Synthesizing waveform sequence-to-sequence to augment training data for sequence-to-sequence speech recognition

Sei Ueno*, Masato Mimura†, Shinsuke Sakai‡ and Tatsuya Kawahara§

Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto, 606–8501 Japan
(Received 20 January 2021, Accepted for publication 2 August 2021)

Abstract: Sequence-to-sequence (seq2seq) automatic speech recognition (ASR) recently achieves state-of-the-art performance with fast decoding and a simple architecture. On the other hand, it requires a large amount of training data and cannot use text-only data for training. In our previous work, we proposed a method for applying text data to seq2seq ASR training by leveraging text-to-speech (TTS). However, we observe the log Mel-scale filterbank (lmfb) features produced by Tacotron 2-based model are blurry, particularly on the time dimension. This problem is mitigated by introducing the WaveNet vocoder to generate speech of better quality or spectrogram of better time-resolution. This makes it possible to train waveform-input end-to-end ASR. Here we use CNN filters and apply a masking method similar to SpecAugment. We compare the waveform-input model with two kinds of lmfb-input models: (1) lmfb features are directly generated by TTS, and (2) lmfb features are converted from the waveform generated by TTS. Experimental evaluations show the combination of waveform-output TTS and the waveform-input end-to-end ASR model outperforms the lmfb-input models in two domain adaptation settings.

Keywords: Speech recognition, Sequence-to-sequence model, Attention-based encoder-decoder model, Speech synthesis, Data augmentation

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) that directly converts acoustic features into a symbol sequence is very attractive since they have a simple architecture, and we can design and develop ASR systems easily. It realizes faster decoding than the conventional DNN-HMM hybrid ASR systems. There are several major approaches to end-to-end ASR systems: connectionist temporal classification (CTC) approaches [1–4] and sequence-to-sequence (seq2seq) approaches such as RNN-transducers [5], attention-based encoder-decoder models [6–11], and transformer-based models [12,13]. These approaches generate a symbol sequence without requiring latent state transition models such as HMMs. Although the end-to-end speech recognition model achieves high performance, it requires a much larger amount of paired data of speech and transcription than the conventional hybrid model.

Several previous works attempted to use unpaired texts for improving end-to-end ASR. Tjandra et al. [14,15] investigated the combination of seq2seq ASR and TTS to realize a deep learning-based speech chain model. Hayashi et al. [16] proposed a back-translation-style data augmentation for ASR. With text-only data, they trained a text encoder to estimate the ASR hidden states. Karita et al. [17,18] used autoencoders to leverage speech-only data and text-only data. These autoencoders learn features from speech-only and text-only datasets by switching the encoders and decoders used in the ASR and TTS models. These methods can be effective for enhancing the ASR models in the same domain. However, it is difficult for the end-to-end model to be adapted to a new target domain, as the vocabulary and language model are often different from one domain to another. When we have a large amount of text data for the domain, it is straightforward to generate speech data from the text for adapting ASR models to a new domain. Inspired by the recent progress of seq2seq TTS, we have investigated this approach.

In our previous work [19,20], we proposed utilizing speech synthesis to generate acoustic features for training end-to-end ASR models from new-domain texts. Recently, seq2seq neural speech synthesis models have also been developed [21–24]. In contrast to conventional text-to-speech (TTS), a seq2seq model realizes TTS with very simple architecture, and its training is much easier.
Moreover, it has been shown to achieve naturalness comparable to human speech \cite{22,24}. TTS efficiently makes it possible to generate paired-data, covering the new target domain. However, speech synthesizers are usually trained with a single speaker and do not have speaker diversity. This may be a serious problem for ASR, which needs to cover a variety of speakers. Therefore, we extended the Tacotron 2-based speech synthesis framework to generate multi-speaker speech using speaker embedding in seq2seq speech synthesis \cite{20}. Training a speech synthesizer with a large number of speakers is expected to generate useful speech data for ASR model training and eventually solve the data sparseness problem.

However, the performance gain by the data augmentation with TTS is still limited compared with that by real speech data. It is because the quality of speech data generated by TTS is not realistic; we observe the spectrum is blurry, particularly on the time dimension. Actually, the time resolution of the artificial spectrum is not sufficient because the TTS model estimates the spectrum of several contiguous frames at one decoding step for stable training and fast inference. While it is not easy to fix this problem in the Tacotron 2 side, we can generate a high-quality speech waveform with the state-of-the-art vocoder. In this work, we adopt the waveform-based data augmentation with TTS. This scheme allows for another option of designing a complete end-to-end ASR model from waveform to a word sequence. Here, we introduce CNN-based feature extraction as a front-end. It is compared with the log Mel-scale filterbank (lmfb)-based ASR systems, where (1) lmfb is generated from the waveform, and (2) lmfb is generated by the Tacotron 2. This comparison of the waveform-based model and the two kinds of lmfb-based model is a major contribution of this paper. As data masking methods such as SpecAugment show a significant effect in the lmfb-based systems, we also design a masking method in the waveform-based end-to-end ASR.

