https://doi.org/10.26599/AIR.2024.9150039 CAAI Artificial Intelligence Research

Pretraining Enhanced RNN Transducer

Junyu Lu[1], Rongzhong Lian[1], Di Jiang[1] ✉, Yuanfeng Song[1], Zhiyang Su[2], Victor Junqiu Wei[2], and Lin Yang[2]

ABSTRACT

Recurrent neural network transducer (RNN-T) is an important branch of current end-to-end automatic speech recognition (ASR). Various promising approaches have been designed for boosting RNN-T architecture; however, few studies exploit the effectiveness of pretrained methods in this framework. In this paper, we introduce the pretrained acoustic extractor (PAE) and the pretrained linguistic network (PLN) to enhance the Conformer long short-term memory (Conformer-LSTM) transducer. First, we construct the input of the acoustic encoder with two different latent representations: one extracted by PAE from the raw waveform, and the other obtained from filter-bank transformation. Second, we fuse an extra semantic feature from the PLN into the joint network to reduce illogical and homophonic errors. Compared with previous works, our approaches are able to obtain pretrained representations for better model generalization. Evaluation on two large-scale datasets has demonstrated that our proposed approaches yield better performance than existing approaches.

KEYWORDS

pretraining; automatic speech recognition; self-supervised learning

End-to-end automatic speech recognition (ASR) systems can directly learn language or pronunciation information without manual alignment labels, which greatly simplifies the complexity of traditional pipeline speech recognition[1–4]. RNN transducer (RNN-T) is one of the promising end-to-end ASR models and has drawn more attention recently[5]. Roughly speaking, RNN-T consists of three major components: an encoder network, a prediction network, and a joint network[6, 7]. The encoder network maps input acoustic frames, i.e., filter-banks or raw waveforms, into a higher-level representation, and the prediction network generates tokens conditioned on the history of previous predictions. Then the joint network merges the latent representations from the encoder network and the prediction network together and finally outputs the proper tokens.

Recently, pretraining enhanced ASR systems have emerged as an effective technique, where the acoustic model is trained on large amounts of unlabeled audios via self-supervise learning. Schneider et al.[8] first introduced a convolutional neural network (CNN) that takes raw audio as input and optimize it via a next-time step prediction task. Since the milestone breakthrough of masked language modeling[9], Baevski et al.[10] presented a Transformer[11]-based framework, Wav2vec 2.0, for self-supervised learning from raw audios by contrastive learning. Zhang et al.[12] further changed Transformer encoder to more advanced Conformer encoder[13] and proposed a recipe for pretraining. Although the pretraining approaches achieve impressive results on the acoustic modeling, few explorations consider well-pretrained models to enhance RNN-T. A common solution to ease the RNN-T training is to initialize the acoustic model with a connectionist temporal classification (CTC)[14] loss trained on labeled audios and pretrain the prediction network on text-only data in advance[15]. Yet, it remains a significant gap between Wav2vec 2.0 and CTC-based approaches due to hard-to-obtain labeled data, which is worth a discussion on how to deploy a pretrained acoustic extractor (PAE) into RNN-T systems. Notably, there are some novel works about how to utilize networks to solve the nonlinear dynamical systems[16–19].

Most early ASR systems[20–27] narrowly consider fixed, handcrafted filter-banks as the input features. With the development of deep learning, several attempts try to learn a convolutional filter directly from the raw waveform[28–31]. Wav2vec 2.0 processes the raw waveform of the speech audio with a CNN-based acoustic extractor to subsample audios. With a large amount of audios pretraining, Wav2vec 2.0 lead to significant improvement on downstream tasks and these latent audio representations are desired to be more robustness for different scenes. Inspired by this, we format our input acoustic features by two convolutional feature extractors. One of them is a 2-D CNN that takes log Mel-filterbanks as input, and the other one is Wav2vec 2.0 feature extractor.

Different from the acoustic encoder, the prediction network is firstly designed to capture linguistic information, similar to a language model (LM). If this is the case, it may be possible to make great strides by pretraining the prediction network via an LM loss in advance. Unfortunately, it seldom improves the performance and can hardly reduce illogical homophonic errors in our experiments. Ghodsi et al.[32] recently revealed removing recurrent layers from the prediction network performs virtually as well as the original model. It can be concluded that the prediction network in RNN-T may serve as a more important role for alignment, not function as language model. Given the above, we fine-tune a pretrained language model, such as GPT-2, with audio transcripts, then fuse contextualize representations in RNN-T. As is shown in Fig. 1, the acoustic features, the predicted alignment features and the pretrained linguistic features are finally merged in the joint network.

