REINFORCEMENT LEARNING BASED SPEECH ENHANCEMENT FOR ROBUST SPEECH RECOGNITION

Yih-Liang Shen[1], Chao-Yuan Huang[1], Syu-Siang Wang[2], Yu Tsao[3] , Hsin-Min Wang[2,4] and Tai-Shih Chi[1]

[1]Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, R.O.C
[2] Joint Research Center for AI Technology and All Vista Healthcare, MOST, Taipei, R.O.C
[3] Research Center for Information Technology Innovation, Academia Sinica, Taipei, R.O.C
[4] Institute of Information Science, Academia Sinica, Taipei, R.O.C

ABSTRACT

Conventional deep neural network (DNN)-based speech enhancement (SE) approaches aim to minimize the mean square error (MSE) between enhanced speech and clean reference. The MSE-optimized model may not directly improve the performance of an automatic speech recognition (ASR) system. If the target is to minimize the recognition error, the recognition results should be used to design the objective function for optimizing the SE model. However, the structure of an ASR system, which consists of multiple units, such as acoustic and language models, is usually complex and not differentiable. In this study, we propose to adopt the reinforcement learning (RL) algorithm to optimize the SE model based on the recognition results. We evaluated the proposed RL-based SE system on the Mandarin Chinese broadcast news corpus (MATBN). Experimental results demonstrate that the proposed SE system can effectively improve the ASR results with a notable 12.40% and 19.23% error rate reductions for signal to noise ratio (SNR) at 0 dB and 5 dB conditions, respectively.

Index Terms— reinforcement learning, automatic speech recognition, speech enhancement, deep neural network, character error rate

1. INTRODUCTION

The performance of automatic speech recognition (ASR) has significantly improved in recent years. However, a long-existing issue still remains: ASR suffers severe performance degradation in noisy environments [1]. Many approaches have been proposed to address the noise issue. One category of these approaches is speech enhancement (SE) [2, 3]. The goal of SE is to generate enhanced speech signals that closly match clean and undistorted speech signals, by removing the noise components from the noisy speech [4, 5, 6]. Traditional SE approaches are designed based on some assumptions of speech and noise characteristics [7, 8]. Generally, these approaches can yield a satisfactory performance in terms of speech quality but may not be directly beneficial in the improvement of the ASR performance [9, 5].

Recently, deep-learning-based SE approaches have received increased attention, and it has been confirmed that they yield better performances than traditional methods in numerous tasks [10, 11, 12]. Because of the deep structure, the deep-learning-based models can effectively characterize the complex transformation of noisy speech to clean speech, or they can precisely estimate a mask to filter out noise components from the noisy speech. To train the deep-learning-based models, the mean square error (MSE)-criterion is usually used as the objective function. Specifically, the model is trained to minimize the MSE of the enhanced speech and clean references. Although it has been proven that the MSE-based objective function is effective for noise reduction, it is not optimal for improving speech quality and intelligibility, or the ASR performance [13, 14, 15, 16].

Clearly, the recognition results should be the optimal objective function for SE if the target task is to achieve good ASR performance. However, most of the commonly used ASR systems consist of multiple modules, such as the acoustic models and language models. Correspondingly, the input–output correlation is extremely complicated and may not be differentiable. Thus, it is difficult to directly use the recognition results to optimize the SE models. Moreover, it takes a considerable amount of resources to build an ASR system, and thus the use of a well-established ASR system from a third party is favorable. In this study, we propose to adopt the reinforcement learning (RL) algorithm to train an SE model to minimize the recognition errors.

The main concept of the RL algorithm is to take an action in an environment in order to maximize some notion of a cumulative reward [17]. Different from supervised and unsupervised learning algorithms, the RL algorithm learns how to attain a (complex) goal in an iterative manner. To date, the RL algorithms have been successfully applied to various tasks, such as robot control [18], dialogue management [19], and computer game playing [20].

