Semi-supervised Training for Sequence-to-Sequence Speech Recognition Using Reinforcement Learning

Hoon Chung, Hyeong-Bae Jeon and Jeon Gue Park
Electronics and Telecommunications Research Institute, Daejeon, Korea
Email: {hchung, hbjeon, jgp}@etri.re.kr

***Abstract*—This paper proposes a reinforcement learning based semi-supervised training approach for sequence-to-sequence automatic speech recognition (ASR) systems. Most recent semi-supervised training approaches are based on multi-loss functions such as cross-entropy loss for speech-to-text paired data and reconstruction loss for speech-text unpaired data.**

Although these approaches show promising results, some considerations still remain: (a) different loss functions are used for paired and unpaired data separately even though the purpose is classification accuracy improvement, and (b) several methods need auxiliary networks that increase the complexity of a semi-supervised training process.

To address these issues, a reinforcement learning based approach is proposed. The proposed approach focuses on rewarding ASR to generate more correct sentences for both paired and unpaired speech data. The proposed approach is evaluated on the Wall Street Journal task domain. The experimental results show that the proposed method is effective by reducing the character error rate from 10.4% to 8.7%.

***Index Terms*—automatic speech recognition, semi-supervised learning, reinforcement learning**

I. Introduction

Recently, automatic speech recognition (ASR) systems using sequence-to-sequence (seq2seq) models have become popular because of their simplicity and state-of-the-art performance. They can integrate separate acoustic, pronunciation, and language models into a single neural network [1], [2], [3], and outperform conventional ASRs in some general tasks [2].

Despite their popularity, these systems have some problems in practical use. Among these problems, this work focuses on the shortage of a speech-to-text paired training corpus. A large amount of speech-to-text paired data is necessary for seq2seq model-based ASR systems to achieve high performance [4], [5], [6], [7], [8]. However, it is expensive and time consuming job to collect a large amount of paired corpus, whereas it is cheap and easy to collect speech-text unpaired corpus in public. Therefore, to handle the shortage of paired corpora, semi-supervised training approaches have been actively conducted as a way to exploit unpaired corpora.

Various semi-supervised training methods for seq2seq models have been proposed, and these methods can be broadly classified into three categories. The first category is the methods generating machine transcriptions for unlabeled speech data using a pre-trained ASR system. The self-training methods [9], [10], [11], [12], [13], and teacher/student learning based approaches [14] fall under this category. The second category is the methods minimizing multi-loss functions for paired and unpaired data. Examples are the speech-chain framework [15] and adversarial training schemes [16], [17]. The third category are the methods minimizing cycling loss. These methods propose an end-to-end differentiable loss composed of cross-entropy loss and reconstruction loss by integrating ASR and Text-to-Speech (TTS) or Text-to-Encoder (TTE) [6], [18]. The methods show that reducing the cycling loss contributes to the decrease in recognition errors.

Although these methods show promising results, some considerations still remain: (a) different loss functions are used for paired data and unpaired data (e.g., cross-entropy loss for paired data and reconstruction loss for unpaired data), (b) reconstruction loss is not directly related to recognition accuracy, and (c) several methods need auxiliary networks (e.g., a TTS or TTE network is required to train unpaired data).

Therefore, to address these issues, we propose a method that focuses on using the same loss function for the paired and unpaired data using reinforcement learning (RL). We conduct the following:

- formulate semi-supervised seq2seq ASR training from the aspect of RL
- investigate hard and soft rewards
- investigate the modified REINFORCE [19] training strategy

The rest of this paper is organized as follows. Section 2 briefly describes seq2seq-based ASR. Section 3 discusses the conventional semi-supervised training methods. Section 4 presents our proposed approach in detail. Section 5 explains the experimental setting and Section 6 presents the experimental results. Section 7 concludes the paper and discusses future works.

