# SynthASR: Unlocking Synthetic Data for Speech Recognition

*Amin Fazel, Wei Yang, Yulan Liu, Roberto Barra-Chicote, Yixiong Meng, Roland Maas, Jasha Droppo*

Alexa Speech, Amazon.com

{aminfaze, wyanamz, lyulan, rchicote, myixiong, rmaas, drojasha}@amazon.com

## Abstract

End-to-end (E2E) automatic speech recognition (ASR) models have recently demonstrated superior performance over the traditional hybrid ASR models. Training an E2E ASR model requires a large amount of data which is not only expensive but may also raise dependency on production data. At the same time, synthetic speech generated by the state-of-the-art text-to-speech (TTS) engines has advanced to near-human naturalness. In this work, we propose to utilize synthetic speech for ASR training (SynthASR) in applications where data is sparse or hard to get for ASR model training. In addition, we apply continual learning with a novel multi-stage training strategy to address catastrophic forgetting, achieved by a mix of weighted multi-style training, data augmentation, encoder freezing, and parameter regularization. In our experiments conducted on in-house datasets for a new application of recognizing medication names, training ASR RNN-T models with synthetic audio via the proposed multi-stage training improved the recognition performance on new application by more than 65% relative, without degradation on existing general applications. Our observations show that SynthASR holds great promise in training the state-of-the-art large-scale E2E ASR models for new applications while reducing the costs and dependency on production data.

**Index Terms**: speech recognition, data efficient machine learning, synthetic speech

## 1. Introduction

End-to-end (E2E) designs, such as those based on connectionist temporal classification (CTC) [1, 2], recurrent neural network transducer (RNN-T) [3], and Listen, Attend and Spell (LAS) [4], have several advantages over the older hybrid designs for automatic speech recognition (ASR) tasks. These designs jointly optimize the model parameters to improve the accuracy at text level, and they learn the tasks directly from data. The highly integrated model structure in E2E designs reduces the overall model size and simplifies both training and inference, making it more attractive to on-device applications [5].

To perform well in real applications, E2E ASR systems need to be trained on thousands of hours of transcribed speech data. One way to meet this data requirement is to learn directly from transcribed production data. To work with production data, one must be careful to handle the data properly. This includes the methods by which the data is collected, transferred, stored, accessed, and deleted. It also includes minimizing the amount of human exposure to the data, such as when it is transcribed.

The goal of this work is to reduce the reliance on human transcribed data by training ASR models on production-like data synthesized from text by text-to-speech (TTS) engines [6, 7, 8, 9]. The contribution of this work comes in three folds. First, we validate the effectiveness of using TTS based synthetic data as a general approach to reduce reliance on transcribed data. Second, we study how to improve ASR models for new applications without relying on corresponding production data. Third, we propose a multi-stage training strategy, and demonstrate that continual learning via this strategy significantly improves ASR performance on the new applications without degradation on existing applications.

## 2. Related Work

Recent research has made significant progress in using synthetic speech data for ASR model training. In [10], a TTS engine based on Tacotron-2 is used to synthesize audio for new vocabulary to teach an acoustic-to-word speech recognition model new words. In a follow-up work [6], multi-speaker TTS is used to improve the acoustic diversity in synthetic data, where speaker embeddings are added to Tacotron-2 architecture. Later in [7], global style token (GST) based embeddings are introduced to modify a version of Tacotron-2 to further increase the acoustic diversity of synthetic data where GST is found superior to i-vector based embeddings. In addition, this work also demonstrates that adding TTS based synthetic data, LM approaches, and general data augmentation method SpecAugment [11] are mostly independent and complementary. In [8], E2E TTS with speaker presentations from a variational autoencoder (VAE) is explored to increase the acoustic diversity in low-resource data. All these previous works in [6, 7, 8] have shown the benefit of increased acoustic diversity in synthetic data for ASR model improvement, especially in low resource scenarios.

