LEVERAGING SEQUENCE-TO-SEQUENCE SPEECH SYNTHESIS FOR ENHANCING ACOUSTIC-TO-WORD SPEECH RECOGNITION

Masato Mimura, Sei Ueno, Hirofumi Inaguma, Shinsuke Sakai, Tatsuya Kawahara

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

ABSTRACT

Encoder-decoder models for acoustic-to-word (A2W) automatic speech recognition (ASR) are attractive for their simplicity of architecture and run-time latency while achieving state-of-the-art performances. However, word-based models commonly suffer from the out-of-vocabulary (OOV) word problem. They also cannot leverage text data to improve their language modeling capability. Recently, sequence-to-sequence neural speech synthesis models trainable from corpora have been developed and shown to achieve naturalness comparable to recorded human speech. In this paper, we explore how we can leverage the current speech synthesis technology to tailor the ASR system for a target domain by preparing only a relevant text corpus. From a set of target domain texts, we generate speech features using a sequence-to-sequence speech synthesizer. These artificial speech features together with real speech features from conventional speech corpora are used to train an attention-based A2W model. Experimental results show that the proposed approach improves the word accuracy significantly compared to the baseline trained only with the real speech, although synthetic part of the training data comes only from a single female speaker voice.

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

1. INTRODUCTION

Deep learning-based hybrid acoustic models have drastically improved the performance of automatic speech recognition (ASR) [1]. It was recently reported that even a human-level recognition performance can be achievable when the hybrid models are coupled with bidirectional LSTMs and very deep convolutional networks with residual connections [2, 3]. However, in exchange for these excellent performances, these ASR systems have very complicated structures consisting of complex decoders, large language models, and carefully designed pronunciation dictionaries. They have a large runtime latency and a limited portability.

In the mean time, we have seen a rapid development of alternative sequence-to-sequence (seq2seq) approaches to speech recognition based on connectionist temporal classification (CTC) [4, 5, 6, 7], attention-based encoder-decoder models [8, 9, 10, 11] and RNN-transducers [12, 13]. Their remarkable advantage is that they get rid of dependency on frame-level probabilistic state transition models such as HMMs. An extreme example of the seq2seq approach is acoustic-to-word (A2W) models [14, 15, 16] which directly map acoustic signals into word sequences. They do not require any external decoders, leading to an extremely simplified architecture of ASR systems and very low latency. Outputting words rather than phones or characters is also an advantage since it requires no post-processing to utilize the ASR output in the subsequent natural language processing such as dialogue, translation and information query.

We have shown in [16] that attention-based models in which the label output probability is explicitly conditioned on the past output are significantly better than CTC-based models for word-level seq2seq speech recognition. We have also demonstrated in [17] that an attention-based A2W model achieved a WER reduction of 25.3% relative from a state-of-the-art hybrid DNN-HMM system with a decoding speed faster by a factor of 50. In this paper, we further seek to improve the attention-based A2W speech recognition system.

Although A2W models are attractive for multiple reasons we pointed out above, they have some drawbacks compared to conventional systems based on phones or characters. The most important problem is that they cannot predict posterior probabilities for unknown words which did not appear in the training data and have no mechanism to add new words to its vocabulary. This is a serious problem, since an ASR system may encounter a number of words specific to the domain which are out of vocabulary (OOV) of the system, when it is deployed in a particular task domain.

In addition to this OOV word problem, there is another issue that A2W models have no way to utilize text data directly to improve its language modeling capability. It is because A2W models are trained in a seq2seq manner from pairs of speech and word labels. In other words, they are constrained to learn a probability distribution over word sequences only from a limited amount of labeled speech, although it could be estimated more reliably from a large collection of texts covering many linguistic phenomena. It also implies that we cannot perform domain adaption of A2W models using relevant texts in a target domain.

Recently, sequence-to-sequence neural speech synthesis models trainable from corpora have been developed and shown to achieve naturalness comparable to recorded human speech [18, 19, 20]. In this paper, we propose a novel approach to address the problems inherent to A2W models exploiting the current seq2seq speech synthesis technologies. In this approach, we perform training data augmentation by generating speech features from a set of target domain texts using a seq2seq speech synthesizer. These artificial speech features together with real speech features from conventional speech corpora are used to train an attention-based A2W model. We only need to prepare relevant texts to the application domain of the speech recognition system, which are much more easily accessible than labeled speech data. We can expand the vocabulary and enhance the language modeling capability of the A2W model using new words and word contexts included in the augmented data. Moreover, our method makes it possible to integrate an A2W model with an RNN-based external language model in a natural way by ensuring that they have the same vocabulary. We also explore an encoder freezing learning technique to prevent the undesirable effects from the uniformness of synthesized speech.

