



Improving Recognition of Out-of-vocabulary Words in E2E Code-switching ASR by Fusing Speech Generation Methods

Lingxuan Ye^{1,2}, Gaofeng Cheng¹, Runyan Yang^{1,2}, Zehui Yang^{1,2},
Sanli Tian^{1,2}, Pengyuan Zhang^{1,2}, Yonghong Yan^{†1,2}

¹Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, CAS, China

²University of Chinese Academy of Sciences, China

{yelingxuan, chenggaofeng, yangrunyan, yangzehui,
tiansanli, zhangpengyuan, yanyonghong}@hcccl.ioa.ac.cn

Abstract

Out-of-vocabulary (OOV) is a common problem for end-to-end (E2E) ASR. For code-switching (CS), the OOV problem on the embedded language is further aggravated and becomes a primary obstacle in deploying E2E code-switching speech recognition (CSSR) systems. Existing recipes for monolingual scenarios typically take advantage of text-to-speech (TTS) synthesis or utilize fine-grained modeling units. However, the sparsity of CS greatly decreases the probability of words to be covered (mainly the embedded language), which hinders the collecting of corresponding CS text for TTS. Using fine-grained units brings limited improvement to the OOV words while increasing the risk of misspelling. In this paper, we propose two distinct CS speech generation methods to improve the recognition of CSSR systems on OOV words. First, we utilize monolingual corpora to generate spliced CS speech containing OOV words. Second, we propose an algorithm to generate CS text containing OOV words, thus enabling using TTS to synthesize CS speech. Both methods are carefully designed to ensure acoustic and semantic smoothness of generated speech. In addition, we provide restrictive methods to suppress the side-effects of using artificially generated data and help avoid misspelling. Finally, we reduced WER on OOV words by 56.3% absolutely on the test set.

Index Terms: speech recognition, code-switching, out-of-vocabulary

1. Introduction

Code-switching (CS) is the alternation of two languages within a single utterance[1, 2]. As globalization is deepening, CS is increasingly becoming a popular phenomenon among multilingual communities. However, building a code-switching speech recognition (CSSR) system is more challenging than a monolingual one.

In the past few years, plenty of works have been targeted to improve end-to-end (E2E) CSSR systems[3, 4, 5, 6, 7]. However, the gap between the experimental results and actual use is aggravated by the distinct nature of CS. CS allows intra-sentential language alternation and introduces a severe sparse problem, which decreases the probability of words to be covered[3]. Even if we assume the CS corpus has achieved a perfect balance that both languages account for an equal amount, at least twice the amount of data for building a monolingual vocabulary is needed[8] to build a sufficient vocabulary

for both involved languages. Also, the vocabulary of the CS speakers varies according to their fluency in languages and depends on their background knowledge[9, 10, 11]. This makes CSSR systems frequently confronted with out-of-vocabulary (OOV) words in real-world development. An extreme case is CS in the Chinese mainland. As shown in Figure 1, Chinese characters take up the main part of a CS utterance while several (usually 1 or 2) English words are embedded in[12]. This distinct characteristic causes the collecting process of CS data in the Chinese mainland to be less efficient and makes the OOV problem for the embedded language - English - be the crucial obstruct. Severe as the OOV problem for CSSR is, there were few works that shed light on this topic.



Figure 1: Examples of CS utterances in Singapore and the Chinese mainland. The example utterances are sampled from SEAME and DT-CS corpus, respectively. CS utterances in the Chinese mainland have an extremely imbalanced language ratio, which aggravates the OOV problem for the English part.

OOV problem is a general challenge for E2E ASR systems. Traditional HMM-based hybrid ASR systems[13, 14] consist of a separated acoustic model (AM) and a language model (LM), which makes word extension more efficient for them. In recent years, E2E models have achieved better performance over hybrid systems[15]. However, E2E ASR models behave faithfully according to the distribution of training data and are prone to degrading performance when facing imbalanced word frequency[16], or to the extreme end, the OOV problem[17].