II. Sequence-to-Sequence ASR

Most seq2seq ASR systems are composed of encoder and decoder networks as depicted in Fig. 1. This model estimates the posterior probability $P_\theta(\mathbf{y}|\mathbf{X})$, where $\mathbf{X} =$

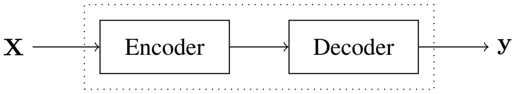

Fig. 1. Block diagram of an encoder-decoder based seq2seq ASR

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ is a sequence of input feature vectors, $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$ is a sequence of output characters, and θ denotes the model parameters. The posterior probability $P_\theta(\mathbf{y}|\mathbf{X})$ is factorized as follows:

$$P_\theta(\mathbf{y}|\mathbf{X}) = \prod_{l=1}^{N} P_\theta(y_n|y_{1:n-1}, \mathbf{X}) \tag{1}$$

where $y_{1:n-1}$ is the sub-sequence $\{y_1, y_2, \ldots, y_{n-1}\}$, and $P_\theta(y_n|y_{1:n-1}, \mathbf{X})$ is calculated by the encoder-decoder network as follows [6]:

$$\begin{aligned}
\mathbf{h}_t &= \text{Encoder}(\mathbf{X}) && (2)\\
\alpha_{nt} &= \text{Attention}(\mathbf{q}_{n-1}, \mathbf{h}_t, \mathbf{a}_{n-1}) && (3)\\
\mathbf{r}_n &= \sum_{t=1}^{T} \alpha_{nt} \mathbf{h}_t && (4)\\
\mathbf{q}_n &= \text{Decoder}(\mathbf{r}_n, \mathbf{q}_{n-1}, y_{n-1}) && (5)\\
P_\theta(y_n|y_{1:n-1}, \mathbf{X}) &= \text{Softmax}(LinB(\mathbf{q}_n)) && (6)
\end{aligned}$$

where α_{nt} is the attention weight, $\mathbf{a}_n$ is the corresponding weight vector, $\mathbf{h}_t$ and $\mathbf{q}_n$ are the hidden states of the encoder and decoder networks, respectively, and $\mathbf{r}_n$ is the character-wise hidden vector. LinB() represents a linear layer with trainable matrix and bias parameters.

In the recognition stage, inference is usually conducted through beam search using an external language model, $p_{LM}(\mathbf{y})$, as follows [20], [21]:

$$\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} \log P_\theta(\mathbf{y}|\mathbf{X}) + \gamma \log P_{LM}(\mathbf{y}) \tag{7}$$

where γ is the language model scale.

III. Semi-supervised ASR training

Semi-supervised seq2seq ASR model training can be treated as a general optimization problem to find the model parameters $\hat{\theta}$, which minimizes the loss function, $\mathcal{L}(\theta)$, for the given speech-to-text paired data, $(\mathbf{X}_l, \mathbf{Y}_l)$, unpaired speech data, $(\mathbf{X}_u)$, and text data $(\mathbf{Y}_u)$, as follows:

$$\hat{\theta} = \text{argmin}_\theta \mathcal{L}(\theta) \tag{8}$$

where the argmin operation is usually conducted using a gradient descent algorithm as follows [22]:

$$\theta_{t+1} = \theta_t - \alpha_t \nabla \mathcal{L}(\theta_t) \tag{9}$$

where α_t is the learning rate.

The semi-supervised seq2seq ASR model training can be dealt with using the loss function design problem. Before describing the proposed approach, we review the two most widely used approaches.

A. Shared encoder loss

The first method is shared encoder loss [8]. This loss aims to learn both the speech-to-text mapping for the paired data, and the shared inter-domain feature extraction between the unpaired speech and text as shown in Fig. 2. Therefore, the method uses cross-entropy loss for speech-to-text paired data and reconstruction loss for unpaired speech/text data, and divergence loss for embedding space between speech and text as follows:

$$\begin{aligned}
\mathcal{L} = {} & \mathcal{L}_{CE}(\text{Dec}(\text{Enc}(\mathbf{X}_l)), \mathbf{Y}_l) && (10)\\
& + \mathcal{L}_{MSE}(\text{Dec}(\text{Enc}(\text{Emb}(\mathbf{Y}_u))), \mathbf{Y}_u) && (11)\\
& + \mathcal{L}_{KLD}(\text{Enc}(\text{Emb}(\mathbf{Y}_u))), \text{Enc}(\mathbf{X}_u)) && (12)
\end{aligned}$$

where $\text{Dec}(\cdot)$, $\text{Enc}(\cdot)$, and $\text{Emb}(\cdot)$ are the decoder, encoder, and embedding components, respectively, and $\mathcal{L}_{CE}, \mathcal{L}_{MSE}, \mathcal{L}_{KLD}$ are cross entropy loss, mean square error loss and KullbackLeibler divergence loss, respectively.