The RL algorithm has also been adopted into the speech signal processing filed [21]. In [22], the RL has been used to improve the ASR performance. Based on hypothesis selections by the users, the system can improve the recognition accuracy as compared to unsupervised adaptation. Meanwhile, the RL has been used for DNN-based source enhancement by optimizing objective sound quality assessment score [23]. The results show that by using the RL algorithm, both perceptual evaluation of the speech quality (PESQ) [24] and the short-time intelligibility measure (STOI) [25] scores can be improved as compared to the MSE-based training criterion [26].

In this study, we adopt the same idea presented in [23] to establish an RL-based SE system to optimize the ASR performance. Instead of estimating the ratio masking as used in [23], the proposed SE system determines the optimal binary mask to minimize the recognition errors. Notably, the ASR system is fixed in the proposed method. This is to simulate most realistic scenarios that a well-trained ASR system is provided by a third party, and an SE is built to generate suitable inputs to the ASR system. We evaluated the proposed RL-based SE system on a Mandarin Chinese broadcast news corpus (MATBN) [27]. According to our experimental results, the proposed RL-based SE system effectively decreases the character error rate (CER) during the testing of the recognition in

the presence of noise. The remainder of this paper is organized as follows. Section 2 reviews relevant techniques. Section 3 introduces the proposed system. Section 4 presents the experimental setup and results. Finally, section 5 provides conclusion remarks.

2. RELATED WORKS

In the time domain, a noisy speech signal $\mathbf{y}$ is formulated by a combination of a clean speech signal $\mathbf{s}$ and an additive noise signal $\mathbf{n}$. By performing short-time Fourier transform (STFT), log–power operation, and mel–frequency-based filtering, the mel–frequency power spectrogram (MPS) of $\mathbf{y}$ can be expressed as:

$$\mathbf{Y} = \mathbf{S} + \mathbf{N}. \tag{1}$$

In this study, p frames of the STFT MPS feature vectors are concatenated to form one chunk vector for $\mathbf{Y}$, $\mathbf{S}$ and $\mathbf{N}$.

$$\hat{\boldsymbol{\mathcal{X}}}_c = [\boldsymbol{\mathcal{X}}_{cp}^{\top}, \boldsymbol{\mathcal{X}}_{cp+1}^{\top}, \cdots, \boldsymbol{\mathcal{X}}_{(c+1)p-1}^{\top}]^{\top}, \ \ \boldsymbol{\mathcal{X}} \in \{\mathbf{Y}, \mathbf{S}, \mathbf{N}\}, \tag{2}$$

where $c = \{0, 1, \cdots, C\}$ is a chunk index, and C is the total number of chunks vectors within $\boldsymbol{\mathcal{X}}$. Note that when $p = 1$, the chunk vector is the STFT MPS feature vector.

2.1. Ideal Binary Mask-based SE System

It has been reported that when the goal is to improve the ASR performance, ideal binary mask (IBM) is more suitable than ideal ratio mask (IRM) or directly mapping [28] to be used to design the SE system. Therefore, we implement an IBM-based SE system in this study. For the IBM-based SE system, the input $\hat{\mathbf{Y}}$ was filtered by IBM to obtain the enhanced output $\hat{\mathbf{S}}'$:

$$\hat{\mathbf{S}}' = \hat{\mathbf{Y}}. \times \hat{\mathbf{B}}, \tag{3}$$

where ".$\times$" represents an element-wise multiplication, and $\hat{\mathbf{B}}$ is the IBM matrix, which is defined as:

$$\hat{\mathbf{B}} = \mathbb{1}\{log(\hat{\mathbf{S}}) - log(\hat{\mathbf{N}})\}, \tag{4}$$

where $\mathbb{1}\{\cdot\}$ is the unit step function applied to each element of $\hat{\mathbf{B}}$.