In this paper, we introduce two new components in RNN-T framework: (1) PAE and (2) pretrained linguistic network (PLN).

1 WeBank Co., Ltd., Shenzhen 518000, China
2 Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong 999077, China
Address correspondence to Di Jiang, dijiang@webank.com

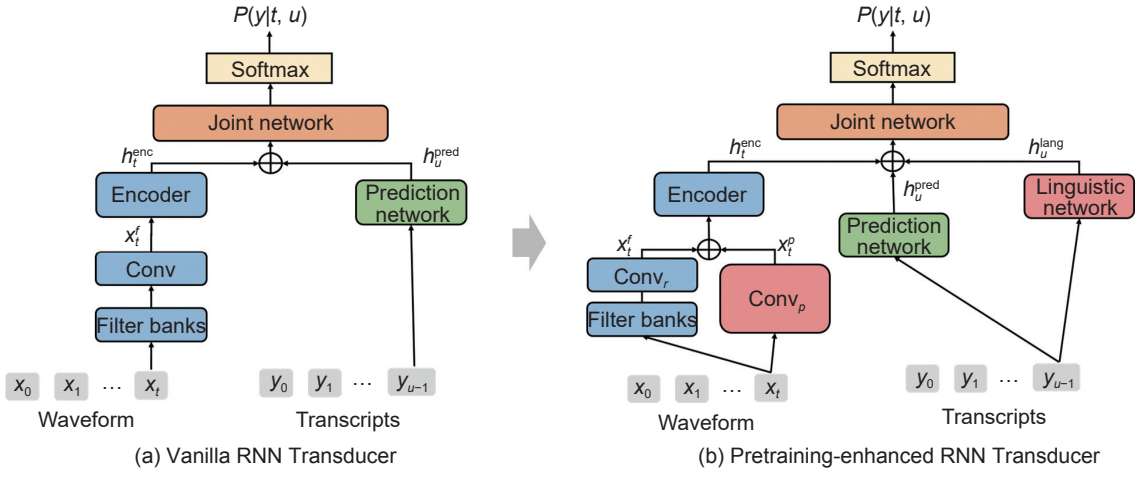

Fig. 1 Our pretrained feature-based framework. The Conv$_p$ and the linguistic network in red color is newly introduced.

It is natural to adopt the Wav2vec 2.0 features to enhance the acoustic encoder, which commonly employs convolution neural network to subsample acoustic frames from raw waveform. Then, we construct input features by summing up latent representation from filter-banks and raw waveform together. Furthermore, in order to produce semantic results, we introduce a linguistic network to provide an extra linguistic feature and fuse it in the joint network. By the way, this linguistic network can be any unidirectional language models that is pretrained on plenty of universal data and fine-tune on domain-specific audio transcripts.

To the best of our knowledge, this paper is the first to utilize PAE and PLN in Conformer long short-term memory (Conformer-LSTM) transducer. Intensive experiments and ablation study show that our methods can achieve better results on large scale datasets compared to existing approaches. We observe in our experiments that the word error rate (WER) drops by average 3.57% in Librispeech[33] and the character error rate (CER) drops 5.75% relatively in AISHELL-1[34] when adopting pretrained features. Besides, the experimental results on different scale of data demonstrate our approaches can improve the performance in low-resource scenario. Last but not least, we compare warm-start and cold-start training procedure and verify that our feature-based framework is a suitable way to enhance RNN-T architecture.

1 Related Work

1.1 Pretrained acoustic models

In the past few years, pretraining neural networks has been proven to be a great way to overcome limited data on computer vision (CV)[35], natural language processing (NLP)[9, 36–41], and ASR[8, 10, 12]. The key idea is to learn general representations and leverage the learned representations on a downstream task for which the amount of data is limited. In computer vision, representations for ImageNet[42] and COCO[43] have also shown promise to initialize models for tasks such as image captioning[44] or pose estimation[45]. In NLP, unsupervised pretraining of language models[9, 36, 38, 39] improves many tasks such as text classification, phrase structure parsing, and machine translation. As we know, most of existing ASR technologies, either pipeline hybrid systems and end-to-end systems, require large amounts of transcribed audio data to attain good performance. This is particularly interesting for ASR tasks where substantial effort is required to obtain labeled data.