Generally, there are two types of methods addressing the OOV problem for E2E ASR in the monolingual scenario: using text-to-speech (TTS) to create speech from textual data containing OOV words[17, 18] or using fine-grained modeling units[19, 20]. However, it is not always feasible for CSSR to apply such strategies. Textual CS data is also rare and obsessed by the sparsity problem. Plenty of works[21, 22, 23] have made efforts to generate CS text. However, neither of them guarantees that the generated texts contain specified words, i.e., generate the context of OOV, which is fundamental for TTS-based

[†]Corresponding author.

This work is partially supported by the National Key Research and Development Program of China (No. 2020AAA0108002)

methods. Using fine-grained units such as byte pair encoding (BPE)-based subwords or multi-level tokens enabled the E2E models to address the open-vocabulary problem. However, only limited improvement is achieved since changing only the tokenization strategy is less helpful for a CSSR model to build the representation of OOV words.

To address the sparsity problem aforementioned and build a strong representation for OOV words, in this work, we propose two CS speech generation methods to promote an E2E CSSR system’s performance on the OOV words. We target the CS in the Chinese mainland where OOV and focus on OOV for the embedded English part. We will introduce our proposed methods and validate their effectiveness in the following sections.

2. Proposed Methods

An overview of our proposed methods is illustrated in Figure 2. We designed two distinct methods to provide CS speech containing the OOV words. The first method is acoustic feature splicing (AFS), which uses monolingual speech and CS speech to generate spliced CS speech. The second method firstly generates CS texts corresponding to OOV words by a statistical machine translation (SMT)-based algorithm and then synthesizes CS speech using a TTS system. The two methods have different characteristics and are complementary to each other.

We aim to promote a well-trained baseline CSSR system by fine-tuning it on the generated CS data. Using artificially generated data has its drawbacks. First, the artifacts of generated data could be harmful to CSSR models’ performance on the other words. We utilize learning without forgetting (LWF) regulation during the fine-tuning stage to preserve the performance on the other words. Second, generated CS data cannot let CSSR models build a strong representation of OOV words and could lead to misspellings. We propose a lexicon constraint method for the decoding stage to prune invalid decoding paths and further promote the performance on the OOV words.

2.1. CS Speech Generation

2.1.1. AFS

We propose to create CS speech by splicing. For accelerating, splicing is directly operated on the extracted acoustic feature. We first extract the alignment information of both the CS and monolingual corpus to get the timestamps for each word. Then, we use the timestamps in the monolingual dataset to build a *feature dictionary* for words. Given a CS utterance, we choose a word in the OOV word list and get its acoustic feature by looking it up in the feature dictionary. Then we choose a word in the CS utterance and replace it with the OOV word to get spliced CS utterance. The splicing procedure is illustrated in Figure 3. AFS has become a popular augmentation method for ASR training recently. In [24, 25], AFS is conducted within datasets to enrich training data. The splicing-based method features its simplicity and can increase speaker diversity, yet the spliced speech loses its smoothness at concatenation points, which can be harmful to CSSR models.

2.1.2. SMT&TTS-based Method

We also propose using TTS to generate the CS data. Firstly, we design an algorithm to generate CS text containing OOV words with bilingual parallel corpora. Then the generated CS text is used to synthesize the corresponding speech with TTS.

The CS text generation algorithm is based on word alignment results from SMT systems. We denote the parallel corpus by $\mathbf{C} = \{\mathbf{c}^i, 1 \leq i \leq k\}$ and $\mathbf{E} = \{\mathbf{e}^i, 1 \leq i \leq k\}$, where \mathbf{C} is the Mandarin corpus and \mathbf{E} is the English corpus. Sentence