2.2. DNN-based SE Model with the MSE Criterion

For the DNN-based SE, a set of noisy-clean training pairs are prepared as the input and reference of a DNN model. For the noisy $\hat{\mathbf{Y}}$, F chunk vectors are then cascaded to include more context information: $\tilde{\mathbf{Y}}_c = [\hat{\mathbf{Y}}_{c-F+1}^{\top}, \hat{\mathbf{Y}}_{c-F+2}^{\top}, \cdots, \hat{\mathbf{Y}}_c^{\top}]^{\top}$. The mapping process of a feedforward DNN with L hidden layers is the formulated as,

$$\begin{aligned} h_1(\tilde{\mathbf{Y}}_c) &= \sigma\{\mathbf{W}_1 log(\tilde{\mathbf{Y}}_c) + \mathbf{b}_1\}, \\ &\vdots \\ h_L(\tilde{\mathbf{Y}}_c) &= \sigma\{\mathbf{W}_L h_{L-1}(\tilde{\mathbf{Y}}_c) + \mathbf{b}_L\}, \\ \hat{\mathbf{S}}_c'' &= \sigma_1\{\mathbf{W}_{L+1} h_L(\tilde{\mathbf{Y}}_c) + \mathbf{b}_{L+1}\}, \end{aligned} \tag{5}$$

where $\mathbf{W}_\ell$ and $\mathbf{b}_\ell$ are the weight matrices and bias vectors,respectively. Both $\sigma\{\cdot\}$ and $\sigma_1\{\cdot\}$ are activation functions, in which $\sigma\{\cdot\}$ is the sigmoid function while $\sigma_1\{\cdot\}$ represents a linear transformation. When the MSE is used as the cost function, the parameter set Θ that consists of all of $\mathbf{W}_\ell$ and $\mathbf{b}_\ell$ in Eq. (5) is estimated by,

$$\Theta^* = \arg\min_{\Theta}(\tfrac{1}{C}\textstyle\sum_{c=1}^{C} \parallel log(\hat{\mathbf{S}}_c) - log(\hat{\mathbf{S}}_c'') \parallel_2^2) \tag{6}$$

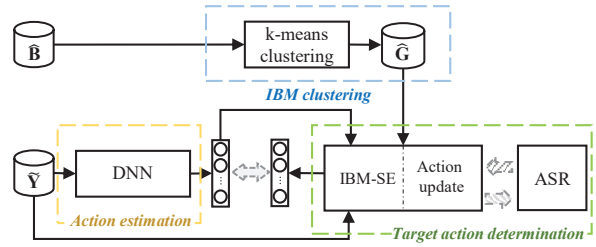

Fig. 1. The block diagram of the proposed SE system, which includes "*IBM clustering*", "*Action estimation*", and "*Target action determination*".

3. PROPOSED METHOD

Figure 1 illustrates the proposed system, which consists of three modules: "*IBM clustering*", "*Action estimation*", and "*Target action determination*".

3.1. IBM clustering module

In the IBM-based SE system, an IBM filter is computed for each feature vector. The IBM clustering module groups the entire set of IBM vectors $\hat{\mathbf{B}}$ collected from the training data to A clusters based on the K-means algorithm. Each cluster is represented as $\hat{\mathbf{g}}_a$ with respect to the cluster index a. The ensemble of these clusters is denoted as $\hat{\mathbf{G}}$. Thus, we have,

$$\hat{\mathbf{G}} = [\hat{\mathbf{g}}_1, \cdots, \hat{\mathbf{g}}_a, \cdots, \hat{\mathbf{g}}_A]. \tag{7}$$

Since the elements in each IBM vector acquire binary values, the Hamming distance [29] is used to compute the distance between the two vectors in this study. Meanwhile, we used 32 clusters $(A = 32)$ to group $\hat{\mathbf{B}}$ based on the k-means algorithm.

3.2. Action estimation module

To effectively use the training data, we first pre-train the DNN model by placing $\tilde{\mathbf{Y}}_c$ at the input and $\hat{\mathbf{B}}_c$ at the output. This pre-trained model was then re-trained with additional hidden layers to compute the A-dimensional action vector $\mathbf{a}_c''$ at cth chunk. Among the A elements in $\mathbf{a}_c''$, the index with the maximum value was determined,

$$a_c = \arg\max_{a \in \mathbb{A}} [\mathbf{a}_c'']_a, \tag{8}$$

where $[\cdot]_a$ represents the ath element of the vector, and $\mathbb{A} = \{1, 2, \cdots, A\}$. In addition, different from the spectral mapping in Eq. (5), the softmax operation is used in the final layer in the re-trained DNN (i.e. σ_1 is now the softmax function). The cost function for the re-training process is expressed as,

$$\Theta^* = \arg\min_{\Theta}(\tfrac{1}{C}\textstyle\sum_{c=1}^{C} \parallel \mathbf{a}_c - \mathbf{a}_c'') \parallel_2^2), \tag{9}$$

where $\mathbf{a}_c$ is the reference target, which is derived from *Target action determination* module and is described in the next section.

