SPEECH COLLAGE: CODE-SWITCHED AUDIO GENERATION BY COLLAGING MONOLINGUAL CORPORA

Amir Hussein^{†1}, Dorsa Zeinali^{†2}, Ondřej Klejch³, Matthew Wiesner¹, Brian Yan⁴, Shammur Chowdhury⁵, Ahmed Ali⁵, Shinji Watanabe⁴, Sanjeev Khudanpur¹

¹Johns Hopkins University, USA, ²Northeastern University, USA, ³ University of Edinburgh, UK, ⁴Carnegie Mellon University, USA, ⁵ Qatar Computing Research Institute, Doha

ABSTRACT

Designing effective automatic speech recognition (ASR) systems for Code-Switching (CS) often depends on the availability of the transcribed CS resources. To address data scarcity, this paper introduces *Speech Collage*, a method that synthesizes CS data from monolingual corpora by splicing audio segments. We further improve the smoothness quality of audio generation using an overlap-add approach. We investigate the impact of generated data on speech recognition in two scenarios: using in-domain CS text and a zero-shot approach with synthesized CS text. Empirical results highlight up to 34.4% and 16.2% relative reductions in Mixed-Error Rate and Word-Error Rate for in-domain and zero-shot scenarios, respectively. Lastly, we demonstrate that CS augmentation bolsters the model's code-switching inclination and reduces its monolingual bias.

Index Terms— Code-switching, ASR, data augmentation, end-to-end, zero-shot learning

1. INTRODUCTION

In multilingual societies, code-switching (CS) is integral to communication, enabling clearer expression and reflecting cultural nuances [1, 2]. While CS is prevalent in daily conversations, it is underrepresented in transcribed datasets. This linguistic phenomenon, where speakers interweave languages within a conversation or utterance, poses challenges for voice technologies like automatic speech recognition (ASR). Given the abundance of monolingual data and scarcity of labeled CS speech, there's a pressing need to harness monolingual resources for CS applications. The prime challenge lies in developing robust ASR systems for CS in zero-shot settings where no CS training data is available.

Several approaches have proposed to build CS ASR directly from monolingual data by utilizing multilingual training [3–8]. Further studies advocate for the joint modeling of CS and monolingual ASR, effectively breaking down bilingual tasks into monolingual components [9–11]. A prominent issue with monolingual training is the model's monolingual bias which impedes seamless language switching [12]. To address this issue, several data augmentation strategies have been proposed including textual data augmentation, text-to-speech synthesis, and concatenation-based speech generation. In [13] authors proposed a methodology to generate the code-switching text from monolingual text to improved ASR performance with language model rescoring. In [8, 14], researchers propose merging monolingual utterance to mimic code-switching. However, this strategy tends to primarily capture inter-sentential switches, often sidelining the nuances of intra-sentential CS. On another front, text-to-speech (TTS) based synthetic audio has gained traction for CS data generation [15–21]. Despite its potential, TTS based augmentation suffer from limited speaker variability compared to real data. Consequently, there's a growing interest in using audio segment splicing as augmentation to covers more speaker variations and acoustic environments [22, 23]. However in the proposed splicing, speech segments and their corresponding words are randomly selected, and the potential of splicing method in code-switching remains unexplored.

In this paper, we introduce Speech Collage¹, a data augmentation technique that constructs synthetic code-switched audio from monolingual data. Our method is inspired by traditional concatenation-based speech synthesis techniques [24, 25]. We demonstrate the efficacy of Speech Collage with two scenarios: a) In-domain CS text: where target-domain CS text is leveraged, and b) Zero-shot CS: where synthesized CS text is used. Our study covers two language pairs: Mandarin-English and Arabic-English. Experimental results show substantial improvements Speech Collage brings to code-switching ASR for both scenarios. Our contributions include: (i) a novel speaker-agnostic CS data augmentation derived from monolingual resources, (ii) further improving ASR performance with enhanced audio quality in generated data, and (iii) propose a of zero-shot learning framework tailored for CS. As an additional contribution, we conduct an ablation study to assess the significance of each component on the final performance. We also perform a modified Code Mixed Index (CMI) analysis to identify where the primary gains achieved through our augmentation method.

