MULTI-SPEAKER SEQUENCE-TO-SEQUENCE SPEECH SYNTHESIS FOR DATA AUGMENTATION IN ACOUSTIC-TO-WORD SPEECH RECOGNITION

Sei Ueno, Masato Mimura, Shinsuke Sakai, Tatsuya Kawahara

Graduate School of Informatics, Kyoto University,
Sakyo-ku, Kyoto, 606–8501, Japan

ABSTRACT

The acoustic-to-word (A2W) automatic speech recognition (ASR) realizes very fast decoding with a simple architecture and achieves state-of-the-art performance. However, the A2W model suffers from the out-of-vocabulary (OOV) word problem and cannot use text-only data to improve the language modeling capability. Meanwhile, sequence-to-sequence neural speech synthesis has also been developed and achieved naturalness comparable to human speech. We investigate leveraging sequence-to-sequence neural speech synthesis to augment training data for the ASR system in a target domain. While speech synthesis model is usually trained with single speaker data, ASR needs to cover a variety of speakers. In this work, we extend the speech synthesizer so that it can output speech of many speakers. The multi-speaker speech synthesizer is trained with a large corpus in the source domain, then used to generate acoustic features from texts of the target domain. These synthesized speech features are combined with real speech features of the source domain to train an attention-based A2W model. Experimental results show that the A2W model trained with the multi-speaker model achieved a significant improvement over the baseline and the single speaker model.

Index Terms— Sequence-to-sequence speech recognition, Sequence-to-sequence speech synthesis, acoustic-to-word model, training data augmentation, multi-speaker speech synthesis

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) systems which directly convert acoustic features into a character or word sequence are very attractive since they have so simple architecture that we can design and develop easily. In the end-to-end ASR systems, connectionist temporal classification (CTC) approaches [1, 2] and sequence-to-sequence (seq2seq) approaches such as RNN-transducers and attention-based encoder-decoder models [3, 4, 5, 6] have been investigated intensively. These approaches generate a symbol sequence without requiring latent state transition models such as HMMs. With regard to the output units of ASR, acoustic-to-word (A2W) models [2, 7, 8, 9] which directly map acoustic features into a word sequence realizes very fast decoding since they do not require any external decoders. We have demonstrated in [10] that an attention-based A2W model achieved word error rate (WER) reduction of 25.3% relative from a state-of-the-art hybrid DNN-HMM system with decoding speed faster by a factor of 50.

However, A2W models have some drawbacks compared to phone-based and character-based models. The most critical problem is that they cannot recognize words which do not appear in the training data by adding new word entries after training. Furthermore, the A2W model requires a huge amount of paired data of speech and transcripts. In conducting domain adaptation, word entries of the source domain are often different from that of the target domain and we cannot assume a large data set for the target domain. These problems imply that it is difficult for the A2W model to be adapted to a new target domain. Even if a reasonably large text corpus is available for the target domain, it cannot be fully utilized to improve the language model, when there is a big mismatch in the vocabulary between two domains.

To address this problem, we investigate utilizing speech synthesis to generate acoustic features for training A2W models from the target domain texts [11]. Recently, seq2seq neural speech synthesis models have also been developed [12, 13, 14, 15]. In contrast to the conventional text-to-speech (TTS), seq2seq speech synthesis realizes TTS with a very simple architecture and its training is much easier. Moreover, it has shown to achieve naturalness comparable to human speech [13, 15]. Therefore, it is expected to synthesize speech data usable for ASR model training. The synthesized data makes it possible to cover the target domain vocabulary and to predict probabilities for words which did not appear in the source domain. However, a speech synthesizer is usually trained with a single speaker and does not have a diversity of speakers. This may be a serious problem for ASR, which needs to cover a variety of speakers. In this work, we extend our speech synthesis framework to contribute to developing multi-speaker speech using speaker embedding in seq2seq speech synthesis. By training the speech synthesizer with a large number of speakers, it is expected to generate more "realistic" speech data for ASR model training, and eventually solve the data sparseness problem.