In the rest of the paper, we first review the seq2seq model for ASR and TTS in Sects. 2 and 3. Section 4 explains the proposed seq2seq speech synthesis for data augmentation for seq2seq ASR. Section 5 describes the waveform-input ASR and the masking method designed in this study. Experimental evaluations using several settings are presented in Sect. 6.

2. ATTENTION-BASED ENCODER-DECODER SPEECH RECOGNITION MODEL

We first review a basic approach to end-to-end speech recognition adopted in this study. The attention-based encoder-decoder model is a seq2seq model \cite{6–11}. This architecture has two distinct networks. One is an encoder network, which maps an acoustic feature sequence to a distributed representation of the same length \( T \). Using this intermediate information, the decoder network predicts a symbol sequence whose length is \( L (L \leq T) \). The decoder network uses only a relevant portion of the encoded sequential representation for predicting a symbol at each time step using the attention mechanism. The encoder is implemented with a multi-layer bidirectional RNN such as LSTM, and the decoder usually consists of a 1-layer unidirectional RNN followed by a softmax output layer.

The attention-based model is formulated as follows. The encoder transforms an acoustic feature sequence \( X = (x_1, \ldots, x_T) \) to intermediate representation vectors \( H = (h_1, \ldots, h_T) \). Then, the decoder predicts \( y = (y_1, \ldots, y_L) \) that denote a length-\( L \) sequence of target symbols, where \( y_l \in \{1, \ldots, K\} \) and \( K \) is the number of target symbols. In the following decoding step, the hidden state (memory) activation of the RNN-based decoder at the \( l \)-th time step is computed as

\[
s_l = \text{Recurrence}(s_{l-1}, g_l, y_{l-1})
\]

where \( g_l \) and \( y_{l-1} \) denote a “glimpse” at the \( l \)-th target label and the predicted symbol at the previous step. The glimpse \( g_l \) is a weighted sum of the encoder output sequence as

\[
g_l = \sum_t \alpha_{l,t} h_t
\]

where \( \alpha_{l,t} \) is an attention weight of \( h_t \). In this work, we use a location-based attention mechanism formulated as follows.

\[
e_{l,t} = w^T \tanh(Ws_{l-1} + Vh_t + Uf_{l,t} + b)
\]

\[
f_l = F \star \alpha_l^{l-1}
\]

\[
\alpha_{l,t} = \frac{\exp(e_{l,t})}{\sum_{t'} \exp(e_{l,t'})}
\]

where \( \star \) denotes a 1-dimensional convolution. Using \( g_l \) and \( s_{l-1} \), the decoder predicts the next symbol \( y_l \) as:

\[
y_l \sim \text{Generate}(s_{l-1}, g_l)
\]

where a Generate function is implemented as follows.

\[
R \tanh(Ps_{l-1} + Qg_l)
\]

The objective function for training the attention models is a cross-entropy loss calculated between the predicted symbol sequences and the target oracle symbol sequences. We prepare special symbols for denoting the start-of-sentence (\( \langle \text{so} s \rangle \)) and end-of-sentence (\( \langle \text{eos} \rangle \)). We use a beam search based on the log posterior probability \( p(y_l|X) \). The decoder completes the process when an \( \langle \text{eos} \rangle \) symbol is output.

2.1. SpecAugment

SpecAugment \cite{25} is a data augmentation method by masking both frequency and time domains randomly. Before input to the ASR model, the mask is applied to lmfb features. We randomly choose the value of \( f \), and then \( f_0 \) consecutive Mel frequency channels \( [f_0, f_0 + f] \) are masked, where \( f \) is first chosen from a uniform distribution from 0 to the frequency mask parameter \( F \), and \( f_0 \) is
chosen from $[0, v - f)$, where $v$ is the number of Mel frequency channels. Time masking is applied so that $t$ consecutive time steps $[t_0, t_0 + t)$ are masked, where $t$ is first chosen from a uniform distribution from 0 to the time mask parameter $T$, and then $t_0$ is chosen from $[0, \tau - t)$. The Lmfb values in the range of masked areas are replaced with 0. We did not use time-warping and used two masks in the time domain and one mask in the frequency domain.