2. SPEECH COLLAGE

We propose a framework designed to splice speech units extracted from monolingual corpora. These units are based on code-switched text, either real or synthesized, as depicted in Figure1. For the merging process, we select word-units for English and Arabic, and characters for Mandarin. While smaller units, such as phones, offer greater adaptability, they tend to degrade audio quality [26]. The constructed data form segment splicing, encompasses variations from multiple speakers and diverse acoustic environments. We first obtain the unit alignments with audio from the monolingual data by training standard Hidden Markov Model-Gaussian Mixture Model

[†]Both authors contributed equally to this research.

¹Visit our repository for audio samples and implementation https://github.com/JSALT2022CodeSwitchingASR/ generating-code-switched-audio



Fig. 1. High level illustration of the proposed Speech Collage CS generation approach.

(HMM-GMM)² using Kaldi ASR toolkit [27]. Utilizing these alignments, in conjunction with the CS text and monolingual audio, our Speech Collage framework generates the CS audio dataset. In cases where the training data possesses multiple segments for a singular unit, a segment is selected at random. The generated audio quality is further enhanced using the overlap-add technique, energy normalization, and n-gram matching, as detailed below. The audio enhancement and segment splicing were implemented using the Lhotse toolkit [28].

2.1. Overlap-add

To enhance the quality of the generated CS audio, we employ the overlap-add with a Hamming window to mitigate discontinuity effects resulting from spliced units. To ensure the segment capture of each unit, we extend the unit-segments by 0.05 seconds at the start and end of the segment. This extension provides an extra 0.05 second which is utilized as overlap in overlap-add process.

2.2. Energy normalization

Additionally, we normalize the synthesized utterance by the average of unit-segments energy to remove artifacts introduced by energy variations between segments. For a speech sequence X of length T, $X = \{x_t \in \mathbb{R} | t = 1, \dots, T\}$, the average energy is calculated as follows:

$$X' = \left\{ \frac{x_t}{\sqrt{\frac{1}{T} \sum_t x_t^2}} | t = 1, \cdots, T \right\}$$
(1)

2.3. N-gram units

To further enhance the quality of the generated CS we explore splicing consecutive units (n-grams), in alignment with selecting longer units in concatenated speech synthesis [29] Given a CS sentence our approach starts by matching the largest consecutive unit from monolingual alignments. If a specific n-gram is unavailable, the algorithm backs off to a smaller unit. It's worth noting that in this study, we only experimented with unigrams and bigrams. A detailed

Algorithm 1 Speech Collage

Require:	Monolingual combined corpus \mathcal{D}	
Require:	Code-switched text \mathcal{T}_{CS}	
Require:	Alignments $\mathcal A$, size of n-grams n	
1: funct	tion SetupSupervisions(\mathcal{A}, n)	
2: f o	or $w_{1:n}$ with timings $(t_s^n, t_e^n) \in \mathcal{A}$ do	
3:	cut, $w_{1:i} \leftarrow \text{GETLONGESTAUDIOCU}$	$\Gamma(w_{1:n}, (t_s^n, t_e^n))$
4:	$\mathcal{D}[w_{1:i}] \leftarrow cut$	
5: r e	eturn \mathcal{D}	
6: funct	tion NORMALIZEENERGY(audioCut)	⊳ Eq 2
7: e	\leftarrow AVGENERGY(audioCut)	
8: r e	eturn audioCut / \sqrt{e}	
9: funct	tion SAMPLEUNIT($\mathcal{D}, w_{1:i}$)	
10: r e	eturn random cut from $\mathcal{D}[w_{1:i}]$	
11: funct	tion GenerateCollage($\mathcal{T}_{CS}, \mathcal{A}, n$)	
12: <i>T</i>	$\mathcal{P} \leftarrow \text{SETUPSUPERVISIONS}(\mathcal{A}, n) \qquad \triangleright$	n: size of n-gram
13: <i>T</i>	$\mathcal{D}_{CS} \leftarrow \emptyset$	0
14: f o	or $text_i \in \mathcal{T}_{CS}$ do	
15:	$W \leftarrow \text{GETCONSECUNITS}(text_i)$	
16:	for $w_i \in W$ do	
17:	$cut \leftarrow \text{SampleUnit}(\mathcal{D}[w_i])$	
18:	$cuts_i \leftarrow \text{OVERLAPADD}(cuts_i, cu)$	<i>t</i>)
19:	$cuts_i \leftarrow \text{NORMALIZEENERGY}(c)$	$uts_i)$
20:	$cuts_i.text \leftarrow cuts_i.text + w_i$	\triangleright Append w_i
21:	\mathcal{D}_{CS} .append($cuts_i$)	
22: r	eturn \mathcal{D}_{CS}	