Recently, there are many pretraining methods emerged in speaker identification[46, 47], speech emotion analysis[48], machine anomalous sound detection[49], phoneme discrimination[50, 51], as well as transferring ASR representations from one language to another[52]. There has been works on unsupervised learning for speech, but the resulting representations have not been applied to improve supervised speech recognition. Inspired by the pretrained language modeling[9], Schneider et al.[8] applied a contrastive loss[51, 53] in a convolution neural network, namely Wav2vec, to leverage pretrained representations for downstream speech recognition. This contrastive loss requires the Wav2vec model to distinguish a true future audio sample from negatives, like a next step prediction task. Furthermore, Baevski et al.[10] proposed an evolution version, Wav2vec 2.0, which consists of Transformer layers and a convolutional feature extractor. It first applies a feature extractor to subsample the raw audio, then builds context representations over the frame representations, and self-attention captures dependencies over the entire sequence of latent representations. The pretraining objective is the sum of a contrastive and a diversity loss, where the model needs to identify the true quantized latent speech representation and encourage the model to use the codebook entries equally. After pretraining, we can add a randomly initialized linear projection on top of Wav2vec 2.0 and finetune by minimizing a CTC[14]/RNN-T[6]/AED[54] loss on the downstream task.

1.2 RNN-T architecture and loss

As is shown in Fig. 1, the vanilla RNN-T framework is composed of an acoustic encoder, a prediction network and a joint network, which is more likely to optimize the acoustic model and the language model jointly. Given an audio x with T acoustic frames and a transcript sequence y with U tokens, RNN-T's encoder network sequentially converts the acoustic feature $x_{0:t}$ into a high-level representation h_t^{enc}. The prediction network produces a high-level representation h_u^{pred} according to the previous non-blank targets $y_{0:u-1}$. The joint network is a feed-forward network that combines the encoder network output h_t^{enc} and the prediction network output h_u^{pred} to generate the joint matrix $z_{t,u}$. In vanilla RNN-T, the encoder network and the prediction network are usually realized by LSTM[55] units. When implemented with an LSTM acoustic encoder and an LSTM prediction network, we formulate this LSTM-LSTM architecture as following:

$$h_t^{\text{enc}} = \text{LSTM}(x_t, h_{t-1}^{\text{enc}}) \tag{1}$$

$$h_u^{\text{pred}} = \text{LSTM}(y_{u-1}, h_{u-1}^{\text{pre}}) \tag{2}$$

$$o_{t,u} = \text{Joint}(h_t^{\text{enc}}, h_u^{\text{pred}}) \tag{3}$$

$$P(y|t,u) = \text{SoftMax}(o_{t,u}) \tag{4}$$

where $t \in [0,T]$, $u \in [0,U]$, and $h_t^{\text{enc}} \in \mathbb{R}^D$ denotes a hidden representation of the LSTM encoder with the t-th acoustic input feature x_t. $h_u^{\text{pred}} \in \mathbb{R}^D$ is a prediction network representation at u-th decoding step. $\text{Joint}(\cdot)$ is the joint network that first concatenates o_t^{enc} and o_t^{pred} together and feeds them to a feed forward network to produce final scores $o_{t,u} \in \mathbb{R}^V$, where V is the vocabulary size.

The loss function in RNN-T framework can be defined as a conditional distribution over all the possible alignments $p(y|t,u)$. It is natural to employ a forward-backward algorithm[6] to compute the total probability $P(y|x)$ of the output token sequence y, conditioned on the input acoustic sequence x.

$$L = -\log P(y|x) = -\sum_{t\in[0,T],u\in[0,U]} \log P(y|t,u) \tag{5}$$

With the advantage of Transformer[11] and Conformer[13], there is a few recent works to upgrade the LSTM-LSTM architecture to Transformer-Transformer transducer[20], Conformer-LSTM transducer and so on. Among them, the Conformer-LSTM transducer is a typical one which. In a nutshell, the major component in a Conformer encoder is a stack of Conformer blocks, each of which contains a series of multi-head self-attention, depth-wise convolution and feed-forward layers.

