MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Cycle-Consistently Training for End-to-End Speech Recognition

Hori, T.; Astudillo, R.; Hayashi, T.; Zhang, Y.; Watanabe, S.; Le Roux, J.

TR2019-002 March 29, 2019

Abstract

This paper presents a method to train end-to-end automatic speech recognition (ASR) models using unpaired data. Although the endto-end approach can eliminate the need for expert knowledge such as pronunciation dictionaries to build ASR systems, it still requires a large amount of paired data, i.e., speech utterances and their transcriptions. Cycle-consistency losses have been recently proposed as a way to mitigate the problem of limited paired data. These approaches compose a reverse operation with a given transformation, e.g., text-to-speech (TTS) with ASR, to build a loss that only requires unsupervised data, speech in this example. Applying cycle consistency to ASR models is not trivial since fundamental information, such as speaker traits, are lost in the intermediate text bottleneck. To solve this problem, this work presents a loss that is based on the speech encoder state sequence instead of the raw speech signal.This is achieved by training a Text-To-Encoder model and defining a loss based on the encoder reconstruction error. Experimental results on the LibriSpeech corpus show that the proposed cycle-consistency training reduced the word error rate by 14.7% from an initial model trained with 100-hour paired data, using an additional 360 hours of audio data without transcriptions. We also investigate the use of textonly data mainly for language modeling to further improve the performance in the unpaired data training scenario.

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

CYCLE-CONSISTENCY TRAINING FOR END-TO-END SPEECH RECOGNITION

Takaaki Hori[1]*, Ramon Astudillo*[2]*, Tomoki Hayashi*[3]*, Yu Zhang*[4]*, Shinji Watanabe*[5]*, Jonathan Le Roux*[1]

[1]Mitsubishi Electric Research Laboratories (MERL),
[2]Spoken Language Systems Lab, INESC-ID, [3]Nagoya University, [4]Google, Inc.,
[5]Center for Language and Speech Processing, Johns Hopkins University

{thori, leroux}@merl.com, ramon@astudillo.com, hayashi.tomoki@g.sp.m.is.nagoya-u.ac.jp,
ngyuzh@google.com, shinjiw@jhu.edu

ABSTRACT

This paper presents a method to train end-to-end automatic speech recognition (ASR) models using unpaired data. Although the end-to-end approach can eliminate the need for expert knowledge such as pronunciation dictionaries to build ASR systems, it still requires a large amount of paired data, i.e., speech utterances and their transcriptions. Cycle-consistency losses have been recently proposed as a way to mitigate the problem of limited paired data. These approaches compose a reverse operation with a given transformation, e.g., text-to-speech (TTS) with ASR, to build a loss that only requires unsupervised data, speech in this example. Applying cycle consistency to ASR models is not trivial since fundamental information, such as speaker traits, are lost in the intermediate text bottleneck. To solve this problem, this work presents a loss that is based on the speech encoder state sequence instead of the raw speech signal. This is achieved by training a Text-To-Encoder model and defining a loss based on the encoder reconstruction error. Experimental results on the LibriSpeech corpus show that the proposed cycle-consistency training reduced the word error rate by 14.7% from an initial model trained with 100-hour paired data, using an additional 360 hours of audio data without transcriptions. We also investigate the use of text-only data mainly for language modeling to further improve the performance in the unpaired data training scenario.

Index Terms— speech recognition, end-to-end, unpaired data, cycle consistency

1. INTRODUCTION

In recent years, automatic speech recognition (ASR) technology has been widely used as an effective user interface for various devices such as car navigation systems, smart phones, and smart speakers. The recognition accuracy has dramatically improved with the help of deep learning techniques [1], and reliability of speech interfaces has been greatly enhanced. However, building ASR systems is very costly and time consuming. Current systems typically have a module-based architecture including an acoustic model, a pronunciation dictionary, and a language model, which rely on phonetically-designed phone units and word-level pronunciations using linguistic assumptions. To build a language model, text preprocessing such as tokenization for some languages that do not explicitly have word boundaries is also required. Consequently, it is not easy for non-experts to develop ASR systems, especially for underresourced languages.