The works in [10, 6, 9] have shown that E2E ASR models can learn new vocabularies from TTS based synthetic data, which is crucial for feature expansion of ASR into new applications. Recent work by [9] pointed out some practical challenges in such vocabulary expansion strategy in terms of learning to recognize new words without recognition degradation on already learned words. Authors have explored several strategies to address this challenge including combining real data with synthetic data with weighted sampling and applying different regularizations on each model components.

Inspired by previous research, this work reduces the dependency of ASR model training on production data by using TTS synthetic data. While a number of existing methods can achieve similar goals, each has their own limitations. For example, in semi-supervised learning (SSL) [12, 13], where speech audio is transcribed by machine rather than human, audio signals are still required to be collected, transferred, and stored in the cloud to generate machine transcribed labels. In federated learning (FL) [14, 15], where many devices collaboratively train a shared global model, comprehensive infrastructure updates are needed, which puts additional cost burdens on the customers for device upgrades in order to support new features.

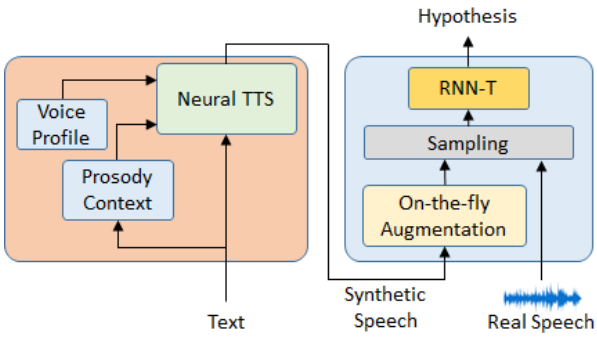


Figure 1: *Diagram of the proposed SynthASR employing TTS synthetic data for ASR training.*

Acknowledging the importance of language model (LM) approaches as alternative methods to utilize text-only data for E2E ASR performance improvement [16, 17, 18], this work focuses on the WER improvement from the first pass ASR model to provide LM approaches a better starting point. In addition, we applied both SpecAugment and classical signal processing based data augmentation methods on TTS synthetic data.

## 3. Technical Approaches

### 3.1. Overview

The schematic diagram of our proposed system is illustrated in Figure 1. It consists of a multi-context TTS engine to generate synthetic speech, and an RNN-T model for speech recognition. The model for TTS engine is trained and evaluated independently from the speech recognition model. RNN-T models are trained in a multi-style training (MST) [19, 20]. With MST, the training data are sampled from a combination of real speech recordings and TTS based synthetic speech audio. The ratio between real recordings and synthetic audio seen during RNN-T training is optimized with sampling weights, as suggested by [9]. This method well mixes the real and synthetic data in each batch so that the ASR model sees both data. In addition, data augmentation is applied at both audio level and feature level for RNN-T training. For one production scale application covered in this work, instead of training RNN-T models from scratch, we adopted continual learning (CL) [21, 22] where an established RNN-T model incrementally learns from data with new information without catastrophic forgetting via the proposed multi-stage training approach.

### 3.2. Multi-context TTS

We use a multi-context TTS to generate clean synthetic speech with diverse speaker and prosody attributes (Figure 2). This system consists of two main modules: a context generation module and a neural vocoder module. The context generation module is an attention-based sequence-to-sequence network [23] that predicts a Mel-spectrogram given an input text. We control the speaker identity with voice profile embeddings, which introduces a bias in the training that makes the reference encoder to be speaker independent. At inference time we guide the TTS with a reference spectrogram generated with a high-fidelity speaker-dependent TTS system trained with more that 20 hours high quality data. This reference spectrogram provides a natural prosodic contour to the attention module. Since the reference encoder has been trained following a variational autoencoder (VAE) approach, where we learn the posterior distributions over the prosody latent space given the reference spectrogram, we later can sample from the posteriors to generate different speech realizations of the same text. This one-to-many capability increases the inter- and intra- speaker variability, which is crucial for ASR training as it increases the speaker diversity of synthetic data. The neural vocoder module consists of the architecture similar to the universal vocoder described in [24] but without any VAE reference. This Universal Neural Vocoder (UNV) was pretrained with more than 100 hours from more than 100 speakers in 27 languages from a proprietary database of paid voice actors. The UNV synthesizes speech audio out of the Mel-spectrograms generated by the first module.