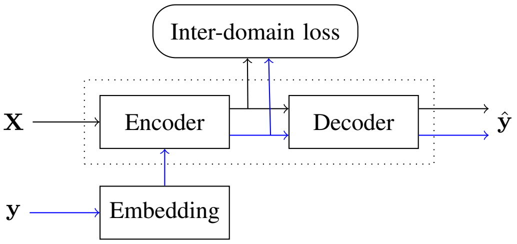

Fig. 2. Shared encoder learning

B. Cycle consistency loss

The second method is cycle consistency loss [6], [18]. This loss focuses on minimizing the cycle loss between speech-to-text (STT) and text-to-speech (TTS) or text-to-encoder (TTE) as shown in Fig. 3. Therefore, this method requires an auxiliary TTS or TTE network that does the reverse work of ASR.

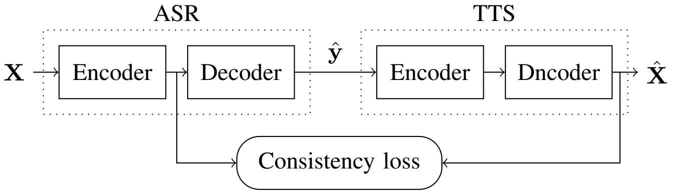

Fig. 3. Cycle consistency learning

Cycle consistency loss has several variants [6], [23]. Fig. 3 shows one example of the variants that minimizes the cycle loss between ASR-to-TTE and TTE-to-ASR as follows:

$$\begin{aligned}
\mathcal{L} = {} & \mathcal{L}_{CE}(\text{Dec}^{\text{ASR}}(\text{Enc}^{\text{ASR}}(\mathbf{X}_l)), \mathbf{Y}_l) && (13)\\
& + \mathcal{L}_{MSE}(\text{Enc}^{\text{ASR}}(\mathbf{X}), \hat{\mathbf{X}}) && (14)
\end{aligned}$$

C. Considerations on the conventional approaches

Both methods use different loss functions for paired and unpaired data separately, especially reconstruction loss, $\mathcal{L}_{MSE}$, for unpaired data. In addition, cycle consistency loss based methods require auxiliary networks, which increases the complexity of the training process.

IV. Reinforcement Learning

In this section, we briefly review RL, which concerns how software agents take actions in an environment to maximize the cumulative reward. RL differs from supervised learning in that it does not need labelled input/output pairs to be presented, and does not need sub-optimal actions to be explicitly corrected. [24], [25]

Fig. 4 illustrates the general interaction between an agent and an environment in an RL setting.

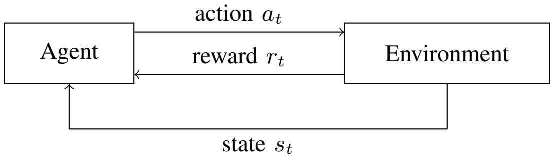

Fig. 4. Block diagram of the RL setting

A. Policy gradient

An agent interacts with the environment via its actions and receives a reward. This transitions the agent into a new state, so that it gives a sequence of states and actions known as a trajectory, τ [25], [26], [27].

$$\tau = (s_0, a_0, \ldots, s_{T+1}) \tag{15}$$

If the total reward for a given trajectory τ is represented as $R(\tau)$, the goal of RL is to maximize the expectation of the reward that it receives from the actions or minimize the negative expectation of the reward as follows:

$$\hat{\theta} = \mathrm{argmin}_\theta \mathcal{L}(\theta) \tag{16}$$

where $\mathcal{L}(\theta) = -\mathbb{E}_{\tau\sim P(\tau|\theta)}[R(\tau)]$, and $P(\tau|\theta)$ is a policy which is a probability distribution of actions given the state as follows:

$$P(\tau|\theta) = \rho_0(s_0)\prod_{t=0}^{T} P(s_{t+1}|s_t, a_t) \tag{17}$$

The gradient of the loss function $\nabla\mathcal{L}(\theta)$ can be derived using the log-trick as follows [27]:

$$\begin{aligned}
\nabla\mathcal{L}(\theta) &= -\nabla\int_\tau P(\tau|\theta)R(\tau) && (18)\\
&= -\int_\tau \nabla P(\tau|\theta)R(\tau) && (19)\\
&= -\int_\tau P(\tau|\theta)\nabla log P(\tau|\theta)R(\tau) && (20)\\
&= -\mathbb{E}_{\tau\sim P(\tau|\theta)}\left[\nabla log P(\tau|\theta)R(\tau)\right] && (21)\\
&= -\mathbb{E}_{\tau\sim P(\tau|\theta)}\left[\sum_{t=0}^{T}\nabla_\theta log P_\theta(a_t|s_t)R(\tau)\right] && (22)
\end{aligned}$$

V. Semi-supervised Seq2seq ASR Training using RL

A. Motivation

RL is used for semi-supervised training for two reasons. First, RL does not require speech-to-text paired corpus but requires a reward function, which is a much relaxed condition. Second, supervised training can be handled in terms of RL because the gradient of cross-entropy loss for a 1-hot target output can be a special case of policy gradient, in which the reward $R(\tau)$ is 1.0 as follows:

$$\nabla_\theta log\pi_\theta(y_t|\mathbf{x}_t) = \nabla_\theta log\pi_\theta(a_t|s_t)\cdot 1.0 \tag{23}$$

B. Semi-supervised training using RL

In the proposed RL-based semi-supervised training, seq2seq ASR is considered an agent that takes an action to select a character for an input feature, and a reward is assigned for the generated character sequence as depicted in Fig. 5. In the figure, action space is composed of discrete characters, state space $\mathcal{S}$ is composed of paired and unpaired speech features $\mathcal{S} = \{\mathbf{X}_l, \mathbf{X}_u\}$, and policy $P(\tau|\theta)$ is defined by a seq2seq model. Therefore, an action and a reward must be taken for the action sequence.

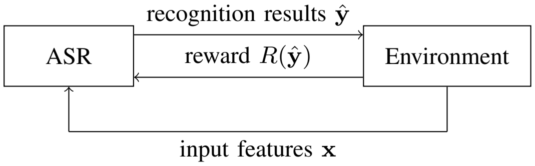
recognition results $\hat{\mathbf{y}}$
ASR
reward $R(\hat{\mathbf{y}})$
Environment
input features $\mathbf{x}$

Fig. 5. ASR from the aspect of RL

The sampling-based approach, $\hat{\mathbf{y}} \sim P_\theta(\mathbf{y}|\mathbf{X})$ is common in RL. However, we use the beam search-based inference (7) because it is the more common decoding action in ASR. For the reward, we use a normalized value in $[0.0, 1.0]$ because supervised training is the same as setting the reward $R(\tau)$ to 1.0 as shown in (23). Disabling the model updating by setting $R(\tau)$ to 0.0 is reasonable if the generated sentence is completely erroneous.

Table I summarizes the action and reward of the proposed semi-supervised training from the aspect of RL.

TABLE I
Semi-supervised training from the aspect of RL

	Paired	**Unpaired**
S	$\mathbf{X}\in\mathbf{X}_l$	$\mathbf{X}\in\mathbf{X}_u$
τ	$\mathbf{y}\in\mathbf{Y}_l$	$\hat{\mathbf{y}} = \mathrm{argmax}_k \mathrm{log}P_\theta(\mathbf{y}\|\mathbf{X}) + \gamma\mathrm{log}p_{LM}(\mathbf{y})$
$R(\tau)$	1.0	$0.0 \le Q(\hat{\mathbf{y}}) \le 1.0$

1) Paired corpus: As shown in Table I, in the case of paired data $(\mathbf{X}, \mathbf{y}) \in (\mathbf{X}_l, \mathbf{Y}_l)$, action and reward are straightforward because text $\mathbf{y}$ is the ground truth action for an input state $\mathbf{X}$. In other words, ASR is assumed to take the exact correct action $\mathbf{y}$ for $\mathbf{X}$, and thus it is reasonable to reward the highest value 1.0 for the paired data.