End-to-end ASR has the goal of simplifying the module-based architecture into a single-network architecture within a deep learning framework, in order to address these issues [2–6]. End-to-end ASR methods typically rely only on paired acoustic and language data, without the need for extra linguistic knowledge, and train the model with a single algorithm. Therefore, this approach makes it feasible to build ASR systems without expert knowledge. However, in the end-to-end ASR framework a large amount of training data is crucial to assure high recognition accuracy. Paired acoustic (speech) and language (transcription) realizations spoken by multiple speakers are needed [7]. Nowadays, it is easy to collect audio and text data independently from the world wide web, but difficult to find paired data in different languages. Transcribing existing audio data or recording texts spoken by sufficient speakers are also very expensive.

There are several approaches that tackle the problem of limited paired data in the literature [8–12]. In particular, cycle consistency has recently been introduced in machine translation (MT) [13] and image transformation [14], and enables one to optimize deep networks using unpaired data. The basic underlying assumption is that, given a model that converts input data to output data and another model that reconstructs the input data from the output data, input data and its reconstruction should be close to each other. For example, suppose an English-to-French MT system translates an English sentence to a French sentence, and then a French-to-English MT system back-translates the French sentence to an English sentence. In this case, we can train the English-to-French system so that the difference between the English sentence and its back-translation becomes smaller, for which we only need English sentences. The French-to-English MT system can also be trained in the same manner using only French sentences.

Applying the concept of cycle consistency to ASR is quite challenging. As is the case in MT, the output of ASR is a discrete distribution over the set of all possible sentences. It is therefore not possible to build an end-to-end differentiable loss that back-propagates error through the most probable sentence in this step. Since the set of possible sentences is exponentially large in the size of the sentence, it is not possible to exactly average over all possible sentences either. Furthermore, unlike in MT and image transformation, in ASR, the input and output domains are very different and do not contain the same information. The output text does not include speaker and prosody information, which is eliminated through feature extraction and decoding. Hence, the speech reconstructed by the TTS system

The work reported here was conducted at the 2018 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies, and supported by Johns Hopkins University with unrestricted gifts from Amazon, Facebook, Google, Microsoft and Mitsubishi Electric. Work by Ramón Astudillo was supported by the Portuguese Foundation for Science and Technology (FCT) grant number UID/CEC/50021/2019.

does not have the original speaker and prosody information and can result in a strong mismatch.

Previous approaches related to cycle consistency in end-to-end ASR [9, 12] circumvent these problems by avoiding back-propagating the error beyond the discrete steps and adding a speaker network to transfer the information not present in the text. Therefore, these methods are not strictly cycle-consistency training, as used in MT and image transformation. Gradients are not cycled both through ASR and TTS simultaneously and only the second step on a ASR-TTS or TTS-ASR chain can be updated.

In this work, we propose an alternative approach that uses an end-to-end differentiable loss in the cycle-consistency manner. This idea rests on the two following principles.

1. Encoder-state-level cycle consistency:
   We use ASR encoder state sequences for computing the cycle consistency instead of waveform or spectral features. This uses a normal TTS Tacotron2 end-to-end model [15] modified to reconstruct the encoder state sequence instead of speech. We call this a *text-to-encoder (TTE) model* [8], which we introduced in our prior work on data augmentation. This approach reduces the mismatch between the original and the reconstruction by avoiding the problem of missing para-linguistic information.
2. Expected end-to-end loss:
   We use an expected loss approximated with a sampling-based method. In other words, we sample multiple sentences from the ASR model, generate an encoder state sequence for each, and compute the consistency loss for each sentence by comparing each encoder state sequence with the original. Then, the mean loss can be used to backpropagate the error to the ASR model via the REINFORCE algorithm [16]. This allows us to update the ASR system when the TTE is used to compute the loss, unlike [9].