A speaker verification system is used to produce the voice profile embeddings for the context generation module. The speaker verification system was trained on an internal Amazon data. This speaker verification model uses the architecture introduced in [25]. This pre-trained speaker verification system is used to provide voice profile embeddings in the Multi-context TTS system as in Figure 2. We first generate the speaker embedding for each utterance of training data in TTS system, then we average all utterance-level embeddings from a given speaker as voice profile embedding for this speaker.

### 3.3. Data augmentation

Reverberation is introduced to TTS synthetic audio by convolving the audio with an acoustic impulse response (AIR) randomly selected from a pool of 10,000 available AIRs estimated from chirp signal measurements in real rooms. Then an audio segment randomly sampled from an in-house dataset is added on top as background noise, with an SNR ranging from 10 to 20 dB. To increase the acoustic diversity, for each on-the-fly audio corruption, there is a 60% probability of reverberation and an independently 60% probability of noise addition. This provides a mixture of clean synthetic speech, synthetic speech with reverberation only, synthetic speech with background noise only, as well as synthetic speech with both reverberation and background noise. In addition, SpecAugment [11] is applied on the log Mel filter bank features generated from both synthetic data and real data for RNN-T training. Two frequency masks are applied to each utterances and the maximal masked frequency percentage is 37.5%. The maximal ratio of each time mask is 5% of the utterance duration, and the number of time masks is proportional to utterance length, i.e. 5% of the frame numbers without being larger than 10. Where masks applies, Gaussian noise is used with the same mean and variance from the masked values.

### 3.4. ASR RNN-T model

RNN-T is an E2E ASR model architecture suitable for streaming applications with proven competitive performance [3, 26]. It consists of a transcription network (or encoder), a prediction network (or decoder), and a joint network. Targeting at streaming applications, we use LSTM [27] layers for both the encoder and decoder. The encoder sequentially maps acoustic feature $\mathbf{x}$ to a high-level feature representation $\mathbf{h} = \mathrm{Enc}(\mathbf{x})$. The decoder network take previous labels in the sequence and generates a high level representation for next prediction. We use a feed-forward network for the joint network to combine the information from both the acoustic representations from encoder and the linguistic representations from decoder to make a joint prediction of next word piece. The loss function for RNN-T pa-

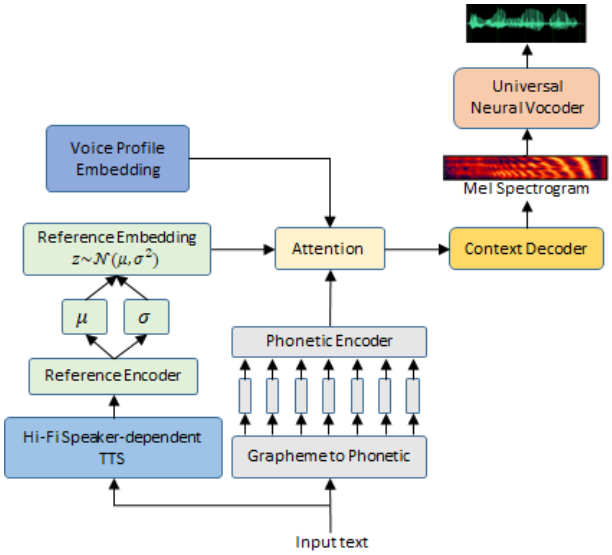


Figure 2: *Architecture of proposed Multi-context TTS system.*

rameter optimization is the negative log-posterior of the target label sequence $\mathbf{y}$:

$$\mathcal{L} = -\log P(\mathbf{y}|\mathbf{h}), \tag{1}$$

where $P(\mathbf{y}|\mathbf{h}) = \sum_{\hat{\mathbf{y}}} P(\hat{\mathbf{y}}|\mathbf{h})$, $\hat{\mathbf{y}} \in \mathcal{A}$. $\mathcal{A}$ is the set of all possible alignments including blank labels between encoder feature representation $\mathbf{h}$ and target label sequence $\mathbf{y}$. The Adam algorithm [28] is used for the numerical optimization during training, with a learning rate scheduler in three stages: linear warm-up, hold and exponential decay.

### 3.5. Continual learning with multi-stage training

Continual learning (CL) [21, 22] imitates the incremental lifelong learning ability in humans and animals by machine, through acquiring data over time, and learning, fine-tuning, and transferring knowledge over various tasks. CL reduces the cost from model training with repeated visit to a large amount of data, and it also reduces or even removes the dependency on previously harvested training data.

In this work, we apply CL and let a general-purpose ASR model learn for a new application task of recognizing medication names, by continually exposing the ASR model to the synthetic speech containing the medication names. To prevent catastrophic forgetting, we propose a multi-stage training strategy for continual learning, including freezing the LSTM layers of encoder during fine-tuning with synthetic speech data and unfreezing all layers in later stages. When parameters are unfrozen, to ensure that the learned parameters do not deviate too much when trained with synthetic speech for a new application, an elastic penalty term is introduced to the RNN-T loss function that minimizes the distance between the new and previous model parameters:

$$\mathcal{J} = \lambda \sum (\theta_{i,pre} - \theta^*_{i,cur})^2, \tag{2}$$

where $i$ indexes the RNN-T decoder parameters, $pre$ and $cur$ are the previous and current training stages, respectively, $*$ indicates the trainable parameters and $\lambda$ adjusts forgetting speed.

Table 1: *WERs on the LibriSpeech.*

| Model | Training sets | | WER | |
|---|---|---|---|---|
| | real | synthetic | test-clean | test-other |
| Benchmark | 960 | - | 7.29 | 17.41 |
| Baseline | 480 | - | 9.90 | 22.64 |
| Baseline + TTS | 480 | 1150 | 8.66 | 20.78 |

## 4. Results

### 4.1. Experiments on LibriSpeech: synthetic speech for reducing transcribed data

To evaluate the effectiveness of synthetic data, we first perform experiments on the LibriSpeech dataset [29]. In these experiments, we use 64-dimensional Log-Mel-Frequency features extracted with 25ms window and 10ms shift. Each feature vector is stacked with 2 frames to the left and downsampled by a factor of 3 corresponding to a frame rate of 30msec. For the experiments on LibriSpeech data, we use six-layer LSTMs with 1024 units in the encoder. The decoder is a two-layer LSTM with 1024 units in each layer. The output size of the recognition encoder and the decoder is set to 640. We use a one-layer feed-forward joint network with 512 units and tanh activation. The output softmax layer dimensionality is 2501 which corresponds to blank label and 2500 word pieces: the most likely subword segmentation from a unigram word piece model [30]. We use an adaptive variant of SpecAugment [11], as proposed in [31]. We use Adam algorithm [28] for optimization of all models, and the learning rate is scheduled based on warm-up, hold and decay strategy as proposed in [11]. For each experimental run, we chose the best model based on its performance on the development set.

LibriSpeech contains 960 hours of read speech data for training [29]. As an ASR baseline with a limited amount of audio data, we assume that only half of the LibriSpeech training data is available, i.e. 480 hours training data randomly selected from the all training data. As shown in Table 1, an RNN-T model trained with 480 hours of data is 35.8% relatively worse on test-clean when compared to an RNN-T model trained with all 960 hours training data. We then synthesize about 1150 hours audio data using our multi-context TTS system, and the input texts for TTS are the transcriptions of the missing 480 hours data. This TTS training set contains about 48k unique input texts, and each text utterance is synthesized with randomly selected 24 voice profiles from the total of 500 available voice profiles. We then trained an RNN-T model using MST combining with 480 hours real data and 1150 hours synthetic speech. Compared to the baseline model trained with 480 hours real data alone, this improves the performance on test-clean by 12.5% relative (Table 1).