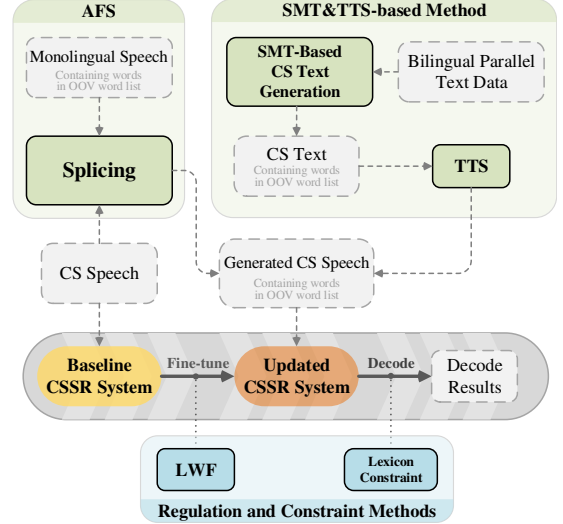


Figure 2: We promote a baseline CSSR system’s performance on OOV words by fine-tuning it on generated CS speech. The green parts are our proposed CS speech generation methods. AFS utilizes monolingual speech and CS speech to create spliced CS speech. The SMT&TTS-based method firstly generates CS text and then synthesizes CS speech with the TTS system. We also propose using LWF regulation and lexicon constrained decoding, which are shown in the blue part. LWF regulation is used in the fine-tuning stage to suppress the side-effect of using artificially generated data. Lexicon constraint is applied on beam search decoding to reduce misspelling of the OOV words.

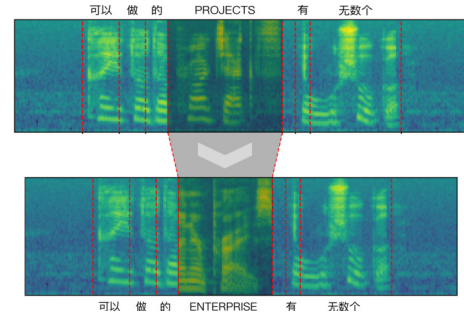


Figure 3: An illustrative example of AFS. The acoustic feature of the English word “PROJECT” in the original CS utterance is replaced by the feature of an OOV word - “ENTERPRISE” - which is extracted from a monolingual corpus.

$\mathbf{c}^i = \{c_j^i \in \Lambda_{\text{man}} | 1 \leq j \leq m_i\}$ in Mandarin consists of m_i words and sentence $\mathbf{e}^i = \{e_j^i \in \Lambda_{\text{eng}} | 1 \leq j \leq n_i\}$ in English consists of n_i words. Λ_{man} and Λ_{eng} are vocabulary sets for the Mandarin and English corpora, respectively. Λ_{OOV} is the OOV word list. In [18], the OOV word list and the context of the OOV words are both extracted from the dev set as the input for TTS system. We only extract the OOV word list Λ_{OOV} from dev set here since the CS context is not available in most cases. The word alignment $\mathbf{A} = \{\mathbf{a}^i \in \{0, 1\}^{m_i \times n_i}, 1 \leq i \leq k\}$ is extracted from the parallel corpus in an unsupervised way. \mathbf{a}^i is a binary matrix indicating whether two words are aligned or not. For sentences \mathbf{c}^i and \mathbf{e}^i , their alignment matrix \mathbf{a}^i is defined as:

$$a_{p,q}^i = \begin{cases} 1 & \text{if } c_p^i \text{ is aligned to } e_q^i, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Different from the application in the machine translation

task, we only focus on the translation of OOV words. So we propose an algorithm to extract phrases corresponding to a single OOV word. Given a Mandarin sentence \mathbf{c}^i , the corresponding English sentence \mathbf{e}^i , and their word alignment matrix \mathbf{a}^i , for all words e_r^i that in the OOV word list, the r -th column of \mathbf{a}^i is extracted as $a_{\cdot,r}^i$. Then we collect all consecutive words as possible translation candidates T^i as:

$$T^i = \left\{ \left\{ b_j^i \mid p \leq j \leq q \right\} \mid \forall 1 \leq p \leq s \leq q \leq m_i, a_{s,r}^i = 1 \right\}. \quad (2)$$

