INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

# Text Only Domain Adaptation with Phoneme Guided Data Splicing for End-to-End Speech Recognition

*Wei Wang, Xun Gong, Hang Shao, Dongning Yang, Yanmin Qian*[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

## Abstract

Adaptation of end-to-end (E2E) automatic speech recognition (ASR) models to unseen domains remains a challenge due to their monolithic construction, which typically necessitates paired data for customization. While neural text-to-speech (TTS) approaches have shown effectiveness for domain adaptation, they come with the drawback of high computational costs during training and inference. In this paper, we propose a model-free audio synthesis pipeline for domain adaptation, which synthesizes audio with text from the target domain and audio pieces from the source domain, allowing ASR models to be adapted with the on-the-fly synthesized audio. Additionally, we apply layer-wise regularization between speech encodings generated by adapted and unadapted models to prevent overfitting. Our experiments adapt from LIBRISPEECH to various domains in GIGASPEECH. The results show a 15-30% relative improvement in target domains compared to shallow fusion, with almost no degradation in the source domain.

**Index Terms**: end-to-end speech recognition, domain adaptation, text to speech, splicing data generation

## 1. Introduction

The field of automatic speech recognition (ASR) has witnessed significant progress with the emergence of end-to-end (E2E) models, which offer a unified and jointly optimized architecture for ASR tasks [1, 2]. In contrast, traditional hybrid systems require multiple separately optimized models, including an acoustic model, a language model (LM), and a pronunciation model [3, 4]. This leads to a complicated pipeline during both training and inference. Despite these advantages, customizing E2E models to new domains remains a challenging task.

Domain adaptation is a crucial topic in ASR that aims to adapt well-trained ASR systems to new domains [5–8]. Recent research has focused on domain adaptation using unpaired text data, which is more practical to collect than speech-text paired data. Modularized hybrid systems can leverage text-only data for customization, whereas E2E models require paired data for training and have limited capacity to exploit unpaired text data. Nonetheless, researchers have recently developed several approaches to address this shortcoming and utilize unpaired text data to customize E2E models.

LM shallow fusion [9] is a commonly adopted approach for domain adaptation where an LM is trained with large text corpora from the target domain and fused with E2E models through score interpolation during inference. Nevertheless, LM fusion based methods often achieve promising results on the target domain at the cost of severe degradation on the source domain.

Another straightforward solution is to synthesize speech from texts in the target domain with text-to-speech (TTS) techniques [10–17]. In [10], a multi-speaker neural TTS model is trained to synthesize speech using the unpaired text data to adapt the RNN-T model to the target domain. [11] compares the different neural TTS models, showing that the diversity of generated speech is crucial for ASR model customization. [13] explores the machine speech chain [12] framework to adapt both TTS and ASR models from the audiobook domain to the presentation domain alternately. [14, 15] improve the recognition accuracy of out-of-vocabulary (OOV) words by training with audio generated from text data containing OOV words. Despite the promising results achieved by exploiting neural TTS models for the customization of ASR models, neural TTS models are computationally expensive during both training and speech generation. Moreover, the speaker variation of generated speech is limited compared with real training data for ASR [18].

To resolve the drawbacks of neural TTS approaches for domain adaptation, [19] proposes splicing data generation (SDG) that concatenates the sampled speech segments corresponding to words in target texts into new utterances. Although the spliced speech suffers from apparent disfluency, the adapted ASR model shows promising improvement on the target domain. In this work, we elaborate the word guided SDG (Word SDG) pipeline in [19] and propose phoneme guided SDG (Phoneme SDG). Instead of using word-level speech segments as in [19], we adopt speech segments guided by phoneme n-grams where $n$ is dynamically determined by a greedy algorithm that minimizes the number of splicing fragments for target texts. The proposed Phoneme SDG scheme improves Word SDG in several aspects: (1) The generated speech includes real word connections or liaison which is absent in Word SDG. (2) Fluency is improved by minimizing the number of speech segments in generated speech. (3) The diversity of speech segments is enriched since the number of phoneme n-grams is much richer than words. Besides, during adaptation of ASR models, encoder freezing strategy is often adopted to prevent overfitting on synthesized data [14, 19]. However, such a strategy reduces the number of trainable parameters and might cause negative impacts to the adaptation results. To avoid the drawbacks of encoder freezing, a regularization term that is similar to [20] is introduced in a layer-wise manner. We validate the effectiveness of the proposed methods with the attention based encoder-decoder (AED) ASR architecture [21, 22].