3. Tacotron 2 Based Sequence-to-Sequence Speech Synthesis

3.1. Tacotron 2

Seq2Seq speech synthesis generates speech from a character or phone sequence. It has a much simpler architecture than conventional speech synthesis approaches that require many modules and considerable manual tuning effort. These systems have recently achieved a very high mean opinion score (MOS), comparable to human speech [22]. In this work, we use a Tacotron 2 [22] based model since it is easier to design a multi-speaker model than Transformer-based speech synthesizer models. It is composed of an encoder-decoder network with an attention mechanism that generates acoustic features or vocoder parameters from a phone or character sequence. The encoder network maps a character sequence into a distributed representation via character embedding, three convolution layers, and one-layer BiLSTM. The decoder network predicts some consecutive frames of Lmfb features at each decoding step using a location-sensitive attention mechanism [7].

To implement the multi-speaker TTS, we add a speaker ID as another input of the TTS model. We compose an embedding layer and a softsign activation function for speaker embedding. The speaker embedding is added as a bias to the 2-layer output in the decoder and the convolution output in the encoder. For fast and stable training, a reduction rate is used as an additional hyper-parameter, which means the number of contiguous frames that the TTS model estimates at one decoding step. When we set a smaller reduction rate such as 2, we observed a large number of failures in speech generation during training the multi-speaker TTS with a spontaneous speech dataset. In this work, we set the reduction rate to 5.

3.2. Vocoder

With Tacotron 2, we can generate Lmfb features from a text sequence. After that, we need a vocoder that converts Lmfb features into a waveform. There are several options for vocoders. We can use Griffin-Lim vocoder [26] and WORLD vocoder [27] without any DNNs. On the other hand, DNN-based vocoders such as WaveNet [28], WaveRNN [29], and WaveGlow [30] can generate high-quality waveforms. In this work, we use the WaveNet model since it can achieve high performance and thus is adopted in many other works [22,31–33]. The WaveNet is composed of a dilated CNN-based autoregressive model that estimates a waveform value. To predict a waveform value, we use the Lmfb features as condition values. At training time, the condition values are the Lmfb features converted from real speech. At inference time, they are the Lmfb features generated by Tacotron 2.

4. TTS for ASR Training

One of the major problems of the end-to-end ASR systems is that it requires a large amount of training data and is difficult to be adapted to a new domain using text-only data. We investigate leveraging seq2seq speech synthesis to augment the training data for speech recognition (Fig. 1).

We collect text from a target domain where we want to perform speech recognition and generate speech using the text. This scheme not only enhances the language model capability but also learns acoustic patterns of the text, including domain-specific words.

However, speech data generated by a single speaker TTS has less diversity. In our seminal work [20], we introduced multi-speaker TTS to acquire speaker diversity by adding speaker embedding to the Tacotron 2 architecture. Similar efforts have been made by Rosenberg [32], Rossenbach [34], and Huang [33] based on speaker-ID embedding for data augmentation. Rosenberg et al. [32] introduced multi-speaker TTS for data augmentation using hierarchical VAE and the WaveNet-based vocoder. Rosenberg et al. [34] used GST-based speaker embedding and the Griffin-Lim vocoder to synthesize speech waveforms. Huang et al. [33] used a multi-speaker TTS for speaker adaptation of ASR and the WaveNet-based vocoder.

---

1Although Tacotron 2 includes not only spectrogram generation network but also vocoder network in the original paper, “Tacotron 2” is referred to the spectrogram generation network in this work.
Although the multi-speaker TTS enhances ASR model training, the quality of the lmfb features generated by the Tacotron 2 based model is not sufficient compared with the real lmfb features. Figure 2 shows the comparison of real and generated speeches (lmfb features) for the same texts. It is observed that the TTS model generates lmfb features reasonably in the low-frequency regions, but the generated lmfb features in the high-frequency regions are vague or almost blurry compared with the real speech. Moreover, we can see the generated speech sometimes includes unnatural silence in the right example (around 250-th frames). Generally, the magnitudes of low-frequency bins are larger than those of the high-frequency bins and the silence parts. When the Tacotron 2 based model focuses on minimizing the L1 loss between predicted and ground-truth lmfb features, the prediction performance in the high-energy parts is sufficient, but the information of low-energy parts and silence tends to be diminished. Moreover, by using the reduction rate to make the training of the TTS model stable and its inference faster, the predicted lmfb features may be over-smoothed.

Considering that we must generate a huge amount of speech data, it is not easy to fix this problem in the Tacotron 2 side. To generate high-quality speech data, we turn to the WaveNet vocoder. Figure 3 shows lmfb features generated by TTS models. In the top (a), the TTS model generates lmfb feature with a 50 ms window and 12.5 ms shift, which TTS models typically use. In the middle (b), we compute the lmfb feature from the waveform generated by applying WaveNet to (a). In the bottom (c), the TTS model generates the lmfb feature with parameters that ASR models typically use. We observe that the lmfb feature, in particular the harmonic structure in low-energy parts, becomes clear after applying the WaveNet vocoder, and thus expected to be effective for ASR model training. Thus, we investigate the waveform-based data augmentation with TTS. Specifically, we compare the following three configurations.