It is worth noting that the collecting process allows overlaps. Then we accumulate all valid translation pairs for all words in Λ_{eng} across the whole corpus and select the top 2 translation pairs as the final translation dictionary. Finally, text pairs in \mathbf{C} and \mathbf{E} containing words in Λ_{OOV} are extracted as \mathbf{C}_{OOV} and \mathbf{E}_{OOV} :

$$\mathbf{E}_{\text{OOV}} = \{ \mathbf{e} \in \mathbf{E} \mid \exists e, e \in \Lambda_{\text{OOV}} \cap \mathbf{e} \},$$

$$\mathbf{C}_{\text{OOV}} = \{ \mathbf{c}^i \in \mathbf{C} \mid \mathbf{e}^i \in \mathbf{E}_{\text{OOV}} \}.$$

We switch the OOV words in \mathbf{C}_{OOV} according to the translation dictionary extracted and get CS text that contains the OOV words. The generated text is further cleaned before feeding to the TTS system.

For the TTS system, we used thinkIT TTS engine[26]. It is built with style transfer and adversarial training techniques to support synthesizing fluent Mandarin-English CS speech.

It is worth noting that the proposed methods only focus on the issue of words. The semantic constraints applied are relatively simple, which are acceptable approximate for CS in the Chinese mainland. For CS cases where the language ratio is relatively balanced, the generation methods should be modified to consider more complications such as the syntactic structure of the involved languages.

2.2. Regulation and Constraint Methods

2.2.1. LWF Regulation

Fine-tuning on artificially generated data could cause the catastrophic forgetting problem, which degrades the model’s performance on the non-OOV words. We applied LWF[27] regulation during fine-tuning to suppress the catastrophic forgetting problem. LWF is first applied in the computer vision domain and is similar to knowledge distillation. LWF implicitly restricts the model’s parameters by constraining the output of fine-tuned model not too far from the original model’s output. We apply LWF on the ASR model’s encoder and write the loss of LWF as

$$\mathcal{L}_{\text{LWF}}(\theta) = 1 - \mathbb{E}_{\mathbf{x} \sim \mathbf{X}} \text{sim}(\text{Enc}(\mathbf{x}, \theta), \text{Enc}(\mathbf{x}, \theta^*)), \quad (3)$$

where $\text{Enc}(\mathbf{x}, \theta)$ is the encoding function, taking acoustic feature \mathbf{x} of an utterance and the encoder’s parameter θ as inputs and outputting embedded feature. θ^* denotes parameters of the original model before fine-tuning. For the feature series similarity function $\text{sim}(\cdot, \cdot)$, we use cosine similarity in our experiment:

$$\text{sim}(\mathbf{a}_{1:T}, \mathbf{b}_{1:T}) = \frac{1}{T} \sum_{t=1}^T \frac{\mathbf{a}_t \cdot \mathbf{b}_t}{|\mathbf{a}_t| |\mathbf{b}_t|}. \quad (4)$$

For training the CSSR systems, we use the hybrid CTC/Attention[28] loss function, which is formulated as:

$$\mathcal{L}_{\text{CTC/Att}}(\theta, \phi) = \lambda_{\text{CTC}} \mathcal{L}_{\text{CTC}}(\theta) + (1 - \lambda_{\text{CTC}}) \mathcal{L}_{\text{Att}}(\theta, \phi). \quad (5)$$

ϕ denotes the parameters in the decoder, λ_{CTC} is set to 0.3 according to the default recipe. Combined with LWF, the final loss function is

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{CTC/Att}}(\theta, \phi) + \lambda_{\text{LWF}} \mathcal{L}_{\text{LWF}}(\theta). \quad (6)$$

2.2.2. Lexicon Constrained Beam Search Decoding

The CSSR learns OOV words from artificially generated data, which might not be so accurate and could let CSSR still misspell the added OOV words. Thus, we propose to optimize the decoding stage by constraining the decoding path on a lexicon. As illustrated in Figure 4, when decoding the embedded language (where the OOV words are located), any path that produces an invalid word is deprecated.

