where \mathbf{x} is a sequence of acoustic features and \mathbf{y} is a sequence of labels. In this work, we use Transformer-based sequence-to-sequence (S2S) model [17, 18] jointly trained with CTC [19, 20]. Specifically, the encoder contains a front-end for subsampling and layers of Transformer blocks [21]. The input \mathbf{x} is passed through the encoder and mapped into a high level representation \mathbf{h} .

$$\mathbf{h} = \text{Encoder}(\mathbf{x}) \quad (1)$$

The decoder contains layers of Transformer blocks, a linear projection layer and a Softmax layer. Here, the Transformer blocks apply two attention mechanisms, i.e. self attention and the attention with encoder output \mathbf{h} . The posterior distribution of the next token is predicted with auto-regressive decoding.

$$p_{s2s}(\mathbf{y}_t; \mathbf{y}_1 \dots \mathbf{y}_{t-1}) = \text{Decoder}(\mathbf{h}, \mathbf{y}_1 \dots \mathbf{y}_{t-1}) \quad (2)$$

Encoder output \mathbf{h} is also passed through another linear projection layer and Softmax layer for CTC. We jointly train the S2S ASR system with CTC, and the loss function is formulated as:

$$L_{ASR} = -\alpha \cdot \log p_{s2s}(\mathbf{y}; \mathbf{x}) - (1 - \alpha) \cdot \log p_{ctc}(\mathbf{y}; \mathbf{x}) \quad (3)$$

where α is a hyper-parameter to tune the relative weights between S2S and CTC losses. In the decoding stage, we predict the output tokens with beam search, which combines the scores of S2S and CTC [22].

3. DATA AUGMENTATION FOR CODE-SWITCHING ASR

In this section, three novel data augmentation methods are developed for code-switching ASR, which can increase the data diversity on both acoustic and linguistic aspects.

3.1. Audio splicing

The first method is named audio splicing, and the process of data augmentation with audio splicing is illustrated in Figure 1(a). The original dataset contains code-switching utterances. We first train an HMM-GMM ASR system on the original dataset and obtain alignments on these code-switching data. According to the alignments, we can cut the individual language-dependent segments from the code-switching utterances. Then, we splice the segments randomly from different original utterances with the same speaker, and generate a new code-switching dataset. Both the original and the newly generated code-switching dataset are then pooled together to train the end-to-end ASR.

Figure 2(a) illustrates an example of audio splicing in Mandarin-English. Specifically, for each utterance \mathbf{X} in the original dataset, we randomly select another utterance \mathbf{Y} of the same speaker. Then we replace the English segment of \mathbf{X} with that of \mathbf{Y} , and thus generate a new code-switching utterance \mathbf{X}' , which will be used in the later ASR model training.

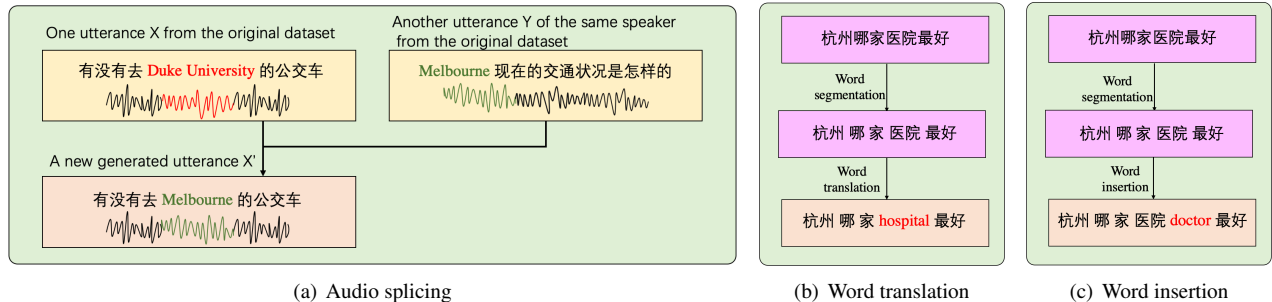


Fig. 2. Augmented examples of the proposed code-switching data augmentation approaches

3.2. Code-switching text with word translation

Code-switching data is limited on both speech and language aspects. Compared to code-switching text, monolingual text is much easier to be collected. To better utilize the monolingual data for code-switching task, we proposed new methods to change the monolingual text to new code-switching text. The first one is named word translation, and Figure 2(b) illustrates an example of generating new code-switching text on Mandarin-English task. Specifically, we first conduct word segmentation and POS tagging on additional Mandarin transcripts. For each transcript, we then randomly select a noun or verb and conduct word-to-word translation with a Mandarin-English dictionary. Consequently, we obtain a new text corpus with code-switching transcripts. These new texts will be then passed to TTS system, which will be described in Section 3.4.