2 Methodology

Our model architecture (as shown in Fig. 1) can be divided into four major components: (1) an encoder network with two feature extractors, (2) an LSTM prediction network, (3) a pretrained linguistic network, and (4) a joint network. The encoder network mainly follows Conformer[13] architecture. Two convolution neural networks are employed as feature extractors to subsample filter-banks and raw waveform, respectively. We denote the randomly initialized extractor as Conv_r and the pretrained one as Conv_p. The prediction network is a unidirectional LSTM network, while the linguistic network is a GPT-2[37] model that pretrained in advance and fine-tune on downstream datasets.

2.1 Problem formalization

Formally, we are given speech frames $x = [x_1, x_2, \ldots, x_T]$ and token sequence $y = [y_1, y_2, \ldots, y_U]$. We denote the number of speech frames by T, with each $x_t \in \mathbb{R}^d$. Our target is to model the alignments between the speech frames x and token sequence y to maximize $P(y|x)$.

2.2 Pretrained acoustic extractor

Speech recognition systems are almost based on the structural speech spectrum obtained by time-frequency analysis. In order to improve the performance of speech recognition systems, it is necessary to identify the temporal and spatial characteristics of spectrum or wave information. A CNN provides translation invariant convolution in time and space to learn more stable representations over windows of acoustic frames, and the invariance of convolution can be used to overcome the diversity of speech signals. From this point of view, CNNs have an advantage in extracting acoustic features, and thus are preferred to speech recognition tasks, notably by Hau and Chen[56], and Abdel-Hamid et al.[57]

In this paper, we introduce two convolution networks, Conv_r and Conv_p, to construct a new pretraining-enhanced acoustic feature. Due to different dimensions of filter-bank features and raw waveform, we apply different kernel sizes in Conv_r and Conv_p. Conv_r is a 2D-CNN for which the filter-bank has two dimensions, while Conv_p uses 1D convolution kernels for raw signals. In details, Conv_r is a two-layer convolution neural network with 3×3 kernel size and 1×1 stride, which takes the filter-banks as input and outputs the high-level representations x_t^f. Conv_r is randomly initialized and updated during training procedure. Conv_p contains the same model structure and parameters as the Wav2vec 2.0 feature extractor[10]. To be specific, the Wav2vec 2.0 feature extractor is just a multi-layer convolutional feature encoder instead of the entire framework with the Transformer blocks. It is also noted that we freeze the parameters of Conv_p in downstream tasks, which is similar to previous works[10]. Correspondingly, Conv_p is composed of seven layers of 1D convolution, whose kernel sizes are gradually changed from 10 to 2 as the number of layers increases. Last but not least, it is necessary to make a slight change to the offset in filter-banks in order to match the number of output frames of Conv_r and Conv_p. We show the detailed architectures of Conv_r and Conv_p in Table 1.

For example, given a raw audio data $\{x_0, x_1, \ldots, x_T\}$, we firstly convert raw waveform to filter-banks and use Conv_r to extract the acoustic features x_t^f. Secondly, we directly apply pretrained Conv_p to extract x_t^p from raw waveform. Last but not least, we sum up x_t^f and x_t^p and feed them into the encoder network as follows:

$$x_t^f = \text{Conv}_r(\text{Filter Bank}(x_t)) \tag{6}$$

$$x_t^p = \text{Conv}_p(x_t) \tag{7}$$

$$h_t^{\text{enc}} = \text{Encoder}(x_t^f + x_t^p) \tag{8}$$

where $t \in [0,T]$, $h_t^{\text{enc}} \in \mathbb{R}^D$ is the t-th output of the encoder network. $x_t^f \in \mathbb{R}^d$ and $x_t^p \in \mathbb{R}^d$ are filter-banks and raw waveform features, respectively. $\text{Encoder}(\cdot)$ represents the encoder network. GELU[58] activation function is applied to all of above models.

2.3 Pretrained linguistic network

One of the key problems in automatic speech recognition systems is to identify the next proper token given a sequence of previous decoded results. The usual practice is using external language models to. In RNN-T, prior research generally confirms that the prediction network is analogous to an internal LM, which can model linguistic information in end-to-end fashion. However, Ghodsi et al.[32] proved that it is ineffective to initialize the prediction network with a pretrained LM. It is also shown that the stateless (i.e. non-recurrent) prediction network depends only on

Table 1 Detail structures of Conv_p and Conv_r.

