ENHANCING CODE-SWITCHING SPEECH RECOGNITION WITH INTERACTIVE LANGUAGE BIASES

Hexin Liu[1], Leibny Paola Garcia[2], Xiangyu Zhang[3], Andy W. H. Khong[1], Sanjeev Khudanpur[2]

[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]CLSP and HLT-COE, Johns Hopkins University, USA
[3]University of New South Wales, Australia

ABSTRACT

Languages usually switch within a multilingual speech signal, especially in a bilingual society. This phenomenon is referred to as code-switching (CS), making automatic speech recognition (ASR) challenging under a multilingual scenario. We propose to improve CS-ASR by biasing the hybrid CTC/attention ASR model with multi-level language information comprising frame- and token-level language posteriors. The interaction between various resolutions of language biases is subsequently explored in this work. We conducted experiments on datasets from the ASRU 2019 code-switching challenge. Compared to the baseline, the proposed interactive language biases (ILB) method achieves higher performance and ablation studies highlight the effects of different language biases and their interactions. In addition, the results presented indicate that language bias implicitly enhances internal language modeling, leading to performance degradation after employing an external language model.

Index Terms— code-switching, automatic speech recognition, interaction, language bias

1. INTRODUCTION

Code-switching (CS) refers to the switching of languages within a spontaneous multilingual recording. Automatic speech recognition (ASR) faces challenges in a code-switching scenario due to the inter- and intra-sentence language varieties compared to its monolingual counterparts [1, 2]. Although conventional ASR approaches can operate on code-switching speech similar to monolingual data, early works identify languages before speech recognition or performs these processes jointly [3, 4, 5]. In contrast, recent CS-ASR techniques tackle language confusion by incorporating language information in modules within the ASR model.

One such approach involves the use of a bi-encoder model that is built on the transformer architecture [6, 7], where modeling of English and Mandarin languages is decoupled by two encoders pre-trained independently on each language. Since the dual-encoder approach has shown to be language discriminative, CS-ASR approaches that adopted similar ar-

chitectures were subsequently proposed [8, 9, 10]. Apart from dual encoders, a language-specific attention mechanism has also been proposed to reduce confusion caused by code-switching contexts [11, 12]. This attention mechanism is employed within the transformer decoders and processes monolingual token embeddings which are separated from code-switching token sequences. In addition, a conditional factorization method factorizes CS-ASR into two monolingual recognitions before composing recognized monolingual segments into a single bilingual sequence which may or may not be code-switched [13].

Although existing approaches mitigate the language confusion for CS-ASR, they are generally stuck in only one module within a CS-ASR model. Since language-aware modules have shown to be effective, it is natural to consider incorporating language information in all modules to further enhance the performance of existing approaches. In addition, these approaches utilize language information either at frame-level (dual-encoder methods) or token-level (transformer-decoder-based approaches) [12, 14]. Since the ASR process aims to align acoustic frames to texts (e.g., characters, words), it is desirable to associate frame- and token-level language information and utilize them jointly for CS-ASR.

Inspired by the success of incorporating language information [14], we propose to enhance language-aware CS-ASR using interactive language biases (ILB). In particular, the proposed method comprises two contributions. Firstly, we bias the connectionist temporal classification (CTC), encoder, and decoder modules jointly within a hybrid CTC/attention CS-ASR model with language posteriors. It is useful to note that the language information transits from frames to tokens (i.e., from the encoder to CTC and decoder) intrinsically. As opposed to existing models, our method utilizes the interaction between frame- and token-level language information resulting in an integrated and language-discriminative model. In addition, the proposed architecture allows the research community to gain insight into how language biases influence a CS-ASR model beyond improving performance. Experiment results suggest that the CS-ASR model is capable of developing a robust internal language model after learning from

language information.

