INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Phoneme-to-Grapheme Conversion Based Large-Scale Pre-Training for End-to-End Automatic Speech Recognition

Ryo Masumura, Naoki Makishima, Mana Ihori,
Akihiko Takashima, Tomohiro Tanaka, Shota Orihashi

NTT Media Intelligence Laboratories, NTT Corporation, Japan
ryou.masumura.ba@hco.ntt.co.jp

Abstract

This paper describes a simple and efficient pre-training method using a large number of external texts to enhance end-to-end automatic speech recognition (ASR). Generally, it is essential to prepare speech-to-text paired data to construct end-to-end ASR models, but it is difficult to collect a large amount of such data in practice. One issue caused by data scarcity is that the performance of ASR on out-of-domain tasks different from those using the speech-to-text paired data is poor, since the mapping from the speech information to textual information is not well learned. To address this problem, we leverage a large number of phoneme-to-grapheme (P2G) paired data, which can be easily created from external texts and a rich pronunciation dictionary. The P2G conversion and end-to-end ASR are regarded as similar transformation tasks where the input phonetic information is converted into textual information. Our method utilizes the P2G conversion task for pre-training of a decoder network in Transformer encoder-decoder based end-to-end ASR. Experiments using 4 billion tokens of Web text demonstrates that the performance of ASR on out-of-domain tasks can be significantly improved by our pre-training.

Index Terms: end-to-end automatic speech recognition, phoneme-to-grapheme conversion, pre-training, Transformer

1. Introduction

End-to-end automatic speech recognition (ASR) systems that directly convert input speech into text is one of the most attractive technologies in speech-related fields. Although conventional hybrid ASR systems have individually optimized component models, i.e., individual acoustic, pronunciation, and language models, end-to-end ASR systems can achieve total optimization in an end-to-end manner. In fact, the reported end-to-end ASR systems have achieved competitive recognition performance in various ASR tasks.

As ways of achieving much higher ASR performance, several modeling methods have been developed in the last few years. The initial studies mainly utilized connectionist temporal classification [1, 2] and recurrent neural network (RNN) encoder-decoder [3–6] for the end-to-end ASR. In addition, recent studies have employed the transformer encoder-decoder, which has shown much stronger performance [7, 8].

While end-to-end ASR systems achieve total optimization, one weakness is that speech-to-text paired data are essential for optimization. In fact, it is difficult to collect a large number of such data in practice. To deal with this problem, several studies have developed methods that utilize unpaired speech data and unpaired text data in semi-supervised learning and in self-supervised pre-training (see Sec. 2). The unpaired text data are usually used for language model (LM) fusion [9, 10]. Unfortunately, the LM fusion approach is not effective to improve ASR performance of out-of-domain tasks which are different from those using speech-to-text paired data because LM fusion cannot learn the mapping from the speech information to the textual information. In addition, LM fusion cannot solve the out-of-vocabulary problem; i.e., tokens that do not appear in the paired data cannot be recognized. To tackle these problems, previous studies have utilized text-to-speech models to create speech-to-text paired data from unpaired text data [11–13]. However, these methods are strongly affected by the performance of the text-to-speech models. In fact, we cannot generate appropriate speech data from text data in unseen linguistic contexts if we construct the text-to-speech part from a limited amount of speech-to-text paired data.

Our idea is to leverage a large number of phoneme-to-grapheme (P2G) paired data, which can be created from external text data and a rich pronunciation dictionary, for enhancing end-to-end ASR modeling. The P2G conversion and end-to-end ASR are regarded as similar transformation tasks, where input phonetic information is converted into textual information. Thus, it can be considered that a decoder network can be shared between the P2G conversion and end-to-end ASR. Different from utilizing text-to-speech models, the input phonetic information is reliable, since it is created by looking it up in the pronunciation dictionary. In fact, a similar idea that utilizes subword-to-word paired data has been used to improve end-to-end ASR performance [14]. However, previous studies have used only a small amount of unpaired text data and the effect of using large-scale unpaired text data such as Web data remains unknown. In addition, it has not been verified that the P2G paired data can improve end-to-end ASR even in modern modelings such as Transformer.