In the rest of the paper, we first review the seq2seq model for ASR and TTS in Section 2. Section 3 gives explanations of the proposed multi-speaker seq2seq speech synthesis for data augmentation in A2W ASR. Experimental evaluations using the Corpus of Spontaneous Japanese (CSJ) are presented in Section 4.

2. SEQ2SEQ MODEL FOR ASR AND TTS

2.1. Attention-based seq2seq model

The attention-based seq2seq encoder-decoder model has two distinct networks. One is an encoder network, which makes a distributed representation from the input sequence. The other is a decoder network, which predicts a label sequence using the intermediate representation. The decoder uses only a relevant portion of the encoded sequential representation for predicting a label at each time step using the attention mechanism. In this work, we use multi-layer bidirectional LSTM for the encoder, and a single layer unidirectional LSTM followed by a softmax layer and the location-sensitive attention mechanism [4] for the decoder. The LSTM-based decoder predicts the next symbol using a history of previous symbols, thus

it substantially includes a language model. The objective function for training the attention model is cross entropy loss between the predicted label sequence and the target label sequences.

2.2. Acoustic-to-Word (A2W) ASR

A2W model maps acoustic features into a word sequence directly. It can realize simple and fast decoding without external processes. However, this model requires a huge amount of pair data of speech and text since the output nodes correspond to all lexical entries. This training problem is mitigated by the multi-task training with the character model [16]. The other problem is that it is not possible to add new word entries unlike subword-based systems. Moreover, the model cannot leverage text-only data which is usually available in a large scale and in a new domain. Although a separate language model can be trained with the text data, it is not straightforward for the A2W model to combine it in decoding as in [17, 18] since the vocabulary of the A2W model cannot be updated according to the language model. This is a serious problem when adapting the A2W model to a new domain using text-only data.

2.3. Seq2seq speech synthesis

Seq2Seq speech synthesis generates speech from a character or phone sequence. It has a much simpler architecture than the conventional speech synthesis, which requires many modules and manpower. Recently, these systems achieve a very high mean opinion score (MOS), comparable to human speech [13].

In this work, we use Tacotron 2 [13] based model, which is composed of an encoder-decoder network with an attention mechanism and a WaveNet-based vocoder network. The encoder-decoder network generates acoustic features or vocoder parameters from a phone or character sequence. The vocoder network synthesizes a waveform from these predicted features. Note that we do not use the vocoder network in this work since we only need mel-spectrogram for training ASR models. The encoder network maps into a distributed representation from a character sequence via character embedding, three convolution layers, and one-layer BiLSTM. The decoder network predicts five consecutive frames of log mel-scale filter bank (lmfb) features at each decoding step using a location-sensitive attention mechanism [4].

3. PROPOSED METHOD

3.1. Leveraging seq2seq speech synthesis for A2W ASR

For efficient training and adaptation of an A2W ASR model to a new domain using text-only data, we have investigated leveraging seq2seq speech synthesis to augment the training data . We collect texts from a target domain where we want to perform speech recognition. The sequence of phones and special symbols representing the word boundaries are fed into the seq2seq speech synthesizer to generate a sequence of lmfb features of the sentence. The set of the synthesized lmfb features and corresponding word sequences are added to the baseline training data of the real speech corpus. This scheme makes it possible to train an A2W model from arbitrary sentences. It also allows for expanding the vocabulary and improves language model. Moreover, this scheme makes it possible to conduct *shallow fusion* [18] of the A2W model and the language model since the vocabulary of the two models becomes matched.

There are several other works to leverage text-only data for end-to-end ASR training. Renduchintala *et. al.* [19] investigated feeding text data to train a seq2seq ASR model via special encoding without converting to speech features. Sriram *et. al.* [20] showed that leveraging a pre-trained language model during seq2seq ASR training made the model converge faster and transfer to a new domain. Moreover, the combination of seq2seq ASR and TTS has been investigated in [21, 22] to realize a deep learning-based speech chain model. The novelty of our work is that we exploit seq2seq speech synthesis for enhancing the A2W model in a direct and effective manner.