2. METHODOLOGY

2.1. Language posterior bias

The language posterior bias approach [14] has been developed on the hybrid CTC/attention ASR model, which comprises an encoder module, a decoder module, and a CTC module [15, 16]. These encoder and decoder modules consist of conformer encoder layers and transformer decoder layers [2, 6], respectively.

Consider a speech signal with its acoustic features $\mathbf{X} = (\mathbf{x}_t \in \mathbb{R}^F | t = 1, \ldots, T)$ and token sequence $W = (w_n \in \mathcal{V} | n = 1, \ldots, N)$, where $\mathcal{V}$ is a vocabulary of size V, T and N are the lengths of acoustic features and token sequence, respectively. The encoder generates output $\mathbf{H} = (\mathbf{h}_t \in \mathbb{R}^D | t = 1, \ldots, T_1)$ from $\mathbf{X}$, which are subsequently fed into the decoder and CTC modules. On the other hand, tokens are first embedded into $\mathbf{W} = (\mathbf{w}_n \in \mathbb{R}^D | n = 1, \ldots, N)$ before being fed into the decoder module along with $\mathbf{H}$. The ASR model is optimized with a language diarization (LD) decoder jointly, where the LD decoder computes a V^{ld}-dimensional token-level language posterior bias $\mathbf{p}(l_{n-1} | w_{1:n-1}, \mathbf{X})$. Here, V^{ld} is the language vocabulary size and l_{n-1} is the language index for the n-th token. The token embedding $\mathbf{w}_{n-1}$ is then biased by its language posterior. The ASR decoder output is subsequently computed via

$$\mathbf{H} = \text{Encoder}\left(\mathbf{X}\right), \tag{1}$$

$$\mathbf{w}'_{n-1} = \text{Concat}\left(\mathbf{w}_{n-1}, \mathbf{p}\left(l_{n-1} | w_{1:n-1}, \mathbf{X}\right)\right), \tag{2}$$

$$p\left(w_n | w_{1:n-1}, \mathbf{X}\right) = \text{Decoder}\left(\mathbf{w}'_{1:n-1}, \mathbf{H}\right), \tag{3}$$
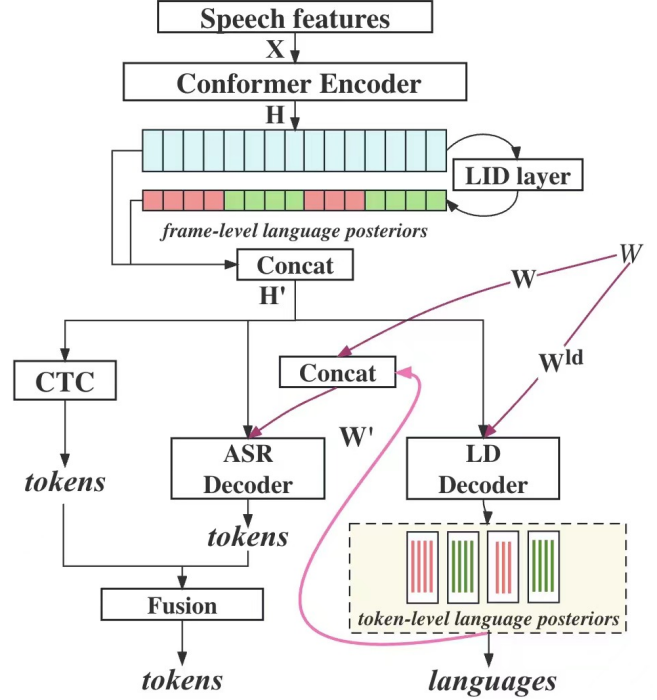
where $\text{Concat}(\cdot)$ denotes the concatenation operation. The matrix $\mathbf{W}' = (\mathbf{w}'_n \in \mathbb{R}^{D+V^{\text{ld}}} | n = 1, \ldots, N)$ consists of input token embeddings of the ASR decoder which are subsequently projected back to D dimensions by a linear layer. The model is optimized via

$$\mathcal{L}_{\text{joint}} = \alpha \mathcal{L}_{\text{ctc}} + (1 - \alpha) \mathcal{L}_{\text{att}} + \beta \mathcal{L}_{\text{ld}} \tag{4}$$

and that the decoding process is similar to (3) but with input token embeddings $\mathbf{W}$ of the ASR decoder being replaced by $\mathbf{W}'$. The integrated model is optimized via a multi-task objective function. Here, β is a multi-task learning parameter and $\mathcal{L}_{\text{ld}}$ is a label-smoothed cross-entropy loss between the predicted and ground-truth language labels for the LD decoder.