In this paper, we propose to use a P2G-conversion-based pre-training with large-scale unpaired text data to enhance the Transformer encoder-decoder based end-to-end ASR. In our method, we first construct a Transformer encoder-decoder based P2G conversion model by using unpaired text data. Next, we transfer the trained decoder network to the Transformer encoder-decoder based end-to-end ASR model and fine-tune it by using speech-to-text paired data. In our experiments on Japanese spontaneous ASR tasks, we used 4 billion tokens of external text collected from the Web. We demonstrate that the P2G-conversion-based large-scale pre-training significantly improve the performance of ASR on out-of-domain tasks. In addition, we verify the relationship between our method and LM shallow fusion.

2. Related Work

Semi-supervised learning: This study is related to semi-supervised learning for end-to-end ASR. In semi-supervised learning, speech-to-text paired data, unpaired speech data, and

http://dx.doi.org/10.21437/Interspeech.2020-1930

unpaired text data are jointly used for optimizing end-to-end ASR models. The dominant approaches use speech chain modeling [15], reconstruction modeling [16–18], consistency training [19] and self-training [20, 21]. In these methods, unlabeled text data are usually used for text-to-speech modeling, so they are not suitable for improving the performance of ASR in out-of-domain tasks. In this study, we utilize unpaired text data and devise a method that allows the text data to be converted into phoneme-to-grapheme paired data by using a pronunciation dictionary.

Self-supervised pre-training: Our method is motivated by the self-supervised pre-training for encoder-decoder-based models. In end-to-end ASR, self-supervised pre-training has been mainly examined for encoder networks that convert input speech into hidden representations [22–24]. On the other hand, self-supervised pre-training for decoder networks has not been examined in the context of end-to-end ASR. In recent natural language processing, encoder-decoder-based models for text-to-text conversion tasks have been pre-trained using unpaired text data in a self-supervised manner. Their self-supervision tasks are made by masking parts of the tokens and rearranging tokens [25, 26]. Our method is regarded as a self-supervised learning that defines the self-supervision task by utilizing the pronunciation dictionary.

3. Proposed Method

We examine P2G-based pre-training to enhance end-to-end ASR models. Here, we can use both speech-to-text paired data $\mathcal{D}_1 = \{(\boldsymbol{X}^1, \boldsymbol{W}^1), \cdots, (\boldsymbol{X}^T, \boldsymbol{W}^T)\}$ and unpaired text data $\mathcal{D}_2 = \{\boldsymbol{W}^{T+1}, \cdots, \boldsymbol{W}^{T+C}\}$. In this study, we convert the unpaired text data into P2G paired data $\mathcal{D}_2 = \{(\boldsymbol{Q}^{T+1}, \boldsymbol{W}^{T+1}), \cdots, (\boldsymbol{Q}^{T+C}, \boldsymbol{W}^{T+C})\}$ by using a pronunciation dictionary. Our objective is to train the model parameter of the end-to-end ASR model from these two data sets. We define a P2G conversion model and an end-to-end ASR model by using the Transformer encoder-decoder, which is a conditional auto-regressive model. Our main idea is to transfer the decoder network in the P2G conversion model to the end-to-end ASR.

3.1. Phoneme-to-Grapheme Conversion

Our phoneme-to-grapheme conversion predicts the generation probability of a grapheme sequence (a token sequence) $\boldsymbol{W} = \{w_1, \cdots, w_N\}$ given a phoneme sequence $\boldsymbol{Q} = \{q_1, \cdots, q_L\}$, where w_n is the n-th token in the grapheme sequence and q_l is the l-th phoneme in the phoneme sequence. In the conditional auto-regressive models, the generation probability of $\boldsymbol{W}$ is defined as

$$P(\boldsymbol{W}|\boldsymbol{Q}; \boldsymbol{\Theta}_{\mathrm{p2g}}) = \prod_{n=1}^{N} P(w_n|\boldsymbol{W}_{1:n-1}, \boldsymbol{Q}; \boldsymbol{\Theta}_{\mathrm{p2g}}), \quad (1)$$

where $\boldsymbol{\Theta}_{\mathrm{p2g}} = \{\boldsymbol{\theta}_{\mathrm{penc}}, \boldsymbol{\theta}_{\mathrm{dec}}\}$ represents the trainable model parameter sets and $\boldsymbol{W}_{1:n-1} = \{w_1, \cdots, w_{n-1}\}$. In our Transformer encoder-decoder based P2G conversion model, $P(w_n|\boldsymbol{W}_{1:n-1}\ \boldsymbol{Q};\ \boldsymbol{\Theta}_{\mathrm{p2g}})$ is computed using a phoneme encoder and a text decoder.