### 4.2. Experiments in real application: synthetic speech for medication names recognition

We then expand a general-purpose ASR model for a new application of medication names recognition. This new application has no available real recordings for ASR model training. In the following set of experiments, we use synthetic speech to teach a general-purpose ASR model to recognize medication names.

We use a slightly different RNN-T architecture from previous LibriSpeech experiments. The encoder now consists of 5 LSTM layers and the softmax layer has an output vocabu-

Table 2: *NWERs for the application of recognition of medication name. The weight on the left in the column* ***Weights%*** *is the percentage of samples in MST from real data and the weight on the right is the percentage of samples from synthetic data.* ***(R, S)*** *indicates whether real (R) or synthetic (S) audio is used during each stage of training.*

| **Model** | **Real Data** (hours) | **Synthetic Data** (hours) | **Weights%** (R, S) | **Freeze Encoder** | **Elastic Penalty** | **NWER** Dev-Gen | Eval-Gen | Eval-Med |
|---|---|---|---|---|---|---|---|---|
| Baseline | 50K | - | - | No | - | 100 | 100 | 100 |
| Stage 1 | 50K | 5k | (95, 5) | Yes | No | 100.99 | 101.32 | 21.06 |
| Stage 2 | 50K | 5k | (98, 2) | No | No | 100.54 | 100.98 | **13.70** |
| Stage 3 | 50K | - | - | No | Yes | 100.14 | 100.89 | 21.47 |
| Stage 4 | 50K | - | - | No | No | **99.18** | **99.72** | 34.56 |

lary size of 4001 word pieces including the blank label. Note that, the results in this section are reported in normalized WER (NWER) numbers, which is the regular word error rate (WER) divided by the WER of the baseline model on the same test set and then multiplied by 100. Therefore, the NWERs for baseline model are 100 for all test sets as shown in Table 2. The baseline RNN-T model for general-purpose application is trained with a dataset of 50,000 hours real human utterances. This dataset is a collection of de-identified production utterances from voice-controlled far-field devices. A development set (Dev-Gen) and an evaluation set (Eval-Gen) are constructed with the same type of utterances, consisting of about 50 hours and 160 hours of data respectively. These two test sets are used to monitor the performance change on existing applications. To evaluate the performance on the new application of medication name recognition, we collected 8 hours of real human data containing utterances with medication names (Eval-Med).

To support feature expansion without real training data, we prepared 5000 hours TTS synthetic data. The texts are generated by randomly combining 150 unique text utterance templates and 600 common medication names. For each text utterance generated, we sample 32 voice profiles from 500 voice profiles. The generated clean synthetic audio is corrupted with noise and reverberation on-the-fly for RNN-T model training. With MST, the corrupted synthetic audio is combined with real recordings with configured data ratio parameters. With continual learning, we start with the baseline general-purpose RNN-T model and fine-tune it so that the recognition performance on general test sets (Eval-Gen) maintains while the performance on the test set for medication names (Eval-Med) is largely improved.

One challenge in such continual learning is a balance between forgetting learned knowledge which causes degradation on Eval-Gen and learning new knowledge which strives to improve performance on Eval-Med. We use multi-stage training to address this challenge, and our experiments concluded with 4 critical stages as shown in Table 2. The first stage is to fine-tune the baseline model with batches of data that contain 95% real data and 5% synthetic data where we fix the RNN-T encoder parameters. We train in this way for 57k iterations, during which process the learning rate decays from 5e-5 to 1e-5. This stages ramps up the parameters for decoder and joint network for the new application of medication name recognition. In the second stage, we further fine-tune the model with both real and synthetic utterances with the portion of 98% and 2% in each batch and fixed learning rate of 1e-5. As shown in Table 2, this improves the recognition performance on Eval-Med further from an NWER of 21.06 to 13.70. In our experiments we found the first training stage with encoder freezing critical, and removing Stage 1 led to performance degradation. At the same time, model at the end of stage 2 showed a small degradation compared to baseline on general test sets.