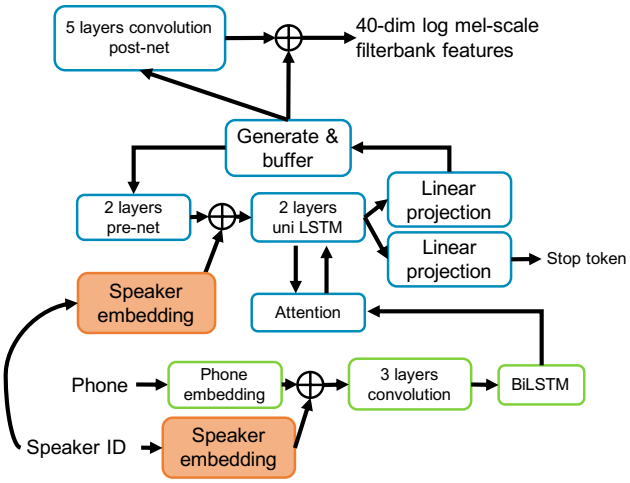

Fig. 1. The multi-speaker speech synthesizer. In the encoder network, convolution inputs are the sum of phone embedding and speaker embedding. In the decoder network, speaker embedding is used to initialize the hidden state of LSTM for each previous time step.

3.2. Multi-speaker seq2seq speech synthesis

As in our previous work [11], the speech synthesizer is trained with only one speaker. The synthesized speech has no diversity and is not appropriate for training ASR models which need to cover a variety of speakers. In [11], we also confirmed the effectiveness of the encoder freezing [23], in which the parameters of the acoustic encoder were copied from the model pre-trained with real speech, and they were fixed during training using the augmented data set consisting of the artificial and real data. This results showed that the monochromatic speech would be harmful for learning the A2W encoder network.

In order to generate a variety of speech for ASR training, we design speech synthesis trained with a large corpus containing many speakers. Several multi-speaker synthesizers have been proposed in the context of TTS. In [12, 14], the speaker embedding was used across encoder, decoder, and vocoder. Jia *et. al.* [24] used a fixed-dimensional embedding vector known as d-vector [25]. Inspired by [12], we add speaker embedding to the Tacotron 2 architecture as shown in Fig. 1. In the encoder network, the speaker embedding is added as a bias to the convolution output after a softsign function. In the decoder network, after a softsign function, the speaker embedding is also added as a bias to the 2-layer pre-net output. Training a speech synthesizer using a multi-speaker corpus is much more difficult than training with a single speaker corpus. In fact, the multi-speaker model with random initialization did not converge in our experiment. In this work, we pre-trained a model with a large single speaker corpus before training the multi-speaker model. Therefore,

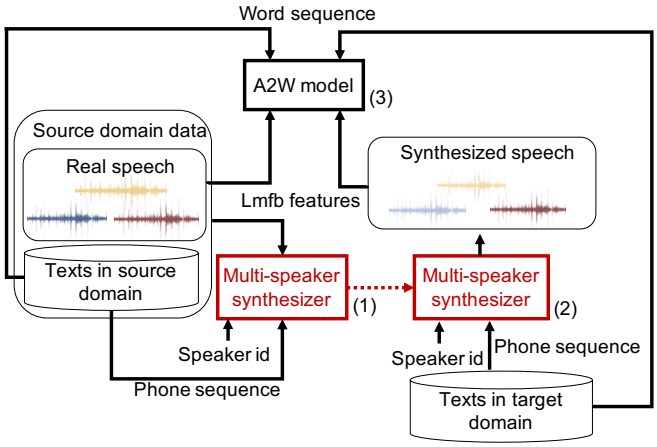

Fig. 2. The overall flow of the data augmentation method leveraging multi-speaker speech synthesis. (1) train multi-speaker synthesizer, (2) generate lmfb features, and (3) learn the A2W model using both of real speech and synthesized speech.

we do not concatenate speaker embedding with the other output, but use it as a bias.

Fig. 2 shows an overall flow of the data augmentation method leveraging the multi-speaker speech synthesis. First, we train the synthesizer using the source domain corpus which has many speakers. In this training, we use speaker embedding. Second, we generate acoustic features from texts in the target domain. We can generate speech of a variety of speakers from the same sentence by randomly choosing the speaker id. Finally, we learn the A2W model using both of real speech and synthesized speech.