3.2. End-to-End ASR

Our end-to-end ASR is modeled by conditional auto-regressive modeling. We predict the generation probability of a token sequence $\boldsymbol{W} = \{w_1, \cdots, w_N\}$ given input speech $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M\}$, where $\boldsymbol{x}_m$ is the m-th acoustic feature in the speech. In the conditional auto-regressive models, the generation probability of $\boldsymbol{W}$ is defined as

$$P(\boldsymbol{W}|\boldsymbol{X}; \boldsymbol{\Theta}_{\mathrm{asr}}) = \prod_{n=1}^{N} P(w_n|\boldsymbol{W}_{1:n-1}, \boldsymbol{X}; \boldsymbol{\Theta}_{\mathrm{asr}}), \quad (2)$$

where $\boldsymbol{\Theta}_{\mathrm{asr}} = \{\boldsymbol{\theta}_{\mathrm{senc}}, \boldsymbol{\theta}_{\mathrm{dec}}\}$ represents the trainable model parameter sets. In our Transformer encoder-decoder based end-to-end ASR model, $P(w_n|\boldsymbol{W}_{1:n-1}\ \boldsymbol{Q};\ \boldsymbol{\Theta}_{\mathrm{p2g}})$ is computed using a speech encoder and a text decoder. Note that $\boldsymbol{\theta}_{\mathrm{dec}}$ is the same one as in Eq. (1). Thus, we use a sharable text decoder between the end-to-end ASR model and the P2G conversion model.

3.3. Transformer encoder-decoder based modeling

The P2G conversion model and the end-to-end ASR model are fully formed from Transformer encoder-decoder networks.

Phoneme encoder: The phoneme encoder converts a input phoneme sequence $\boldsymbol{Q}$ into hidden representations $\boldsymbol{S}^{(K)}$ by using K Transformer encoder blocks. The k-th Transformer encoder block composes the k-th hidden representations $\boldsymbol{S}^{(k)}$ from the lower layer inputs $\boldsymbol{S}^{(k-1)}$, as

$$\boldsymbol{S}^{(k)} = \texttt{TransformerEncoderBlock}(\boldsymbol{S}^{(k-1)}; \boldsymbol{\theta}_{\mathrm{penc}}), \quad (3)$$

where $\texttt{TransformerEncoderBlock}()$ is a Transformer encoder block that consists of a scaled dot product multi-head self-attention layer and a position-wise feed-forward network [7]. The hidden representations $\boldsymbol{S}^{(0)} = \{\boldsymbol{s}_1^{(0)}, \cdots, \boldsymbol{s}_L^{(0)}\}$ are produced by

$$\boldsymbol{s}_l^{(0)} = \texttt{AddPositionalEncoding}(\boldsymbol{q}_l), \quad (4)$$

$$\boldsymbol{q}_l = \texttt{Embedding}(q_l; \boldsymbol{\theta}_{\mathrm{penc}}), \quad (5)$$

where $\texttt{AddPositionalEncoding}()$ is a function that adds a continuous vector in which position information is embedded. $\texttt{Embedding}()$ is a linear layer that embeds the input token in a continuous vector.

Speech encoder: The speech encoder converts the input acoustic features $\boldsymbol{X}$ into hidden representations $\boldsymbol{H}^{(I)}$ by using I Transformer encoder blocks. The i-th Transformer encoder block composes the i-th hidden representations $\boldsymbol{H}^{(i)}$ from the lower layer inputs $\boldsymbol{H}^{(i-1)}$, as

$$\boldsymbol{H}^{(i)} = \texttt{TransformerEncoderBlock}(\boldsymbol{H}^{(i-1)}; \boldsymbol{\theta}_{\mathrm{senc}}), \quad (6)$$

The hidden representations $\boldsymbol{H}^{(0)} = \{\boldsymbol{h}_1^{(0)}, \cdots, \boldsymbol{h}_{M'}^{(0)}\}$ are produced by

$$\boldsymbol{h}_{m'}^{(0)} = \texttt{AddPostionalEncoding}(\boldsymbol{h}_{m'}), \quad (7)$$

$$\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{M'}\} = \texttt{ConvPool}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M; \boldsymbol{\theta}_{\mathrm{senc}}), \quad (8)$$

where $\texttt{ConvPool}()$ is a function composed of convolution layers and pooling layers. M' is the subsampled sequence length depending on the function.