2) Unpaired corpus: In the case of unpaired speech data $\mathbf{X} \in (\mathbf{X}_u)$, a beam search-based inference is used for the agent's action, and two types of rewards are investigated to reward the generated character sequences $\hat{\mathbf{y}}$ that contain errors. The first scheme is the hard reward, which assigns a constant value as a reward regardless of the number of errors in the generated character sequence. The second scheme is the soft reward, which attempts to assign a lower value as more errors are included in the generated character sequence to make ASR generate sentences with fewer errors. For this purpose, we use the perplexity-based soft reward. In this work, the following reward function is used to combine these two schemes:

$$Q(\hat{\mathbf{y}}) = \alpha \cdot (\frac{min(PPL(\mathbf{Y}_l))}{PPL(\hat{\mathbf{y}})})^{\beta} \tag{24}$$

where $PPL(\hat{\mathbf{y}}) = P(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_N)^{-1/N}$ is the perplexity [28], [29], $min(PPL(\mathbf{Y}_l))$ is the lowest perplexity in the paired text for normalization purpose, and α and β control contribution of the hard and soft rewards. The hard reward is investigated by setting $\beta = 0.0$ and controlling α, and the soft reward is investigated by setting $\alpha = 1.0$ and controlling β. Fig. 6 shows the relation between the perplexity-based reward $Q(\hat{\mathbf{y}})$ and the character error rate (CER) used in our experiments. The higher the error rate is, the lower the reward.

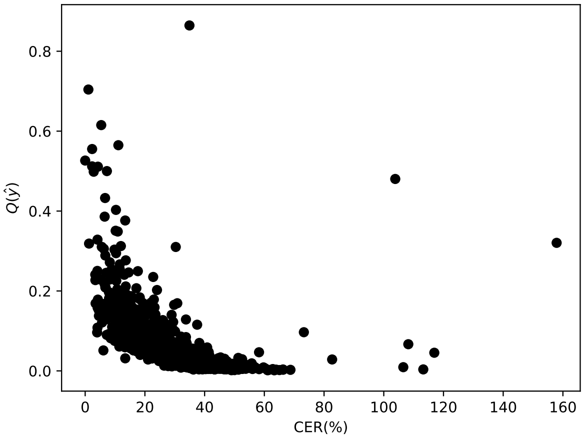

Fig. 6. Relation between the perplexity based reward and the error rate

C. Semi-supervised training procedure

Algorithm 1 describes the proposed semi-supervised training procedure. The algorithm consists of pre-training, interleaved training, and fine-tuning. Pre-training performs supervised training using the paired data to generate a seed model, and interleaved training performs semi-supervised learning [14], [30]. For the paired data, gradients of cross entropy loss are used to update the model parameters and the gradients of reward loss are used for unpaired speech data. In this work, $(\mathbf{X}_l, \mathbf{Y}_l)^i$ and $(\mathbf{X}_u)^i$ indicate the ith mini-batch, and N_l, and N_u are the number of paired and unpaired batches, respectively.

The proposed algorithm is affected by k, γ, α, and β, where

- k controls the number of top-k generated character sequences.
- γ controls the weight of language model.
- (α, β) controls the contribution of the rewards.

Algorithm 1 Semi-supervised training procedure

Require: A training set $(\mathbf{X}_l, \mathbf{Y}_l), (\mathbf{X}_u)$, initial values θ_0

```
1. Pre-training
while not converged do
    for $i = 0$ to $N_l$ do
        Select paired data $(\mathbf{X}, \mathbf{y}) \in (\mathbf{X}_l, \mathbf{Y}_l)^i$
        $\tau \leftarrow \mathbf{y}$
        $G \leftarrow 1.0$
        $\theta_{t+1} \leftarrow \theta_t - \alpha_t G \nabla log P_\theta(\tau|\mathbf{X})$
    end for
end while
2. Interleaved training
while not converged do
    for $i = 0$ to $N_u$ do
        Select unpaired data $\mathbf{X} \in (\mathbf{X}_u)^i$
        $\hat{\mathbf{y}} = \mathrm{argmax}_k P_\theta(\mathbf{y}|\mathbf{X}) + \gamma \mathrm{log} p_{LM}(\mathbf{y})$
        for $j = 0$ to $k$ do
            $\tau \leftarrow \hat{\mathbf{y}}_j$
            $G \leftarrow Q(\tau)$
            $\theta_{t+1} \leftarrow \theta_t - \alpha_t G \nabla log P_\theta(\tau|\mathbf{X})$
        end for
    end for
    for $i = 0$ to $N_l$ do
        Select paired data $(\mathbf{X}, \mathbf{y}) \in (\mathbf{X}_l, \mathbf{Y}_l)^i$
        $\tau \leftarrow \mathbf{y}$
        $G \leftarrow 1.0$
        $\theta_{t+1} \leftarrow \theta_t - \alpha_t G \nabla log P_\theta(\tau|\mathbf{X})$
    end for
end while
3. Fine-tuning
for $i = 0$ to $N_l$ do
    Select paired data $(\mathbf{X}, \mathbf{y}) \in (\mathbf{X}_l, \mathbf{Y}_l)^i$
    $\tau \leftarrow \mathbf{y}$
    $G \leftarrow 1.0$
    $\theta_{t+1} \leftarrow \theta_t - \alpha_t G \nabla log P_\theta(\tau|\mathbf{X})$
end for
```