The proposed approach allows therefore training with unpaired data, even if only speech is available. Furthermore, since error is backpropagated into the ASR system from a TTS-based loss, additional unsupervised losses can be used, such as language models. We demonstrate the efficacy of the proposed method in a semi-supervised training condition on the LibriSpeech corpus.

2. CYCLE-CONSISTENCY TRAINING FOR ASR

2.1. Basic concept

The proposed method consists of an ASR encoder-decoder, a TTE encoder-decoder, and consistency loss computation as shown in Fig. 1. In this framework, we need only audio data for backpropagation. In a first step, the ASR system transcribes the input audio feature sequence into a sequence of characters. In addition to this, a encoder state sequence is obtained. In a second step, the TTE system reconstructs the ASR encoder state sequence fro the character sequence. Finally, the cycle-consistency loss is computed by comparing the original state sequence and the reconstructed one. Backpropagation is performed with respect to this loss to update the ASR parameters.

2.2. Attention-based ASR model

The ASR model used is the well known attention-based encoder-decoder [17]. This model directly estimates the posterior $p_{\text{asr}}(\mathbf{C}|\mathbf{X})$, where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T | \mathbf{x}_t \in \mathbb{R}^D\}$ is a sequence of input D-dimensional feature vectors, and $\mathbf{C} = \{c_1, c_2, \ldots, c_L | c_l \in \mathcal{U}\}$ is

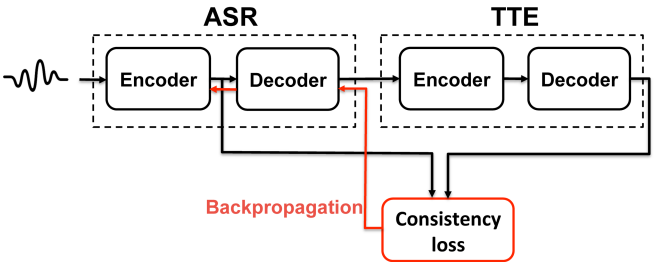

Fig. 1: Cycle-consistency training for ASR.

a sequence of output characters in the label set $\mathcal{U}$. The posterior $p_{\text{asr}}(\mathbf{C}|\mathbf{X})$ is factorized the probability chain rule as follows:

$$p_{\text{asr}}(\mathbf{C}|\mathbf{X}) = \prod_{l=1}^{L} p_{\text{asr}}(c_l|c_{1:l-1}, \mathbf{X}), \tag{1}$$

where $c_{1:l-1}$ represents the subsequence $\{c_1, c_2, \ldots c_{l-1}\}$, and $p_{\text{asr}}(c_l|c_{1:l-1}, \mathbf{X})$ is calculated as follows:

$$\begin{aligned}
\mathbf{h}_t^{\text{asr}} &= \text{Encoder}^{\text{asr}}(\mathbf{X}), && (2)\\
a_{lt}^{\text{asr}} &= \text{Attention}^{\text{asr}}(\mathbf{q}_{l-1}^{\text{asr}}, \mathbf{h}_t^{\text{asr}}, \mathbf{a}_{l-1}^{\text{asr}}), && (3)\\
\mathbf{r}_l^{\text{asr}} &= \Sigma_{t=1}^{T} a_{lt}^{\text{asr}} \mathbf{h}_t^{\text{asr}}, && (4)\\
\mathbf{q}_l^{\text{asr}} &= \text{Decoder}^{\text{asr}}(\mathbf{r}_l^{\text{asr}}, \mathbf{q}_{l-1}^{\text{asr}}, c_{l-1}), && (5)\\
p_{\text{asr}}(c_l|c_{1:l-1}, \mathbf{X}) &= \text{Softmax}(\text{LinB}(\mathbf{q}_l^{\text{asr}})), && (6)
\end{aligned}$$