Convolution network	Type	Layer	Kernel size	Stride	Initialization
Conv_r	2D	2	$[3\times3, 3\times3]$	$[1\times1, 1\times1]$	-
Conv_p	1D	7	[10, 3, 3, 3, 3, 2, 2]	[5, 2, 2, 2, 2, 2, 2]	Wav2vec 2.0

the last output symbol y_{u-1}, and works virtually as well as the original RNN-T architecture.

Inspired by Deep Fusion[59] that integrates the external LM into neural machine translation (NMT), we introduce an extra GPT-2 feature into RNN-T architecture to provide the semantic information. We consider an LSTM prediction network and a GPT-2 linguistic network in our pretraining-enhanced RNN-T architecture. For example, at the u-th decoding step, we need to generate latent representations h_u^{pred} and h_u^{lang} by the prediction network and the linguistic network respectively. The prediction network predicts y_u according to left-to-right predicted words $y_0, y_1, \ldots, y_{u-1}$.

$$h_u^{\text{pred}} = \text{PredNet}(y_0, y_1, \ldots, y_{u-1}) \tag{9}$$

where $y_0, y_1, \ldots, y_{u-1}$ is the transcript tokens, while h_u^{pred} is the output of the prediction network. $\text{PredNet}(\cdot)$ indicates a unidirectional LSTM prediction network in our case.

Likewise, we choose an autoregressive model as the PLN to produce the next token representation according to the previous ones. GPT-2[37] is such an autoregressive model which performs well for text generation and compatible to the manner of the prediction network. Hence, we finetune a pretrained GPT-2 model on domain-specific transcripts and use it to produce the u-th linguistic feature conditioned on previous tokens $y_0, y_1, \ldots, y_{u-1}$. We formulate the linguistic feature h_u^{lang} as below:

$$h_u^{\text{lang}} = \text{PLN}(y_0, y_1, \ldots, y_{u-1}) \tag{10}$$

As illustrated in Fig. 1, the joint network merge latent representations of the encoder h_t^{enc}, the prediction network h_u^{pred}, and the linguistic network h_u^{lang} together at the (t,u)-th step, and maps the combination to a distribution $p(y|t,u)$.

$$o_{t,u} = \text{JointNet}(h_t^{\text{enc}}, \lambda h_u^{\text{pred}}, (1-\lambda) h_u^{\text{lang}}) \tag{11}$$

$$o_{t,u} = \text{FFN}_3(\text{FFN}_0(z_t^{\text{enc}}) + (1-\lambda)\text{FFN}_1(z_u^{\text{pred}}) + \lambda \text{FFN}_2(z_u^{\text{lang}})) \tag{12}$$

where $t \in [0, T]$, $u \in [0, U]$, and λ is a hyperparameter to adjust the importance of the pretrained linguistic feature. $h_t^{\text{enc}} \in \mathbb{R}^{D_0}$, $h_u^{\text{pred}} \in \mathbb{R}^{D_1}$, and $h_u^{\text{lang}} \in \mathbb{R}^{D_2}$ are the latent representations for the encoder network, the prediction network, and the linguistic network at (t,u)-th step, respectinel. $\text{Joint}(\cdot)$ is the joint network, which can be factorized as four feed forward networks.

The entire framework is trained jointly to optimize the RNN-T loss[6], which marginalizes over all alignments of target labels with blanks, and is computed using dynamic programming.

$$p(y|t,u) = \text{SoftMax}(o_{t,u}) \tag{13}$$

3 Experiment and Result

In this section, we conduct experiments upon two public corpora: (1) LibriSpeech[1] consists of 970 hours of labeled English recording derived from the LibriVox project and an additional 800 million word token text-only corpus for building language model. (2) AISHELL-1[2] includes 178 hours of Chinese Mandarin speech from different accents in China. Both above two datasets contain speech records in wav format with a sampling rate of 16 kHz, and have been divided into training sets, validation sets and test sets in advance. We conduct experiments according to the existing divisions.

We adopt two common metrics of the performance of an automatic speech recognition system, WER and CER, in LibriSpeech and AISHELL-1. WER measures the Levenshtein distance between the ground truth and our recognition hypothesis according to the number of words in ground truth. Correspondingly, CER is similar to WER, but operates on characters instead of words.

3.1 Experimental settings

In this part, we first describe the experimental settings of comparable benchmarks in details. We consider the Conformer-LSTM transducer[13] as our baseline, which takes only filter-bank features as input and uses Conv_r to extract subsampling features. Specifically, we extracte 80-channel filter-banks computed from a 25 ms window with a stride of 5 ms. Speed perturbation is applied to each audio with 0.9× and 1.1× speed. SpecAugment[60] method is applied to avoid overfitting. The mask parameter in SpecAugment is 27, and the maximum size of the time mask is set to 20.