3.3. Target action determination module

Figure 2 shows the flowchart of the *Target action determination* module. First, $\mathbf{a}_c''$, which is estimated from the action estimation module is used to determine the cluster index a_c in Eq. (8). Then, the IBM selection function selects $\mathbf{g}_a$ from $\hat{\mathbf{G}}$ with respect to index $a = a_c$. Next the SE function uses the selected $\mathbf{g}_a$ to enhance the input $\tilde{\mathbf{Y}}_c$ (i.e. $\tilde{\mathbf{Y}}_c. \times \mathbf{g}_a$). After enhancing all C chunk vectors, both the input noisy and the IBM-enhanced STFT–MPS features are reconstructed back to the time domain signals, and then provide the

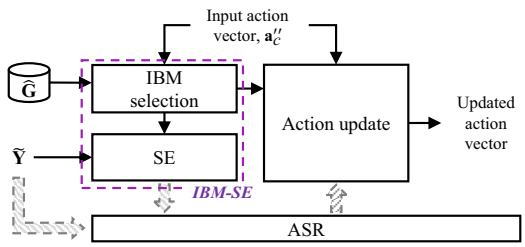

Fig. 2. The flowchart of *Target action determination* module, which is used to update the input action vector.

ASR to calculate the utterance-based error rates (ERs), $z_{\mathbf{y}}$ and $z_{\mathbf{s}'}$, respectively. Both $z_{\mathbf{y}}$ and $z_{\mathbf{s}'}$ are used in the *Target action determination* function, which is a two-stage operations, namely, the reward calculation and action update.

3.3.1. *Reward calculation*

Rather than directly use $z_{\mathbf{s}'}$ as the reward, we applied the relative value between $z_{\mathbf{y}}$ and $z_{\mathbf{s}'}$ in Eq. (10).

$$R = \tanh\{\alpha(z_{\mathbf{y}} - z_{\mathbf{s}'})\}, \tag{10}$$

where $\alpha > 0$ is a scalar factor, which is set to 10 in this study. For this equation, the positive R denotes a larger ER of $z_{\mathbf{y}}$ than that of $z_{\mathbf{s}'}$, thus suggesting that the enhanced speech can provide better recognition results. On the other hand, a negative R denotes a smaller $z_{\mathbf{y}}$ than $z_{\mathbf{s}'}$, suggesting that the enhanced speech gives worse recognition performance.

In addition to the utterance-based rewards R, we also consider a chunk-based reward because the action for each chunk vector may act and contribute differently to $z_{\mathbf{s}'}$. That is, an effective enhancement can cause positive contribution on the ASR performance. Therefore, we defined a time-varied reward r_c as:

$$\hat{E}_c = (log(\hat{\mathbf{S}}_c) - log(\hat{\mathbf{S}}_c'))^{\top}(log(\hat{\mathbf{S}}_c) - log(\hat{\mathbf{S}}_c')), \tag{11}$$

$$\tilde{E}_c = \frac{\hat{E}_c}{\max_{0\le c\le C-1}(\hat{E}_c)}, \tag{12}$$

$$r_c = \begin{cases} (1-\tilde{E}_c)R, & R > 0, \\ \tilde{E}_c R, & R \le 0. \end{cases} \tag{13}$$

From Eqs. (11)– (13), the weighting factor $\tilde{E}_c$, $0 \le \tilde{E}_c \le 1$, at the cth chunk is the normalized square error. When selecting a erroneous IBM vector, the normalized error $\tilde{E}_c$ in(12) is large, and accordingly r_c is small, which penalizes this wrong action, as to be introduced in the next sub-section.