4. EXPERIMENTAL EVALUATIONS

4.1. Data sets and tasks

We use a standard large-scale Japanese corpus named the Corpus of Spontaneous Japanese (CSJ) [26]. CSJ has two distinct sub-corpora, CSJ-APS and CSJ-SPS. CSJ-APS consists of academic presentation speeches and has 986 speakers (male: 809, female: 177) with the total duration of 247.9 hours for training. CSJ-SPS consists of simulated presentation speeches on three general themes and has 1704 speakers (male: 799, female: 905) with the total duration of 281 hours. These subsets have their own official test sets, namely, CSJ-TESTSET1 and CSJ-TESTSET3. The vocabulary consists of all distinct words occurring more than twice in the training data and special tokens such as ⟨sos⟩, ⟨eos⟩, and ⟨UNK⟩. The vocabulary sizes are 19146 for APS, 24826 for SPS, and 34331 for APS and SPS respectively. The number of shared word entries in APS and SPS is 11446.

4.2. System configuration

4.2.1. A2W model

We used a 40-channel log mel-scale filter bank (lmfb) outputs as acoustic features for A2W models. Non-overlapping frame stacking [27] was applied to these features, in which we stacked and skipped three frames to make a new super-frame. The acoustic encoder consists of five layers of bidirectional LSTMs with 320 cells. The dropout [28] rate was 0.2 for training each BiLSTM layer. The attention-based A2W decoder consists of one-layer unidirectional LSTM with 320 cells, a hidden layer with 320 tanh nodes, and a softmax output layer for word entries. We used Adam [29] optimizer with a standard setting and gradient clipping with a threshold of 5.0. We also used label smoothing [30] for regularization. The beam width was set to be 4 in decoding. For the language model integration with shallow fusion, we trained neural language models with 3 layers of unidirectional LSTMs with 256 memory cells. Each word is mapped to a 512-dimensional continuous vector before fed to LSTMs. We used PyTorch [31] to implement the A2W models.

4.2.2. Multi-speaker speech synthesizer

In the original Tacotron 2, the input is a character sequence, the output is an 80-dim mel-spectrogram, and the loss function is the mean squared error (MSE) of the mel-spectrogram. In this work, however, the input is a phone sequence, the output is 40-dim lmfb features, and the loss function is L1 loss. As the 40-dim lmfb features are used for our A2W recognition system, we can use an output sequence from the synthesizer directly as input to the A2W model.

For word segmentation and pronunciation annotation of texts, we used Mecab[1], a CRF-based Japanese morphological analyzer. We used 33 phone classes including special tokens for pause, word boundary and the end of a sentence. The phone encoder consists of a 512-dimensional phone embedding, a 256-dim speaker embedding, three convolution layers with 512 filters and one-layer bidirectional LSTM with 256 cells. The encoder outputs are summarized using the location-sensitive attention mechanism [4]. The attention weight at each decoding step is calculated using the 128-dimensional projected vectors of the decoder LSTM state, the encoder output sequence, and the location features. The location features are calculated by convolving 32 one-dimensional convolution filters with length 31 to the cumulative vector of the attention weights in all past decoding steps. Meanwhile, the last one frame of prediction in the last time step is passed through a pre-net consisting of two fully-connected layers with 256 ReLU units. This pre-net output is summed with the speaker embedding and the encoded representation with the attention vector to be provided to 2-layer unidirectional LSTMs with 1024 memory cells. The LSTM outputs together with the attention context vector are passed through a linear projection layer to predict five frames of the target lmfb features. We also used PyTorch [31] to implement the speech synthesizer.

We first pre-trained a model using the JSUT (Japanese speech corpus of Saruwatari Laboratory, University of Tokyo) corpus [32], which is a recording of 7,607 prompt texts read aloud by a female speaker with the total duration of ten hours. This single speaker model is also used for a reference. After that, we fine-tuned the pre-train model using the source domain corpus of a thousand speakers. When we synthesize speech features, a speaker id is chosen randomly to generate speech features given a sentence text.