Sharable text decoder: The text decoder computes the generative probability of a token from the preceding tokens and the hidden representations of the phoneme information or the speech. The predicted probabilities of the n-th token w_n are calculated as

$$P(w_n|\boldsymbol{W}_{1:n-1}, \boldsymbol{O}) = \texttt{Softmax}(\boldsymbol{u}_{n-1}^{(J)}; \boldsymbol{\theta}_{\mathrm{dec}}), \quad (9)$$

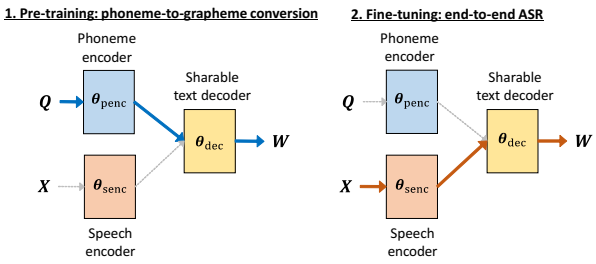

Figure 1: *Training procedure of our method.*

$$\boldsymbol{O} = \begin{cases} \boldsymbol{Q} & \text{if input is phoneme} \\ \boldsymbol{X} & \text{if input is speech,} \end{cases} \quad (10)$$

where $\texttt{Softmax}()$ is a softmax layer with a linear transformation. The input hidden vector $\boldsymbol{u}_{n-1}^{(J)}$ is computed from J Transformer decoder blocks. The j-th Transformer decoder block composes the j-th hidden representation $\boldsymbol{u}_{n-1}^{(j)}$ from the lower layer inputs $\boldsymbol{U}_{1:n-1}^{(j-1)} = \{\boldsymbol{u}_1^{(j-1)}, \cdots, \boldsymbol{u}_{n-1}^{(j-1)}\}$, as

$$\boldsymbol{u}_{n-1}^{(j)} = \texttt{TransformerDecoderBlock}(\boldsymbol{U}_{1:n-1}^{(j-1)}, \boldsymbol{Z}; \boldsymbol{\theta}_{\text{dec}}), \quad (11)$$

$$\boldsymbol{Z} = \begin{cases} \boldsymbol{S}^{(K)} & \text{if input is phoneme} \\ \boldsymbol{H}^{(I)} & \text{if input is speech,} \end{cases} \quad (12)$$

where $\texttt{TransformerDecoderBlock}()$ is a Transformer decoder block that consists of a scaled dot product multi-head masked self-attention layer, a scaled dot product multi-head source-target attention layer, and a position-wise feed-forward network [7]. The hidden representations $\boldsymbol{U}_{1:n-1}^{(0)} = \{\boldsymbol{u}_1^{(0)}, \cdots, \boldsymbol{u}_{n-1}^{(0)}\}$ are produced by

$$\boldsymbol{u}_{n-1}^{(0)} = \texttt{AddPositionalEncoding}(\boldsymbol{w}_{n-1}), \quad (13)$$

$$\boldsymbol{w}_{n-1} = \texttt{Embedding}(w_{n-1}; \boldsymbol{\theta}_{\text{dec}}). \quad (14)$$

As mentioned above, the text decoder is sharable between the P2G conversion model and the end-to-end ASR model.

3.4. Pre-Training and Fine-Tuning

In this study, these networks are trained in two stages by using the speech-to-text paired data $\mathcal{D}_1 = \{(\boldsymbol{X}^1, \boldsymbol{W}^1), \cdots, (\boldsymbol{X}^T, \boldsymbol{W}^T)\}$ and the P2G paired data $\mathcal{D}_2 = \{(\boldsymbol{Q}^{T+1}, \boldsymbol{W}^{T+1}), \cdots, (\boldsymbol{Q}^{T+C}, \boldsymbol{W}^{T+C})\}$. Figure 1 shows the training procedure of our method. In the first stage, we train the P2G conversion model with the P2G paired data. The model parameters for the P2G conversion model are optimized by

$$\hat{\boldsymbol{\theta}}_{\text{penc}}, \hat{\boldsymbol{\theta}}_{\text{dec}} = - \underset{\boldsymbol{\theta}_{\text{penc}}, \boldsymbol{\theta}_{\text{dec}}}{\text{argmin}} \sum_{c=1}^{C} \sum_{n=1}^{N^{T+c}} \log P(w_n^{T+c} | \boldsymbol{W}_{1:n-1}^{T+c}, \boldsymbol{Q}^{T+c}; \boldsymbol{\theta}_{\text{penc}}, \boldsymbol{\theta}_{\text{dec}}), \quad (15)$$

where w_n^{T+c} is the n-th token for $\boldsymbol{W}^{T+c}$ and $\boldsymbol{W}_{1:n-1}^{t} = \{w_1^t, \cdots, w_{n-1}^t\}$. N^{T+c} is the number of tokens in $\boldsymbol{W}^{T+c}$.

