Constrained output embeddings for end-to-end code-switching speech recognition with only monolingual data

Author Name[1], *Co-author Name*[2], *Co-author Name*[2], *Co-author Name*[2], *Co-author Name*[2], *Co-author Name*[2], *Co-author Name*[2]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Temasek Laboratories, Nanyang Technological University, Singapore
[3]Co-author Affiliation

author@university.edu, coauthor@company.com

Abstract

In spite of recent progress in code-switching speech recognition, the lack of code-switch data still remains a major challenge. Different from the previous works which highly rely on the availability of code-switch data, we aim to build an end-to-end code-switching automatic speech recognition (E2E-CS-ASR) system using only monolingual data. While greatly mitigating the code-switch data scarcity problem, the E2E-CS-ASR will fail to learn language switch-points due to the absence of cross-lingual signal. Indeed, we investigate the E2E-CS-ASR model and found that the embedding feature representations of output tokens of code-switching languages are concentrated in disjoint clusters. We hypothesize that a gap between these clusters hinders the E2E-CS-ASR from switching between languages, leading to sub-optimal performance. To address this issue, we propose embedding feature matching approaches based on Jensen-Shannon divergence and cosine distance constraints. The proposed constraints will act as a cross-lingual signal enforcing the disjoint clusters to be similar. The experiment results performed on Mandarin-English code-switching language pair from the SEAME corpus demonstrate high effectiveness of the proposed method.

Index Terms: code-mixing, code-switching, feature matching, speech recognition, end-to-end

1. Introduction

The code-swtiching (CS) is a practice of using more than one language within single discourse which poses a serious problem to many speech and language processing applications. Recently, the end-to-end code-switching automatic speech recognition (E2E-CS-ASR) gained increasing interest where impressive improvements have been reported [1, 2, 3]. The improvements are mainly achieved for CS languages where sufficient amount of transcribed CS data is available such as Mandarin-English [4]. Unfortunately, for vast majority of other CS languages the CS data remains too small or non-existent.

Several attempts have been made to alleviate the CS data scarcity problem. Notably, [5, 6] used semi-supervised approaches to utilize untransrcibed CS speech data. On the other hand, [2, 3] employed transfer learning techniques where additional monolingual corpora are either used for pre-training or joint-training. On the account of increased training data, these approaches achieved significant improvements. However, all these approaches rely on the cross-lingual signal imposed by some CS data and other linguistic resources such as word aligned parallel corpus.

In this work, we attempt to build an E2E-CS-ASR using only monolingual data without any form of cross-lingual resource. The only assumption we make is an availability of monolingual speech corpus for each of the CS languages. This set up is important and common to many low-resource CS languages, but has not received much research attention. Besides, it will serve as a strong baseline performance that any system trained on CS data should reach.

One of the major drawbacks of using only monolingual data to train E2E-CS-ASR is the absence of cross-lingual signal. As a result, the E2E-CS-ASR will fail to learn language switch-points. Indeed, we examined the shared embedding feature space learned by E2E-CS-ASR and observe that output token representations of CS languages are concentrated in disjoint clusters (see Figure 3a). We hypothesize that a gap between these clusters hinders the E2E-CS-ASR from switching between languages, leading to sub-optimal performance.

To address this problem, we propose to bridge the gap between clusters using the feature matching approaches [7] based on Jensen-Shannon divergence and cosine distance constraints. These constraints are incorporated into the objective function of E2E-CS-ASR where they will act as a cross-lingual signal source forcing the embedding feature representations of CS languages to be similar. In addition, the constraint will act as a regularization term to prevent overfitting. Our method is inspired by [8, 9] where intermediate feature representations of text and speech are forced to be close to each other. We evaluate our method on Mandarin-English CS language pair from the SEAME [4] corpus where we removed all CS utterances from the training data. Experiment results show that our method significantly improves the recognition accuracy of E2E-CS-ASR built using only monolingual data.

The rest of the paper is organized as follows. In section 2, we review related works addressing the CS data scarcity problem. In section 3, we briefly describe the baseline E2E-CS-ASR model. In section 4, proposed embedding feature matching approaches are presented. Section 5 describes the experiment setup and discusses obtained results. Lastly, section 6 concludes the paper.