The experimental evaluations show that the proposed approach implemented with a speech synthesizer trained using speech data from a single female speaker significantly improved the speech recognition performance compared to the baseline models trained using only the real speech data.

2. ATTENTION-BASED SPEECH RECOGNITION

This section presents a brief review on attention-based seq2seq speech recognition, including a decoding algorithm based on beam search. In attention-based speech recognition, we model seq2seq mapping between speech and label sequences using an encoder-decoder architecture [8, 9]. This architecture has two distinct sub-networks. One is the encoder which transforms an acoustic feature sequence of length T to a sequential representation. Based on this encoded acoustic information, the other decoder sub-network predicts a label sequence whose length L is usually shorter than the input length T. The decoder uses only a relevant portion of the encoded sequential representation for predicting a label at each time step using the attention mechanism. The encoder is implemented as multi-layer bidirectional RNN such as LSTM [21], and the decoder usually consists of a single layer of unidirectional LSTM followed by a softmax output layer.

The attention-based models are formulated as follows. The encoder transforms input acoustic features $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ to a sequential representation $\boldsymbol{H} = (\boldsymbol{h}_1, ..., \boldsymbol{h}_T)$ that summarizes the characteristics of the input. In the following decoding step, the hidden state activation of the RNN-based decoder at the l-th time step is computed as:

$$\boldsymbol{r}_l = Recurrency\left(\boldsymbol{r}_{l-1}, \boldsymbol{g}_l, \boldsymbol{y}_{l-1}\right), \tag{1}$$

where g_l and y_{l-1} denote the "glimpse" at the l-th time step and the predicted label at the previous step, respectively. The glimpse g_l is a weighted sum of the encoder output sequence:

$$\boldsymbol{g}_l = \sum_t \alpha_{l,t} \boldsymbol{h}_t, \tag{2}$$

where $\alpha_{l,t}$ is an attention weight for $\boldsymbol{h}_t$. It is calculated as:

$$e_{l,t} = Score(\boldsymbol{r}_{t-1}, \boldsymbol{h}_t, \boldsymbol{\alpha}_{l-1}), \tag{3}$$

$$\alpha_{l,t} = \exp(e_{l,t}) / \sum_{t'=1}^{T} \exp(e_{l,t'}). \tag{4}$$

There are many choices for implementation of the score function (4). In this paper, we adopt the hybrid location and content-based attention mechanism [9] as follows:

$$e_{l,t} = \boldsymbol{w}^T tanh(\boldsymbol{W}\boldsymbol{r}_{t-1} + \boldsymbol{V}\boldsymbol{h}_t + \boldsymbol{U}\boldsymbol{f}_{l,t} + \boldsymbol{b}), \tag{5}$$

$$\boldsymbol{f}_l = \boldsymbol{F} * \boldsymbol{\alpha}_{l-1}, \tag{6}$$

where $*$ denotes one-dimensional convolution. Using $\boldsymbol{g}_l$ and $\boldsymbol{r}_{l-1}$, the decoder predicts the next label y_l as:

$$y_l \sim Generate\left(\boldsymbol{r}_{l-1}, \boldsymbol{g}_l\right), \tag{7}$$

where the Generate function is implemented as:

$$\boldsymbol{R} \tanh\left(\boldsymbol{P}\boldsymbol{r}_{l-1} + \boldsymbol{Q}\boldsymbol{g}_l\right). \tag{8}$$

The objective for training the attention models is a cross entropy loss calculated between the predicted label sequences and the target label sequences.

Algorithm 1 ForwardBeamSearch(B, $\boldsymbol{X}$)