where a_{lt}^{asr} represents an attention weight, $\mathbf{a}_l^{\text{asr}}$ the corresponding attention weight vector, $\mathbf{h}_t^{\text{asr}}$ and $\mathbf{q}_l^{\text{asr}}$ the hidden states of the encoder and decoder networks, respectively, $\mathbf{r}_l^{\text{asr}}$ a character-wise hidden vector, which is a weighted summarization of the hidden vectors $\mathbf{h}_t^{\text{asr}}$ using the attention weight vector $\mathbf{a}_l^{\text{asr}}$, and $\text{LinB}(\cdot)$ represents a linear layer with a trainable matrix and bias parameters.

All of the above networks are optimized using back-propagation to minimize the following objective function:

$$\begin{aligned}
\mathcal{L}_{\text{asr}} &= -\log p_{\text{asr}}(\mathbf{C}|\mathbf{X}) \\
&= -\Sigma_{l=1}^{L} \log p_{\text{asr}}(c_l^{\text{asr}}|c_{1:l-1}^{\text{asr}}, \mathbf{X}),
\end{aligned} \tag{7}$$

where $c_{1:l-1}^{\text{asr}} = \{c_1^{\text{asr}}, c_2^{\text{asr}}, \ldots, c_{l-1}^{\text{asr}}\}$ represents the ground truth for the previous characters, i.e. teacher-forcing is used in training. In the inference stage, the character sequence $\hat{\mathbf{C}}$ is predicted as

$$\hat{\mathbf{C}} = \underset{\mathbf{C} \in \mathcal{U}^+}{\text{argmax}} \log p_{\text{asr}}(\mathbf{C}|\mathbf{X}). \tag{8}$$

where $\mathcal{U}^+$ is the set of all sentences formed from the original character vocabulary $\mathcal{U}$.

2.3. Tacotron2-based TTE model

For the TTE model, we use the Tacotron2 architecture, which has demonstrated superior performance in the field of text-to-speech synthesis [15]. In our framework, the network predicts the ASR encoder state $\mathbf{h}_t^{\text{asr}}$ and the end-of-sequence probability s_t at each frame t from a sequence of input characters $\mathbf{C} = \{c_1, c_2, \ldots, c_L\}$ as follows:

$$\begin{aligned}
\mathbf{h}_l^{\text{tte}} &= \text{Encoder}^{\text{tte}}(\mathbf{C}), && (9)\\
a_{tl}^{\text{tte}} &= \text{Attention}^{\text{tte}}(\mathbf{q}_{t-1}^{\text{tte}}, \mathbf{h}_l^{\text{tte}}, \mathbf{a}_{t-1}^{\text{tte}}), && (10)\\
\mathbf{r}_t^{\text{tte}} &= \Sigma_{l=1}^{L} a_{tl}^{\text{tte}} \mathbf{h}_l^{\text{tte}}, && (11)
\end{aligned}$$

$$\mathbf{v}_{t-1} = \mathrm{Prenet}(\mathbf{h}^{\mathrm{asr}}_{t-1}), \quad (12)$$
$$\mathbf{q}^{\mathrm{tte}}_{t} = \mathrm{Decoder}^{\mathrm{tte}}(\mathbf{r}^{\mathrm{tte}}_{t}, \mathbf{q}^{\mathrm{tte}}_{t-1}, \mathbf{v}_{t-1}), \quad (13)$$
$$\hat{\mathbf{h}}^{b,\mathrm{asr}}_{t} = \tanh(\mathrm{LinB}(\mathbf{q}^{\mathrm{tte}}_{t})), \quad (14)$$
$$\mathbf{d}_{t} = \mathrm{Postnet}(\mathbf{q}^{\mathrm{tte}}_{l}), \quad (15)$$
$$\hat{\mathbf{h}}^{a,\mathrm{asr}}_{t} = \tanh(\mathrm{LinB}(\mathbf{q}^{\mathrm{tte}}_{l}) + \mathbf{d}_{t}), \quad (16)$$
$$\hat{s}_{t} = \mathrm{Sigmoid}(\mathrm{LinB}(\mathbf{q}^{\mathrm{tte}}_{t})), \quad (17)$$