VI. Experiments

A. Settings

We used the Wall Street Journal (WSJ) dataset LDC93S6B and LDC94S13B [31] to evaluate the proposed training approach. The dataset is composed of a small 15-hour (7138 utterances) dataset called si84, and a large 81-hour (37416 utterances) dataset called si284. We use si84 as a paired dataset and si284 as unpaired dataset, respectively. We employ the official test dataset dev93 for a hyper-parameter and decoding parameter search and eval92 for performance evaluation.

An 83-dimensional filter-bank with pitch features are used as the input feature. The encoder-decoder network utilizes location-aware attention [1], [32]. The encoder comprises 6 bi-directional Long Short Term Memory (LSTM) layers [3], [33], [34] each with 320 units and the decoder comprises 1 (uni-directional) LSTM layer with 300 units. The cross entropy and Connectionist Temporal Classification (CTC) [4], [35], [36] objective is optimized using AdaDelta [37] with an initial learning rate set to 1.0. The training batch size is 5 and the number of training epochs is 15. ESPnet [38] is used to implement and execute all our experiments. We pre-train a seed model with the si84 dataset in a supervised manner and then retrain the model with si84 and unpaired si284 in a semi-supervised manner. The performance is measured by character error rates (CER), and the performance is compared between two conventional methods: shared encoder [8] and cycle consistency loss [7].

B. Baseline performance

Table II shows the performance of the baseline systems or seed models trained only on the si84 corpus in a supervised manner. The shared encoder [8] and cycle consistency [7] reported 15.8% and 10.2% CER, respectively. Our re-implementation of the seq2seq model for the shared encoder is achieved at 10.4%.

TABLE II
CERs(%) OF BASELINE SYSTEMS TRAINED USING WSJ-SI84 CORPUS

System	**dev93**	**eval92**
Shared encoder [8]	25.4	15.8
Cycle consistency [7]	-	10.2
This work	15.2	10.4

The different CERs are due to the different numbers of encoder layers, numbers of decoder units, batch sizes and numbers of epochs. Although the proposed seq2seq model architecture is the same as that of the shared encoder model [8], its performance is better because different batch shuffling schemes, learning rate scheduling, and batch size are used. Table III summarizes the differences.

TABLE III
DIFFERENCES AMONG BASELINE SETTINGS

System	#. Enc layers	#. Dec units	Batch size	#. of epochs
Shared encoder [8]	6	300	15	15
Cycle consistency [7]	8	320	30	20
This work	6	300	5	15

C. Semi-supervised training performance

Table IV shows the performance of semi-supervised training. The shared encoder and cycle consistency methods achieve 14.4% and 9.1% CER, respectively for the eval92 testset. The proposed method achieves 8.7% CER at the best hyper-parameter setting.