description of n-gram Speech Collage implementation is described in Algorithm 1. Using the alignments from monolingual data and maximum n-gram size, SETUPSUPERVISIONS(·) creates a collection \mathcal{D} of audio segments corresponding to each n-gram unit. Consecutive n-gram units are matched from alignments, starting with n and progressing to unigrams. If an n-gram is absent, the algorithm backs off to an (n - 1) unit. In GENERATECOLLAGE(·), the function GETCONSECUNITS(·) returns all consecutive (1 : n)units. Each n-gram unit is randomly drawn from its respective collection using SAMPLEUNIT(·). These segments are appended to the current spliced utterance with OVERLAPADD(·) described in §2.1, and the resulting combined utterance undergoes energy normalization NORMALIZEENERGY(·) from Eq1.

²https://github.com/kaldi-asr/kaldi/tree/master/ egs/aishell/s5

https://github.com/kaldi-asr/kaldi/tree/master/
egs/mgb2_arabic/s5

2.4. Zero-shot CS framework

In this case study we focus on generating Arabic-English codeswitching (CS) data, operating under the assumption that no Arabic-English CS training data is available. To generate speech data using the Speech Collage method, we require CS text. We generate the CS text from monolingual resources using the lexicon-based (Random) replacements approach described in [13]. The approach entails the following steps:

- 1. **Parallel Text Translation**: We leverage a public Arabic-English Machine Translation System³ to generate the parallel English text from the Arabic transcription.
- 2. Word Level Alignments: After translation, we fine-tune multilingual BERT (mBERT) [30] to obtain the word-level alignments.
- 3. **Random Replacement**: Given the alignments, Arabic words are randomly substituted with their corresponding English words at a rate of 20%, as suggested by [13].

2.5. End-to-End Speech Recognition

In this work, we utilized the end-to-end (E2E) ASR conformer architecture [31], with the ESPNET toolkit [32]. The E2E-ASR implementation consists of a conformer encoder and a transformer decoder. Both are multiblocked self-attention architectures with the encoder further enhanced by an additional convolution module. The ASR task is formulated as Bayesian decision finding the most probable target word sequence $\hat{\mathbf{Y}}$, from all possible outputs \mathbf{Y}^* , by selecting the sequence which maximizes the posterior likelihood $P(\mathbf{Y}|\mathbf{X})$, given T-length sequence of D-dimensional speech features, $\mathbf{X} = {\mathbf{x}_t \in \mathbb{R}^D | t = 1, \dots, T}$. For text tokenization, we used word-piece byte-pair-encoding [33]. The total loss function \mathcal{L}_{asr} is a multi-task learning objective that combines the decoder cross-entropy (CE) loss \mathcal{L}_{ce} and the CTC loss [34] \mathcal{L}_{ctc} .

$$\mathcal{L}_{asr} = \alpha \mathcal{L}_{ctc} + (1 - \alpha) \mathcal{L}_{ce} \tag{2}$$

where α is used for interpolation. In our approach, the conformer is initially pre-trained on monolingual data and subsequently fine-tuned on monolingual and synthetic CS speech combined.

2.6. Code-Mixing Index

To quantify the amount of code-switching we use *Code-Mixing Index* (CMI) metric [35]. The CMI for an utterance is defined as:

$$CMI = \frac{\frac{1}{2} * (N - \max_i) + \frac{1}{2}P(x)}{N}$$
(3)

Where max_i represents the number of words in the dominant language *i*, *N* is the total word count in the utterance, *P* is the number of code alternation points, with the constraint $0 \le P < N$. A low CMI score indicates monolingualism in the text whereas the high CMI score implies high degree of code-mixing in the text.

3. DATA AND EXPERIMENTAL SETUP

In-domain: The target domain we are considering is the Mandarin-English code-switching, specifically SEAME [36]. In this scenario, we utilize monolingual training data from Chinese AISHELL-1 [37], 100h of English data randomly sampled from Tedlium3 [38] and SEAME text [36] to generate 62.2 hours of CS data. Evaluation is performed on SEAME test sets (devman and devsge), measuring mixed error-rate (MER) that considers word-level English and character-level Mandarin. We also report WER on monolingual English and CER on monolingual Chinese subsets.

Zero-shot: For this scenario, we use monolingual training data from MGB-2 [39] and Tedlium3. We generate 80 hours of CS data using synthetic CS text described in §2.4. Evaluation is conducted on ES-CWA [8], which is a real Arabic-English CS dataset.