4.1. Lmfb-output TTS and Lmfb-input ASR

In this method, lmfb features are directly generated by TTS (Fig. 4A). This method is computationally efficient since we do not generate waveforms. However, we change the setting of TTS to match the typical ASR systems. While typical TTS systems predict lmfb features based on 80 channels using a 50 ms window with a shift of 12.5 ms, we adopt a 25 ms window with a shift of 10 ms to compute
40-channel Lmfb, which is typically used in the ASR model. We train the Tacotron 2 model with this ASR-matched setting. We also train the Tacotron 2 model with TTS-matched setting and the ASR model with TTS-matched setting in an experiment.

4.2. Waveform-output TTS and Lmfb-input ASR

As general ASR systems input Lmfb features, we convert waveforms generated by TTS to Lmfb (Fig. 4B). As shown in Fig. 3, this Lmfb has better quality than the Lmfb generated by TTS. We compute 40-channel Lmfb features using a 25 ms window with a shift of 10 ms for the ASR input. It is the widely-used setting in ASR.

4.3. Waveform-output TTS and Waveform-input ASR

In this study, we design a seq2seq ASR system whose input is a raw waveform (Fig. 4C). This ASR system is an end-to-end ASR from a waveform to a word sequence. We describe the detail of the feature extraction part and the masking method in Sect. 5. The Tacotron 2-based model predicts 80-channel Lmfb features using a 50 ms window with a shift of 12.5 ms, and then the WaveNet model generates waveforms from these Lmfb features.

5. WAVEFORM-INPUT ASR

There are many previous works on learning acoustic models from raw waveforms [35–40]. Jaitly et al. [35] modeled a waveform using a restricted Boltzmann machine. Palaz et al. [36,38] used CNN for extracting acoustic features from speech signals. Tüské et al. [37] analyzed which acoustic features, including waveforms, were effective for training acoustic models. Sainath et al. [39] proposed feature extraction that consisted of two convolution layers for reducing temporal variations and reducing frequency variations. Ravanelli et al. [40,41] used SincNet, which is a learnable filter based on the band-pass filter. Recently, many works proposed feature extraction in an end-to-end manner. Tjandra et al. [42] proposed a CNN-based feature extraction architecture for an attention-based model and pretrained the model by minimizing the mean squared distance between Lmfb features and the output of the model. Zeghidour et al. [43] presented an end-to-end speech recognition system based on convolution layers without RNNs.

5.1. Feature Extraction

We adopt a CNN-based model proposed by Zeghidour et al. [43]. First, we apply a $2 \times 1$ filter initialized by a pre-emphasis filter ($[-0.97, 1]$). We then apply a CNN with several channels (40 in this work) and a 1-time sample stride. After taking absolute values, the Hanning window is applied to unify time frames. After processing with $\log(1 + \text{abs}())$, instance normalization [44] is applied to normalize across each channel. Finally, we use frame stacking [3] in which we stack and skip some frames to make a new super-frame. Figure 5 shows this feature extraction part and the masking method in Sect. 5.2.

After frame stacking, we apply the attention-based encoder-decoder model described in Sect. 2 to predict the symbol sequence. Based on the cross-entropy between the predicted sequence and the ground-truth sequence, we can update not only the parameters of the attention-based model but also those of the feature extraction part. We expect that the model can extract more effective features for speech recognition than Lmfb features. In this model, we do not conduct any pretraining or particular initialization except for the pre-emphasis filter. The CNN filters are initialized with random values drawn by He initialization [45].

5.2. Data Augmentation by Masking

In the attention-based model using Lmfb features, we can apply data augmentation methods such as SpecAugment [25]. In the waveform-input model, we apply data augmentation after instance normalization in a similar manner. However, in a preliminary experiment, we observe that the masking was not effective when updating all parameters including feature extraction during the training. Therefore, we first train the entire waveform-input ASR model without masking. We then fix the parameters of the feature extraction part and fine-tune the attention-based model part by using masking. The masks are randomly chosen since the order of filters is meaningless, unlike Lmfb features.
6. EXPERIMENTAL EVALUATIONS

6.1. Datasets and Tasks

All experiments are conducted with Japanese TTS and ASR systems. We use the JSUT corpus [46] to train a single speaker Tacotron 2 model. JSUT has a recording of 7,607 utterances of prompt texts read aloud by a female speaker with a total duration of ten hours. We converted the sampling rate to 16 kHz for all datasets. We used 33 phone classes as input. They include special tokens for pause, word boundary, and end of the sentence. For extracting word segmentation and phone sequence of texts, we used MeCab\(^2\), a CRF-based Japanese morphological analyzer.