2. Related works

An early approach to build CS-ASR using only monolingual data are so-called "multi-pass" systems [10]. The multi-pass systems are based on traditional ASR and consist of thee main steps. First, the CS utterances are split into monolingual speech segments using the language boundary detection system. Next, obtained segments are labeled into specific languages using the language identification system. Lastly, labeled segments are de-

coded using corresponding monolingual ASR system. However, this approach is prone to error-propagation between different steps, not to mention that language boundary detection and language identification are difficult tasks.

More recently, the semi-supervised approaches have been explored to circumvent the CS data scarcity problem. For instance, [5] used their best CS-ASR to transcribe a raw CS speech, the transcribed speech is then used to re-train the CS-ASR. In the similar manner, [6] employed their best CS-ASR to re-transcribe the poorly transcribed portion of the the training set and then re-build the system. The semi-supervised approaches are promising direction for increasing CS data, however, they depends on the availability of transcribed CS data and other systems such as language identification.

In the context of end-to-end ASR models, the transfer learning techniques are widely used to alleviate the CS data scarcity issue. For example, [2] used monolingual corpora to pretrain the model followed by the fine-tuning with CS data. On the other hand, [3] used both CM and monolingual data for joint-training followed by the standard fine-tuning with the CS only data. While being effective, the transfer learning based techniques highly rely on the CS data.

Generating synthesized CS data using only monolingual corpora has been also explored in [11, 12, 13, 14], however, they only address the textual data scarcity problem.

3. Baseline E2E-CS-ASR

Figure 1 illustrates the baseline E2E-CS-ASR model based on hybrid CTC/Attention architecture [15] which incorporates the advantages of both attention-based encoder-decoder model [16] and Connectionist Temporal Classification (CTC) model [17]. Specifically, the attention-based decoder and CTC modules share a common encoder network and are jointly trained.

Encoder. The shared encoder network takes a sequence of T-length speech features $\boldsymbol{x} = (x_1, \ldots, x_T)$ and transforms them into L-length high level representations $\boldsymbol{h} = (h_1, \ldots, h_L)$ where $L < T$. The encoder is modeled as a deep convolutional neural network (CNN) based on the VGG network [18] followed by several bidirectional long short-term memory (BLSTM) layers.

$$\boldsymbol{h} = \text{BLSTM}(\text{CNN}(\boldsymbol{x})) \tag{1}$$

CTC module. The CTC sits on top of encoder and computes the posterior distribution $P_{\text{CTC}}(\boldsymbol{y}|\boldsymbol{x})$ of N-length output characters sequence $\boldsymbol{y} = (y_1, \ldots, y_N)$. To compute $P_{\text{CTC}}(\boldsymbol{y}|\boldsymbol{x})$, CTC introduces framewise letter sequence with an additional "blank" symbol $\boldsymbol{z} = (z_1, \ldots, z_T)$ and factorizes $P_{\text{CTC}}(\boldsymbol{y}|\boldsymbol{x})$ using conditional independence assumption as follows:

$$P_{\text{CTC}}(\boldsymbol{y}|\boldsymbol{x}) \approx \sum_{\boldsymbol{z}} \prod_{t} P(z_t|z_{t-1}, \boldsymbol{y}) P(z_t|\boldsymbol{x}) P(\boldsymbol{y}) \tag{2}$$

where three distribution components are: state transition probability $P(z_t|z_{t-1}, \boldsymbol{y})$, framewise posterior distribution $P(z_t|\boldsymbol{x})$ and character-level language model $P(\boldsymbol{y})$. The state transition probability enforces the monotonic alignment between speech and character sequences, and is obtained using the set of predefined rules (see Eq. (21) in [19]), whereas framewise posterior distribution is modeled as follows:

$$P(z_t|\boldsymbol{x}) = \text{Sofmax}(\text{Lin}(\boldsymbol{h})) \tag{3}$$

where $\text{Lin}(\cdot)$ is a linear projection layer with learnable matrix and bias parameters. Lastly, the Eq. (2) is efficiently computed