where $\mathrm{Prenet}(\cdot)$ is a shallow feed-forward network to convert the network outputs before feedback to the decoder, $\mathrm{Postnet}(\cdot)$ is a convolutional neural network to refine the network outputs, and $\hat{\mathbf{h}}^{b,\mathrm{asr}}_{t}$ and $\hat{\mathbf{h}}^{a,\mathrm{asr}}_{t}$ represent predicted hidden states of the ASR encoder before and after refinement by Postnet. Note that the indices t and l of the encoder and decoder states are reversed compared to the ASR formulation in Eqs. (2)-(6), and that we use an additional activation function $\tanh(\cdot)$ in Eqs. (14) and (16) to avoid range mismatch in the outputs, in contrast to the original Tacotron2 [15].

All of the networks are jointly optimized to minimize the following objective function:

$$\begin{aligned}\mathcal{L}_{\mathrm{tte}} = &\mathrm{MSE}(\hat{\mathbf{h}}^{a,\mathrm{asr}}_{t}, \mathbf{h}^{\mathrm{asr}}_{t}) + \mathrm{MSE}(\hat{\mathbf{h}}^{b,\mathrm{asr}}_{t}, \mathbf{h}^{\mathrm{asr}}_{t}) \\ &+ \mathrm{L1}(\hat{\mathbf{h}}^{a,\mathrm{asr}}_{t}, \mathbf{h}^{\mathrm{asr}}_{t}) + \mathrm{L1}(\hat{\mathbf{h}}^{b,\mathrm{asr}}_{t}, \mathbf{h}^{\mathrm{asr}}_{t}) \\ &+ \tfrac{1}{T}\Sigma^{T}_{t=1}(s_t \ln \hat{s}_t + (1-s_t)\ln(1-\hat{s}_t)), \quad (18)\end{aligned}$$

where $\mathrm{MSE}(\cdot)$ represents mean square error, $\mathrm{L1}(\cdot)$ represent an L1 norm, and the last two terms represent the binary cross entropy for the end-of-sequence probability.

2.4. Cycle-consistency training

In this work, we use the TTE reconstruction loss $\mathcal{L}_{\mathrm{tte}}$ in Eq. (18) to measure the cycle consistency. The loss compares the ASR encoder state sequence with the encoder sequence reconstructed from the ASR output by the TTE. However, the argmax function in Eq. (8) to output the character sequence is not differentiable, and the consistency loss cannot be propagated through TTE to ASR directly. To solve this problem, we introduce the expected loss

$$\mathcal{L}_{\mathrm{ette}} = \mathbb{E}_{\mathbf{C}|\mathbf{X}}\left[\mathcal{L}_{\mathrm{tte}}(\hat{\mathbf{H}}^{\mathrm{asr}}(\mathbf{C}), \mathbf{H}^{\mathrm{asr}}(\mathbf{X}))\right], \quad (19)$$

where $\hat{\mathbf{H}}^{\mathrm{asr}}(\mathbf{C})$ denotes the state sequence $\{\hat{\mathbf{h}}^{a,\mathrm{asr}}_{t}, \hat{\mathbf{h}}^{b,\mathrm{asr}}_{t}, \hat{s}_t | t = 1, \ldots, T\}$ predicted by the TTE model for a given character sequence $\mathbf{C}$, and $\mathbf{H}^{\mathrm{asr}}(\mathbf{X})$ denotes the original state sequence $\{\mathbf{h}^{\mathrm{asr}}_{t}, s_t | t = 1, \ldots, T\}$ given by the ASR encoder for the input feature sequence $\mathbf{X}$.