3.3. Code-switching text with word insertion

The second method is named word insertion. We could alternatively use word insertion to transform monolingual texts into code-switching ones. An example of word insertion is illustrated in Figure 2(c). Specifically, we conduct word segmentation on additional Mandarin transcripts. For each utterance, we then randomly select an English word from an English lexicon, and insert the word into the utterance at a random position. Similarly, we obtain a new text corpus with code-switching transcripts, which will be then passed to TTS system.

3.4. Text-to-speech with new code-switching texts

With these new generated code-switching texts, a TTS system is performed to generate the speech, and the complete process of TTS with word translation or insertion is shown in Figure 1(b). Provided with the code-switching transcripts from word translation or word insertion, we synthesize the corresponding speech with a TTS model. Then the synthetic dataset and the original dataset are pooled together for end-to-end ASR training.

Here, the TTS model is based on Fastspeech [23] architecture and is trained with the original code-switching dataset. Duration targets \mathbf{d}^t of standard Fastspeech are extracted from the attentions of an encoder-attention-decoder TTS model. However, we find that the attention alignments are sometimes poor. Therefore, in order to improve the quality and stability of synthetic speech, we train our Fastspeech-based TTS system with duration targets \mathbf{d}^t from the alignments of an HMM-GMM ASR system.

Instead of using vectors from a speaker embedding table to indicate speakers, we sample speaker embeddings from a latent variable \mathbf{z} of a variational auto-encoder (VAE) [24]. This is designed to improve the speaker diversity in synthetic speech [25]. Prior distribution of \mathbf{z} is set to an isotropic standard Gaussian $N(\mathbf{0}, \mathbf{I})$. Our audio encoder architecture is similar to [26], which takes mel spectrogram \mathbf{y} as input and outputs the mean and log variance of the posterior probability distribution of latent variables. \mathbf{z} is sampled from the posterior distribution $q(\mathbf{z}|\mathbf{y})$ in training and from the prior distribution $p(\mathbf{z})$ in generation.

The loss function for TTS training can be formulated as

$$L_{TTS} = L_{MSE}(\mathbf{d}, \mathbf{d}^t) + \mathbf{E}_{q(\mathbf{z}|\mathbf{y})}[\log p(\mathbf{y}; \mathbf{x}, \mathbf{z})] + \lambda \cdot D_{KL}[q(\mathbf{z}|\mathbf{y})||N(\mathbf{0}, \mathbf{I})] \quad (4)$$

where λ is the hyper-parameter to tune the relative weights. The first term denotes the MSE loss between predicted and extracted durations. The second term denotes the reconstruction loss between generated mel spectrograms and corresponding targets. The third term is KL-divergence between $q(\mathbf{z}|\mathbf{y})$ and $N(\mathbf{0}, \mathbf{I})$. We use Griffin-Lim algorithm [27] to reconstruct waveforms from predicted Mel spectrograms.

4. EXPERIMENT AND RESULT

4.1. Dataset

The dataset in ASRU 2019 Code-Switching ASR Challenge is used in our experiments, and it contains 200 hours Mandarin-English code-switching data and 500 hours Mandarin data,

denoted as ASRU-CS and ASRU-MANDARIN here. Our experiments are mainly based on ASRU-CS, in which Mandarin is the host language and English is the guest language. The audio of ASRU-CS is recorded in a relatively quiet indoor environment with 16kHz sampling rate. Additional 20 hours Mandarin-English data is used for the evaluation, and we divide it into Mandarin part CER, English part WER and the total mix error rate (MER) as those in ASRU2019 Challenge.

4.2. Baseline systems

We use ESPnet [28] toolkit to train our end-to-end ASR systems. 3003 Mandarin characters and 1000 English byte pair encodings (BPE) [29] are used as modeling units. We set α in equation 3 to 0.2 and use the same learning rate scheduling as in [30] for all experiments with initial learning rate of 5.0. We use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$.