Our contributions can be summarized as follows: (1) we propose a splicing data generation pipeline that enables the adaptation of well-trained ASR models on on-the-fly synthesized speech. (2) Performance of ASR model adapted with spliced speech surpasses neural TTS synthesized speech in terms of WER on the target domain. (3) We introduce a layer-

[†] corresponding author


10.21437/Interspeech.2023-1349

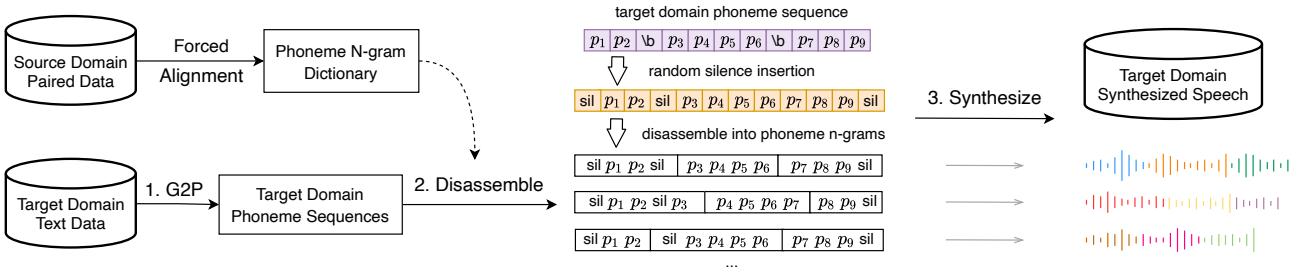


Figure 1: *Proposed splicing data generation pipeline guided by n-grame phonemes:* $p_1, p_2, \cdots$ *are phonemes derived from original word sequences by querying lexicon,* $\backslash b$ *indicates word boundary. The dashed line means disassembling target domain phoneme sequences requires querying phoneme n-gram dictionary.*

wise regularization term during ASR model adaptation that shows better results on both the source domain and target domains than the widely adopted encoder freezing strategy. (4) We setup a benchmark for domain adaptation from LIBRISPEECH to multiple domains of GIGASPEECH. The effectiveness of our proposed methods is further demonstrated on this benchmark.

## 2. Methodology

In order to address disfluency and enhance the diversity of synthesized speech generated by Word SDG, we propose a Phoneme SDG pipeline. To prevent overfitting on synthesized speech, we introduce a layer-wise distance regularization technique between speech encodings produced by the adapted and unadapted models. In this section, we provide a detailed explanation of these methods.

### 2.1. Phoneme Guided Splicing Data Generation

The Phoneme SDG pipeline requires the construction of a phoneme n-gram dictionary, which is effectively a mapping from phoneme n-grams to their corresponding speech segments in source domain training data. In this work, we build the dictionary with $3 \leq n \leq 10$ by processing the forced alignment results. The dictionary is denoted as $\mathbb{P}$ and the set of its keys is denoted as $\mathbb{S}$. As shown in Fig. 1, the proposed pipeline is accomplished with 3 stages.

#### *2.1.1. grapheme to phoneme (g2p)*

In this stage, word sequences in target texts are converted to phoneme sequences by querying the word lexicon. For words with multiple entries in the lexicon (i.e. heteronyms), we randomly select one of the entries during each synthesis. Word boundaries are preserved and marked as $\backslash b$ for subsequent silence insertion. The outcome phoneme sequence after g2p is illustrated in lilac in Fig. 1.

#### *2.1.2. disassemble into phoneme n-grams*

In the forced alignment results of real speech data, there exists occasional silence between words and constant silence at both ends of all utterances. To imitate real speech data, we also append silence at both ends of the phoneme sequences. Besides, word boundaries are randomly removed or replaced with silence according to the statistics in the forced alignment results. The resulting phoneme sequence after random silence insertion is illustrated in orange in Fig. 1.