We primarily used the the Corpus of Spontaneous Japanese (CSJ) [47] to train the ASR model and multi-speaker Tacotron 2 model. CSJ has two distinct sub-corpora, Academic Presentation Speeches (APS) and Simulated Public Speeches (SPS). APS consists of academic presentation speeches in nine different academic societies (engineering, humanities, and social and behavioral sciences). It has 986 speakers (male: 809, female: 177) with 162,259 utterances and 247.9 hours. SPS consists of simulated presentation speeches about everyday topics. It has 1,704 speakers (male: 799, female: 905) with 238,108 utterances and 281 hours. We added all distinct words that occur more than twice in the training data and special tokens ((sos), (eos), and (UNK)) to the vocabulary. The vocabulary sizes were 24,286 for SPS and 34,305 for combination of APS and SPS. CSJ provides the official testsets of APS and SPS. The APS testset has 1.83 hours of speech and 26,028 words, and the SPS testset has 1.31 hours of speech and 17,134 words. CSJ contains many hours of speech and 26,028 words, and the SPS testset has 1.31

We also trained neural language models with 3 layers of unidirectional LSTMs with 256 filters, and one-layer BiLSTM with 256 cells. The location-sensitive attention mechanism [7] summarizes the encoder outputs. The attention weight at each decoding step is calculated by 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 a length of 31 to the cumulative vector of the attention weights in all past decoding steps. The pre-net consists of two fully-connected linear layers with 256 ReLU units. We sum the pre-net output, the speaker embedding, and the encoded representation with the attention vector. The decoder network consists of 2-layer unidirectional LSTMs with 1,024 memory cells. The decoder LSTM outputs, together with the attention context vector, were passed through a linear projection layer to predict five frames of the target Lmf features.

Figure 6 shows examples of generated speech of the multi-speaker model. We generated these pieces of speech from the same text and different speaker IDs. It is confirmed that the multi-speaker model could produce various speakers’ speech as these Lmf features and waveform were different in terms of the length of speech and the spectrum patterns because the Tacotron 2 based model not only generates the Lmf features but also estimates the lengths of them.

For synthesizing the waveform, we used WaveNet [28] vocoder\(^3\) with conditioning on 80-dimensional Lmf features of the TTS setting\(^5\). The upsampling layer assumes

---

\(^2\)http://taku910.github.io/mecab

\(^3\)We evaluated the quality of TTS model in [48].

\(^5\)https://github.com/NVIDIA/nv-wavenet
the lmfb features with a frame shift of 12.5 ms. The dilation was 1, 2, 4, ..., 128, 1, ..., 128 repeatedly, and the total number of layers is 16. Each dilated CNN has 128 channels with a kernel size of 2. We used the PyTorch [50] toolkit to train all networks with Nvidia TITAN RTX.

The TTS system often fails to generate some pieces of speech correctly. For example, some synthesized speech samples were silent, and their lengths were too short. On the other hand, the lengths of some speech samples were too long. We discarded speech samples if they are not within some thresholds.

5In a preliminary experiment, we observed that the WaveNet with the typical ASR setting (40-channel, and 10 ms shift and 25 ms window) does not generate a waveform properly.

### 6.3. Results of Waveform-input ASR vs. Lmfb-input ASR

We compared the performance of two ASR systems in Table 1: one is the standard ASR, whose input is lmfb features. The other is the waveform-input ASR, whose input is raw waveforms. In this table, we evaluated the models trained using the SPS (281 hours) and APS+SPS (528.9 hours) datasets on the APS and SPS test sets. We also applied SpecAugment to the lmfb-input ASR, and the masking method presented in Sect. 5.2 to the waveform-input ASR. The waveform-input ASR performed comparable to or better than the lmfb-input ASR in all cases. There are significant differences in the WERs for the APS testset using SPS and the SPS testset using APS+SPS. For the APS testset using SPS training set, the waveform-input ASR with masking achieved WER of 20.92%, which is much better than the lmfb-input ASR with SpecAugment. We use the waveform-input ASR with masking and the lmfb-input ASR with SpecAugment as the baseline. We also use the models using APS+SPS dataset as the oracle model.

### 6.4. Results of Simulated Domain Adaptation to APS

In order to simulate a domain adaptation scenario, we chose SPS as a source domain and APS as a target domain since domain adaptation to academic topics from general topics is often required. Using the source domain data, we trained the baseline ASR model and the multi-speaker speech synthesizer. For the target domain, we assumed only text transcription data for adaptation. We generated speech data from the texts of the target domain to retrain the ASR model.

The results are shown in Table 2. By using the text-only data of APS, we synthesized about 209 hours of speech data. The baseline WERs were over 20%, as shown in Table 2. When we applied an LM shallow fusion using the APS text, the improvement was limited since the APS topics is different from the SPS topics. If we can prepare real speeches for the APS dataset, both the lmfb-input and waveform-input ASR models achieved WER of 9.23%. When we used the single-speaker Tacotron 2 model for synthesizing the additional training data, a large improvement from the baseline is obtained. Among them, "wave-
form-output TTS and lmfb-input ASR” is better than “lmfb-output TTS and lmfb-input ASR.” The best WER in the single speaker setting was 12.90% with the “waveform-output TTS and waveform-input ASR.” After the domain adaptation, the LM fusion leads to a large improvement.