1: F : set of completed label sequences
2: $NewSeqs$: set of label sequences at current output time
3: $Seqs$: set of label sequences up to the last output time
4: $F \Leftarrow \{\phi\}$, $score(\langle \text{sos} \rangle) = 0$, $Seqs \Leftarrow \{(\langle \text{sos} \rangle)\}$
5: $b \Leftarrow B$
6: **while** $b > 0$ **do**
7: $NewSeqs \Leftarrow \{\phi\}$
8: **for** sequence $\boldsymbol{s} \in Seqs$ **do**
9: $Y \Leftarrow$ The b best words in terms of $p(y|\boldsymbol{s}, \boldsymbol{X})$
10: **for** word $y \in Y$ **do**
11: $\boldsymbol{s}^+ \Leftarrow concat(\boldsymbol{s}, y)$
12: $score(\boldsymbol{s}^+) = score(\boldsymbol{s}) + \log(p(y|\boldsymbol{s}, \boldsymbol{X}))$
13: **if** $y = \langle \text{eos} \rangle$ **then**
14: add $\boldsymbol{s}^+$ to F
15: $b \Leftarrow b - 1$
16: **else**
17: add $\boldsymbol{s}^+$ to $NewSeqs$
18: **end if**
19: **end for**
20: **end for**
21: $Seqs \Leftarrow$ The b best sequences in $NewSeqs$ in terms of $score(\boldsymbol{s}^+)$
22: **end while**
23: **return** set of sentence candidates F

A runtime decoding algorithm for word-level attention-based models is presented in Algorithm 1. This algorithm returns the B-best sentence candidates using a decreasing beam width initialized with B. It is simple since we do not need to incorporate external dictionaries or language models. $\langle \text{sos} \rangle$ and $\langle \text{eos} \rangle$ are special symbols for the start and end of a sentence. The posterior probability of a word at each decoding step $p(y|\boldsymbol{s}, \boldsymbol{X})$ on line 9 and 12 is calculated using formulas from (1) to (8). After performing Algorithm 1, we rescore each sentence candidate $\boldsymbol{s}$ in F using an insertion penalty λ as follows:

$$rescore(\boldsymbol{s}) = score(\boldsymbol{s}) - \lambda|\boldsymbol{s}|, \tag{9}$$

where $|\boldsymbol{s}|$ is the length of sequence $\boldsymbol{s}$, and $score(\boldsymbol{s})$ is the value calculated on line 12 of Algorithm 1. We output the sentence with the largest $rescore(\boldsymbol{s})$.

3. SEQUENCE-TO-SEQUENCE SPEECH SYNTHESIS

Seq2seq speech synthesis is a technology for generating speech directly from text which does not require complex multistage pipelines unlike traditional text-to-speech (TTS) systems. Although a number of distinct architectures have been proposed [22, 18, 19, 20], these systems are commonly composed of an attention-based feature prediction network which maps character embedding to mel-scale spectrograms or vocoder parameters, followed by a vocoder which synthesize time-domain waveforms from these predicted features. In this paper, we use Tacotron 2 [19] which has a relatively simple architecture similar to our A2W model and was shown to achieve striking naturalness. Here, we describe in depth the feature prediction network of Tactron 2 along with the network configurations we used in our experiments. Note that we do not need a vocoder, since what we need is mel-scale spectrograms for training a recognizer rather than waveforms.

The feature prediction network consists of the character encoder and the attention-based decoder subnetworks. The former summa-

rizes the input character sequence and outputs a sequential representation of the same length as the input sequence. The latter predicts a sequence of Mel-spectrograms in an autoregressive way conditioned on the encoder outputs.

In the encoder subnetwork, each input character is first mapped to a 512-dimensional continuous vector. This mapping is performed via a learnable character embedding layer. These character embeddings are fed to a stack of 3 convolutional layers. Each layer convolves 512 filters of size $(5, 1)$ to its input, followed by batch normalization [23] and ReLU activations. The output of the final convolutional layer is input to a one-layer bidirectional LSTM with 256 memory cells in each direction to generate the encoded features.

The decoder subnetwork predicts five consecutive frames of Mel-spectrograms at each decoding step based on the encoder outputs and the final frame of the predicted features at the previous step, as follows. The encoder outputs are summarized using the location-sensitive attention mechanism [9]. 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. The sum of these vectors is normalized using a formula equivalent to (4) in the last section after applying tanh activation to generate the attention weight of the current decoding step. Meanwhile, the last one frame of the 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 and the attention vector are concateneted to be provided to a 2-layer unidirectional LSTM with 1024 memory cells. The LSTM outputs together with the attention context vector are passed through a linear projection layer to predict the 5 frames of the target Mel-spectrograms.

After finishing all decoding steps, the predicted Mel-spectrogram sequence are processed by a 5-layer convolutional post-net which predicts a residual to add to the original prediction. Each convolutional layer has 512 filters of size $(5, 1)$. Batch normalization and tanh activations are applied on all but the final layer. The entire network is trained using the L1 distance between the predictions and the target spectrograms as the loss function[1].