Then we search the dictionary $\mathbb{P}$ to find phoneme n-grams with maximized averaged length (i.e. minimized number of phoneme n-grams) that assemble to a phoneme sequence. This is achieved with a greedy algorithm demonstrated in Algorithm 1 that operates in a divide-and-conquer manner. Note that the symbol $\times$ in line 13 denotes Cartesian Product. It can be proved by mathematical induction that Algorithm 1 always returns sequences comprised of a minimum number of phoneme n-grams if such a sequence exists, which well preserves the fluency of synthesized audio. We randomly take 10 disassembled sequences if more are returned and discard the input sequences that cannot be disassembled with such a procedure.

#### *2.1.3. audio synthesis*

Finally, we convert phoneme n-grams into actual speech segments by randomly selecting one of the speech segments corresponding to each phoneme n-gram from the dictionary $\mathbb{P}$. The speech segments are then concatenated into complete speech.

An example that converts a target domain text to a phoneme n-gram sequence is demonstrated in Table 1. Note that the liaison in the target text 'SWIM AGAIN' is reflected in the highlighted part of disassembled phoneme n-grams. The number of tokens is reduced from 7 (i.e. number of words in Word SDG) to 5, which also improves fluency.

**Algorithm 1** Disassemble a phoneme sequence into n-grams

**Input:** $x$, the phoneme sequence
**Output:** $y$, the list of disassembled n-gram sequences
**Require:** $\mathbb{S}$, the set of all phoneme n-grams in the dictionary $\mathbb{P}$
**Require:** $n_{min}, n_{max}, n_{min} \leq n \leq n_{max}$ for $\mathbb{S}$

```
1:  function DISASSEMBLE-SEQUENCE(x)
2:      if length(x) = 0 then
3:          return []
4:      y ← []
5:      for n ← n_max to n_min do
6:          for i ← 0 to length(x) − n do
7:              t ← x[i : i + n]
8:              if t ∈ S then
9:                  y_pre ← DISASSEMBLE-SEQUENCE(x[0 : i])
10:                 y_post ← DISASSEMBLE-SEQUENCE(x[i + n :])
11:                 if y_pre is ∅ or y_post is ∅ then
12:                     continue
13:                 y.Append( y_pre × [t] × y_post )
14:         if length(y) ≠ 0 then
15:             return y
16:     return ∅
```

### 2.2. Layer-wise Encoding Distance Regularization (LEDR)

It has been shown in previous works that encoder freezing is an effective approach to prevent the model from overfitting on

Table 1: *Phoneme SDG pipeline demonstrated with text from utterance YOU0000001275_S0000060 in* GIGASPEECH

| | |
|---|---|
| target text | UM LIKE GREAT I'LL NEVER SWIM AGAIN |
| g2p | AH1 M \b L AY1 K \b G R EY1 T \b AY1 L \b N EH1 V ER0 \b S W IH1 M \b AH0 G EH1 N |
| insert silence | SIL AH1 M L AY1 K SIL G R EY1 T AY1 L N EH1 V ER0 S W IH1 M AH0 G EH1 N SIL |
| dis-assemble | (SIL AH1 M L AY1 K), (SIL G R EY1 T), (AY1 L N EH1 V ER0), (S W IH1), (M AH0 G EH1 N SIL) |

synthesized data [14, 19]. However, due to the reduced trainable parameters, adapting a well-trained model with a frozen encoder to new domains can be more challenging.