TABLE IV
BEST CERs(%) OF THE PROPOSED SEMI-SUPERVISED TRAINING METHOD AT BEST HYPER-PARAMETER SETTINGS

System					**dev93**	**eval92**
Shared encoder [8]					24.8	14.4
Cycle consistency [7]	-					9.1
	α	k	γ	β		
Hard reward	0.05	1	0.0	0.0	14.3	9.8
Top-k	0.05	2	0.0	0.0	13.6	9.3
Language model	0.05	2	0.02	0.0	13.1	9.3
Soft reward	1.00	2	0.02	6.0	**13.0**	**8.7**

The hyper-parameters are tuned sequentially. The steps are shown in Table IV. The hard reward is first tuned by varying α with the β fixed at 0.0. It obtains the lowest CER at $\alpha = 0.05$. Then, the number of actions is tuned by changing the k while fixing the α to 0.05. The CER decreases from 9.8% to 9.3% when using the 2-best actions. During fixing α, k, γ, is tuned, but there is no improvement for the eval92 test set. Then, β is tuned to reflect perplexity. CER is further reduced from 9.3% to 8.7%.

VII. Conclusions

Although conventional semi-supervised seq2seq ASR training approaches report promising results, there are still some considerations when using reconstruction loss for classification improvement of ASR systems.

To deal with this problem, we propose an RL based semi-supervised training approach. In RL, the speech-to-text paired corpus for training is not a mandatory condition and only reward is sufficient. This is the relaxed condition, and it makes semi-supervised training straightforward to handle both paired and unpaired speech data. We evaluate the proposed approach on the WSJ domain. The experimental results show that the proposed method outperforms the conventional methods. The experimental results show that the proposed method is effective by reducing the CER from 10.4% to 8.7%.

The proposed approach is characterized by a top-k action selection, a language model integration, and a perplexity-based reward. The top-k action and soft reward are important factors for improvement. We consider the improvement is because top-k action helps to solve the exploitation and exploration issues in RL, and give a reasonable reward.

In future studies we intend to investigate more suitable reward functions and action schemes.

VIII. Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2019-0-00004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners).

References

[1] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[2] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[3] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[4] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[5] Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A comparison of sequence-to-sequence models for speech recognition.," in *Interspeech*, 2017, pp. 939–943.

[6] Takaaki Hori, Ramon Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, and Jonathan Le Roux, "Cycle-consistency training for end-to-end speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6271–6275.

[7] Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, and Jan Černocký, "Self-supervised sequence-to-sequence asr using unpaired speech and text," *arXiv preprint arXiv:1905.01152*, 2019.

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

[9] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[10] Frank Wessel and Hermann Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 23–31, 2005.

[11] Jeff Ma and Richard Schwartz, "Unsupervised versus supervised training of acoustic models," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[12] Kai Yu, Mark Gales, Lan Wang, and Philip C Woodland, "Unsupervised training and directed manual transcription for lvcsr," *Speech Communication*, vol. 52, no. 7-8, pp. 652–663, 2010.

[13] Bo Li, Tara N Sainath, Ruoming Pang, and Zelin Wu, "Semi-supervised training for end-to-end models via weak distillation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2837–2841.

[14] Sree Hari Krishnan Parthasarathi and Nikko Strom, "Lessons from building acoustic models with a million hours of speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6670–6674.

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

[16] Jennifer Drexler and James Glass, "Combining end-to-end and adversarial training for low-resource speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 361–368.

[17] Shigeki Karita, Atsunori Ogawa, Marc Delcroix, and Tomohiro Nakatani, "Sequence training of encoder-decoder model using policy gradient for end-to-end speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5839–5843.

[18] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6281–6285.

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

[20] Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 369–375.

[21] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *arXiv preprint arXiv:1805.03294*, 2018.

[22] Sebastian Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[23] 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," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 426–433.

[24] Richard S Sutton, Andrew G Barto, et al., *Introduction to reinforcement learning*, vol. 2, MIT press Cambridge, 1998.

[25] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[26] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[27] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[28] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker, "Perplexitya measure of the difficulty of speech recognition tasks," *The Journal of the Acoustical Society of America*, vol. 62, no. S1, pp. S63–S63, 1977.

[29] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[30] Ladislav Mošner, Minhua Wu, Anirudh Raju, Sree Hari Krishnan Parthasarathi, Kenichi Kumatani, Shiva Sundaram, Roland Maas, and Björn Hoffmeister, "Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6475–6479.

[31] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[32] 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*, 2015, pp. 577–585.

[33] Jürgen Schmidhuber and Sepp Hochreiter, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[35] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[36] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.

[37] Matthew D Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[38] 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.