The multi-speaker models obtained over 0.6% WER reduction from the single speaker models in all settings. In this experiment, we compared two settings of “lmfb-output TTS and lmfb-input ASR”: ASR-matched setting (40-channels using a 25 ms window with a shift of 10 ms) and TTS-matched setting (80-channels using a 50 ms window with a shift of 12.5 ms). It is shown that the former is outperformed by the latter, but they are behind the performance of the waveform-output using the WaveNet vocoder, which improved the quality of lmfb features. In this case, too, the best WER of the augmented model was 12.27% with the “waveform-output TTS and waveform-input ASR” with the multi-speaker model. This model improved the WER by 41.01% relatively from the model without domain adaptation. These results show that the waveform-output TTS using WaveNet makes the ASR model better than the lmfb-output TTS. The waveform-input ASR realized further improvement.

Table 3 shows how the adapted model recognizes unknown words that do not appear in the target domain. When we use only SPS vocabulary, 1,143 words of the APS testset cannot be recognized. Among them, 905 words (80.05%) are included in the augmented (SPS + APS) vocabulary. The “data augmentation method (waveform-output TTS and waveform-input ASR)” in Table 2 correctly recognized 782 out of 1,143 words (68.42%). It enhanced the ASR model’s language model capability and recognized a majority of new words.

### 6.5. Results of Adaptation to Newspaper Domain

**Leveraging a Large Amount of Newspaper Texts**

Next, we conducted adaptation to a newspaper domain. For this experiment, we used the JNAS dataset [51]. JNAS is a read speech corpus of Japanese newspaper articles. It has 85.5 hours for training data and 20 minutes for testset. For synthesizing training data for the JNAS testset, we also used an external language resource. We collected 500k sentences randomly from newspaper articles of Mainichi Shinbun, one of the major newspapers in Japan.

Table 4 shows the evaluation of adaptation to the newspaper domain using the JNAS testset. In this evaluation, we used the real speech of both CSJ-APS and CSJ-SPS (528.8 hours) to train the ASR and multi-speaker TTS models. We generated 58 hours speeches from the JNAS transcript. The original JNAS data provides 85.5 hours of real speeches. The generated speech is much shorter than the real one because we do not use the silence as an input label of TTS, and the silence, including short pause, is not inserted in the synthesized speech. We also generated 768.9-hour data from 500k newspaper articles. The lmfb-input and waveform-input ASR using only the CSJ dataset obtained WERs of 15.79% and 13.79%. They are very high because the CSJ domain is different from the JNAS domain. When we mixed the CSJ and original JNAS dataset, the WERs were 5.24% and 5.52%. The models that used the CSJ and synthetic JNAS dataset improved approximately 4~5% in WERs from the baseline CSJ model. When a large amount of speech was generated
using newspaper articles, a large improvement is achieved. The best WER was 7.95% with the “waveform-output TTS and waveform-input ASR.” This result showed that the waveform-input ASR enhanced with the data augmentation is most effective when preparing a large amount of synthesized data.

7. CONCLUSION

In this paper, we have presented the waveform-based data augmentation method for end-to-end ASR systems. To realize the waveform-output TTS, we use the WaveNet vocoder. The WaveNet vocoder makes better lmfb features, improving the ASR performance. To fully utilize waveform-output TTS, we have also designed the waveform-input ASR and a fine-tuning method by masking. We have shown that the masking method for the waveform-input ASR achieved comparable or better performance than the standard lmfb-input ASR with SpecAugment. We have also demonstrated that the waveform-output TTS and waveform-input ASR achieved better performance than the waveform-output TTS and lmfb-input ASR in two domain adaptation scenarios. Future work includes improvement of multi-speaker TTS model for spontaneous speech for generating better and more data set.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI under Grants 19J21031 and 20H00602.

REFERENCES

[1] A. Graves, S. Fernandez, F. Gomez and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” Proc. Int. Conf. Mach. Learn. (ICML) 2006, pp. 369–376 (2006).

[2] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” Proc. Int. Conf. Mach. Learn. (ICML) 2014, pp. 1764–1772 (2014).

[3] H. Sak, A. Senior, K. Rao and F. Beaufays, “Fast and accurate recurrent neural network acoustic models for speech recognition,” Proc. INTERSPEECH 2015, pp. 1468–1472 (2015).