In this work, instead of freezing the encoders, we incorporate a regularization term into the ASR loss that is similar to [20] for each real speech sample $\mathbf{x}$. The regularization term penalizes $L_1$ and cosine distance between the normalized real speech encodings produced by adapted model $\phi_l(x)$ and unadapted model $\phi'_l(x)$ at layer $l$. Different from [20] where speech encodings come from clean-noisy speech pairs encoded by the same model, the speech encodings in Eq (1) are produced by adapted and unadapted models using the same real speech input:

$$\mathcal{L}_d\left(\mathbf{x};\theta,\theta'\right)=\sum_{l=1}^{L}\Bigg(\left\|\phi_l(\mathbf{x})-\phi'_l(\mathbf{x})\right\| + \frac{\phi_l(\mathbf{x})\cdot\phi'_l(\mathbf{x})}{\|\phi_l(\mathbf{x})\|\cdot\|\phi'_l(\mathbf{x})\|}\Bigg) \tag{1}$$

where $L$ is the total number of encoder layers, $\theta$ and $\theta'$ are parameters of adapted and unadapted models, respectively.

We adopt the joint CTC/attention training framework [22], where the multi-task learning based ASR loss is denoted as $L_{joint}$. The final loss is formulated as:

$$\mathcal{L}=\begin{cases}\mathcal{L}_{joint}(\mathbf{x};\theta)+\alpha\mathcal{L}_d(\mathbf{x};\theta,\theta'), & \text{if } \mathbf{x} \text{ is real speech}\\ \mathcal{L}_{joint}(\mathbf{x};\theta), & \text{if } \mathbf{x} \text{ is synthetic speech}\end{cases} \tag{2}$$

where $\alpha$ is the weight of the regularization term.

# 3. Experiments

## 3.1. Experiment Setup

### *3.1.1. Data*

We conduct experiments by adapting the ASR model trained on LIBRISPEECH [23] to a variety of domains in GIGASPEECH [24]. GIGASPEECH is a recently published ASR corpus comprised of 10,000 hours of transcribed speech. In this work, we use the YouTube partition of GIGASPEECH XL subset. 4 different domains with a comparable amount of data are selected as target domains. 5 hours development set and 10 hours test set are split from training data as shown in Table 2. For all experiments except the upper bound, only text data in target domains are used for audio synthesis and model training. The phoneme n-gram dictionary $\mathbb{P}$ in Section 2.1 is constructed with the forced alignment results obtained from a Chain Time Delayed Neural Network (TDNN) model [25, 26].

Table 2: *Duration of* GIGASPEECH *target domains (hours)*

| domain | train | dev | test |
|---|---|---|---|
| Science | 313.93 | 5.73 | 9.30 |
| News | 358.78 | 5.90 | 9.29 |
| People | 383.78 | 4.27 | 10.87 |
| Entertainment | 284.64 | 4.71 | 10.31 |

### *3.1.2. ASR Model*

The source domain ASR model is trained on the full 960 hours of LIBRISPEECH data. We adopt 12 layers of Conformer speech encoder [27] and 6 layers of Transformer decoder with 2048 hidden units. Each layer is equipped with 8 heads of 64 dimension self-attention layer [28]. The kernel size for convolution modules is 31. For joint-CTC-attention training, the weight for CTC and attention is set to 0.3 and 0.7 empirically. The weight $\alpha$ in Eq (2) is set to 150 for most experiments and we also investigate the impact of $\alpha$ in Section 3.2.3. We use an 80-dimensional log Mel-filterbank with a 25ms window length computed every 10ms as inputs of the speech encoder. Specaugment [29] is applied for all experiments. During adaptation, the model is trained on the real speech from the source domain and synthetic speech generated from texts in the target domain alternately for each batch. Encoders are frozen during adaptation unless LEDR is applied. The Adam [30] optimizer is adopted with 0.001 initial learning rate and 20,000 warmup steps. We also adopt the joint-CTC-attention decoding strategy [31]. The weight for CTC and attention during inference is set to 0.2 and 0.8, which was tuned to achieve the best decoding results on LIBRISPEECH development sets. The number of modeling units (BPE) for text sequence in the decoder is 10,000 [32]. The LM applied in shallow fusion are Transformer LMs trained on corresponding texts in target domains following official ESPnet setup[1]. The source domain model was trained for 150 epochs and fine-tuned for 70 epochs on each target domain. Experiments are carried out with ESPnet toolkit [33]

### *3.1.3. Neural TTS Model*

We adopt a single-speaker and multi-speaker neural TTS systems for comparison. Both systems are comprised of a FastSpeech2 [34] acoustic encoder and a HiFi-GAN vocoder [35]. The single and multi-speaker TTS systems are pretrained with LJSPEECH [36] and LIBRITTS [37], respectively. We follow the ESPnet pipeline[2], where models and details regarding audio synthesis can be found.