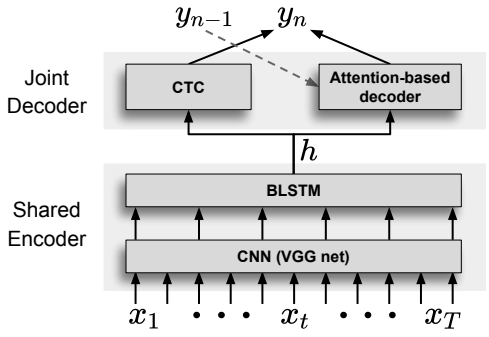


Figure 1: *Hybrid CTC/Attention end-to-end ASR architecture.*

using dynamic programming. The CTC loss is defined as a negative log-likelihood of the ground truth character sequences $\boldsymbol{y}^*$:

$$\mathcal{L}_{\text{CTC}} = -\log P_{\text{CTC}}(\boldsymbol{y}^*|\boldsymbol{x}) \tag{4}$$

Attention-based decoder module. Unlike the CTC module, the attention-based decoder directly computes the $P_{\text{ATT}}(\boldsymbol{y}|\boldsymbol{x})$ based on the chain rule:

$$P_{\text{ATT}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{n} P(y_n|y_{<n}, \boldsymbol{x}) \tag{5}$$

$$P(y_n|y_{<n}, \boldsymbol{x}) = \text{Softmax}(\text{Lin}(s_n)) \tag{6}$$

$$s_n = \text{LSTM}(s_{n-1}, \text{Lin}(y_{n-1}), c_n) \tag{7}$$

$$c_n = \text{Attention}(s_{n-1}, c_{n-1}, \boldsymbol{h}) \tag{8}$$

s_n is a hidden state produced by unidirectional long short-term memory (LSTM) which accepts previous hidden state s_{n-1}, previously emitted character y_{n-1} and context vector c_n. The context vector c_n encapsulates the information in the input speech features required to generate the next character and is produced by Attention$(\cdot)$ module. The loss function of attention-based decoder module is computed using Eq. (5) as:

$$\mathcal{L}_{\text{ATT}} = -\log P_{\text{ATT}}(\boldsymbol{y}^*|\boldsymbol{x}) \tag{9}$$

Finally, the CTC and attention-based decoder modules are jointly trained within multi-task learning (MTL) framework as follows:

$$\mathcal{L}_{\text{MTL}} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda)\mathcal{L}_{\text{ATT}} \tag{10}$$

Our proposed method will augment the Eq. (10) and mainly impact the learnable matrix parameter of linear projection layer in Eq. (6) as will be explained in the following section.

4. Embedding feature matching

In this work, we aim to build E2E-CS-ASR using only monolingual data. This setup is essential for vast majority of CS languages for which CS data is non-existent. However, an E2E-CS-ASR model trained on monolingual data will fail to learn language switch-points between and within utterances, due to the absence of cross-lingual signal. We investigate the E2E-CS-ASR and found that the shared feature spaces of output tokens of CS languages in CTC module (Eq. (3)) and input embedding matrix (Eq. (7)), modeled by linear projection layers Lin$(\cdot)$, to be highly similar. However, the output token representations learned by output embedding matrix (Eq. (6)) of attention-based decoder module are concentrated in two disjoint clusters (see Figure 3a). We hypothesize that a gap between clusters restricts the E2E-CS-ASR model from switching between languages.

To bridge the gap between these clusters, we propose to employ feature matching approaches [7] based on Jensen-Shannon divergence (JSD) and cosine distance (CD) constraints. These constrains will typically act as a cross-lingual signal source which will force output token embedding feature representations of CS languages to be similar. Specifically, JSD will enforce the learned output token embedding feature representations of CS languages to posses similar distribution. On the other hand, CD will enforce the centroids of two clusters to be close to each other.

Jensen-Shannon divergence. First, we assume that learned output token representations of CS language pair L_1 and L_2 follow a z-dimensional multivariate Gaussian distribution:

$$L_1 \sim Normal(\mu_1, \Sigma_1) \tag{11}$$

$$L_2 \sim Normal(\mu_2, \Sigma_2) \tag{12}$$

The JSD between these distributions is then computed as:

$$\begin{aligned}\mathcal{L}_{\text{JSD}} =& tr(\Sigma_1^{-1}\Sigma_2 + \Sigma_1\Sigma_2^{-1}) \\ &+ (\mu_1 - \mu_2)^T(\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2) - 2z\end{aligned} \tag{13}$$

Lastly, we fuse the JSD constraint with the loss function of E2E-CS-ASR using Eq. (10) as follows:

$$\mathcal{L}_{\text{MTL}} = \lambda\mathcal{L}_{\text{CTC}} + (1-\lambda)(\alpha\mathcal{L}_{\text{ATT}} + (1-\alpha)\mathcal{L}_{\text{JSD}}) \tag{14}$$

where $\alpha \in [0, 1]$ controls the importance of the constraint.