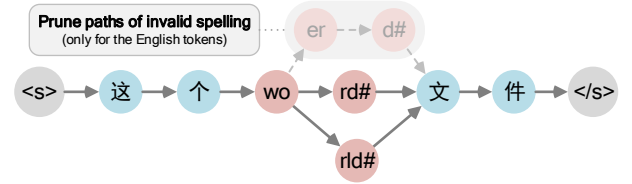


Figure 4: An illustrative example of lexicon constrained beam search decoding. When confronted with an English token (a red circle), the subsequent decoding path is restricted on a lexicon. The invalid spelling “woerd” is deprecated in this example.

3. Experiments

3.1. Corpora

Two speech corpora and one parallel textual corpus are used in our experiments. For the CS speech corpus, we used DT-CS, a 200-hour Mandarin-English CS speech corpus provided by DataTang¹. The corpus contains 187k utterances recorded from mainland Chinese speakers. Librispeech[29] is used as the monolingual speech corpus to extract voice fragments of OOV words. We also extract the utterances from Librispeech that contain the OOV words and form a subset eng_{OOV} . The detail of the speech corpora and their partitions are listed in Table 1. For parallel text corpus, we use Translation2019 [30], a large Mandarin-English parallel corpus containing 5.2M sentence pairs (1.1GB) of text data.

3.2. Experimental Setup

All the experiments are conducted on the ESPnet2 toolkits [31]. We follow the Conformer recipe for AISHELL-1. General data augmentation methods such as SpecAugment or speed perturbation are not applied since they will slow down the training process. Also, shallow fusion decoding with LM is not used.

As for modeling, characters are used for Mandarin, and BPE-based subword units are used for English. We generated 1050 BPE units for English and used 4969 units totally.

Conventionally, mixed error rate (MER) on the test set is reported, which stands for a mixture of character error rate (CER) for Chinese characters and word error rate (WER) for English words. We also calculate WER on the English part and only on OOV words to give a close look at OOV performance.

¹DT-CS corpus was served as the training set of the ASRU 2019 Code-Switching Challenge. The additional 40-hour development set is not accessible according to DataTang. We re-partitioned DT-CS thus the results are NOT comparable with works that used the development subset such as [5]. The detail of our partition is released at <https://github.com/winlaic/CSOOV-Partition>.

Table 1: Details of partitions of used corpora.

Partition	Source	#utterances	Duration(h)
train	DT-CS	167136	180
dev	DT-CS	9731	10
test	DT-CS	9687	10
eng _{OOV}	Librispeech	19463	70

Table 2: MER and WER results on the test set with AFS applied. We use the following abbreviations in column “AFS Type”. **R**: Randomly replace a word in CS utterances without considering language. **E**: Only replace English words in CS utterances. **P**: Only replace the word with the same POS tag.

Fine-tune Partition	AFS Type	MER _{ALL}	WER	
			ENG	OOV
(w/o fine-tuning)	-	11.5	39.3	82.1
train + eng _{OOV}	-	11.2	36.3	70.8
train + AFS	R	9.9	28.8	48.6
	R+P	9.9	29.1	48.7
	E	9.8	28.7	46.9
	E+P	9.8	28.3	46.3

3.3. Baselines and Fine-tuning on AFS Generated Speech

We set up two baselines in our experiments. The first baseline system is directly trained on the train set and used to initialize all the subsequent experiments. We fine-tune the first system on a combination of train and eng_{OOV} set to get the second baseline system. This system has access to the speech of OOV words, but the context is in English. The performance of the baseline systems is listed in the first two rows of Table 2.

In AFS, we select one word in the CS utterance and replace the word’s acoustic feature and the corresponding transcript with that from an OOV word to create spliced CS speech for the fine-tuning stage. We only perform such replacement on 40% of utterances in a mini-batch. Four different word replacement strategies are studied in our experiments. Firstly, we try randomly selecting one among all words in a CS utterance and replace it. Considering that random replacement could deform the semantic structure, we try constraining to replace words with the same part-of-speech (POS) tag. We also try replacing only English words to keep the language ratio in CS utterances. The results are shown in the rest rows of Table 2.