3.3.2. *Action update*

To update the action vector, $\mathbf{a}_c''$, we first determine two different action indices, $a_{\hat{\mathbf{B}}_c}$ and a_c. To obtain $a_{\hat{\mathbf{B}}_c}$, we first follow Eq. (4) to determine an IBM vector, which is then used to locate the closest cluster in $\hat{\mathbf{G}}$; the located cluster index is $a_{\hat{\mathbf{B}}_c}$. On the other hand, the cluster index a_c is determined by Eq. (8), as presented in the *Action estimation* module.

With the determined action indices $a_{\hat{\mathbf{B}}_c}$ and a_c, the input action vector $\mathbf{a}_c''$ is updated for the output $\mathbf{a}_c$ based on the following equations:

$$[\mathbf{a}_c]_{a_c} = \begin{cases} r_c + max_{a_c\in\mathbb{A}}[\mathbf{a}_c'']_{a_c}, & R > 0, \\ [\mathbf{a}_c'']_{a_c}, & R = 0, \end{cases} \tag{14}$$

and

$$[\mathbf{a}_c]_{a_{\hat{\mathbf{B}}_c}} = [\mathbf{a}_c'']_{a_{\hat{\mathbf{B}}_c}} - r_c, \quad R < 0. \tag{15}$$

Fig. 3. The block diagram of testing part for the proposed algorithm.

3.4. Testing procedure

After performing the training on DNN with the associated objective function in Eq. (9), Fig. 3 illustrates the block diagram of the testing process. From the figure, the trained DNN model is applied on a noisy STFT–MPS $\hat{\mathbf{X}}$, which is first extracted from the time-domain signal $\mathbf{x}$. The estimated IBM indices are then used in combination with Eq. (8) for each chunk to further enhance the input noisy and provide $\hat{\mathbf{S}}_{\mathbf{x}}$ in the output of the IBM–SE function. The waveform $\mathbf{s}_{\mathbf{x}}'$ is reconstructed from $\hat{\mathbf{S}}_{\mathbf{x}}$ is then applied to ASR to get recognition results.

4. EXPERIMENTS

4.1. Experimental setup

We conducted our experiments on the MATBN task, which was an 198-hour Mandarin Chinese broadcast news corpus [27]. The utterances in MATBN were originally recorded at a 44.1 kHz sampling rate and were further down-sampled to 16 kHz. A 25-hour gender-balanced subset of the speech utterances was used to train a set of CD-DNN-HMM acoustic models, which were used in our previous work [30]. A set of trigram language models was trained on a collection of text news documents published by the Central News Agency (CNA) between 2000 and 2001 (the Chinese Gigaword Corpus released by LDC) with the SRI Language Modeling Toolkit [31]. The overall ASR system was implemented on the Kaldi [32] toolbox. Each speech waveform was parameterized into a sequence of 40-dimensional filter-bank features. The DNN structure for the acoustic models was consisted of six hidden layers, and each layer had 2048 nodes. The dimensions for the input and output layers were 440 ($40 \times (2 \times 5 + 1)$) and 2596, respectively [30]. The evaluated results are reported as the average CER. To train the RL–SE system, another 460 utterances were selected from the MATBN corpus. The overall RL–SE and ASR systems were evaluated using another 30 utterances, where the utterance lengths were around 1 to 6 seconds, from the MATBN testing set. In this study, we used the baby-cry noise as the background noise. The baby-cry noise waveform was divided into two parts, the first part was artificially added to the 460 training utterances with signal-to-noise ratio (SNR) level at 5 dB; the second part was artificially added to the 30 testing utterances at 0 and 5 dB SNR levels. Notably, the training and testing utterances were simulated using different segments of the noise source waveform, and thus the properties were slightly different. Finally, we have prepared 460 noisy–clean pairs to train the RL-based SE system. For all of the training and testing data, the applied frame size and the shift for STFT were 32 and 16 ms in length, respectively. The 64-dimensional MPS features were then extracted from all noisy and clean utterances. Next, we established two RL-based SE models, with two different parameters p for the chunk vectors: the systems with $p = 1$ and $p = 2$ are termed $RLSE_1$ and $RLSE_2$, respectively. Both $RLSE_1$ and $RLSE_2$ were composed of one hidden layer with 64 nodes, and 32 for the output nodes. The input dimensions of $RLSE_1$ was 704 ($64 \times 1 \times 11$), and that of $RLSE_2$ was 640 ($64 \times 2 \times 5$), in which the 11 and 5 are values of the parameter F, and is used for providing the context information (as mentioned in Section 2.2).