Cosine distance. We first compute the average output token representation vectors C_1 and C_2 corresponding to the cluster centroids of CS language pair L_1 and L_2 respectively. The cosine distance between two centroids is then computed as:

$$\mathcal{L}_{\text{CD}} = 1 - \frac{C_1 \cdot C_2}{\|C_1\| \, \|C_2\|} \tag{15}$$

The CD constraint is integrated into the loss function in a similar way as Eq. (14).

5. Experiment

5.1. Dataset

We evaluate our method on Mandarin-English CS language pair from the SEAME [4] corpus (Table 1). We used standard data splitting[1] on par with previous works [1, 6] which consists of 3 sets: train, test$_{man}$ and test$_{eng}$. To match the no CS data scenario, where we assume that we only posses monolingual data for each of the CS languages, we removed all CS utterances from the train set. The test$_{man}$ and test$_{eng}$ sets were used for evaluation. Both evaluation sets are gender balanced and consist of 10 speakers, but matrix[2] language of speakers is different, i.e. Mandarin for test$_{man}$ and English for test$_{eng}$.

5.2. E2E-CS-ASR model configuration

We used ESPnet toolkit [20] to train our baseline E2E-CS-ASR model. The encoder module consists of 4 CNN layers followed by 6 BLSTM layers each with 512 units. The attention-based decoder module consists of single LSTM layer with 512 units and employs hybrid attention mechanism [21]. The CTC module consists of single linear layer with 512 units and its weight in Eq. (10) is set to 0.2. The network was optimized using Adadelta with gradient clipping. During the decoding stage, the beam size was set to 30.

[1] https://github.com/zengzp0912/SEAME-dev-set

[2] The dominant language into which elements from the embedded language are inserted.

Table 1: *SEAME dataset statistics after removing the CS utterances from the train set. 'Man' and 'Eng' refer to Mandarin and English languages, respectively.*

	train		test$_{man}$	test$_{eng}$
	Man	Eng		
# tokens	216k	109k	96k	54k
# utterances	21,476	17,925	6,531	5,321
(# CS utterances)	(0)	(0)	(4,418)	(2,652)
Duration	15.8 hr	11.8 hr	7.5 hr	3.9 hr

5.3. Results and analysis

The experiment results are shown in Table 2. We split the test sets into monolingual and CS utterances to analyze the impact of proposed method on each of them. We first report the mixed error rate (MER)[3] performance of traditional ASR model built using Kaldi toolkit [22] (row 1). The MER performance of the baseline E2E-CS-ASR model is shown in the second row. Following the recent trends [1, 2, 3], we applied speed perturbation (SP) based data augmentation technique [23] and used byte pair encoding (BPE) based subword units [24] to balance Mandarin and English characters (rows 3 and 4). We tried different vocabulary sizes for BPE and found $4k$ units to work best in our case, resulting in much stronger baseline model.

Table 2: *The MER (%) performance of different ASR models built using monolingual data. The test sets are further split into monolingual (mono) and code-switching (CS) uttearances.*

Model	test$_{man}$			test$_{eng}$		
	mono utts.	CS utts.	all	mono utts.	CS utts.	all
Kaldi-TDNN	-	-	39.1	-	-	45.2
E2E-CM-ASR	57.7	73.3	70.6	73.7	80.6	78.3
+ SP	39.4	56.0	53.2	54.2	65.9	62.2
+ BPE (4k)	38.1	51.8	49.5	52.9	61.4	58.9
+ CD	34.4	49.0	46.3	**47.2**	58.5	55.1
+ JSD	34.9	48.8	46.3	47.8	57.6	54.6
+ CD	**34.0**	**48.1**	**45.6**	**47.2**	**57.4**	**54.4**

The performance of models employing proposed CD and JSD constraints are shown in rows 5 and 6, the interpolation weights for CD and JSD are set to 0.9 and 0.97, respectively. Both constraints gain considerable MER improvements. Notably, we found that CD constraint is more effective on monolingual utterances, whereas JSD constraint is more effective on CS utterances. To complement advantages of both constraints, we combined them as follows:

$$\begin{aligned}\mathcal{L}_{\text{MTL}} =& \lambda\mathcal{L}_{\text{CTC}} + (1-\lambda)(\alpha\mathcal{L}_{\text{ATT}} + \\ &(1-\alpha)(\beta\mathcal{L}_{\text{JSD}} + (1-\beta)\mathcal{L}_{\text{CD}}))\end{aligned} \tag{16}$$

where α and β are set to 0.05 and 0.9, respectively. The combination of two constraints significantly improves the MER over the strong baseline model by 3.9% and 4.5% on test$_{man}$ and test$_{eng}$, respectively (row 7). These results suggest that proposed feature matching approaches are effective.