To recover the degradation, in the third and fourth stages, we only fine-tune models with real human speech. In the third stage, we include an elastic penalty as described in section 3.5 to minimize the deviation of the model parameters from the previous stage as the model has well learned medication names. We further fine-tune the model in the fourth stage without such regularization but with a small learning rate of 1e-5 to ensure the performance of the model doesn't degrade from the baseline. Note that in our experiments, we found the third stage critical and directly jumping from Stage 2 to Stage 4 led to worse results. As shown in Table 2, the final model from Stage 4 achieved slightly better performance on both general test sets compared to baseline model, and at the same time the recognition performance on Eval-Med is more than 65% better than the baseline model. This is achieved with 5k hours of synthetic training data without real recordings for medication names .

## 5. Conclusions

In this work, we propose to use synthetic speech for E2E ASR model training to reduce both data costs and production data reliance. In addition, we demonstrated how to effectively and incrementally improve ASR for a new application that customer audio data is not available at all. Using continual learning with our proposed multi-stage training, the best system relatively improves the WER on the new application by more than 65% without compromise on the existing application. While the value of synthetic speech as ASR training data remains less than that of real speech, but synthetic speech shows great promise in training large-scale ASR for new applications.

## 6. Acknowledgements

We would like to thank Charles Chang and Paul MacCabe for the high-level support of this research. In addition, we would like to acknowledge the Alexa Data Synthetic and Alexa ASR teams for providing the infrastructure that this work has benefited from.

## 7. References

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 369–376.

[2] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st In-*

*ternational Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1764–1772.

[3] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.

[4] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[5] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. yiin Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[6] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6161–6165.

[7] N. Rossenbach, A. Zeyer, R. Schlter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7069–7073.

[8] C. Du and K. Yu, "Speaker augmentation for low resource speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7719–7723.

[9] 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.

[10] 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.

[11] 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*, Sep 2019.

[12] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end ASR: from supervised to semi-supervised learning with modern architectures," *CoRR*, vol. abs/1911.08460, 2019. [Online]. Available: http://arxiv.org/abs/1911.08460

[13] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6429–6433.

[14] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[15] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3*, 2018.

[16] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*. IEEE, 2019, pp. 434–441.

[17] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (hat)," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6139–6143.

[18] Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong, "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 243–250.

[19] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12, 1987, pp. 705–708.

[20] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[21] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[22] A. Gepperth and C. Karaoguz, "A bio-inspired incremental learning architecture for applied perceptual problems," *Cognitive Computation*, vol. 8, pp. 924 – 934, 2016.

[23] N. Prateek, M. Lajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood, "In other news: a bi-style text-to-speech model for synthesizing newscaster voice with limited data," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 2 (Industry Papers)*, A. Loukina, M. Morales, and R. Kumar, Eds. Association for Computational Linguistics, 2019, pp. 205–213.

[24] J. Rohnke, T. Merritt, J. Lorenzo-Trueba, A. Gabrys, V. Aggarwal, A. Moinet, and R. Barra-Chicote, "Parallel WaveNet conditioned on VAE latent vectors," 2020. [Online]. Available: https://arxiv.org/abs/2012.09703

[25] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Proc. Interspeech 2020*, 2020, pp. 2977–2981.

[26] B. Li, S. yiin Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, "Towards fast and accurate streaming end-to-end ASR," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6069–6073.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, 2015.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[30] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Nov. 2018, pp. 66–71.

[31] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "SpecAugment on large scale datasets," *ICASSP*, May 2020.