To compute the gradients with respect to the expectation in Eq. 19, we utilize the REINFORCE algorithm [16]. This yields the following expression for the gradient

$$\nabla\mathcal{L}_{\mathrm{ette}} \approx \frac{1}{N}\sum_{\substack{\mathbf{C}^n \sim p_{\mathrm{asr}}(\cdot|\mathbf{X}),\\ n=1,\ldots,N}} \mathrm{T}\left(\mathbf{C}^n, \mathbf{X}\right) \nabla \log p_{\mathrm{asr}}(\mathbf{C}^n|\mathbf{X}), \quad (20)$$

where the weight for each sample $\mathbf{C}^n$ is defined as

$$\mathrm{T}\left(\mathbf{C}^n, \mathbf{X}\right) = \mathcal{L}_{\mathrm{tte}}(\hat{\mathbf{H}}^{\mathrm{asr}}(\mathbf{C}^n), \mathbf{H}^{\mathrm{asr}}(\mathbf{X})) - B(\mathbf{X}, \mathbf{C}^n) \quad (21)$$

and $B(\mathbf{X}, \mathbf{C}^n)$ is a baseline value used to reduce the estimate variance [16]. We used the mean value of $\mathbf{H}^{\mathrm{asr}}(\mathbf{C}^n)$ over N samples for $B(\mathbf{X}, \mathbf{C}^n)$ in this work.

3. RELATED WORK

The algorithm introduced in this paper is related to existing works on data augmentation and chain-based training. Our prior work [8] introduced the TTE model but used the synthesized encoder state sequences to train the ASR decoder from text data only. This is equivalent to back-translation in MT [18] and builds a non-differentiable TTE-ASR chain as opposed to the end-to-end differentiable ASR-TTE chain proposed here.

The work in [11] introduces a model consisting of a text-to-text auto-encoder and a speech-to-text encoder-decoder sharing the speech and text encodings. This model can also be trained jointly using paired and unpaired data but uses a simpler text encoder. Furthermore speech-only data is used to enhance the speech encodings, but not used to reduce recognition errors unlike our cycle-consistency approach. Finally, the text encoder is much simpler than our TTE model. In our work, the TTE model can hopefully generate better speech encodings to compute the consistency loss.

The speech chain model [9] is the most similar architecture to ours. As described in Section 1, the ASR model is trained with synthesized speech and the TTS model is trained with ASR hypotheses for unpaired data. Therefore, the models are not tightly connected with each other, i.e., one model cannot be updated directly with the help of the other model to reduce the recognition or synthesis errors. Our approach utilizes an end-to-end differentiable loss that allows TTS or other loss to be used *after* ASR for unsupervised training. We introduce as well the TTE model, which benefits from the reduction of speaker variations in the loss function and of computational complexity. With regard to cycle-consistency approaches in other disciplines, our approach is most similar to the dual learning approach in MT [13]. This paper combines alternating losses as in [9] using REINFORCE to compute expected translation losses.

4. EXPERIMENTS

4.1. Conditions

We conducted several experiments using the LibriSpeech corpus [19], consisting of two sets of clean speech data (100 hours + 360 hours), and other (noisy) speech data (500 hours) for training. We used 100 hours of the clean speech data to train the initial ASR and TTE models, and the audio of 360 hours set for unsupervised re-training of the ASR model with the cycle-consistency loss. We used five hours of clean development data as a validation set, and five hours of clean test data as an evaluation set.

The open source speech recognition toolkit Kaldi [20] was used to extract 80-dimensional log mel-filter bank acoustic vectors with three-dimensional pitch features. The ASR encoder had an eight-layered bidirectional long short-term memory with 320 cells including projection layers [21] (BLSTMP), and the ASR decoder had a one-layered LSTM with 300 cells. In the second and third layers from the bottom of the ASR encoder, sub-sampling was performed to reduce the utterance length from T down to $T/4$. The ASR attention network used location-aware attention [4]. For decoding, we used a beam search algorithm with beam size of 20. We set the maximum and minimum lengths of the output sequence to 0.2 and 0.8 times the length of the subsampled input sequence, respectively.