2.2. Interactive language biases

We propose to extend the language posterior bias method to frame-level language information. We note that frame-level language identification (LID) is undesirable since the LID performance generally degrades with shorter speech [17, 18, 19]. However, since acoustic frames are tightly associated

Fig. 1. The hybrid CTC/attention model with interactive language biases.

with tokens in ASR, frame-level language identification over $\mathbf{H}$ may benefit from token-level language diarization in (2). The frame-level language posteriors, therefore, enhance the hidden output $\mathbf{H}$ to achieve high language discrimination before being fed into the ASR and LD decoders. Consequently, frame- and token-level language posteriors interact and jointly improve the model performance in CS-ASR.

With reference to Fig. 1, the frame-level language bias is achieved through a LID layer before being concatenated with the hidden output. In particular, the biased hidden output $\mathbf{H}'$ is computed via

$$\mathbf{h}'_t = \text{Concat}\left(\mathbf{h}_t, \mathbf{p}\left(l_t | \mathbf{h_t}\right)\right), \tag{5}$$

which subsequently replaces $\mathbf{H}$ in (3) to facilitate the interaction between language information among frames and tokens. In addition, $\mathbf{H}'$ is also employed to develop a language-aware CTC module. The ASR decoder output is next achieved via

$$p\left(w_n | w_{1:n-1}, \mathbf{X}\right) = \text{Decoder}\left(\mathbf{w}'_{1:n-1}, \mathbf{H}'\right). \tag{6}$$

During training, the frame-level LID is optimized in an unsupervised manner similar to [7] (i.e., frame-level language annotations are not provided during training). However, frames in $\mathbf{H}'$ are trained to be aligned with their corresponding token-level language labels within the language diarization decoder. An assumption made here is that an accurate frame-to-token alignment enriches the unsupervised LID process with supervised information through backpropagation. Optimization of the model is achieved similarly to that of (4).

During inference, the frame- and token-level language posterior are computed before biasing the hidden output and the ASR decoding, respectively. The decoding process is similar to that presented in [15], which is defined to maximize the linear combination of the logarithmic CTC and attention objectives, i.e.,

$$\widehat{W} = \underset{W}{\mathrm{argmax}} \left\{ \alpha \log p_{\mathrm{ctc}}\left(W|\mathbf{X}\right) + (1-\alpha) \log p_{\mathrm{att}}\left(W|\mathbf{X}\right) \right\}. \quad (7)$$

3. DATASET, EXPERIMENTS, AND RESULTS

3.1. Dataset and experiment setup

All experiments are conducted on datasets from the ASRU 2019 Mandarin-English code-switching speech recognition challenge [20]. This challenge comprises four datasets, including a 500-hour Mandarin-only training set, a 200-hour intra-sentence English-Mandarin code-switching training set, a 40-hour intra-sentence English-Mandarin code-switching development set, and a 20-hour intra-sentence English-Mandarin code-switching test set. We employed ESPnet [1] to train all models on the 200-hour CS training set, which are validated on the development set and evaluated on the test set [21].