The decoder simultaneously predicts if the output sequence has completed or not at each time step. It is judged based on the decoder LSTM output and the attention context. More precisely, the concatenation of these vectors are projected down to a scalar and applied with sigmoid activation to calculate the probability that the decoding is complete. In the runtime of speech synthesis, we stop generating acoustic features if this probability exceeds 0.5.

Seq2seq speech synthesis techniques are commonly shown to achieve very high mean opinion scores (MOS) using unified simple architectures which are easy to implement and optimize. They also have an advantage that they generate a sequence of acoustic features we can use directly as the input to speech recognition systems. These are the reasons why we adopt seq2seq speech synthesis in our proposed approach instead of conventional unit selection-based concatenative methods or HMM-based statistical parametric methods.

4. PROPOSED METHOD

Before introducing our proposed methods, we recap the problems with A2W models. First, they cannot predict posterior probabilities of words which did not appear in the training data. Therefore, they are never able to recognize a sentence with these OOV words correctly. Second, since the entire network of an A2W model is trained in a seq2seq manner from pairs of speech and word sequences, there is no direct way to enhance language modeling capability of the model even if a large collection of texts is available.

A simple way to address the OOV word problem is to combine A2W models with character-based models [24, 16]. In this approach, speech is decoded using both of an A2W model and an acoustic-to-character (A2C) model. When an OOV word is detected with the A2W model, the character sequence from the A2C model appearing in the speech segment corresponding to the OOV word is output instead of the OOV symbol. These methods assume that OOV words are detected with a high recall rate. A more recent approach called the modular training of A2W models aims to resolve both of the two problems [25]. An A2W model is factorized into two modules, an acoustic-to-phoneme (A2P) model and a phoneme-to-word (P2W) model. This is similar to the traditional hybrid systems which combine acoustic, transition, pronunciation and language models in a modular way. The P2W model maps phone sequences to word sequences and can be trained from text data without speech.

In this paper, we propose a more direct approach for enhancing A2W models inspired by the recent progress in seq2seq speech synthesis. Our method makes it possible to train A2W models from arbitrary sentences leveraging a state-of-the-art speech synthesis technique. It does not require any modification to the simple architecture of A2W models or additional lower-level models retaining the full strength of vanilla A2W models.

4.1. Training data augmentation

We exploit seq2seq speech synthesis reviewed in the previous section for augmenting training data for attention-based A2W models using relevant texts to a target domain. Addition of the text with right words and word contexts unseen to the baseline model, trained only with an available real speech corpus, will contribute to expand the vocabulary and improve the language modeling capability of the A2W model.

In our first attempt of the proposed approach in this paper, we work with the Japanese language. Since the number of distinct characters in Japanese are a few thousands, unlike languages such as English where the number is a few tens, it is obvious that we will have too many number of parameters in the model as well as a serious sparse data problem if characters are used as input units. Therefore, we opted to choose phones as the input unit rather than characters in the original Tacotron2.

We use a 40-dimensional vector consisting of 40-channel log Mel-filterbank (lmfb) outputs as the target of the synthesizer network, which is the same acoustic feature we use for our speech recognition systems. This is because we want to use an output sequence from the synthesizer directly as input to the A2W model in order to avoid an artifact caused by performing additional feature conversion. The frame window length and frame shift are set to be 10 ms and 25 ms following the standard setting in speech recognition.

The procedure of the data augmentation is as follows. We collect texts from a target domain where we want to perform speech recognition. Since words are not separated by spaces in Japanese, each sentence in the collected data is first processed by a Japanese morphological analyzer to separate it into words and simultaneously obtain the pronunciation of each word. The sequence of phones and special symbols representing word boundaries are fed into the

[1]The original Tacotron 2 used the mean squared error (MSE), which gave slightly worse results than the L1 loss in our preliminary experiments.

seq2seq speech synthesizer to generate a mel-spectrogram for the sentence. The set of synthesized log mel-spectral features and corresponding word sequences are added to the conventional training data coming from the real speech corpora to train the attention-based A2W speech recognition model.

4.2. Encoder freezing learning of attention model

One concern with the data augmentation using a speech synthesizer is the possibility that artificial speech is much less acoustically diverse than real speech. This is more likely when the synthesizer is trained using a typical speech synthesis corpus consisting of voices from a single speaker. We thought that the augmented artificial data can be harmful for training the acoustic encoder subnetwork of A2W models, while they should be essential for enhancing the decoder.