Table 1. WER (%) on Mandarin-English test dataset of baseline systems

Data Augmentation	WER		
	CN	EN	TOTAL
None	11.15	33.31	13.56
Speed Perturb	10.86	32.77	13.23
Monolingual TTS	11.13	31.61	13.35
SpecAug	9.60	30.18	11.84

The baseline system without data augmentation is trained on ASRU-CS. Moreover, we also implement some usual data augmentations to improve the CS ASR, and three basic methods are compared, including speech perturb, monolingual TTS and SpecAug. The speed perturbation uses speed ratios 0.9, 1.0 and 1.1. In the monolingual TTS method, we synthesize 200k monolingual utterances with 100k Mandarin transcripts from ASRU-MANDARIN and 100k English transcripts from Librispeech. Finally, the recent popular SpecAugment [15] is also performed. The frequency mask parameter F and the time mask parameter T of SpecAugment is set to 30 and 40 respectively, the number of frequency masks m_F and time masks m_T are both set to 2, and the time warp parameter W is set to 5.

As shown in Table 1, only very limited gains are obtained by speed perturbation or monolingual TTS for this code-switching ASR task, and in contrast SpecAugment can yield significant improvement compared with the baseline.

4.3. Evaluations on the proposed data-aug methods

The proposed data augmentation methods for code-switching ASR are evaluated here. The Mel spectrograms of the augmented samples using different methods are shown and compared in Figure 3. For the audio splicing in Figure 3(b), the replaced segment is between the two red lines.

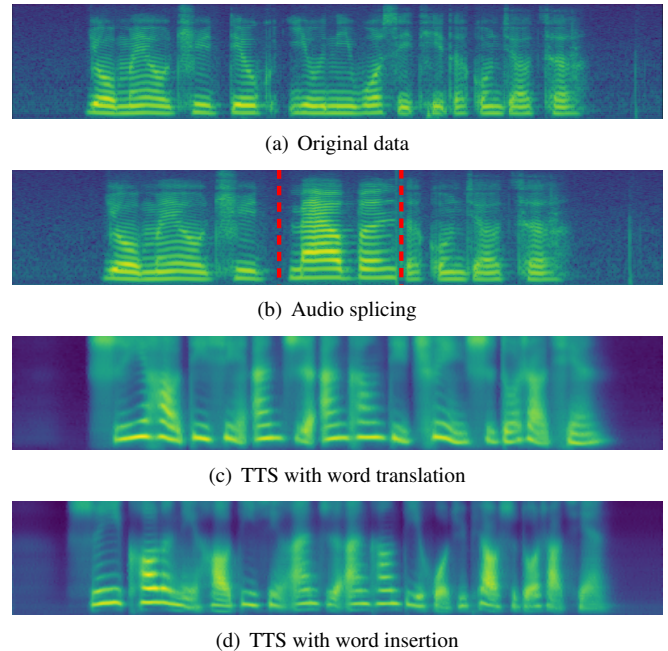


Fig. 3. Mel spectrograms of the augmented samples

4.3.1. Audio splicing

In order to splice audios, we need alignments between transcripts and speech. We train a GMM-HMM system with a Kaldi [31] recipe. It starts from a monophone model with MFCC features, finally obtains a tied triphone model. Alignments of ASRU-CS are then calculated with Viterbi beam-search algorithm. According to the alignments, we replace the English segment of each utterance with that of another randomly selected utterance from the same speaker.

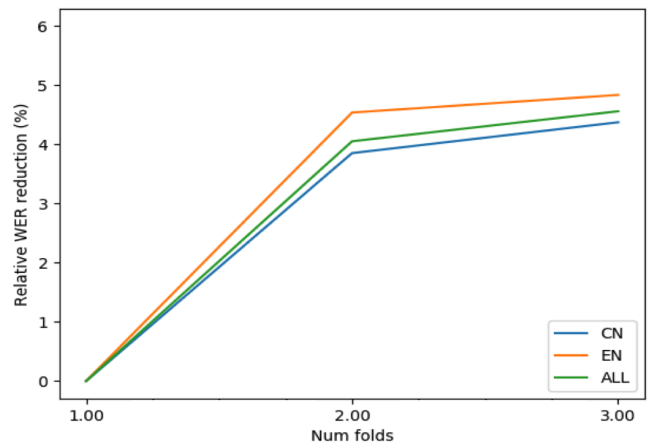


Fig. 4. Relation between WER relative reduction and the amount of augmented data using the proposed audio splicing approach.