## 3.2. Experiment Results and analysis

The performance for different system setups is shown in Table 3. In the first line, the unadapted model trained on LIBRISPEECH is tested on 4 target domains of GIGASPEECH. The last line shows the upper bound performance on target domains by training ASR models with paired target domain data. Although shallow fusion yields promising improvement on target domains by incorporating an external LM, adapting the ASR model on synthesized speech-text pairs achieves better performance. Besides, by training on a mixture of source domain real speech and target domain synthesized speech, performance degradation on the source domain can be restrained.

[1] https://zenodo.org/record/3966501/

[2] https://colab.research.google.com/github/espnet/notebook/blob/master/espnet2_tts_realtime_demo.ipynb

Table 3: *WER (%) comparison of different setups. The left results in each column show WERs on* LIBRISPEECH *test clean / test other sets. The right results in each column show WERs on* GIGASPEECH *dev / test sets for the corresponding domain.*

| Method | Libri → Giga Science | | Libri → Giga News | | Libri → Giga People | | Libri → Giga Entertainment | |
|---|---|---|---|---|---|---|---|---|
| | clean / other | dev / test | clean / other | dev / test | clean / other | dev / test | clean / other | dev / test |
| 1. Unadapted Model | 2.5 / 5.4 | 17.1 / 19.0 | 2.5 / 5.4 | 18.1 / 16.8 | 2.5 / 5.4 | 23.3 / 17.7 | 2.5 / 5.4 | 24.1 / 24.1 |
| 2. + SF | 2.9 / 6.5 | 15.3 / 16.4 | 2.9 / 6.4 | 16.0 / 14.8 | 3.0 / 6.3 | 21.6 / 16.2 | 3.0 / 6.2 | 23.3 / 22.9 |
| 3. Multi-spkr neural TTS | 2.6 / 5.9 | 11.6 / 12.9 | 2.6 / 5.9 | 13.9 / 12.3 | 2.6 / 5.7 | 17.4 / 13.4 | 2.9 / 5.9 | 20.2 / 20.7 |
| 4. Single-spkr neural TTS | 2.7 / 5.7 | 12.2 / 13.3 | 2.7 / 5.7 | 14.1 / 12.4 | 2.6 / 5.6 | 18.2 / 13.7 | 2.8 / 5.7 | 20.4 / 20.8 |
| 5. Word SDG | 2.5 / 5.6 | 12.6 / 14.0 | 2.6 / 5.9 | 14.1 / 12.8 | 2.6 / 5.5 | 19.7 / 14.7 | 2.4 / 5.6 | 21.5 / 21.6 |
| 6. Phoneme SDG | 2.5 / 5.8 | 11.4 / 12.1 | 2.6 / 5.6 | 13.5 / 11.9 | 2.6 / 5.6 | 17.8 / 13.4 | 2.4 / 5.5 | 20.7 / 20.6 |
| 7. + LEDR | 2.5 / 5.6 | 10.8 / 11.7 | 2.5 / 5.6 | 12.9 / 11.6 | 2.5 / 5.5 | 17.5 / 13.2 | 2.4 / 5.4 | **19.6** / 19.4 |
| 8. ++ SF | 2.8 / 6.2 | **10.6 / 11.4** | 2.8 / 6.2 | **12.6 / 11.3** | 2.8 / 6.0 | **17.2 / 13.1** | 2.6 / 5.9 | **19.6 / 19.3** |
| 9. Upper bound | - | 7.6 / 7.7 | - | 8.8 / 7.3 | - | 11.4 / 8.5 | - | 13.6 / 13.7 |

Comparing the third and fourth rows, adapting models on multi-speaker TTS data constantly yields better results than on single-speaker TTS data, showing that speaker diversity plays an important role in neural TTS based text-only domain adaptation. Although the quality of synthesized speech can be slightly worse for multi-speaker TTS than single-speaker TTS, the synthesized speech is richer in speaker diversity, which prevents the adapted model from overfitting to a single speaker.