The architecture of the TTE model followed the original Tacotron2 [15]. It use 512-dimensional character embeddings, the TTE encoder consisted of a three-layered 1D convolutional neural network (CNN) containing 512 filters with size 5, a batch normalization, and rectified linear unit (ReLU) activation function, and

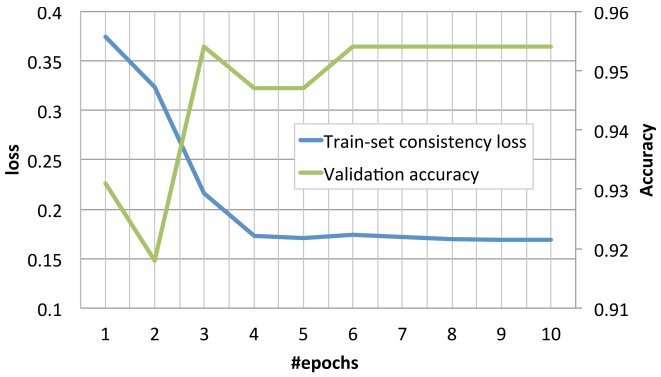

Fig. 2: Learning curve.

a one-layered BLSTM with 512 units (256 units for forward processing, the rest for backward processing). Although the attention mechanism of the TTE model was based on location-aware attention [4], we additionally accumulated the attention weight feedback to the next step to accelerate attention learning. The TTE decoder consisted of a two-layered LSTM with 1024 units. Prenet was a two-layered feed forward network with 256 units and ReLU activation. Postnet was a five-layered CNN containing 512 filters with the shape 5, a batch normalization, and tanh activation function except in the final layer. Dropout [22] with a probability of 0.5 was applied to all of the convolution and Prenet layers. Zoneout [23] with a probability of 0.1 was applied to the decoder LSTM. During generation, we applied dropout to Prenet in the same manner as in [15], and set the threshold value of the end-of-sequence probability at 0.75 to prevent from cutting off the end of the input sequence.

In cycle-consistency training, five sequences of characters were drawn from the ASR model for each utterance, where each character was drawn repeatedly from the Softmax distribution of ASR until it encountered the end-of-sequence label '<eos>'. During training, we also used the 100-hour paired data to regularize the model parameters in a teacher-forcing manner, i.e., the parameters were updated alternately by cross-entropy loss with paired data and the cycle-consistency loss with unpaired data.

All models were trained using the end-to-end speech processing toolkit ESPnet [24] on a single GPU (Titan Xp). Character error rate (CER) and word error rate (WER) were used as evaluation metrics.

4.2. Results

First, we show the changes of the consistency loss for training data and the validation accuracy for development data in Fig. 2, where the accuracy was computed based on the prediction with ground truth history. The consistency loss successfully decreased as the number of epochs increased. Although the validation accuracy did not improve smoothly, it reached a better value than that for the first epoch. We chose the 6th-epoch model for the following ASR experiments.

Table 1 shows the ASR performance using different training methods. Compared with the baseline result given by the initial ASR model, we can confirm that our proposed cycle-consistency training reduced the word error rate from 25.2% to 21.5%, a relative reduction of 14.7%. Thus, the results demonstrate that the proposed method works for ASR training with unpaired data. To verify the effectiveness of our approach, we further examined more straightforward methods, in which we simply used cross-entropy (CE) loss for unpaired data, where the target was chosen as the one best ASR hypothesis or sampled in the same manner as the cycle-consistency training. To alleviate the impact of the ASR errors, we weighted the CE loss by 0.1 for unpaired data while we did not down-weight the paired data. However, the error rates increased significantly in the 1-best condition. Even in the 5-sample condition, we could not obtain better performance than the baseline. We also conducted additional experiments under an oracle condition, where the 360-hour paired data were used together with the 100-hour data using the standard CE loss. The error rates can be considered the upper bound of this framework. We can see that there is still a big gap to the upper bound and further challenges need to be overcome to reach this goal.