For the baseline, the encoder network consists of 12 Conformer blocks while all Conformer blocks' kernel size is 15 and the number of attention head for each layer is 4, while the dimension of encoder is set to 256 and feed forward expansion factor is 4. The prediction network is a single unidirectional LSTM layer with only 1024 cells. And the joint network has three feed forward layers: two of them first project different hidden representations to the same 512-dimensional hidden states, then merge all representations, and finally the third feed forward layer projects it to a final distribution of vocabulary plus a "BLANK" token. Different from the baseline, our pretraining-enhanced framework further adopts a 768-dimensional pretrained linguistic network

Table 2 Experiment results on LibriSpeech and AISHELL-1 task. The metric used in LibriSpeech taskes is WER, while we evaluate CER in AISHELL-1 (Dev: Development; Avg: Avergage). ❊ means no additional training parameters compared to the baseline.

Transducer	Parameter (10^6)	Error rate (%)					
		LibriSpeech				AISHELL-1	
		Dev	Test-clean	Test-other	Avg (test)	Dev	Test
LSTM-LSTM [6]	130	-	3.2	7.8	5.5	10.1	11.8
Transformer-Transformer [20]	139	-	2.4	5.6	4.0	-	-
Baseline	57	7.9	2.9	7.6	5.3	4.9	4.5
+PAE	❊	**7.3**	**2.8**	**7.2**	5.0 (↓ 7.4%)	4.4	4.3 (↓ 4.4%)
+PAE +PLN	❊	7.4	**2.8**	7.3	5.1 (↓ 5.6%)	**4.3**	**4.2** (↓ 6.6%)

1 https://www.openslr.org/12
2 https://openslr.org/33/

and a feed forward layer with 512 hidden units according to Eq. (12).

We use the same hyperparameters in all experiments. During training, we suggest 128 batch size and fine-tuning for 60 epochs. We apply $\lambda = 0.3$ in Eq. (12) and dropout $P_{\text{drop}} = 0.1$ in each unit of the Conformer, i.e., the output activation and the attention weight of each module. An l_2 regularization with 10^{-6} weight is added to all the trainable weights in the network. AdamW optimizer[61] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$, and a Transformer learning rate schedule[11] are used in training. We consider 10% warm-up steps and set peak learning rate to $0.005/\sqrt{d}$ where d is the model dimension in Conformer encoder.

We choose the deep learning platform PyTorch[62] and open source toolkit Huggingface[63] [3] to establish our ASR system. The pretrained model shared by Huggingface can be utilized as our proposed PAE and PLF. In the English task, we initialize our pretrained acoustic extractor Conv_p with the large Wav2vec 2.0 model[4] fine-tuned on 960 hours of Libri-Light and Librispeech on 16 kHz sampled speech audio. And we use a Chinese version of Wav2vec 2.0 from ydshieh[5] in the AISHELL-1. The GPT-2 linguistic network follows an English[6] and a Chinese[7] version from Huggingface repository. It is also noted that we use the same tokenizer and vocabulary in the prediction network and the linguistic network.

3.2 Experiment on pretrained features

In this part, we perform a study on the final recipe to tease out the importance of the pretrained acoustic extractor and pretrained linguistic network. All the experiments do not consider the external language model for the purpose of exploring the effectiveness of pretrained linguistic network in RNN-T.

Firstly, we incorporate only pretrained acoustic extractor in the baseline model. Note that our model with PAE and PLN has only 57 million trainable parameters. With a large gap of model complexity, the WER drops by average 7.4% on Librispeech test sets after employing pretrained acoustic extractor. And we meet the same observation that the character error rate also drops about 4.4% relatively on AISHELL-1 test set, indicating that using the features from a large pretrained acoustic model could be a suitable way to enhance model performance.

Then, we employ pretrained acoustic extractor and pretrained linguistic network together, which further achieves about 6.6% relative CER drops on AISHELL-1 task compared to the baseline. In order to visualize the effect of pretrained linguistic network, we compare the results between PLN and non-PLN approaches. Non-PLN approach just removes PLN from our framework, and an easy way to eliminate the influence of PLN is setting $\lambda = 0$ in Eq. (12). For examples, in Fig. 2, we observe that the decoding results are sometimes incoherent and make no sense according to the baseline results, although their pronunciations are similar. It is obvious that we obtain a more reasonable and impressive decoding results after fusing PLN in both English and Chinese tasks.