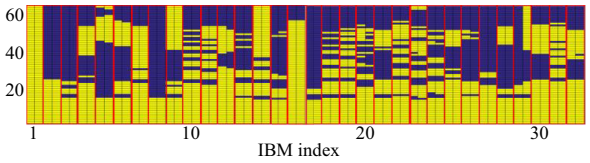

Fig. 4. Clustered IBMs were derived by the k-means algorithm.

4.2. Experimental results

Figure. 4 shows all the 32 IBM vectors, each with 64-dimensions. The IBM in Eq. (7) used in the $RLSE_2$ system. Bright yellow elements in the figure denote ones (in terms of their binary values) and the blue elements denote zeros. From the figure, we observe that low-dimensional MPS features are dominated by speech components. One possible explanation is that the noise signals did not mask the human speech in the low-frequency regions. In addition, the entire first column consisted of ones, thus suggesting that the silence frames were also contained in the baby-cry noise.

We then compared the averaged CER results of the $RLSE_1$ and $RLSE_2$ systems, and the corresponding results are listed in Table 1. The unprocessed noisy speech was also recognized by an ASR system, and the corresponding results are denoted as "Noisy". To test the effectiveness of RL learning, we designed two other sets of experiments with the same 32 IBM vectors: (1) the one-nearest-neighbor ($1nnSE$) (2) the DNN ($DNNSE$) methods were used to determine the IBM vectors for enhancement. Please note that for $DNNSE$, only the MSE term is used to train the DNN model. The enhanced speech was then recognized by the same ASR system; the corresponding results were denoted as $1nnSE$ in Table 1.

When the recognition was tested using the 30 clean testing utterances, the CER was 11.50%. However, as shown in Table 1, when there was noise involved in the background, the CER was dropped considerably to 56.14% and 81.40%, respectively, for 5 dB and 0 dB SNR levels. We then noted that $1nnSE$ could not provide any improvements over Noisy, thus showing that one-nearest-neighbor method could not select the optimal IBM vectors for SE to improve the ASR performance. In addition, $DNNSE$ provided better CER than Noisy at 0dB but worse at 5dB SNR; the results confirm that the MSE criterion for $DNNSE$ does not always improve the recognition accuracy. Furthermore, both $RLSE_1$ and $RLSE_2$ provided better recognition results than those of Noisy, $DNNSE$ and $1nnSE$, and $RLSE_2$ outperformed $RLSE_1$. The relative CER reductions of $RLSE_2$ over Noisy are 12.40% (from 56.14% to 49.18%) at the 5 dB SNR level, and 19.23% (from 81.40% to 65.75%) for the 0 dB SNR level. The results in Table 1 clearly demonstrate the effectiveness of RL-based SE for improving ASR performance in the presence of noise.

To visually analyze the effect of the derived RL-based SE system, we presented the spectrograms of one noisy utterance at the 5 dB SNR level (as shown in Fig. 5 (a)), as well as its clean and enhanced versions by $RLSE_1$ and $RLSE_2$ (as shown in Fig. 5 (b), (c), and (d), respectively). From the figure, noise components of noisy datasets were effectively removed by $RLSE_1$ and $RLSE_2$, thus showing that despite the fact that the goal was to improve the ASR performance, the RL-based SE also performed denoising on the input speech.

Table 1. The average CERs of Noisy (the baseline), $1nnSE$, $DNNSE$, $RLSE_1$, and $RLSE_2$ at 0 and 5 dB SNR conditions.