[3] The term "mixed" refers to different token units used for English (words) and Mandarin (characters).

5.3.1. Changing interpolation weight

We repeat the experiment with different interpolation weights for CD and JSD constraints (hyperparameter α in Eq. (14)) to investigate its effect on MER performance. Figure 2 shows that proposed constraints consistently improve the MER and best results are achieved for interpolation weights in range 0.8-0.99.

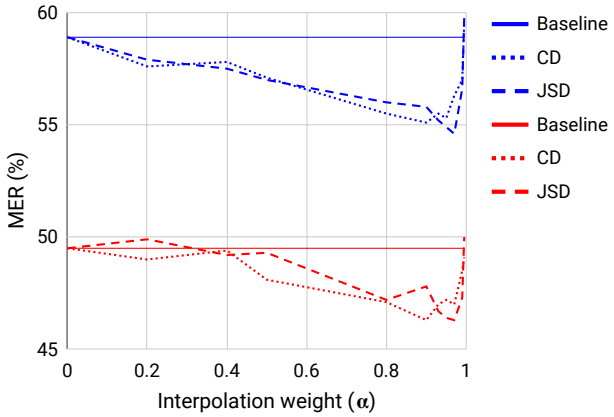
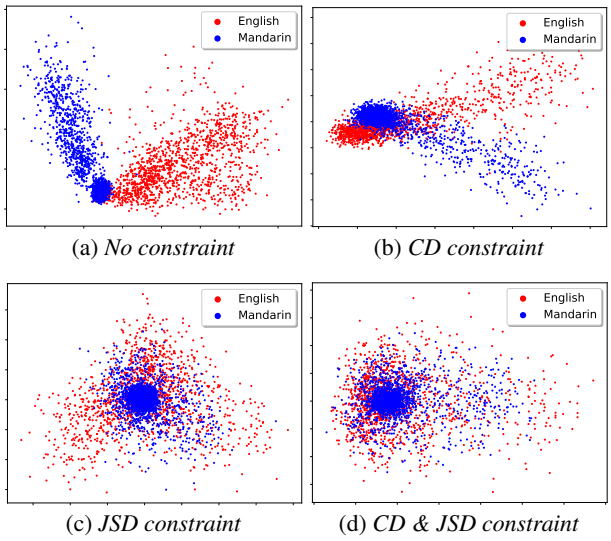


Figure 2: *The impact of constraint interpolation weights on MER performance for test$_{eng}$ (blue) and test$_{man}$ (red) sets.*

5.3.2. Visualization of shared embedding feature space

To gain insights from the effects of proposed method on the shared embedding feature space, we visualize the learned output token representations using dimensionality reduction technique based on principle component analysis (PCA). Figure 3 shows the shared embedding feature space without (3a) and with (3b,3c,3d) proposed constraints. Note that the learned representations are split into two clusters when proposed constraints are not employed. Visualization of shared embedding feature space confirms that our method is effective at bridging the gap between two clusters.



(a) *No constraint* (b) *CD constraint*

(c) *JSD constraint* (d) *CD & JSD constraint*

Figure 3: *PCA visualization of shared embedding feature space of output token representations without (a) and with (b,c,d) proposed constraints.*

5.3.3. Applying language model

To examine whether proposed constraints are complementary with language models (LM), we employed LMs during the decoding and rescoring stages (see Table 3). All LMs were trained on monolingual transcripts of train set and applied to our best model employing both JSD and CD constraints. In the decoding stage, we used shallow fusion technique [25] to integrate subword-level recurrent LSTM LM [26]. During the rescoring stage, we examined Kneser-Ney smoothed 3-gram LM and recurrent LSTM LM, both LMs are mixed word-character level and used to rescore 50-best hypotheses. Obtained MER improvements show that proposed constraints and LMs complement each other.