Table 2. WER (%) on Mandarin-English test dataset of proposed methods

Data Augmentation	WER		
	CN	EN	TOTAL
None	11.15	33.31	13.56
Audio Splicing	10.75	31.74	13.02
+ SpecAug	9.23	28.81	11.36
Word translation with TTS	10.61	31.25	12.85
+ SpecAug	8.70	28.18	10.81
Word insertion with TTS	10.54	32.11	12.88
+ SpecAug	8.51	28.17	10.65
All three proposed + SpecAug	8.29	26.74	10.29

We generate a new dataset with the same size as ASRU-CS for data augmentation. The augmented data is used to train a CS E2E ASR system, and the results are shown in Table 2. A relative WER reduction of 4.0% was observed compared to the system without data augmentation, and it is better than the speed perturbation and monolingual TTS methods. We also find that further improvement can be achieved by combining audio splicing with SpecAugment, reducing WER by relative 16.2% compared to the system without data augmentation, and also relative 4.1% compared to the system with SpecAugment.

We also investigate how the amount of additional spliced data affects the performance. Shown in Figure 4, 1-fold means the baseline without data augmentation. It shows that we can first get a significant gain with 2-fold data by data augmentation, but only limited improvement when further increasing the augmented data. In all the following experiments, we use 2-fold data for system construction considering both the accuracy and computational cost.

4.3.2. TTS with word translation

We apply word translation to 200k ASRU-MANDARIN transcripts for code-switching data augmentation. Specifically, we use Jieba [32] toolkit for word segmentation and a bilingual dictionary provided by [33] for Mandarin-to-English word translation. The generated code-switching transcripts is then passed to the TTS system trained on ASRU-CS. The synthetic data and original data are then pooled for end-to-end ASR training.

As shown in Table 2, TTS with word translation leads to a relative WER reduction of 5.2% compared to the baseline result, slightly better than the audio splicing. Compared to the monolingual TTS, synthesizing code-switching data from word translation is more helpful for end-to-end code-switching ASR. Additionally, we find that combining with SpecAugment could obtain a further large improvement.

4.3.3. TTS with word insertion

Finally TTS with word insertion is performed, and we use 200k ASRU-MANDARIN transcripts for this experiment. Specifically, we also apply word segmentation with Jieba [32] toolkit and randomly insert an English word from an English lexicon containing words that appear more than 10 times in Librispeech [34] to each Mandarin transcript. Similarly, we pass the generated code-switching transcripts to TTS system and synthesize corresponding speech. Both the synthetic and original dataset are used for end-to-end ASR training.

Table 2 shows that the similar observation and the same conclusion as word translation are obtained by TTS with word insertion, and it can also improve the code-switching E2E ASR significantly.

4.3.4. Combination of all the data-aug methods

The augmented data from different methods are further combined to increase the diversity of the training data, and the system using all kinds of augmented data is finally constructed. The results are shown as the last line of Table 2. It shows that combining different augmented data outperforms all the individual system that uses only a single data augmentation method, indicating that the different data augmentation approaches do not conflict with each other. The final system significantly reduces WER by relative 24.0% compared to the baseline, and still relative 13.0% compared to the system with only SpecAugment.

5. CONCLUSIONS

In this paper, we propose three new data augmentation approaches for code-switching ASR. We use audio splicing to generate new utterances with only existing utterances, and also attempt to leverage additional monolingual texts by transforming the texts into code-switching ones with word translation or word insertion which is then followed by a TTS system. Experimental results show that all the three approaches yield improvements in code-switching ASR. Additionally, ASR still benefits from our approaches when SpecAugment is further combined. Finally all the different data augmentation approaches are all combined, and the code-switching ASR system can be significantly improved.

6. ACKNOWLEDGEMENTS

This work was supported by the China NSFC projects (No. 62071288 and No. U1736202) and the innovation project from CAS (No. 2014DP173025). Experiments have been carried out on the PI supercomputers at Shanghai Jiao Tong University.