Therefore, we adopted the *encoder freezing learning* of attention-based models which we investigated for domain transfer learning of A2W models [26]. In this framework, the parameters of the acoustic encoder are copied from a model pretrained on real speech, and they are fixed during the training using the augmented data set consisting of the artificial and real data. In other words, the encoder subnetwork for summarizing the acoustic information is trained only on real speech data, while the decoder layer which is responsible for predicting word transition probabilities is tuned using the full set of the augmented data. We show the procedure of the encoder freezing learning in Fig. 1. The decoder is designed to have an softmax output layer whose size is the same as the expanded vocabulary and its parameters are initialized with random values. This framework aims to prevent the undesirable effects from the uniformness of synthesized speech, while taking advantage of the diversity coming from a large text corpus.

4.3. Language model integration

It was shown to be very effective to incorporate external language models in speech recognition using conventional seq2seq models based on characters [27, 28, 29]. This is because language models can be trained on a large set of texts covering much richer linguistic information than a limited amount of labeled speech data used for training the seq2seq models.

On the other hand, integration of A2W models and external word-level language models has not been investigated well, mainly because the vocabularies of the A2W model and the language model trained on larger data are inevitably different. It is far from trivial to combine them to calculate scores for words in each decoding step. However, based on our data augmentation method, we can easily achieve a shallow fusion of the A2W model and the language model, because we can train both of them on the same word sequences and ensure that they have the same vocabulary. For decoding with the external language model, a small modification is required to line 12 in Algorithm 1 as:

$$score(\boldsymbol{s}^+) = score(\boldsymbol{s}) + \log(p(y|\boldsymbol{s}, \boldsymbol{X})) + \mu \log(p_{LM}(y|\boldsymbol{s})), \quad (10)$$

where $p_{LM}(y|\boldsymbol{s})$ is the posterior probability of word y given word history $\boldsymbol{s}$, which is calculated with a LSTM-based neural language model. It is trained using all texts from the source and target domain. μ is the weight for the language model score. The A2W model and the external language model use context in different ways and are expected to complementarily contribute to improve speech recognition performance[2].

[2]For example, A2W models tend to give similar probabilities to words with similar pronunciations, while language models make predictions independent of word pronunciations.

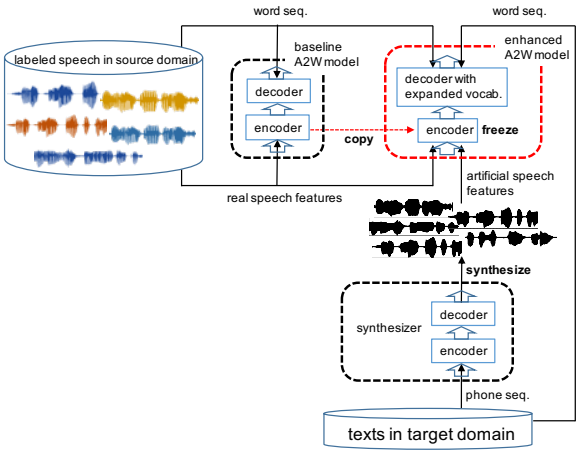

Fig. 1. Encoder freezing learning for enhancing A2W model using real and artificial training data

5. EXPERIMENTAL EVALUATIONS

5.1. Data

We evaluated our methods through speech recognition tasks using two standard Japanese corpora: the Corpus of Spontaneous Japanese (CSJ) [30] and Japanese Newspaper Article Sentences (JNAS). CSJ includes two distinct subcorpora, namely, CSJ-APS and CSJ-SPS. CSJ-APS consists of academic presentation speeches on several topics such as science, engineering, humanities and social science. CSJ-SPS consists of simulated presentation speeches on three general themes. These subsets have their own official test sets, namely, CSJ-TESTSET1 and CSJ-TESTSET3. While CSJ comprises spontaneous utterances, JNAS consists of newspaper articles read aloud.