All the AFS strategies lead to a significant improvement on OOV words. AFS can reduce the WER of OOV words from 82.1% to less than 50%. Also, fine-tuning on AFS generated data is superior to fine-tuning on a mixture of monolingual and CS data, indicating that the CS context is fundamental for training. Only performing AFS on the English words leads to a better result, indicating that keeping the language ratio is essential. On the other hand, constraining POS leads to a marginal improvement and even slightly degrades the performance of random replacing. We conclude that the model is more sensitive to language distribution than semantic coherence.

3.4. CS Text Generation and Fine-tuning on TTS Synthesized Speech

For CS text generating, the fast-align[32] toolkit is used to extract alignment information. The generated raw CS text is filtered that we only keep texts with approximately the same lengths as utterances in the training set. Then we utilize the TTS engine mentioned above to synthesize the speech. There are 70 hours of speech from 20k generated CS sentences synthe-

Table 3: MER and WER results on the test set when adding synthesized data based on the best configuration of AFS. “SYN” indicates the total duration of the synthesized speech.

SYN(h)	MER _{ALL}	WER	
		ENG	OOV
0	9.8	28.3	46.3
14	9.5	27.0	42.7
28	9.3	26.2	39.5
42	9.3	26.2	39.5
56	9.3	26.0	38.0
70	9.4	26.3	38.7

Table 4: MER and WER results on the test set when applying LWF and lexicon constraint (referred to as “LC”) based on the best configuration in previous experiments.

Methods	MER _{ALL}	WER	
		ENG	OOV
AFS+SYN(56h)	9.3	26.0	38.0
+LWF	9.0	25.1	37.7
+LC	8.2	20.1	26.8
+LWF+LC	8.0	19.4	25.8

sized. Each sentence is spoken by a male speaker and a female speaker. The synthesized speech is equally split into five parts.

Based on the best configuration of AFS, we gradually add TTS-synthesized data to the training set to see how it influences the performance on the OOV words. The results are shown in Table 3. We can see that incorporating synthesized CS data does help the recognition. The WER on OOV words reaches a minimum when adding about 56 hours of synthesized data. This indicates that synthesized data is helpful for OOV performance, but excess synthesized data can be harmful to all the words.

3.5. Applying Regulation and Constraining Decoding

Based on the best configuration of the two augmentation methods, we incorporate the LWF regulation. We performed several experiments to tune λ_{LWF} in Eq. 6 and finally set it to 0.5. Limited by space, the detailed results are not shown. We also exert the proposed lexicon constraint in decoding stage.

We report the results in Table 4. We can see that the LWF regulation improves the overall performance while retaining the performance on OOV words, which indicates that LWF successfully suppressed the catastrophic forgetting problem. The lexicon-constrained decoding strategy significantly improves all the results and benefits OOV words part (relatively 29.5%) more than all the English part (relatively 23.1%), indicating that the proposed method is more helpful on OOV words.

4. Conclusion

In this work, we designed two distinct CS speech generation methods to improve the recognition of OOV words in CSSR. Considering the acoustic and semantic smoothness, we carefully designed splicing-based and SMT&TTS-based methods to generate reasonable CS speech for the OOV words. We also proposed restrictive methods to suppress the side-effects of using generated data and alleviate the misspelling problem. The performance on OOV words is significantly improved over the baseline systems. The performance could be further promoted by creating more smooth spliced speech or increasing the speaker diversity in the TTS synthesized speech, which is remained to be explored in future works.