7. REFERENCES

- [1] Joyce Y. C. Chan, P. C. Ching, Tan Lee, and Helen M. Meng, "Detection of language boundary in code-switching utterances by bi-phone probabilities," 2004, pp. 293–296.
- [2] K Bhuvanagiri and Sunil Kopparapu, "An approach to mixed language automatic speech recognition," *Oriental COCODA, Kathmandu, Nepal*, 2010.
- [3] Kiran Bhuvanagiri and Sunil Kumar Kopparapu, "Mixed language speech recognition without explicit identification of language," *American Journal of Signal Processing*, vol. 2, pp. 92–97, 2012.
- [4] Zhiping Zeng, Yerbolat Khassanov, Van Tung Pham, Haihua Xu, Eng Siong Chng, and Haizhou Li, "On the end-to-end solution to mandarin-english code-switching speech recognition," *arXiv preprint arXiv:1811.00241*.
- [5] Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie, "Investigating end-to-end speech recognition for mandarin-english code-switching," in *Proc. IEEE ICASSP*, 2019, pp. 6056–6060.
- [6] Ke Li, Jinyu Li, Guoli Ye, Rui Zhao, and Yifan Gong, "Towards code-switching ASR for end-to-end CTC models," in *Proc. IEEE ICASSP*, 2019, pp. 6076–6080.
- [7] Shuai Zhang, Jiangyan Yi, Zhengkun Tian, Jianhua Tao, and Ye Bai, "Rnn-transducer with language bias for end-to-end mandarin-english code-switching speech recognition," *arXiv preprint arXiv:2002.08126*.
- [8] Pengcheng Guo, Haihua Xu, Lei Xie, and Eng Siong Chng, "Study of semi-supervised approaches to improving english-mandarin code-switching speech recognition," in *Proc. ISCA Interspeech*, 2018, pp. 1928–1932.
- [9] Emre Yilmaz, Mitchell McLaren, Henk van den Heuvel, and David A van Leeuwen, "Semi-supervised acoustic model training for speech with code-switching," *Speech Communication*, vol. 105, pp. 12–22, 2018.
- [10] Yerbolat Khassanov, Haihua Xu, Van Tung Pham, Zhiping Zeng, Eng Siong Chng, Chongjia Ni, and Bin Ma, "Constrained output embeddings for end-to-end code-switching speech recognition with only monolingual data," *arXiv preprint arXiv:1904.03802*.
- [11] Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung, "Towards end-to-end automatic code-switching speech recognition," *arXiv preprint arXiv:1810.12620*.
- [12] Michael L. Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE ICASSP*, 2013, pp. 7398–7402.
- [13] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Proc. ISCA Interspeech*, 2015, pp. 3586–3589.
- [14] Navdeep Jaitly and Geoffrey E Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proc. ICML*, 2013, vol. 117.
- [15] 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*.
- [16] Hiroshi Seki, Shinji Watanabe, Takaaki Hori, Jonathan Le Roux, and John R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," in *Proc. IEEE ICASSP*, 2018, pp. 4919–4923.
- [17] Laura Cross Vila, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà, "End-to-end speech translation with the transformer.," in *Proc. ISCA Interspeech*, 2018, pp. 60–63.
- [18] Shigeki Karita, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, Wangyou Zhang, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyang Jiang, Masao Someki, Nelson Enrique Yalta Soplín, and Ryuichi Yamamoto, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE ASRU*, 2019, pp. 449–456.
- [19] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.
- [20] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.
- [22] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1240–1253, 2017.

- [23] 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 *Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.
- [24] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” in *ICLR*, 2014.
- [25] Chenpeng Du and Kai Yu, “Speaker augmentation for low resource speech recognition,” in *Proc. IEEE ICASSP*, 2020.
- [26] Yuxuan Wang, Daisy Stanton, Yu Zhang, R. J. Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” in *Proc. ICML*, 2018, pp. 5167–5176.
- [27] Daniel Griffin and Jae Lim, “Signal estimation from modified short-time fourier transform,” *Proc. IEEE ICASSP*, pp. 236–243, 1984.
- [28] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplín, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., “Espnet: End-to-end speech processing toolkit,” *arXiv preprint arXiv:1804.00015*.
- [29] Taku Kudo and John Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proc. EMNLP*, 2018, pp. 66–71.
- [30] Linhao Dong, Shuang Xu, and Bo Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in *Proc. IEEE ICASSP*, 2018, pp. 5884–5888.
- [31] 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 *Proc. IEEE ASRU*, 2011.
- [32] “Jieba: a chinese word segmentation module implemented in python,” .
- [33] Yo Joong Choe, Kyubyong Park, and Dongwoo Kim, “word2word: A collection of bilingual lexicons for 3,564 language pairs,” in *Proc. International Conference on Language Resources and Evaluation (LREC)*, 2020.
- [34] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. IEEE ICASSP*, 2015, pp. 5206–5210.