5.2. A2W model

A 40-dimensional vector consisting of 40-channel log Mel-scale filterbank (lmfb) outputs is used as acoustic features for attention-based A2W models. Non-overlapping frame stacking [6] was applied to these features in which we stack and skip three frames to make a new super frame. The acoustic encoders in our attention models consist of 5-layers of bidirectional LSTMs with 320 cells. Dropout [31] was used for training each LSTM layer with a dropout rate of 0.2. The decoder consists of a 1-layer LSTM with 320 cells and a softmax output layer with the nodes for vocabulary words, ⟨sos⟩, ⟨eos⟩ and ⟨OOV⟩ special token for words which appeared less than 3 times in the training sets. The vocabularies of the A2W models consist of words which appearerd more than two times in the training sets. We used Adam [32] for optimizing network parameters. We also used gradient clipping with a threshold of 5.0. All network parameters were initialized with random values drawn from a uniform distribution with range (-0.1, 0.1). We also used scheduled sampling [33] and label smoothing [34] to improve the optimization. In the experiments for integrating A2W models and RNN-based external language models, we used 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 [35] to implement the A2W models

Table 1. ASR performance for two CSJ test sets (WER(%))

	TESTSET1 (APS)	TESTSET3 (SPS)
SPS (real) (baseline)	24.92	11.43
+ language model (SPS)	25.37	11.27
SPS (real) + APS (synthesized)	**19.40**	**10.42**
+ language model (APS + SPS)	**18.74**	**10.22**
SPS (real) + APS (synthesized) with encoder freezing learning	**19.09**	**11.29**
+ language model (APS + SPS)	**18.38**	**11.07**

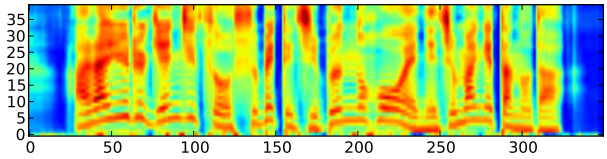

Fig. 2. An example of synthesized lmfb features from a Japanese sentence *"arayuru geNjitsu o subete jibuN no ho: e nejimage ta no da"*. The vertical axis designates lmfb channel numbers and the horizontal axis designates the time frames with 10ms shift.

and the LSTM-based language models. The beam width B was set to be 4 in all speech recognition experiments.

5.3. Seq2seq speech synthesizer

As we described in section 4.1, The seq2seq speech synthesizer is trained to output 40-channel log mel-filterbank features. For word segmentation and pronunciation annotation of texts, we used Mecab[3], a CRF-based Japanese morphological analyzer, combined with the "Unidic" word dictionary[4]. As training data, we adopted JSUT (Japanese speech corpus of Saruwatari-lab., University of Tokyo) corpus [36]. It is a recording of 7,607 prompt texts read aloud by a female speaker with total duration of ten hours. The prompt sentences were tokenized using Mecab and the corresponding phone sequences were obtained. These automatically generated phone labels are used without human checking and, therefore, are expected to include a certain amount of erroneous ones. We used 33 phone classes including special tokens for pause, word boundary and the end of a sentence. We also used PyTorch [35] to implement the Tacotron2-based feature prediction network using phones as input units. We used dropout with a dropout rate of 0.5 in all convolutional layers following the recipe in [19]. The LSTM layers in the decoder subnetwork are regularized using zoneout [37] with a probability of 0.1. As in the training of A2W models, we used the Adam optimizer. Fig. 2 depicts an example of speech synthesized from a sentence in the training set of JNAS [5].

5.4. Adaptation between two spontaneous speech domains

We first examined how proposed approach works effectively in adapting a seq2seq ASR model trained with spoken presentations in general everyday life topic domain to the academic presentation domain on audio, speech and language processing, using SPS ("Simulated" Presentation Speech) and APS (Academic Presentation Speech) subcorpora of the CSJ corpus.

The baseline model was trained using SPS training set consisting of 281 hours of spontaneous speech. Its vocabulary consisting of all the distinct words occurring more than twice in the training set comprises 24,826 words. For the adaptation for Academic presentation domain, we synthesized 282,235 utterances from the transcript of the APS training set using the seq2seq speech synthesis model of single female voice described in the last subsection. The ASR model for the target domain was trained by adding this 255 hours of synthetic speech to the baseline 281-hour SPS training set. By the addition of the synthetic APS set, the vocabulary of the model was expanded by around 10k words and reached 34,331 words. The training was performed with and without encoder freezing described in Section 4.2.