[4] H. Sak, F. de Chaumont Quirty, T. Sainath and K. Rao, “Acoustic modelling with CD-CTC-SMBR LSTM RNNs,” Proc. Automatic Speech Recognition and Understanding (ASRU) 2015, pp. 604–609 (2015).

[5] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur Yi Li, H. Liu, S. Satheesh, A. Sriman and Z. Zha, “Exploring neural transducers for end-to-end speech recognition,” Proc. Automatic Speech Recognition and Understanding (ASRU) 2017, pp. 206–213 (2017).

[6] J. Chorowski, D. Bahdanau, K. Cho and Y. Bengio, “End-to-end continuous speech recognition using attention-based recurrent NN: First results,” Proc. Advances in Neural Information Processing Systems (NIPS) Workshop on Deep Learning, 10 pages (2014).

[7] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho and Y. Bengio, “Attention-based models for speech recognition,” Proc. Advances in Neural Information Processing Systems (NIPS) 2015, pp. 577–585 (2015).

[8] W. Chan, N. Jaitly, Q. Le and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2016, pp. 4960–4964 (2016).

[9] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2016, pp. 4945–4949 (2016).

[10] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao and N. Jaitly, “An analysis of “attention” in sequence-to-sequence models,” Proc. INTERSPEECH 2017, pp. 3702–3706 (2017).

[11] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2018, pp. 4774–4778 (2018).

[12] L. Dong, S. Xu and B. Xu, “Speech-transformer: A non-recurrence sequence-to-sequence,” Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2018, pp. 5884–5888 (2018).

[13] A. Zeyer, P. Bahar, K. Irie, R. Schlüter and H. Ney, “A comparison of transformer and LSTM encoder decoder models for ASR,” Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) 2019, pp. 8–15 (2019).

[14] A. Tjandra, S. Sakti and S. Nakamura, “Listening while speaking: Speech chain by deep learning,” Proc. IEEE Automatic Speech Recognition and Understanding (ASRU) 2017, pp. 301–308 (2017).

[15] A. Tjandra, S. Sakti and S. Nakamura, “Machine speech chain with one-shot speaker adaptation,” Proc. INTERSPEECH 2018, pp. 887–891 (2018).

[16] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo and K. Takeda, “Back-translation-style data augmentation for end-to-end asr,” Proc. Workshop on Spoken Language Technology (SLT) 2018, pp. 426–433 (2018).
17. S. Karita, S. Watanabe, T. Iwata, A. Ogawa and M. Delcroix, “Semi-supervised end-to-end speech recognition,” Proc. INTERSPEECH 2018, pp. 2–6 (2018).

18. S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa and T. Nakatani, “Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2019, pp. 6166–6170 (2019).

19. M. Mimura, S. Ueno, H. Inaguma, S. Sakai and T. Kawahara, “Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition,” Proc. Workshop on Spoken Language Technology (SLT) 2018, pp. 477–484 (2018).

20. 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,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2019, pp. 6161–6165 (2019).

21. 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,” Proc. Advances in Neural Information Processing Systems (NIPS) 2017, pp. 2962–2970 (2017).

22. J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis and Y. Wu, “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” Proc. INTERSPEECH 2017, pp. 4779–4783 (2017).

23. 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,” Proc. Int. Conf. Learning Representations (ICLR) 2018, 16 pages (2018).

24. 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).

25. 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,” Proc. INTERSPEECH 2019, pp. 2613–2617 (2019).

26. D. Griffin, D. Daddick and J. Lim, “Speech synthesis from short-time fourier transform magnitude and its application to speech processing,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 74, Vol. 9, pp. 61–64 (1984).

27. M. Morise, F. Yokomori and K. Ozawa, “World: A vocoder-based high-quality speech synthesis system for real-time applications,” IEICE Trans. Inf. Syst., E99.D, 1877–1884 (2016).

28. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” Arxiv (2016). [Online]. Available: https://arxiv.org/abs/1609.03499

29. N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman and K. Kavukcuoglu, “Efficient neural audio synthesis,” CoRR, vol. abs/1802.08435 (2018). [Online]. Available: http://arxiv.org/abs/1802.08435

30. R. Prenger, R. Valle and B. Catanzaro, “WaveGlow: A flow-based generative network for speech synthesis,” Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2019, pp. 3617–3621 (2019).

31. N. Li, S. Liu, Y. Liu, S. Zhao and M. Liu, “Neural speech synthesis with transformer network,” Proc. 33rd AAAI Conf. Artificial Intelligence (AAAI-19) 2019, pp. 6706–6713 (2019).

32. A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu and Z. Wu, “Speech recognition with augmented synthesized speech,” Proc. Automatic Speech Recognition and Understanding (ASRU) 2019, pp. 996–1002 (2019).

33. Y. Huang, L. He, W. Wei, W. Gale, J. Li and Y. Gong, “Using personalized speech synthesis and neural language generator for rapid speaker adaptation,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2020, pp. 7394–7398 (2020).