In the second stage, we transfer the trained decoder parameter $\hat{\boldsymbol{\theta}}_{\text{dec}}$ to that for the end-to-end ASR model and fine-tune the end-to-end ASR model by using the speech-to-text paired data. The model parameters for the end-to-end ASR model are optimized by

$$\hat{\boldsymbol{\theta}}_{\text{senc}}, \hat{\hat{\boldsymbol{\theta}}}_{\text{dec}} = - \underset{\boldsymbol{\theta}_{\text{senc}}, \boldsymbol{\theta}_{\text{dec}}}{\text{argmin}} \sum_{t=1}^{T} \sum_{n=1}^{N^t} \log P(w_n^t | \boldsymbol{W}_{1:n-1}^{t}, \boldsymbol{X}^t; \boldsymbol{\theta}_{\text{senc}}, \boldsymbol{\theta}_{\text{dec}}), \quad (16)$$

Table 1: *Speech-to-text paired data sets.*

	Domain	Data size (Hours)	Number of characters
Train 1	CSJ-A	252.5	6,747,386
Train 2	CSJ-S	263.1	6,679,489
Test 1	CSJ-A	1.8	48,064
Test 2	CSJ-A	1.9	47,970
Test 3	CSJ-S	1.3	32,089
Test 4	CCDC	3.5	65,843
Test 5	CJLC	4.1	84,641

where $\hat{\hat{\boldsymbol{\theta}}}_{\text{dec}}$ is the fine-tuned decoder parameter after the pre-training. In addition, we examine the fine-tuning while freezing the trained decoder parameter.

4. Experiments

Our experiments used three Japanese ASR corpora. One is the Corpus of Spontaneous Japanese (CSJ) [27], which includes two training data sets (Train 1 and 2) and three test data sets (Test 1, 2 and 3). Train 1, Test 1 and 2 are academic presentation speech (CSJ-A). Train 2 and Test 3 are simulated public speech (CSJ-S). The other two corpora are our home-made Japanese contact center dialogue corpus (CCDC) and the corpus Japanese classroom lecture speech contents (CJLC), each of which were used for only testing in the out-of-domain tasks. We denote the CCDC as Test 4, and the CJLC as Test 5. Details of the data sets are shown in Table 1. Note that this paper uses characters as the tokens. We assessed the following two ASR setups.

- *Setting A*: Training data are drawn from Train 1. Thus, Test 1 and 2 are in-domain tasks and Test 3, 4 and 5 are out-of-domain tasks.
- *Setting B*: Training data are drawn from Train 1 and 2. Thus, Test 1, 2 and 3 are in-domain tasks and Test 4 and 5 are out-of-domain tasks.

In addition, we prepared large-scale Japanese Web text as unpaired text data. The Web text was downloaded from various topic Web pages by using our home-made crawler. The downloaded pages were filtered for excluding HTML tags, Javascript codes and other parts that were not useful for the ASR modeling. As a result, about 0.2 billion sentences with 4 billion characters were prepared. The Japanese Web text was used for both P2G-conversion-based pre-training and language modeling for LM fusion. For the experiments, we created subsets of the collected Web data. We randomly sampled sentences from the full data and composed 0.4 billion and 0.04 billon tokens of subsets. To perform the P2G-conversion-based pre-training, we converted all Web data into phoneme sequences by using our home-made morphological analyzer with a rich Japanese pronunciation dictionary.