5. References

- [1] S. Poplack, "Sometimes I'll start a sentence in Spanish Y TERMINO EN ESPAÑOL: toward a typology of code-switching," *Linguistics*, 2013.
- [2] J. Milroy *et al.*, *One speaker, two languages: Cross-disciplinary perspectives on code-switching*. Cambridge University Press, 1995, vol. 10.
- [3] 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*, 2019, pp. 2165–2169.
- [4] X. Zhou, E. Yilmaz, Y. Long, Y. Li, and H. Li, "Multi-Encoder-Decoder Transformer for Code-Switching Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 1042–1046.
- [5] 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*, 2020, pp. 4766–4770.
- [6] S.-P. Chuang, H.-J. Chang, S.-F. Huang, and H.-y. Lee, "Non-autoregressive mandarin-english code-switching speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 465–472.
- [7] S. Zhang, J. Yi, Z. Tian, J. Tao, and Y. Bai, "Rnn-transducer with language bias for end-to-end mandarin-english code-switching speech recognition," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.
- [8] 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 (Volume 1: Long Papers)*, 2018, pp. 1543–1553.
- [9] N. T. Vu, H. Adel, and T. Schultz, "An investigation of code-switching attitude dependent language modeling," in *Statistical Language and Speech Processing*, A.-H. Dediu, C. Martín-Vide, R. Mitkov, and B. Truthe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 297–308.
- [10] Y. Li and P. Fung, "Code-switch language model with inversion constraints for mixed language speech recognition," in *Proceedings of COLING 2012*, 2012, pp. 1671–1680.
- [11] S. Sitaram, K. R. Chandu, S. K. Rallabandi, and A. W. Black, "A survey of code-switched speech and language processing," *arXiv preprint arXiv:1904.00784*, 2019.
- [12] G. Liu and L. Cao, "Code-switch speech rescoring with monolingual data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6229–6233.
- [13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldii speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.
- [14] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with lstms," in *Proc. Interspeech 2017*, 2017, pp. 1586–1590. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2017-129>
- [15] G. Cheng, H. Miao, R. Yang, K. Deng, and Y. Yan, "Ete: Unified attention-based end-to-end asr and kws architecture," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–1, 2022.
- [16] K. Deng, G. Cheng, R. Yang, and Y. Yan, "Alleviating asr long-tailed problem by decoupling the learning of representation and classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 340–354, 2022.
- [17] M. Mimura, S. Ueno, H. Inaguma, S. Sakai, and T. Kawahara, "Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 477–484.
- [18] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5674–5678.
- [19] J. Li, G. Ye, R. Zhao, J. Droppo, and Y. Gong, "Acoustic-to-word model without oov," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 111–117.
- [20] A. Das, J. Li, G. Ye, R. Zhao, and Y. Gong, "Advancing acoustic-to-word ctc model with attention and mixed-units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1880–1892, 2019.
- [21] C.-Y. Li and N. T. Vu, "Improving Code-Switching Language Modeling with Artificially Generated Texts Using Cycle-Consistent Adversarial Networks," in *Proc. Interspeech 2020*, 2020, pp. 1057–1061.
- [22] C.-T. Chang, S.-P. Chuang, and H.-Y. Lee, "Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation," in *Proc. Interspeech 2019*, 2019, pp. 554–558. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-3214>
- [23] Y. Gao, J. Feng, Y. Liu, L. Hou, X. Pan, and Y. Ma, "Code-Switching Sentence Generation by Bert and Generative Adversarial Networks," in *Proc. Interspeech 2019*, 2019, pp. 3525–3529. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-2501>
- [24] J. Sun, Z. Tang, H. Yin, W. Wang, X. Zhao, S. Zhao, X. Lei, W. Zou, and X. Li, "Semantic Data Augmentation for End-to-End Mandarin Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 1269–1273.
- [25] T. K. Lam, M. Ohta, S. Schamoni, and S. Riezler, "On-the-Fly Aligned Data Augmentation for Sequence-to-Sequence ASR," in *Proc. Interspeech 2021*, 2021, pp. 1299–1303.
- [26] Z. Shang, Z. Huang, H. Zhang, P. Zhang, and Y. Yan, "Incorporating Cross-Speaker Style Transfer for Multi-Language Text-to-Speech," in *Proc. Interspeech 2021*, 2021, pp. 1619–1623.
- [27] Z. Li and D. Hoiem, "Learning without forgetting," in *Computer Vision - 14th European Conference, ECCV 2016, Proceedings*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Germany: Springer, 2016, pp. 614–629.
- [28] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [29] 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.
- [30] B. Xu, "Nlp chinese corpus: Large scale chinese corpus for nlp," Sep. 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.3402023>
- [31] 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 *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2018-1456>
- [32] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 644–648. [Online]. Available: <https://aclanthology.org/N13-1073>