34. N. Rossenbach, A. Zeyer, R. Schluter and H. Ney, “Generating synthetic audio data for attention-based speech recognition systems,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2020, pp. 7064–7068 (2020).

35. N. Jaitly and G. Hinton, “Learning a better representation of speech waveforms using restricted Boltzmann machines,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2011, pp. 5884–5887 (2011).

36. D. Palaz, R. Collobert and M. Magimai-Doss, “Evaluating phoneme class conditional probabilities from raw speech signals using convolutional neural networks,” Proc. INTERSPEECH 2013, pp. 1766–1770 (2013).

37. Z. Tüske, P. Golik, R. Schlüter and H. Ney, “Acoustic modeling with deep neural networks using raw time signal for LVCSR,” Proc. INTERSPEECH 2014, pp. 890–894 (2014).

38. D. Palaz, M. Magimai-Doss and R. Collobert, “Convolutional neural networks-based continuous speech recognition using raw audio,” Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2015, pp. 4295–4299 (2015).

39. T. N. Sainath, J. R. Weiss, A. Senior, K. W. Wilson and O. Vinyals, “Learning the speech front-end with raw waveform CLDNNs,” Proc. INTERSPEECH 2015, pp. 1–5 (2015).

40. M. Ravanelli and Y. Bengio, “Speech and speaker recognition from raw waveform with SincNet,” arXiv preprint arXiv:1812.05920 (2018).

41. M. Ravanelli and Y. Bengio, “Speaker recognition from raw waveform with SincNet,” Proc. 2018 IEEE Spoken Language Technology Workshop (SLT) 2018, pp. 1021–1028 (2018).

42. A. Tjandra, S. Sakti and S. Nakamura, “Attention-based wav2text with feature transfer learning,” Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) 2017, pp. 309–315 (2017).

43. N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve and R. Collobert, “Fully convolutional speech recognition,” arXiv preprint arXiv:1812.06984 (2018).

44. D. Ulyanov, A. Vedaldi and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” arXiv preprint arXiv:1607.08022 (2016).

45. K. He, X. Zhang, S. Ren and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR) 2015, pp. 1026–1034 (2015).

46. 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).

47. K. Maekawa, H. Koiso, S. Furui and H. Isahara, “Semi-supervised end-to-end speech recognition,” Proc. Automatic Speech Recognition and Understanding Workshop (ASRU) 2019, pp. 947–9520 (2000).

48. S. Ueno, M. Mimura and T. Kawahara, “End-to-end speech synthesis for multiple speakers using the corpus of spontaneous Japanese,” Proc. Autumn Meet. Acoust. Soc. Jpn., pp. 1085–1086 (2018) (in Japanese).

49. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint, 1412.6980, pp. 1–15 (2014).

50. 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,” Proc. Advances in Neural Information Processing Systems (NIPS) Workshop on Deep Learning, 4 pages (2017).
Sei Ueno received the B.E. from Doshisha University, Kyoto, Japan in 2017 and M.E. degrees from Kyoto University in 2019. He is currently working toward the Ph.D. degree in Kyoto University. His research interests include automatic speech recognition and speech synthesis. He is a member of the Acoustical Society of Japan (ASJ) and the IEEE.

Masato Mimura received the B.E. and M.E. degrees from Kyoto University, Kyoto, Japan, in 1996 and 2000, respectively. He is currently a researcher in the School of Informatics, Kyoto University.

Shinsuke Sakai received his B.E. in 1982, M.E. in 1984, and Ph.D. in 2012 from Kyoto University, Kyoto, Japan. Between 1984–2002, he worked on speech and natural language processing at NEC Corp. Between 1991–1993 and 2002–2004, he worked at MIT, MA., U.S.A. as a visiting scientist at LCS and as a research engineer at CSAIL, respectively. He worked as a senior researcher at ATR in Kyoto, Japan between 2006–2009 and as an expert researcher at NICT, between 2009–2011. Currently, he is a researcher at Graduate School of Informatics, Kyoto University. His research interests include ASR and Speech Translation. He is a member of the IPSJ, the ASJ, and the IEEE.

Tatsuya Kawahara received B.E. in 1987, M.E. in 1989, and Ph.D. in 1995, all in information science, from Kyoto University, Kyoto, Japan. Currently, he is a Professor and the Dean of the School of Informatics, Kyoto University. He was also an Invited Researcher at ATR and NICT. He has published more than 400 academic papers on speech recognition, spoken language processing, and spoken dialogue systems. He has been conducting several projects including speech recognition software Julius, the automatic transcription system deployed in the Japanese Parliament (Diet), and the autonomous android ERICA. Dr. Kawahara is a board member of APSIPA and ISCA, and a Fellow of IEEE.