4.1. Setups

In our experiments, we modeled both the P2G conversion models and end-to-end ASR models using Transformer encoder-decoders. For the P2G conversion models and the end-to-end ASR models, the Transformer blocks were composed under the following conditions: the dimensions of the output continuous representations were set to 256, the dimensions of the inner outputs in the position-wise feed forward networks were set to 2,048, and the number of heads in the multi-head attentions was

Table 2: *Experimental results in terms of character error rate (%). ID means in-domain and OOD means out-of-domain.*

	Language model shallow fusion	P2G-based pre-training	Decoder freezing	Data size of external text	Test 1	Test 2	Test 3	Test 4	Test 5
Setting A					ID	ID	OOD	OOD	OOD
Baseline	-	-	-	-	9.8	7.9	16.1	32.6	27.1
Baseline + LM	✓	-	-	4 billion	9.6	7.7	15.5	32.2	26.7
Our method	-	✓	freeze	0.04 billion	12.3	8.7	16.3	33.4	28.4
Our method	-	✓	unfreeze	0.04 billion	9.6	7.5	14.3	31.3	26.1
Our method	-	✓	freeze	0.4 billion	11.2	7.9	14.7	29.1	26.7
Our method	-	✓	unfreeze	0.4 billion	9.5	7.3	13.7	29.5	26.2
Our method	-	✓	freeze	4 billion	10.7	7.6	14.1	28.8	26.2
Our method	-	✓	unfreeze	4 billion	9.2	7.0	13.3	28.4	25.0
Our method + LM	✓	✓	unfreeze	4 billion	**9.0**	**6.8**	**12.9**	**28.0**	**24.6**
Setting B					ID	ID	ID	OOD	OOD
Baseline	-	-	-	-	6.8	5.0	6.0	26.5	23.8
Baseline + LM	✓	-	-	4 billion	6.7	4.9	5.8	26.2	23.5
Our method	-	✓	freeze	0.04 billion	10.4	7.8	7.5	25.9	25.9
Our method	-	✓	unfreeze	0.04 billion	6.7	4.9	5.9	25.3	23.1
Our method	-	✓	freeze	0.4 billion	9.5	7.3	7.0	25.3	25.9
Our method	-	✓	unfreeze	0.4 billion	6.6	4.7	5.6	24.3	22.8
Our method	-	✓	freeze	4 billion	8.5	5.8	6.4	24.6	25.4
Our method	-	✓	unfreeze	4 billion	6.5	4.4	5.4	23.9	22.4
Our method + LM	✓	✓	unfreeze	4 billion	**6.4**	**4.3**	**5.3**	**23.5**	**22.0**

set to 4. In the nonlinear transformational functions, the GELU activation was used. For the speech encoder, we used 40 log mel-scale filterbank coefficients appended with delta and acceleration coefficients as acoustic features. The frame shift was 10 ms. The acoustic features passed two convolution and max pooling layers with a stride of 2, so we down-sampled them to $1/4$ along with the time axis. After these layers, we stacked 8-layer transformer encoder blocks. For the phoneme encoder, we used 256-dimensional phoneme embeddings where the vocabulary size was set to 86. We stacked 4-layer transformer encoder blocks. In the text decoder, we used 256-dimensional character embeddings where the vocabulary size was set to 5,777.

For the training, we used the Radam optimizer [28]. The training steps were stopped based on early stopping using part of the training data. We set the mini-batch size to 64 sentences/utterances and the dropout rate in the Transformer blocks to 0.1. In addition, for optimizing the P2G conversion models and end-to-end ASR models, we introduced label smoothing [29] and scheduled sampling [30]. For the label smoothing, a smoothing parameter was set as 0.1. Our scheduled sampling-based optimization process used the teacher forcing at the beginning of the training steps, and we linearly ramped up the probability of sampling to the specified probability at the specified epoch. Furthermore, for optimizing the end-to-end ASR models, we used SpecAugment [31]. Our SpecAugment only applied frequency masking and time masking, where the number of frequency masks and time step masks were set to 2, the frequency masking width was randomly chosen from 0 to 20 frequency bins, and the time masking width was randomly chosen from 0 to 100 frames. When we examined fine-tuning based on the P2G-conversion-based pre-training, we evaluated two setups: decoder-freezing and decoder-unfreezing transfers. For testing, we used a beam search algorithm in which the beam size was set to 20. In addition, to compare the P2G-conversion-based pre-training with other methods using unpaired text, we examined log-linear interpolation LM shallow fusion [9, 10] using a two-layer LSTM-based LM trained from all the Web data. The number of units in each LSTM was set to 512. The weight factor for the log-linear interpolation was set to 0.1.