3.3 Experiment on low-resource data

Since end-to-end models have achieved important improvements on the task of ASR, however, labeled data can hardly satisfy the demand of end-to-end models for low resource ASR tasks. One of the most popular ways to tackle the low-resource problem is the self-supervised pretraining method. Recently, self-supervised acoustic pretraining has already shown its amazing ASR performance, while the transcription is still inadequate for language modeling in end-to-end models.

Dataset	Method	Result
Librispeech	B/L	He **saw his violent** (×) deeds not with the hand but with the suck.
	Ours	He **sows** (✓) violent deeds not with the hand but with the sack.
AISHELL-1	B/L	楼市调控的行政手段意见不一家 (×) 。
	Ours	楼市调控的行政手段宜减不宜加 (✓) 。

Fig. 2 Examples of decoding results. B/L refers to baseline.

In this part, in order to verify our proposed pretrained acoustic extractor andpretrained linguistic network can improve the performance in low-resource data, we investigate a further discussion upon low-resource scenario. We carry out our experiments on AISHELL-1 by randomly sampling 20 hours and 50 hours subsets. As the results shown in Fig. 3, the pretrained acoustic extractor can get better results than the well-trained baseline model both on different sizes of AISHELL-1, achieving CER of 17.22% with 10 hours data, 11.08% with 50 hours, and 4.26% with 178 hours. When incorporating a pretrained linguistic network, CERs are further dropped to 16.48%, 10.75%, and 4.19%, respectively. Overall, our approaches outperform the baseline by relative 9.1% and 7.8% improvement on 10 hours and 50 hours, respectively.

3.4 Experiment on cold-start model

Recent developments in pretrained representations have been accompanied by large and expensive models that leverage vast amounts of general-domain text through self-supervised pretraining. Nonetheless, few researches focus on the procedure of "pretraining and fine-tuning" paradigm. In this part, we analyze how much our approaches benefits from "pretraining and fine-tuning" in Table 3. To investigate the importance of pretraining procedure, we first adopt the pretrained model parameters to initialize PAE and PLN, and then train other parts of our framework jointly except PAE and PLN. It is realized that GPT-2 is also fine-tuned on AISHELL-1 transcripts before initialing PLN.

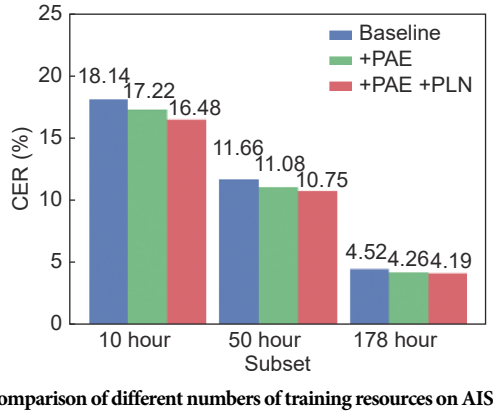

Fig. 3 Comparison of different numbers of training resources on AISHELL-1.

3 https://github.com/huggingface/transformers
4 https://huggingface.co/facebook/wav2vec2-large-960h-lv60
5 https://huggingface.co/ydshieh/wav2vec2-large-xlsr-53-chinese-zh-cn-gpt
6 https://huggingface.co/gpt2
7 https://huggingface.co/uer/gpt2-chinese-cluecorpussmall

We call this approach with pretrained parameters as "warm-start" in Table 3. At the same time, we conduct comparative "cold-start" experiments by initializing PAE and PLN randomly instead of using a pretrained one. In this setting, PAE without the pretrained parameters from Wav2vec 2.0 only achieves 4.57% CER score in the AISHELL-1 after 100 epochs, which can not even catch up with the baseline.