Finally, we combined the ASR model with a character-based language model (LM) in a shallow fusion technique [25]. An LSTM-based LM was trained using text-only data from the 500-hour noisy set excluding audio data, and used for decoding. As shown in Table 2, the use of text-only data yielded further improvement reaching 19.5% WER (an 8% error reduction), which is the best number we have achieved so far for this unpaired data setup.

Our baseline WER is much worse than that reported in [19] for the 100-hour training setup. This is because we did not use any pronunciation lexicon or word-based language model for end-to-end ASR. Such end-to-end systems typically underperform conventional DNN/HMM systems with n-gram language model when using this size of training data.

Table 1: ASR performance using different training methods.

	CER / WER [%]	
	Validation	Evaluation
Baseline	11.2 / 24.9	11.1 / 25.2
Cycle-consistency loss	**9.5 / 21.5**	**9.4 / 21.5**
CE loss (1 best)	47.8 / 86.8	48.8 / 89.3
CE loss (5 samples)	13.3 / 28.2	12.3 / 27.7
Oracle	4.7 / 11.4	4.6 / 11.8

Table 2: ASR performance with LM shallow fusion.

	CER / WER [%]	
	Validation	Evaluation
Baseline + LM	11.9 / 22.6	11.9 / 22.9
Cycle consistency + LM	**10.2 / 19.6**	**9.9 / 19.5**

5. CONCLUSION

In this paper, we proposed a novel method to train end-to-end automatic speech recognition (ASR) models using unpaired data. The method employs an attention-based ASR model and a Tacotron2-based text-to-encoder (TTE) model to compute a cycle-consistency loss using audio data only. Experimental results on the LibriSpeech corpus demonstrated that the proposed cycle-consistency training reduced the word error rate by 14.7% from an initial model trained with 100-hour paired data, using an additional 360 hours of audio-only data without transcriptions. We also investigated the use of text-only data from 500-hour utterances for language modeling, and obtained a further error reduction of 8%. Accordingly, we achieved 22.7% error reduction in total for this unpaired data setup. Future work includes joint training of ASR and TTE model using both sides of the cycle-consistency loss, and the use of additional loss functions to make the training better.

6. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, 2014, pp. 1764–1772.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, speech and signal processing (ICASSP)*, 2013, pp. 6645–6649.

[4] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 577–585.

[5] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE International Conference on Acoustics, speech and signal processing (ICASSP)*, 2017, pp. 4835–4839.

[6] Takaaki Hori, Shinji Watanabe, and John R. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[7] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[8] Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, and Kazuya Takeda, "Back-translation-style data augmentation for end-to-end asr," *arXiv preprint arXiv:1807.10893*, 2018.

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

[10] Adithya Renduchintala, Shuoyang Ding, Matthew Wiesner, and Shinji Watanabe, "Multi-modal data augmentation for end-to-end ASR," in *Proc. Interspeech*, 2018, pp. 2394–2398.

[11] Shigeki Karita, Shinji Watanabe, Tomoharu Iwata, Atsunori Ogawa, and Marc Delcroix, "Semi-supervised end-to-end speech recognition," *Proc. Interspeech*, pp. 2–6, 2018.

[12] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Machine speech chain with one-shot speaker adaptation," in *Proc. Interspeech 2018*, 2018, pp. 887–891.

[13] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma, "Dual learning for machine translation," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 820–828. 2016.

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.

[15] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *arXiv preprint arXiv:1712.05884*, 2017.

[16] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[17] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE International Conference on Acoustics, speech and signal processing (ICASSP)*, 2015.

[18] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[19] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[20] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2011.

[21] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Aaron Courville, and Chris Pal, "Zoneout: Regularizing rnns by randomly preserving hidden activations," *arXiv preprint arXiv:1606.01305*, 2016.

[24] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "ESPnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[25] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. Interspeech*, 2017.