*3.2.1. splicing data generation*

We compared the proposed Phoneme SDG with Word SDG. In our experiments, Word SDG does not surpass neural TTS approaches, which is inconsistent with the results in [19]. We assume that this is caused by the reduction of source domain data from 65,000 hours in [19] to 960 hours in LIBRISPEECH, leading to a significantly degraded diversity of word guided speech segments. This defect is mitigated in the proposed Phoneme SDG pipeline since phoneme n-grams are significantly richer than words. Besides, the proposed pipeline also enables different ways of disassembling target texts into phoneme n-grams rather than always disassembling them into underlying words. Compared to Word SDG, results on Phoneme SDG show consistent improvement on target domains. Moreover, most adapted models with Phoneme SDG yield better results on the source domain than neural TTS approaches, and show similar or even better performance (i.e. Libri → Giga Entertainment) on the source domain than unadapted models. We attribute this to the fact that synthesized speech from SDG is comprised of speech segments from the source domain, and that SDG approaches significantly increase the diversity of training data with on-the-fly data generation.

*3.2.2. layer-wise encoding distance regularization*

The seventh row shows the results by replacing the encoder freezing strategy with the layer-wise regularization term in Eq (1). The overall performance is improved on both the source domain and target domains since the number of trainable parameters is increased by unfreezing the encoder and the overfitting problem can be prevented with the regularization term.

Models adapted with the proposed methods achieve the largest relative improvement compared to unadapted models on GIGASPEECH Science dev / test sets (i.e. 36.3% / 40.0%), which can probably be ascribed to the large domain bias of text data in the science domain. The smallest relative improvement is achieved on Entertainment dev / test sets (i.e. 18.7% / 19.9%), which can be explained by the shortage of text data in Entertainment domain compared with other domains.

We further validate the compatibility of the proposed methods with shallow fusion. Results show that the performance on target domains can be further improved with LM shallow fusion by sacrificing recognition accuracy on the source domain.

*3.2.3. Investigation on weight of regularization term*

Table 4: *WER (%) comparison of different weights $\alpha$ in proposed layer-wise encoding distance regularization*

| Method | $\alpha$ | Librispeech | | Giga Science | |
|---|---|---|---|---|---|
| | | clean | other | dev | test |
| Phoneme SDG + LEDR | 50 | 2.7 | 6.0 | 12.3 | 12.8 |
| | 150 | 2.5 | 5.6 | 10.8 | 11.7 |
| | 300 | 2.5 | 5.7 | 10.9 | 11.7 |
| | 500 | 2.5 | 5.7 | 11.2 | 11.8 |

In the last experiments, we investigate the impact of the weight $\alpha$ of the regularization term in Eq (2) and the results are shown in Table 4. When $\alpha$ is small ($\alpha = 50$), the regularization is too weak to prevent the adapted model from overfitting to synthesized speech. We achieve the best results by increasing $\alpha$ to 150. As $\alpha$ becomes larger ($\alpha = 300, 500$), the regularization term is more likely to freeze the encoder and the results become closer to the sixth row of Table 2 which adopts the encoder freezing strategy.

## 4. Conclusions

This paper introduces a novel approach for text-only domain adaptation that utilizes data splicing pipeline guided by phoneme n-grams to generate speech from texts in target domains. Due to the model-free nature of the proposed pipeline, it has negligible computational cost compared to neural TTS approaches, allowing the ASR model to be trained with on-the-fly synthesized speech. Moreover, a layer-wise regularization term is introduced to prevent the ASR model from overfitting to synthesized speech. We validate the effectiveness of the proposed methods by adapting a well-trained model on LIBRISPEECH to 4 different domains in the YouTube partition of GIGASPEECH XL subset. Results show around 15% to 30% relative WER reduction on test sets from target domains, while almost without deterioration on test sets from the source domain. By combining the proposed methods with LM shallow fusion, performance on target domains can be further improved at the cost of minor WER degradation on the source domain. Overall, our proposed methods offer a promising solution for improving ASR performance in text-only domain adaptation scenarios

## 5. Acknowledgements

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

# 6. References


[1] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," *Interspeech 2018*, Sep 2018.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE ICASSP*. IEEE, 2016, pp. 4960–4964.