SNR	*Noisy*	$1nnSE$	$DNNSE$	$RLSE_1$	$RLSE_2$
5 dB	56.14	73.09	60.76	55.60	**49.18**
0 dB	81.40	85.79	73.88	77.20	**65.75**

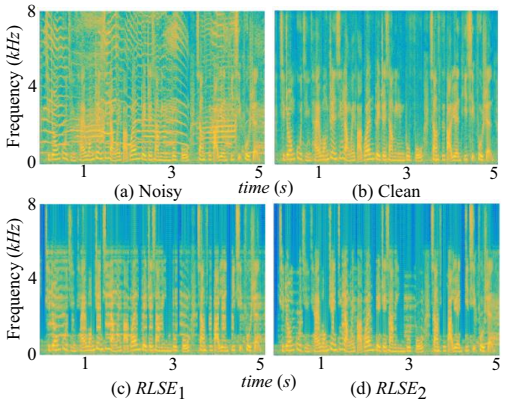

Fig. 5. The spectrograms of (a) Noisy speech, (b) clean speech, (c) enhanced speech by $RLSE_1$, and (d) enhanced speech by $RLSE_2$.

Recent studies have reported a positive correlation between objective intelligibility scores and ASR performance [28, 33]. In Table 2, we show the STOI and PESQ scores of enhanced speech processed by $RLSE_1$ and $RLSE_2$ at SNR levels of0 and 5 dB. The results of the unprocessed noisy speech, shown as Noisy, are also listed for comparison. From this table, we show that both $RLSE_1$ and $RLSE_2$ elicit higher STOI scores than Noisy and $RLSE_2$ provides again clear improvements over $RLSE_1$. From Tables 1 and 2, we can clearly note positive correlations between the STOI scores and ASR performances. As for the PESQ scores, $RLSE_2$ outperformed Noisy but $RLSE_1$ slightly underperformed Noisy. It can be noted that the correlation of the PESQ scores with ASR results is not as strong as that of the STOI scores and the ASR results.

5. CONCLUSION

We present an RL-based SE for robust speech recognition without retraining the ASR system in this study. By using the recognition errors as the objective function, the RL-based SE can effectively reduce CERs by 12.40% and 19.23% at 5 and 0 dB SNR conditions, respectively. Notably, the current work adopts the noisy–clean pair to prepare the IBMs and to train the action estimation module. We also noted that although the objective is to improve ASR performance, the enhanced speech presented denoised properties and was with improved STOI scores. This study serves as a pioneering work for building an SE system with the aim to directly improve ASR performance. The designed scenario is practical in many real-world applications where an ASR engine is supplied by a third-party. In the future, more noise types and SNR levels will be considered to build the RL-based SE system. We will also try to implement the whole system with only noisy speech without the paired clean speech.

Table 2. The STOI and PESQ scores of $RLSE_1$, $RLSE_2$, and Noisy at 0 and 5 dB SNR conditions.

SNR	STOI			PESQ		
	Noisy	$RLSE_1$	$RLSE_2$	*Noisy*	$RLSE_1$	$RLSE_2$
5 dB	0.82	0.82	**0.86**	1.85	1.67	**1.96**
0 dB	0.74	0.77	**0.81**	1.45	1.42	**1.59**

6. REFERENCES

[1] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Academic Press, 2015.

[2] B. Li, Y. Tsao, and K. C. Sim, "An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition.," in *Proc. INTERSPEECH*, pp. 3002–3006, 2013.

[3] Z.-Q. Wang and D. Wang, "A joint training framework for robust automatic speech recognition," *IEEE/ACM TASLP*, vol. 24, no. 4, pp. 796–806, 2016.

[4] A. Acero and R. M. Stern, "Environmental robustness in automatic speech recognition," in *Proc. ICASSP*, pp. 849–852, 1990.

[5] H.-J. Hsieh, B. Chen, and J.-w. Hung, "Employing median filtering to enhance the complex-valued acoustic spectrograms in modulation domain for noise-robust speech recognition," in *Proc. ISCSLP*, pp. 1–5, 2016.