SpecAugment is applied to augment the training data [22]. Words are transformed into a total of $V = 6,923$ tokens that include 3,000 English byte-pair encoding (BPE) tokens, 3,920 Mandarin characters, and three special tokens for *unk*, *blank*, and *sos/eos*. All tokens are transformed to language labels building $\mathcal{V}^{\mathrm{ld}}$, which comprises e for English BPEs, m for Mandarin characters, and *sos/eos*. Language labels in $\mathcal{V}^{\mathrm{ld}}$ are used as LD outputs. We extracted $F = 83$ dimensional features comprising 80-dimensional log-fbanks and 3-dimensional pitch for each speech sample before applying global mean and variance normalization.

We chose a hybrid CTC/Attention ASR model comprising twelve conformer encoder layers and six transformer decoder layers as the baseline model [2, 6, 23]. In addition, we adopted the multi-task learning model and the language posterior bias approach as our benchmark [14]. All self-attention encoder and decoder layers have four attention heads with input and output dimensions being $D = 256$, and the inner layer of the position-wise feed-forward network is of 2048 dimensions. During training, we set parameters $\alpha = 0.3$ and $\beta = 0.8$ in (4), while a label smoothing factor of 0.1 is used for all cross-entropy losses. The ten best models during validation are averaged for inference. All models are trained on two GeForce RTX 3090 GPUs, where the baseline was trained for seventy epochs, while other models were trained for eighty epochs due to their higher number of parameters.

During inference, we set parameters $\alpha = 0.4$ in (7). Ten-best beam search is used before selecting the best hypothesis. The language model (LM) used in this paper is a sixteen-layer transformer model with each attention layer comprising eight

---

[1] Source code: https://github.com/Lhx94As/interactive_language_biases

Table 1. Performance comparison of models utilizing various-level language information without using external language model by employing MER (%)

Index	Method		MER
1.0	Baseline	Hybrid CTC/attention	12.8
1.1	Multi-task	Multi-task with LD	12.4
1.2	Token-level	Decoder LPB	12.4
1.3	Frame-level	Encoder LPB	12.8
1.4	Interactive	Encoder + CTC LPB	12.4
1.5		Encoder + Decoder LPB	12.1
1.6		Encoder+ Decoder + CTC LPB	**11.8**

heads. The proposed systems are evaluated by employing mix error rate (MER) comprising word error rate (WER) for English and character error rate (CER) for Mandarin.

3.2. Baseline and single language-biased models

The results of the benchmark models are shown in Table 1 as systems 1.0, 1.1, and 1.2. Compared to the vanilla hybrid CTC-attention CS-ASR model, incorporating an auxiliary language diarization task and employing token-level LPB proposed in [14] lead to higher performance. These indicate that incorporating language information benefits the CS-ASR process, which is consistent with the observation presented in [14]. However, the token-level LPB approach shows no performance improvement over the multi-task optimization since Mandarin is the primary language in this dataset and languages do not switch frequently.

The above data characteristics also result in performance degradation for model configuration 1.3 when frame-level LID is not sufficiently accurate. To prevent the CTC outputs from interacting with the unsupervised frame-level LID, the input of CTC is set to $\mathbf{H}$ while the input of the ASR decoder is set to $\mathbf{H}'$ in model configuration 1.3. As mentioned in Section 2.2, frame-level LID is generally less accurate than token-level LID. Those incorrect language posteriors may increase language confusion when being transmitted into the ASR decoder module.

3.3. Results of models with interactive language biases

We next investigate how the interaction between frame- and token-level language information improves the model performance using systems 1.4, 1.5, and 1.6. In model configuration 1.4, as opposed to model configuration 1.3, the input of CTC is set to $\mathbf{H}'$ so as to bias the CTC module with language information. The CTC performs frame-level classification before computing the optimal alignment, where the language biases are infused with acoustic features and combined intrinsically when generating tokens. Therefore, the performance improvement shown in Table 1 when comparing model configuration 1.4 with 1.3 indicates that language-biased frames can also perform better than vanilla frames. This underpins

Table 2. Performance comparison of models using external language model during inference by employing MER (%), where "Reduction" denotes the absolute MER reduction compared to their no-LM counterparts