Data pre-processing: All audios are augmented with speed perturbations (0.9, 1.0 and 1.1) and transformed into 83-dimensional feature frames (80 log-mel filterbank coefficients plus 3 pitch features). Additionally, we augment the features with specaugment, with mask parameters (mT, mF, T, F) = (5, 2, 27, 0.05) and bi-cubic time-warping.

Models: the conformer encoder consists of 12 blocks, each with 2048 feed-forward dimensions, 256 attention dimensions, and 4 attention heads. The transformer decoder has 6 blocks with configurations similar to the encoder. We combine 2622 Mandarin characters with 3000 English BPE units for **In domain** scenario. As for the **Zero-shot** scenario we use a shared Arabic-English vocabulary of size 5000 BPE. Our training configuration utilizes Adam optimizer with a learning rate of 0.001, warmup-steps of 25K, a dropout-rate of 0.1 and 40 epochs. We use joint training with hybrid CTC/attention by setting CTC weight α , Eq 2, to 0.3. During inference, we use a beam size of 10 with no length penalty. For the language model (LM), we train a long short term memory (LSTM) with 4 layers, each of 2048 dimensions, over 20 epochs. When integrating LM with E2E-ASR, we apply an LM weight of 0.2.

4. RESULTS AND ANALYSIS

4.1. In-domain CS text

We examine the impact of augmenting data with generated CS speech from monolingual data, particularly by integrating in-domain CS text. The results, presented in Table 2, are based on the SEAME evaluation. The results from *Mono*, obtained by training on monolingual Chinese and English data, act as our baseline. A shallow fusion with a *SEAME-LM*, trained on SEAME text data, results in a marginal relative reduction: up to 2% in MER. However, simple CS augmentation using unigram units yields up to 15.3% relative reductions in MER, compared to *Mono*. By further enhancing the audio quality of the generated data, we achieve an overall relative improvement of up to 34.4% in MER compared to the *Mono*. Finally comparing our best results to ASR trained on SEAME, the

Table 1. Comparison of the CER/WER/MER results on SEAME. CS: generated CS using in-domain SEAME text. Mono: baseline trained on monolingual data, (Unigram, Bigram): generated CS using (unigram, bigram) units, SE: signal enhancement from §2, SEAME-ASR: topline model trained on SEAME.

Model	DevMan			DevSge		
	CER-MAN	WER-EN	MER	CER-MAN	WER-EN	MER
Mono	37.2	67.4	32.9	56.7	47.5	38.4
+ SEAME-LM	36.4	65.9	32.2	55.2	46.5	37.6
+ CS-Unigram	31.5	55.3	28.4	47.5	42.2	34.4
+ CS-Unigram-SE	29.7	53.7	27.2	44.0	40.9	33.0
+ CS-Bigram-SE	27.2	47.9	25.4	39.7	38.1	31.4
SEAME-ASR (topline)	15.1	28.8	16.5	21.7	28.7	23.5

³API access available from https://mt.qcri.org/api

Table 2. Comparison of the CER/WER results on ESCWA. CS: data generated using synthetic CS text. Mono: baseline trained on monolingual data, (Unigram, Bigram): generated CS using (unigram, bigram) units, SE: signal enhancement from §2

Model	MGB-2		TED3		ESCWA	
	CER	WER	CER	WER	CER	WER
Mono	6.1	12.9	4.4	8.5	31.1	48.7
+ CS-LM	6.3	12.5	4.6	8.7	38.0	57.0
+ CS-Unigram	6.9	14.6	5.2	10.1	24.0	42.7
+ CS-Unigram-SE	7.0	14.7	10.4	5.4	23.1	42.0
+ CS-Bigram-SE	7.0	14.7	10.2	5.2	22.5	40.8

absolute gap is up to 8.9% MER. Given that we utilize SEAME text for data generation, this gap can be attributed to audio mismatches. Thus, we anticipate that further enhancements in audio quality to align with SEAME will bridge this gap.

4.2. Zero-shot CS

We investigate the effects of augmenting the dataset with CS speech, generated from monolingual data and synthetic CS text. This synthetic CS text is produced from the monolingual Arabic MGB-2 and English Tedlium3 datasets, as described in §2.4. Our evaluations, detailed in Table 2, utilize the ESCWA dataset. Operating under our assumption that we do not have access to real CS data, we use the merged evaluation sets from MGB-2 and Tedlium3 to select the best epochs for the model. The observations align with those from §4.1: the CS-Unigram yields relative reductions of 12.3% in WER and 22.8% in CER. Interestingly, the results from shallow fusion with Mono + CS-LM consistently underperform when compared to Mono. Moreover, enhancing the quality of the generated audio further improves results, leading to an overall relative improvement of 16.2% in WER and 27.6% in CER compared to Mono. It's noteworthy that, on monolingual data, performance deteriorates with CS augmentation. This suggests model bias towards code-switching and a reduced inclination for monolingual data. We further analyze this observation in §4.4.