[6] H. Zhang, X. Zhang, and G. Gao, "Training supervised speech separation system to improve stoi and pesq directly," in *Proc. ICASSP*, pp. 5374–5378, 2018.

[7] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE signal processing letters*, vol. 9, no. 1, pp. 12–15, 2002.

[8] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE TASSP*, vol. 28, no. 2, pp. 137–145, 1980.

[9] J. Du, Q. Wang, T. Gao, Y. Xu, L.-R. Dai, and C.-H. Lee, "Robust speech recognition with speech enhanced deep neural networks," in *Proc. INTERSPEECH*, pp. 616–620, 2014.

[10] D. Baby, J. F. Gemmeke, T. Virtanen, *et al.*, "Exemplar-based speech enhancement for deep neural network based automatic speech recognition," in *Proc. ICASSP*, pp. 4485–4489, 2015.

[11] A. J. R. Simpson, "Probabilistic binary-mask cocktail-party source separation in a convolutional deep neural network," *CoRR*, vol. abs/1503.06962, 2015.

[12] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM TASLP*, vol. 26, no. 10, pp. 1702–1726, 2018.

[13] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM TASLP*, vol. 23, no. 1, pp. 7–19, 2015.

[14] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder.," in *Proc. INTERSPEECH*, pp. 436–440, 2013.

[15] Z. Meng, J. Li, Y. Gong, *et al.*, "Adversarial feature-mapping for speech enhancement," *arXiv preprint arXiv:1809.02251*, 2018.

[16] Z. Meng, J. Li, Y. Gong, *et al.*, "Cycle-consistent speech enhancement," *arXiv preprint arXiv:1809.02253*, 2018.

[17] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, vol. 12, no. 2, pp. 19–22, 1992.

[18] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proc. ICRA*, vol. 3, pp. 2619–2624, 2004.

[19] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue management with reinforcement learning: Experiments with the njfun system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 105–133, 2002.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–333, 2015.

[21] R. Fakoor, X. He, I. Tashev, and S. Zarar, "Reinforcement learning to adapt speech enhancement to instantaneous input signal quality," *CoRR*, vol. abs/1711.10791, 2017.

[22] T. Kala and T. Shinozaki, "Reinforcement learning of speech recognition system based on policy gradient and hypothesis selection," in *Proc. ICASSP*, pp. 5759–5763, 2018.

[23] Y. Koizumi, K. Niwa, Y. Hioka, K. Kobayashi, and Y. Haneda, "Dnn-based source enhancement self-optimized by reinforcement learning using sound quality measurements," in *Proc. ICASSP*, pp. 81–85, 2017.

[24] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. ICASSP*, pp. 749–752, 2001.

[25] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE/ACM TASLP*, vol. 19, no. 7, pp. 2125–2136, 2011.

[26] Y. Koizumi, K. Niwa, Y. Hioka, K. Koabayashi, and Y. Haneda, "Dnn-based source enhancement to increase objective sound quality assessment score," *IEEE/ACM TASLP*, vol. 26, no. 10, pp. 1780–1792, 2018.

[27] H.-M. Wang, B. Chen, J.-W. Kuo, and S.-S. Cheng, "Matbn: A mandarin chinese broadcast news corpus," *IJCLCLP*, vol. 10, no. 2, pp. 219–236, 2005.

[28] A. H. Moore, P. P. Parada, and P. A. Naylor, "Speech enhancement for robust automatic speech recognition: Evaluation using a baseline system and instrumental measures," *Computer Speech & Language*, vol. 46, pp. 574–584, 2017.

[29] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Proc. NIPs*, pp. 1061–1069, 2012.

[30] S.-S. Wang, P. Lin, Y. Tsao, J.-W. Hung, and B. Su, "Suppression by selecting wavelets for feature compression in distributed speech recognition," *IEEE/ACM TASLP*, vol. 26, no. 3, pp. 564–579, 2018.

[31] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE TASSP*, vol. 35, no. 3, pp. 400–401, 1987.

[32] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[33] S. Xia, H. Li, and X. Zhang, "Using optimal ratio mask as training target for supervised speech separation," in *Proc. APSIPA*, pp. 163–166, 2017.