Index	Method	MER	Reduction
2.0	Hybrid CTC/attention	12.6	0.2
2.1	Multi-task with LD	12.5	-0.1
2.2	Decoder LPB	12.6	-0.2
2.4	Encoder LPB	12.9	-0.1
2.5	Encoder + CTC LPB	12.5	-0.1
2.6	Encoder + Decoder LPB	12.3	-0.2
2.7	Encoder + Decoder + CTC LPB	**11.9**	-0.1

the efficacy of the frame-level language bias when being used for CTC.
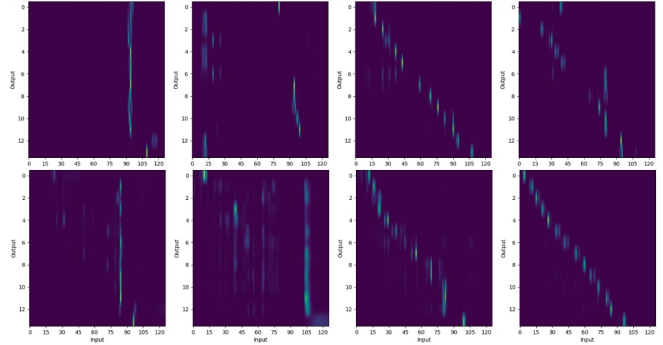
Model configuration 1.5 employs frame- and token-level language biases jointly but excludes the CTC module from being biased. Model configuration 1.5 shows significantly higher performance than single-language-biased models 1.2 and 1.3. This implies that the use of token-level language bias compensates for the inaccurate frame-level LID especially when model configuration 1.3 degrades the performance of model 1.1, which demonstrates that the interactive language biases are effective for CS-ASR.

Model configuration 1.6 further combines two language biases with the CTC module and achieves the highest performance among all model considerations, with a 7.8% relative improvement compared to the baseline model. It is not surprising that this configuration achieves higher performance than model configurations 1.4 and 1.5 since biasing CTC with language information improves the performance over the encoder LPB approach. In addition, the above implies that enriching all modules within a CS-ASR model with language information obtains higher gain compared to a single module, which is consistent with our proposition in Section 1.

3.4. Results of external language modeling

Since the end-to-end ASR approaches internally perform language modeling, we explore whether the internal LM is stronger than the external LM when being trained on the same corpora.

We present the results with respect to external language models in Table 2. The vanilla hybrid CTC/attention model shows higher performance after being integrated with external LM during inference. However, the results show that all language-aware CS-ASR models suffer from performance degradation compared to the baseline model. This implies that the CS-ASR model biased by language information could develop a more robust internal language model compared to an external model trained on the same text data. Since training an external language model can be time-consuming, robust internal language modeling can thus be concluded as an advantage of the proposed interactive language biases approach.

Text: 我 对 这 人 没 兴 趣 但 不 想 显 得 jerk
Language labels: m m m m m m m m m m m m e

Fig. 2. Comparison between attention matrices with respect to the frame-to-token alignment within language diarization decoder after employing token-level LPB (above) and interactive language biases (below).

4. DISCUSSION

Although the language diarization decoder adopted in this work does not generate timestamps for language changes, the frame-to-language alignment can be obtained from the attention matrices within the LD decoder as shown in Fig. 2.

The token-level LPB and interactive language biases (model configurations 1.2 and 1.6) are selected to compare single language bias with interactive language biases. As illustrated in Fig. 2, the attention mechanism identifies language changes in the first and second heads, and captures sequential information in the third and fourth heads. Compared to the token-level LPB, the attention matrices of our proposed interactive language biases approach exhibit clearer vertical language boundaries and smoother diagonal frame-to-token alignment. This indicates that the proposed approach improves not only ASR but also language diarization performance being consistent with our assumption in Section 2.2.