4.3. Results

We chose one sub-corpora as a source domain and the other as a target domain. Using the source domain data, we trained the baseline ASR model and the multi-speaker speech synthesizer. In the target domain, we use only transcription data for adaptation. We generated speech data from the texts of the target domain and re-trained the A2W model using the augmented data.

Fig. 3 shows synthesized lmfb features generated by the multi-speaker model. The multi-speaker model was trained using CSJ-

[1] http://taku910.github.io/mecab

Table 1. ASR performance (WER (%)) for CSJ-APS and CSJ-SPS testset. In this table, "Source" is APS and "Target" is SPS. "(single speaker)" means that we trained speech synthesizer using a single speaker female corpus. "(multi-speaker)" means that we trained speech synthesizer using the source-domain corpus. "+ LM integration" is shallow fusion with the language model which is trained using both of the source and target domain.

Training data	Source (APS)	Target (SPS)	+ LM integration
Source real + target real (oracle)	10.35	9.06	9.00
Source real only [baseline]	12.30	19.22	18.84
Source real + target TTS (single speaker) [reference]	11.89	14.64	14.16
Source real + target TTS (multi-speaker in source domain) [proposed]	**11.43**	**13.37**	**13.27**

Table 2. ASR performance (WER (%)) for CSJ-APS and CSJ-SPS testset. In this table, "Source" is SPS and "Target" is APS.

Training data	Source (SPS)	Target (APS)	+ LM integration
Source real + target real (oracle)	9.06	10.35	10.24
Source real only [baseline]	9.69	23.30	23.14
Source real + target TTS (single speaker) [reference]	9.86	18.74	18.24
Source real + target TTS (multi-speaker in source domain) [proposed]	**9.36**	**16.68**	**15.94**

SPS, and generated acoustic features from a sentence text in CSJ-APS for three speakers. We first confirmed that the multi-speaker model could output speech of a variety of speakers as these lmfb features were different in the length of speech and the spectrum characteristics.

Table 1 and Table 2 show the ASR performance in word error rate (WER) for the CSJ-APS and the CSJ-SPS test sets. In Table 1, the source domain is CSJ-APS and the target domain is CSJ-SPS with 213-hour synthesized data. The OOV rate in the CSJ-SPS test set with the baseline APS model is 3.53%, but it is reduced to 1.21% by incorporating the texts of SPS. In Table 2, the source domain is CSJ-SPS and the target domain is CSJ-APS with 209-hour synthesized data. The OOV rate in the CSJ-APS test set with the baseline SPS model is 4.28%, but it is reduced to 0.85% by incorporating the texts of APS. The duration of synthesized speech is different from that of real speech since the speech synthesizer also estimates the length of speech for a given text.

We confirmed that the WER of the baseline model was very high, and the model augmented with the synthesized speech using the single speaker achieved a large improvement as the model can recognize the words which do not appear in the original training data. Our proposed multi-speaker model achieved a further significant improvement for the target domain and also yielded a small improvement for the source domain. This shows the multi-speaker synthesized speech provides more meaningful training data than the single speaker when the multi-speaker model generate the same amount of augmented data. However, when we generate speech features with two speakers from each sentence, the proposed models did not improve the performance. In other words, synthesizing speech from a sentence for the first time is effective for enhancing the language model capability. There is still a gap from the oracle performance trained with the real speech. This suggests that the synthesized speech does not have a variety of the real speech.

In addition to the A2W model, we can use the external language model with the enhanced vocabulary in shallow fusion. The language model using both of APS and SPS was trained. But the vocabulary was adjusted when it was applied to the source-only model. We can see that integration of the A2W models and the external language model was effective in all cases. However, the fusion with the source-only model is still far behind the proposed method. The result demonstrates the language model integration alone cannot conduct so effective adaptation as the data augmentation.