The second column in Table 1 ("TESTSET1 (APS)") shows the result for the target domain test set[6]. TESTSET1 is composed of ten academic presentation speeches by ten male speakers. The baseline model trained only with the SPS training data gave a relatively high WER of 24.9% and integration of the language model trained on the SPS training set transcript rather degraded the accuracy slightly. On the other hand, with the enhanced model trained by the proposed approach using synthetic speech, the WER for the target domain test set was reduced by 5.5 points to 19.4%. The encoder freezing gave a further reduction of 0.31 points. The LSTM language model trained with the transcripts of both SPS and APS training data consistently yielded significant improvements. The best model gave an WER of 18.38%, which corresponds to a 26.2% relative improvement over the baseline. We observed that a number of words unknown to the baseline model were correctly recognized with the enhanced models with a enlarged vocabulary. These words include, for example, "efuzero" (F0) and "oNcho" (tone) that often appears in the context of audio, speech, and language processing. It is noteworthy that we had these substantial improvements with the all male-speaker test set by augmenting the training data with synthesized speech of a single female voice. It may be because the synthetic speech by the seq2seq synthesizer has a high enough naturalness to contribute to the training of the decoder part of the A2W model as well as it does not hurt the training of the encoder part of the model which has a inherent discriminative nature resistant to the quantitative dominance of a single speaker.

We also looked at the influence of the domain adaptation to the original source domain (SPS) test set and show the results on the third column of Table 1. The test set of CSJ-SPS, TESTSET3, is composed of ten simulated presentation speeches by five male and five female speakers. Although the WER obtained with the baseline model was already low, the data augmentation using artificial data gave a further significant improvement. We understand that this

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

[4]http://unidic.ninjal.ac.jp/

[5]We cannot assess the naturalness of the synthesized features directly, because it is difficult to reconstruct waveforms from 40-dimensional lmfb features. However, we found that the synthesized utterances were recognized almost perfectly using an A2W model trained on real speech.

[6]The OOV rates of TESTSET1 for the baseline and enhanced models are 4.48% and 0.96%, respectively.

Table 2. ASR performance for JNAS test set

	training data amount (hours)	WER (%)
CSJ (real) (baseline)	528.8	17.71
JNAS (real) (oracle)	85.5	21.16
CSJ (real) + JNAS (real) (oracle)	614.2	5.16
CSJ (real) + JNAS (synthesized)	596.5	**11.21**
CSJ (real) + JNAS (synthesized) with encoder freezing learning	596.5	**9.01**
+ language model (CSJ + JNAS)	-	**8.71**
CSJ (real) + JNAS (synthesized) + Mainichi (synthesized) with encoder freezing learning	1502.10	**7.78**
+ language model (CSJ + JNAS + Mainichi)	-	**7.40**

gain is due to the improved capability for estimating word transition probabilities learned from enhanced training data based on a considerable amount of text. However, a different tendency was observed in the encoder freezing learning compared to the cross-domain results. For the SPS test set, encoder freezing training gave an improvement from the baseline, but was not as good as training of the whole model. Again, the language model integration was effective for both of the enhanced models.

5.5. Adaptation to newspaper domain

We also attempted a more challenging domain adaptation from spontaneous presentation to read speech of newspaper articles, where the difficulty comes from the large differences in speaking style and information content. This time, the baseline model was trained using the whole real speech training data of the CSJ corpus comprising both of APS and SPS. Its vocabulary consists of 32,573 words. For the target domain ASR model, we synthesized a set of utterances from 49,576 news paper prompt texts of the JNAS training data and added them to CSJ training set to train the A2W model. By the addition of synthetic JNAS data, its vocabulary turned out to be 35,795 words. In order to manage the consistency of word boundary definitions in using two distinct corpora, which is not obvious as a matter of fact with the Japanese language, we tokenized and generated pronunciations of the both corpora using Mecab. The label error rate of the automatically generated phone sequences calculated using the manual transcriptions as reference was 6.29%. The results of the speech recognition experiment using the test set of JNAS are summarized in Table 2. The test set is composed of 200 sentences spoken by 22 male and 22 female speakers.

The enhanced model with the proposed method gave a much lower WER than the baseline trained on all utterances in CSJ. We also observed an interesting fact that the encoder freezing learning was quite effective in this experiment which gave an additional improvement of 2.2 points. One possible reason for this is that the speaking style of the test set speech is more different from the added synthetic speech than what we expect about two sets that are both coming from read speech. Integrating the external language model also yielded a WER reduction, resulting in an improvement of 50.1% relative over the baseline.

We also tried to further increase the amount of training data using an external language resource. We randomly picked 500k sentences from articles of Mainichi Shimbun, one of the major newspapers in Japan, published in 2000 and 2001. The artificial speech features synthesized from these newspaper sentences are used for training a new A2W model together with CSJ and the artificial data generated from sentences of JNAS. The vocabulary size of the new model is 41,890. By performing this additional data augmentation, the WER was further reduced by 1.23 points as shown in the seventh row in Table 2. The language model integration was also effective. From these results, we confirm that relevant texts from outside of the target corpus can be utilized to improve the performance of the A2W model as well.