4.2. Results

Table 2 shows the results in terms of character error rate for settings A and B. The baseline represents results that only used the speech-to-text data, while the other results used both the speech-to-text data and the unpaired text data. First, the results show freezing the trained decoder parameter in the fine-tuning achieved moderate ASR performance. This indicates that the decoder network trained via the P2G conversion modeling matches the end-to-end ASR even though its source-target attention mechanism is trained so as to handle phoneme sequences. Next, the results show that our methods with unfreezing of the decoder network had better ASR performance than the baseline on each test set. In particular, our approach significantly improved ASR performance on the out-of-domain test sets. This suggests that the pre-training using large-scale Web text is effective for learning unknown mappings from input phonetic information to textual information. The results also show the effectiveness of increasing the amount of unpaired text data. Furthermore, our methods significantly outperformed the LM fusion approach even when we used the same unpaired text data. This is because our methods can directly capture the mapping from the input phonetic information to textual information. The highest results were attained by combining the P2G-conversion-based pre-training with LM shallow fusion. These confirm that our pre-training method can yield effective performance improvements even when combining with LM shallow fusion.

5. Conclusions

We presented a phoneme-to-grapheme conversion based large-scale pre-training to improve end-to-end ASR systems. The strength of our method is that it learns an unknown mapping from phonetic information to textual information by utilizing large-scale Web text. Our experiments using 4 billion tokens of Web text demonstrated that ASR performance on the out-of-domain tasks can be significantly improved by using P2G-conversion-based large-scale pre-training. In addition, we showed that our method can yield performance improvements even when combining with LM shallow fusion.

6. References

[1] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4805–4809, 2017.

[2] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 959–963, 2017.

[3] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4945–4949, 2015.

[4] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3249–3253, 2015.

[5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

[6] R. Masumura, T. Tanaka, T. Moriya, Y. Shinohara, T. Oba, and Y. Aono, "Large context end-to-end automatic speech recognition via extension of hierarchical recurrent encoder-decoder models," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5661–5665, 2019.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *In Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

[8] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5884–5888, 2018.

[9] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 523–527, 2017.

[10] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 949–953, 2017.

[11] M. Mimura, S. Ueno, H. Inaguma, S. Sakai, and T. Kawahara, "Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition," *In Proc. Spoken Language Technology Workshop (SLT)*, pp. 477–484, 2018.

[12] R. Masumura, H. Sato, T. Tanaka, T. Moriya, Y. Ijima, and T. Oba, "End-to-end automatic speech recognition with a reconstruction criterion using speech-to-text and text-to-speech encoder-decoders," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1606–1610, 2019.

[13] N. Rossenbach, A. Zeyer, R. Schluter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7064–7068, 2020.

[14] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multimodal data augmentation for end-to-end ASR," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2394–2398, 2018.

[15] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," *In Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 301–308, 2017.

[16] S. Karita, S. Watanabe, T. Iwata, A. Ogawa, and M. Delcroix, "Semi-supervised end-to-end speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2–6, 2018.

[17] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. L. Roux, "Cycle-consistency training for end-to-end speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6271–6275, 2019.

[18] M. K. Baskar, S. Watanabe, R. Astudillon, T. Hori, L. Burget, and J. Cernocky, "Semi-supervised sequence-to-sequence ASR using unpaired speech and text," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3790–3794, 2019.

[19] R. Masumura, M. Ihori, A. Takashima, T. Moriya, A. Ando, and Y. Shinohara, "Sequence-level consistency training for semi-supervised end-to-end automatic speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7049–7053, 2020.

[20] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-End ASR: from supervised to semi-supervised learning with modern architectures," *In Proc. International Conference on Learning Representations (ICLR) Workshop on Self-supervision in Audio and Speech*, 2020.

[21] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7079–7083, 2020.

[22] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3465–3469, 2019.

[23] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6884–6888, 2020.

[24] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acousitc representations for semi-supervised speech recognition," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6424–6428, 2020.

[25] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mass: Masked sequence to sequence pre-training for language generation," *In Proc. International Conference on Machine Learning (ICML)*, pp. 5926–5936, 2019.

[26] L. Wang, W. Zhao, R. Jia, S. Li, and J. Liu, "Denoising based sequence-to-sequence pre-training for text generation," *In Proc. Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4003–4015, 2019.

[27] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," *In Proc. International Conference on Language Resources and Evaluation (LREC)*, pp. 947–952, 2000.

[28] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *In Proc. International Conference on Learning Representations (ICLR)*, 2020.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *In Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[30] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *In Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1171–1179, 2015.

[31] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2613–2617, 2019.