Fig. 2. WER/MER at different percentages of generated CS data where 0%: represents Monolingual, 100%: represents Monolingual with all generated CS.

Table 3. Comparison of the average CMI. Mono: baseline trained on monolingual data, SE: Signal enhancement from §2, Ref: reference, (Uni, Bi): generated CS using (unigram, bigram) units.

Dataset	Ref	Mono	CS-Uni	CS-Uni-SE	Bi-SE
ESCWA	15.6	8.7	10.6	11.6	10.5
SEAME	10.4	3.3	5.4	6.2	7.3

4.3. Generated CS data size

We explore the impact of the amount of generated CS data size on ASR system performance. Figure 2 illustrates the WER at different percentages of generated CS data. In this experiment, we generated CS data with bigrams at 10%, 50%, and 100%. The 0% represents monolingual condition, while 100% corresponds to 80 hours for Arabic-English and 62.2 hours for Chinese-English. It can be observed that there is a substantial improvement when using 10% of generated CS data. However, as the percentage of generated CS data increases, the rate of improvement decreases. This suggests that with more data, further gains can be expected, albeit at a diminishing rate.

4.4. Analysis

To understand the effect of our proposed CS augmentation, we measure the average CMI. Notably, the conventional CMI doesn't account for the accuracy of the sentence. To address this, we select predictions that closely align with the reference using a WER with heuristic threshold set at $\leq 20\%$. It can be observed from Table 3, that employing CS data augmentation consistently elevates the CMI. This affirms our assumption that CS augmentation enhances the model's aptitude for code-switching.

5. CONCLUSION

We introduced a framework that generates synthetic code-switched data from monolingual corpora. Our findings demonstrate that integrating this CS data augmentation yields substantial improvements that surpass results from training exclusively on monolingual sources or simply combining with a code-switched language model. The enhancement of the generated audio's quality further improves the performance. Additionally, in a zero-shot learning scenario, our CS augmentation is superior to solely monolingual training. Finally, we show that improvements from using CS data augmentation stem from the model's increased propensity for code-switching and a decreased bias towards monolingual input.

6. ACKNOWLEDGEMENTS

This work was carried out during the 2022 Jelinek Memorial Summer Workshop on Speech and Language Technologies at Johns Hopkins University, which was supported by Amazon, Kanari AI, Microsoft and Google. This work was also partially supported by NSF CCRI Grant No 2120435.