For comparison, we also built two oracle models. One is the model trained using the real utterances in JNAS, which gave a poor speech recognition performance due to the small amount of training data. The other is the A2W model trained on real data from both of CSJ and JNAS. This model yielded a very low WER. It is encouraging to note that the enhanced model using artificial augmented data with our proposed method gave a gain corresponding to as large as 70% of the gain coming from adding the true speech data of the domain in this oracle model.

6. CONCLUSION

We showed that we can significantly enhance the speech recognition performance of A2W models using only text data via phone-based seq2seq speech synthesis. The proposed method makes it possible to train an A2W model from arbitrary sentences and effectively expand the vocabulary and improve language modeling capability of the model. We demonstrated the effectiveness of the method thorough two cross-domain speech recognition experiments.

Previously a combination of seq2seq speech synthesis and recognition has been investigated in [38]. The novelty of our contribution is that we exploited seq2seq speech synthesis for enhancing the A2W model and demonstrated that we can significantly improve speech recognition performance using artificial training data, while the main issue in [38] was investigating a deep learning-based speech chain model.

Although our method extremely expands the vocabulary of the A2W model, it is still not totally free from the OOV word problem. It can be combined with existing methods for addressing problems of A2W models such as [24, 16, 25] to achieve a further improvement. While one of the important findings in this paper is that we can significantly improve the performance of A2W models using a speech synthesizer trained on a typical speech synthesis corpus consisting of speech from a single speaker, it would achieve a further improvement if we can train a synthesizer on a large speech corpus containing many speakers and many speaking styles. Another possible direction is, for example, investigating joint training of the seq2seq synthesizer and the A2W model.

7. ACKNOWLEDGEMENTS

This work was supported by ERATO Ishiguro Symbiotic Human-Robot Interaction program (Grant Number JPMJER1401), Japan.

8. REFERENCES

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

[2] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," in *arXiv preprint arXiv:1610.0525*, 2016.

[3] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *arXiv preprint arXiv:1703.02136*, 2017.

[4] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks," in *Proc. of the 23st International Conference on Machine Learning*, 2006, pp. 369–376.

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. of the 31st International Conference on Machine Learning*, 2014, pp. 1764–1772.

[6] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *arXiv preprint arXiv:1607.06947*, 2015.

[7] H. Sak, F. de Chaumont Quitry, T. Sainath, and K. Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNs," in *ASRU*, 2015, pp. 604–609.

[8] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: first results," in *NIPS: Workshop Deep Learning and Representation Learning Work- shop*, 2014.

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

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

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

[12] A. Graves, "Sequence transduction with recurrent neural networks," in *LCML*, 2012, pp. 4945–4949.

[13] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

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

[15] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Interspeech*, 2017, pp. 3707–3711.

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

[17] M. Mimura, S. Sakai, and T. Kawahara, "Forward-backward attention decoder," in *INTERSPEECH*, 2018.

[18] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017, pp. 4006–4010.

[19] 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," in *ICASSP*, 2018, pp. 4779–4783.

[20] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, and S. Narang, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *ICLR*, 2018.

[21] S.Hochreiter and J.Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *ICLR*, 2017.

[23] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of ICML*, 2015, pp. 448–456.

[24] J. Li, G. Ye, R. Zhao, J. Droppo, and Y. Gong, "Acoustic-to-word model without OOV," in *ASRU*, 2017.

[25] Z. Chen, Q. Liu, H. Li, and K. Yu, "On modular training of neural acoustics-to-word model for LVCSR," in *ICASSP*, 2018, pp. 4754–4758.

[26] 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 *INTERSPEECH*, 2018.

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

[28] T. Hori, S. Watanabe, and J. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *ASRU*, 2017.

[29] 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 *ICASSP*, 2018, pp. 5824–5828.

[30] K.Maekawa, "Corpus of Spontaneous Japanese: Its design and evaluation," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003, pp. 7–12.

[31] 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*, vol. 15, pp. 1929–1958, 2014.

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

[33] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *NIPS*, 2015, pp. 1171–1179.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[35] 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," 2017.

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

[37] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing RNNs by randomly preserving hidden activations," in *LCLR*, 2017.

[38] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *ASRU*, 2017.