Table 3: *The MER performance after applying language model during the decoding and rescoring stages.*

Decode LM	Rescore LM	MER (%)	
		test$_{man}$	test$_{eng}$
No	No	45.6	54.4
LSTM (subword)	No	45.1	53.7
LSTM (subword)	3-gram (mixed)		
LSTM (subword)	LSTM (mixed)		

6. Conclusions

In this work, we proposed embedding feature matching approaches for E2E-CS-ASR models trained on monolingual data. Specifically, we examined two approaches based on Jensen-Shannon divergence and cosine distance constraints which are incorporated into the objective function of the E2E-CS-ASR models. The former one is used to enforce learned embedding representations of CS languages to poses similar distributions, while the later one is used to pull together output token representation centroids of CS languages. We evaluated proposed method on Mandarin-English CS language pair from the SEAME corpus where CS utterances were removed from the train set. The experiment results show that the proposed method outperforms the strong baseline model by a large margin, i.e. absolute 3.9% and 4.5% MER improvement on test$_{man}$ and test$_{eng}$, respectively. The visualization of shared embedding feature space confirms the effectiveness of the proposed method. In addition, our method is complementary with language models where further MER improvements can be achieved. Importantly, all these improvements are achieved without using any additional linguistic resources such as word aligned parallel corpus or language identification system.

We believe that proposed method of matching embedding feature representations of output tokens of CS languages can be easily adapted to other scenarios and benefit other CS language processing applications. For the future work, we plan to test the proposed method on scenarios with larger amount of monolingual data and explore efficient ways to further improve MER performance using CS text or speech only data.

7. Acknowledgements

This work is supported by the project of Alibaba-NTU Singapore Joint Research Institute.

8. References

[1] 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," *arXiv preprint arXiv:1811.00241*, 2018.

[2] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Investigating end-to-end speech recognition for mandarin-english code-switching," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, 2019*, 2019.

[3] N. Luo, D. Jiang, S. Zhao, C. Gong, W. Zou, and X. Li, "Towards end-to-end code-switching speech recognition," *arXiv preprint arXiv:1810.13091*, 2018.

[4] D. Lyu, T. P. Tan, E. Chng, and H. Li, "SEAME: a mandarin-english code-switching speech corpus in south-east asia," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, 2010, pp. 1986–1989.

[5] E. Yilmaz, M. McLaren, H. van den Heuvel, and D. A. van Leeuwen, "Semi-supervised acoustic model training for speech with code-switching," *Speech Communication*, vol. 105, pp. 12–22, 2018. [Online]. Available: https://doi.org/10.1016/j.specom.2018.10.006

[6] P. Guo, H. Xu, L. Xie, and E. S. Chng, "Study of semi-supervised approaches to improving english-mandarin code-switching speech recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 1928–1932.

[7] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 2226–2234.

[8] S. Karita, S. Watanabe, T. Iwata, A. Ogawa, and M. Delcroix, "Semi-supervised end-to-end speech recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 2–6.

[9] J. Drexler and J. Glass, "Combining end-to-end and adversarial training for low-resource speech recognition," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, 2018, pp. 361–368.

[10] D. Lyu, R. Lyu, Y. Chiang, and C. Hsu, "Speech recognition on code-switching among the chinese dialects," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*, 2006, pp. 1105–1108.

[11] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali, "Language modeling for code-mixing: The role of linguistic theory based synthetic data," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 1543–1553.

[12] C. Chang, S. Chuang, and H. Lee, "Code-switching sentence generation by generative adversarial networks and its application to data augmentation," *CoRR*, vol. abs/1811.02356, 2018.

[13] G. I. Winata, A. Madotto, C. Wu, and P. Fung, "Learn to code-switch: Data augmentation using copy mechanism on language modeling," *CoRR*, vol. abs/1810.10254, 2018.

[14] E. Yilmaz, H. van den Heuvel, and D. A. van Leeuwen, "Acoustic and textual data augmentation for improved ASR of code-switching speech," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 1933–1937.

[15] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-Attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 949–953.

[16] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, 2016, pp. 4945–4949.

[17] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 2006, pp. 369–376.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

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

[20] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 2207–2211.

[21] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 577–585.

[22] 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 *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.

[23] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015, pp. 3586–3589.

[24] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[25] S. Toshniwal, A. Kannan, C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, 2018, pp. 369–375.

[26] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 194–197.