7. REFERENCES

- C. M. Scotton, "The possibility of code-switching: Motivation for maintaining multilingualism," *Anthropological linguistics*, pp. 432– 444, 1982.
- [2] S. Sitaram *et al.*, "A survey of code-switched speech and language processing," *ArXiv preprint*, vol. abs/1904.00784, 2019.
- [3] K. Taneja *et al.*, "Exploiting monolingual speech corpora for codemixed speech recognition," in *Proc. Interspeech*, G. Kubin and Z. Kacic, Eds., 2019, pp. 2150–2154.
- [4] G. Liu and L. Cao, "Code-switch speech rescoring with monolingual data," in *Proc. ICASSP*, 2021, pp. 6229–6233.
- [5] S. Chuang, T. Sung, and H. Lee, "Training code-switching language model with monolingual data," in *Proc. ICASSP*, 2020, pp. 7949– 7953.
- [6] S. Shah *et al.*, "Learning to recognize code-switched speech without forgetting monolingual speech recognition," *ArXiv preprint*, vol. abs/2006.00782, 2020.
- [7] S. A. Chowdhury *et al.*, "Towards one model to rule all: Multilingual strategy for dialectal code-switching arabic ASR," in *Proc. Interspeech*, H. Hermansky *et al.*, Eds., 2021, pp. 2466–2470.
- [8] A. Ali *et al.*, "Arabic code-switching speech recognition using monolingual data," in *Proc. Interspeech*, H. Hermansky *et al.*, Eds., 2021, pp. 3475–3479.
- [9] X. Zhou *et al.*, "Multi-encoder-decoder transformer for code-switching speech recognition," *ArXiv preprint*, vol. abs/2006.10414, 2020.
- [10] B. Yan *et al.*, "Joint modeling of code-switched and monolingual asr via conditional factorization," in *Proc. ICASSP*, 2022, pp. 6412–6416.
- B. Yan *et al.*, "Towards zero-shot code-switched speech recognition," in *Proc. ICASSP*, 2023, pp. 1–5.
- [12] A. Hussein *et al.*, "Balanced end-to-end monolingual pre-training for low-resourced indic languages code-switching speech recognition," *arXiv e-prints*, arXiv–2106, 2021.
- [13] A. Hussein *et al.*, "Textual data augmentation for arabic-english codeswitching speech recognition," in *Proc. SLT*, 2023, pp. 777–784.
- [14] H. Seki et al., "An end-to-end language-tracking speech recognizer for mixed-language speech," in *Proc. ICASSP*, 2018, pp. 4919–4923.
- [15] S. Nakayama et al., "Japanese-english code-switching speech data construction," in 2018 Oriental COCOSDA - International Conference on Speech Database and Assessments, 2018, pp. 67–71.
- [16] Y. Sharma *et al.*, "Improving low resource code-switched ASR using augmented code-switched TTS," in *Proc. Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds., 2020, pp. 4771–4775.
- [17] S. Murthy, D. Sitaram, and S. Sitaram, "Effect of TTS generated audio on OOV detection and word error rate in ASR for low-resource languages," in *Proc. Interspeech*, B. Yegnanarayana, Ed., 2018, pp. 1026–1030.
- [18] C. Peyser *et al.*, "Improving performance of end-to-end ASR on numeric sequences," in *Proc. Interspeech*, G. Kubin and Z. Kacic, Eds., 2019, pp. 2185–2189.
- [19] A. Rosenberg *et al.*, "Speech recognition with augmented synthesized speech," in ASRU, 2019.
- [20] G. Wang *et al.*, "Improving speech recognition using consistent predictions on synthesized speech," in *Proc. ICASSP*, 2020, pp. 7029– 7033.
- [21] H. Yu et al., "Code-switching text generation and injection in mandarin-english asr," in Proc. ICASSP, 2023, pp. 1–5.
- [22] T. K. Lam *et al.*, "On-the-fly aligned data augmentation for sequenceto-sequence ASR," in *Proc. Interspeech*, H. Hermansky *et al.*, Eds., 2021, pp. 1299–1303.
- [23] R. Zhao et al., "On addressing practical challenges for rnn-transducer," in Proc. ASRU, 2021, pp. 526–533.

- [24] A. Hunt *et al.*, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, vol. 3, 1996, pp. 233–258.
- [25] J. Wouters and M. W. Macon, "Control of spectral dynamics in concatenative speech synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 1, pp. 30–38, 2001.
- [26] R. A. Khan and J. Chitode, "Concatenative speech synthesis: A review," *International Journal of Computer Applications*, vol. 136, no. 3, pp. 1–6, 2016.
- [27] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.
- [28] P. Żelasko *et al.*, "Lhotse: A speech data representation library for the modern deep learning ecosystem," *ArXiv preprint*, vol. abs/2110.12561, 2021.
- [29] A. W. Black and P. A. Taylor, "Automatically clustering similar units for unit selection in speech synthesis.," 1997.
- [30] Z.-Y. Dou and G. Neubig, "Word alignment by fine-tuning embeddings on parallel corpora," in *Proc. ACL*, 2021, pp. 2112–2128.
- [31] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds., 2020, pp. 5036–5040.
- [32] P. Guo *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *Proc. ICASSP*, 2021, pp. 5874–5878.
- [33] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP*, 2018, pp. 66–71.
- [34] A. Graves et al., "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, W. W. Cohen and A. W. Moore, Eds., ser. ACM International Conference Proceeding Series, vol. 148, 2006, pp. 369–376.
- [35] B. Gambäck and A. Das, "Comparing the level of code-switching in corpora," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 1850– 1855.
- [36] D.-C. Lyu *et al.*, "Seame: A mandarin-english code-switching speech corpus in south-east asia," in *Proc. Interspeech*, 2010.
- [37] H. Bu et al., "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in 2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA), 2017, pp. 1–5.
- [38] F. Hernandez et al., "Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20, 2018, pp. 198– 208.
- [39] A. Ali *et al.*, "The MGB-2 challenge: Arabic multi-dialect broadcast media recognition," in *SLT*, 2016.