[3] F. Zhang, Y. Wang, X. Zhang, C. Liu, Y. Saraf, and G. Zweig, "Faster, simpler and more accurate hybrid asr systems using wordpieces," 10 2020, pp. 976–980.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohman, F. Beaufays, and Y. He, "Large-scale asr domain adaptation using self- and semi-supervised learning," in *ICASSP 2022 IEEE*, 2022, pp. 6627–6631.

[6] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," *2018 IEEE ICASSP*, pp. 5949–5953, 2018.

[7] X. Gong, Y. Lu, Z. Zhou, and Y. Qian, "Layer-wise fast adaptation for end-to-end multi-accent speech recognition," in *Proc. Interspeech 2021*. Proc. Interspeech 2021, 2021, pp. 1274–1278.

[8] T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu, "Cluster adaptive training for deep neural network," in *2015 IEEE ICASSP*, Apr. 2015, pp. 4325–4329.

[9] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *2018 ICASSP*, 2018, pp. 1–5828.

[10] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao, and Y. Gong, "Developing RNN-T Models Surpassing High-Performance Hybrid Models with Customization Capability," in *Proc. Interspeech 2020*, 2020, pp. 3590–3594.

[11] Y. Deng, R. Zhao, Z. Meng, X. Chen, B. Liu, J. Li, Y. Gong, and L. He, "Improving RNN-T for Domain Scaling Using Semi-Supervised Training with Neural TTS," in *Proc. Interspeech 2021*, 2021, pp. 751–755.

[12] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 301–308.

[13] F. Yue, Y. Deng, L. He, and T. Ko, "Exploring machine speech chain for domain adaptation and few-shot speaker adaptation," *ArXiv*, vol. abs/2104.03815, 2021.

[14] 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," *ICASSP, IEEE - Proceedings*, vol. 2021-June, pp. 5674–5678, 2021.

[15] 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," 09 2018, pp. 1026–1030.

[16] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. Moreno, and G. Wang, "Tts4pretrain 2.0: Advancing the use of text and speech in asr pretraining with consistency and contrastive losses," in *ICASSP 2022 IEEE*, 2022, pp. 7677–7681.

[17] W. Wang, Z. Zhou, Y. Lu, H. Wang, C. Du, and Y. Qian, "Towards data selection on tts data for children's speech recognition," in *ICASSP 2021 IEEE*, 2021, pp. 6888–6892.

[18] Y. Deng, L. He, and F. K. Soong, "Modeling multi-speaker latent space to improve neural tts: Quick enrolling new speaker and enhancing premium voice," *ArXiv*, vol. abs/1812.05253, 2018.

[19] R. Zhao, J. Xue, J. Li, W. Wei, L. He, and Y. Gong, "On addressing practical challenges for rnn-transducer," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 526–533.

[20] D. Liang, Z. Huang, and Z. C. Lipton, "Learning noise-invariant representations for robust speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 56–63.

[21] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[22] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE ICASSP*. IEEE, 2017, pp. 4835–4839.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," 04 2015, pp. 5206–5210.

[24] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang, M. Jin, S. Khudanpur, S. Watanabe, S. Zhao, W. Zou, X. Li, X. Yao, Y. Wang, Y. Wang, Z. You, and Z. Yan, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," in *Proc. Interspeech 2021*, 2021.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.

[26] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *Interspeech*, 2016, pp. 2751–2755.

[27] 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," *arXiv preprint arXiv:2005.08100*, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[29] 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," *Interspeech 2019*, Sep 2019.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 518–529.

[32] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," *arXiv preprint arXiv:1804.10959*, 2018.

[33] 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.

[34] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[35] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.

[36] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[37] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," in *Proc. Interspeech*, Sep. 2019.