5. CONCLUSION

We proposed an interactive language biases approach to improve CS-ASR through the interaction between frame- and token-level language information. Experiment results presented indicate that the proposed approach outperforms the benchmark in CS-ASR. We next visualized the attention matrices within the LD decoder. The proposed interactive language biases achieve higher language diarization performance compared with single token-level language bias, highlighting the efficacy of the proposed interactive language biases approach. In addition, the results show that a language-aware CS-ASR model can develop a robust internal LM, resulting in performance degradation when using an external language model during inference.

6. REFERENCES

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011.

[2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020.

[3] N. T. Vu, D.-C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, "A first speech recognition system for mandarin-english code-switch conversational speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4889–4892.

[4] Z. Zeng, Y. Khassanov, V. T. Pham, H. Xu, E. S. Chng, and H. Li, "On the end-to-end solution to Mandarin-English code-switching speech recognition," in *Proc. Interspeech*, 2019, pp. 2165–2169.

[5] H. Liu, L. P. G. Perera, X. Zhang, J. Dauwels, A. W. H. Khong, S. Khudanpur, and S. J. Styles, "End-to-end language diarization for bilingual code-switching speech," in *Proc. Interspeech*, 2021, pp. 1489–1493.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[7] Y. Lu, M. Huang, H. Li, J. Guo, and Y. Qian, "Bi-encoder transformer network for Mandarin-English code-switching speech recognition using mixture of experts," in *Proc. Interspeech*, 2020, pp. 4766–4770.

[8] M. S. Mary N J, V. M. Shetty, and S. Umesh, "Investigation of methods to improve the recognition performance of Tamil-English code-switched data in transformer framework," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7889–7893.

[9] S. Dalmia, Y. Liu, S. Ronanki, and K. Kirchhoff, "Transformer-transducers for code-switched speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5859–5863.

[10] T. Song, Q. Xu, M. Ge, L. Wang, H. Shi, Y. Lv, Y. Lin, and J. Dang, "Language-specific characteristic assistance for code-switching speech recognition," in *Proc. Interspeech*, 2022, pp. 3924–3928.

[11] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5884–5888.

[12] S. Zhang, J. Yi, Z. Tian, J. Tao, Y. T. Yeung, and L. Deng, "Reducing multilingual context confusion for end-to-end code-switching automatic speech recognition," in *Proc. Interspeech*, 2022, pp. 3894–3898.

[13] B. Yan, C. Zhang, M. Yu, S.-X. Zhang, S. Dalmia, D. Berrebbi, C. Weng, S. Watanabe, and D. Yu, "Joint modeling of code-switched and monolingual asr via conditional factorization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6412–6416.

[14] H. Liu, H. Xu, L. P. Garcia, A. W. H. Khong, Y. He, and S. Khudanpur, "Reducing language confusion for code-switching speech recognition with token-level language diarization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[15] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1240–1253, 2017.

[16] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[17] H. Liu, L. P. G. Perera, A. W. H. Khong, S. J. Styles, and S. Khudanpur, "PHO-LID: A unified model incorporating acoustic-phonetic and phonotactic information for language identification," in *Proc. Interspeech*, 2022, pp. 2233–2237.

[18] S. O. Sadjadi, T. Kheyrkhah, C. S. Greenberg, E. Singer, D. A. Reynolds, L. P. Mason, and J. Hernandez-Cordero, "Performance analysis of the 2017 NIST language recognition evaluation," in *Proc. Interspeech*, 2018, pp. 1798–1802.

[19] L.-H. Tseng, Y.-K. Fu, H.-J. Chang, and H.-y. Lee, "Mandarin-English code-switching speech recognition with self-supervised speech representation models," *arXiv preprint arXiv:2110.03504*, 2021.

[20] X. Shi, Q. Feng, and L. Xie, "The ASRU 2019 Mandarin-English code-switching speech recognition challenge: Open datasets, tracks, methods and results," *arXiv preprint arXiv:2007.05916*, 2020.

[21] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[22] 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," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[23] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. Interspeech*, 2019, pp. 1408–1412.