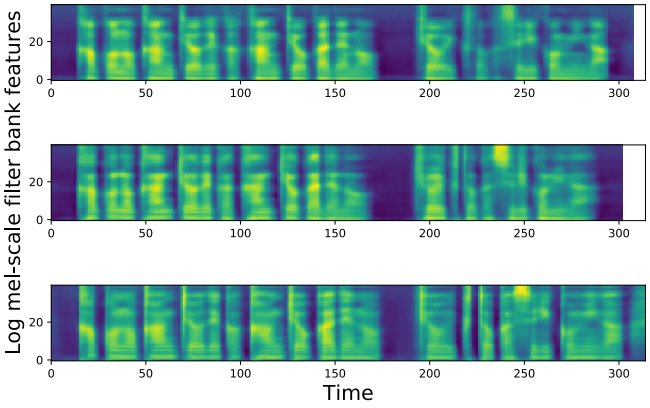

Fig. 3. Sample of synthesized log mel-scale filter bank features. We selected different speakers to generate acoustic features from a disfluent transcript "kako no kaNkyo: de ka kaNkyo: ka de no, kyo:iku to ka surikomi ga".

5. CONCLUSION

In this work, we designed a multi-speaker seq2seq speech synthesis for augmenting training data for the A2W ASR model. We could train it with a large speech corpus containing many speakers to generate speech data of a variety of speakers. This augmentation method achieved a large improvement in domain adaptation, and the multi-speaker model improved the ASR performance of the A2W model compared to the single speaker model, showing that it can generate useful data for ASR training. Moreover, we also demonstrated that integration of the language model with shallow fusion yielded a further improvement. We are investigating the further improvement of speaker representation in multi-speaker speech synthesis to fill the gap from real speech data.

6. REFERENCES

[1] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks," in *Proc ACM*, 2006, pp. 369–376.

[2] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 4280–4284.

[3] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur Yi Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*, 2017, pp. 206–213.

[4] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

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

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-End attention-based large vocabulary speech recognition," in *Proc. ICASSP*, 2016, pp. 4945–4949.

[7] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *Proc. ICASSP*, 2016, pp. 5060–5064.

[8] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 959–963.

[9] Z. Chen, Q. Liu, H. Li, and K. Yu, "On modular training of neural acoustics-to-word model for LVCSR," *arXiv preprint arXiv:1803.01090*, 2018.

[10] M. Mimura, S. Sakai, and T. Kawahara, "Forward-backward attention decoder," in *Proc. INTERSPEECH*, 2018, pp. 2232–2236.

[11] 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 *Proc. SLT*, 2018, accepted.

[12] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Proc. NIPS*, 2017, pp. 2962–2970.

[13] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. INTERSPEECH*, 2017, pp. 4779–4783.

[14] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICRL*, 2018.

[15] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to human quality TTS with transformer," *arXiv preprint, 1809.08895*, 2018.

[16] S. Ueno, H. Inaguma, M. Mimura, and T. Kawahara, "Acoustic-to-word attention-based model complemented with character-level CTC-based model," in *Proc. ICASSP*, 2018, pp. 5804–5808.

[17] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. INTERSPEECH*, 2017, pp. 523–527.

[18] 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 *Proc. ICASSP*. IEEE, 2018, pp. 5824–5828.

[19] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end ASR," in *Proc. INTERSPEECH*, 2018, pp. 2394–2398.

[20] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *Proc. INTERSPEECH*, 2018, pp. 387–391.

[21] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. ASRU*, 2017, pp. 301–308.

[22] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain with one-shot speaker adaptation," in *Proc. INTERSPEECH*, 2018, pp. 887–891.

[23] S. Ueno, T. Moriya, M. Mimura, S. Sakai, Y. Shinohara, Y. Yamaguchi, Y. Aono, and T. Kawahara, "Encoder transfer for attention-based acoustic-to-word speech recognition," in *Proc. INTERSPEECH*, 2018, pp. 2424–2428.

[24] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint, 1806.04558*, 2018.

[25] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, 2016, pp. 5115–5119.

[26] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese.," in *LREC*, 2000, pp. 947–9520.

[27] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *INTERSPEECH*, 2015, pp. 1468–1472.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, pp. 1929–1958, 2014.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint, 1412.6980*, pp. 1–15, 2014.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, 2016, pp. 2818–2826.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[32] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv preprint, 1711.00354*, 2017.