We illustrate the convergence curves of both methods in Fig. 4, where the training loss of different warm-start and cold-start models are reported. Overall, the training loss first decrease in a very slow way with a small initial learning rate, then start to learn key knowledge quickly via the warm-up learning rate, and converges at the end. It can be observed that when employing only the PAE into our framework, both warm-start and cold-start techniques are able to converge stably, so that the training loss is greatly reduced after 10 000 training steps. What is more, we find that the warm-start loss is slightly lower than the cold-start one, which can explain that transferring pretrained acoustic extractor to downstream tasks contribute to model generalization. However, the cold-start model with PAE and PLN remains a significant gap to the warm-start one, even after five times of training steps. It is difficult to optimize cold-start GPT-2 architecture in our framework, as the training loss seems hard to converge. To clarify this problem in cold-start scenario, we visualize the output tokens of the linguistic network during the training procedure. Surprisingly, we observe cold-start GPT-2 model produce the same token at the beginning and still get lost in the final step. On the contrary, warm-start GPT-2 provides closed next-token prediction and draws a jagged loss compared to non-PLN approach, which may server as a regularizer in downstream tasks.

4 Conclusion

In this paper, we suggest two kinds of pretrained features from PAE and PLN to enhance the Conformer-LSTM transducer. PAE handles diverse speech signals by pretraining on a large amount of unlabeled audio, while PLN aids in semantic decoding with the prediction network. Our methods achieve lower CER on two public corpora compared to the Conformer-LSTM transducer without external language models. Additionally, further experiments demonstrate effectiveness in low-resource scenarios and highlight the essential role of pretraining in enhancing RNN-T models.

Table 3 CER comparison of cold-start and warm-start models on AISHELL-1. (%)

Model	Cold-start	Warm-start
Baseline	4.52	-
+PAE	4.57	4.26
+PAE +PLN	97.89	4.19

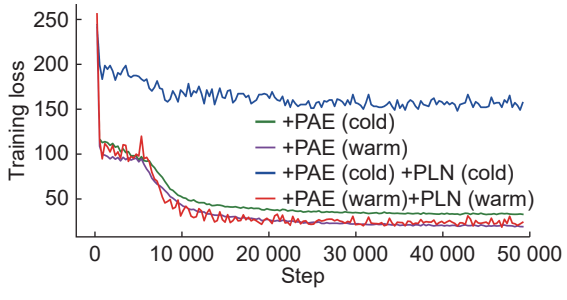

Fig. 4 Trends of the training loss on AISHELL-1.

Acknowledgment

We are grateful to the anonymous reviewers for their constructive comments on this paper. The research was supported in part by the Guangdong Basic and Applied Basic Research Foundation (No. GDST23EG32).

Article History

Received: 22 February 2024; Revised: 28 June 2024; Accepted: 23 July 2024

References

[1] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al., Deep Speech: Scaling up end-to-end speech recognition, arXiv preprint arXiv: 1412.5567, 2014.

[2] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, et al., Deep Speech 2: End-to-end speech recognition in English and Mandarin, in *Proc. 33rd Int. Conf. Machine Learning*, New York, NY, USA, 2016, pp. 173–182.

[3] Y. Song, D. Jiang, X. Wu, Q. Xu, R. C. W. Wong, and Q. Yang, Topic-aware dialogue speech recognition with transfer learning, in *Proc. 20th Annu. Conf. Int. Speech Communication Association* (*INTERSPEECH 2019*), Graz, Austria, 2019, pp. 829–833.

[4] X. Wu, R. Lian, D. Jiang, Y. Song, W. Zhao, Q. Xu, and Q. Yang, A phonetic-semantic pre-training model for robust speech recognition, *CAAI Artif. Intell. Res.*, vol. 1, no. 1, pp. 1–7, 2022.

[5] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, et al., Streaming end-to-end speech recognition for mobile devices, arXiv preprint arXiv: 1811.06621, 2018.

[6] A. Graves, Sequence transduction with recurrent neural networks, arXiv preprint arXiv: 1211.3711, 2012.

[7] K. Rao, H. Sak, and R. Prabhavalkar, Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer, in *Proc. 2017 IEEE Automatic Speech Recognition and Understanding Workshop* (*ASRU*), Okinawa, Japan, pp. 193–199.

[8] S. Schneider, A. Baevski, R. Collobert, and M. Auli, wav2vec: unsupervised pre-training for speech recognition, arXiv preprint arXiv: 1904.05862, 2019.

[9] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, E. Hulburd, D. Liu, M. Wang, A. G. Catlin, M. Lei, J. Zhang, et al., BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805, 2018.

[10] A. Baevski, H. Zhou, A. Mohamed, M. Auli, N. Vaessen, and D. A. van Leeuwen, wav2vec 2.0: A framework for self-supervised learning of speech representations, in *Proc. 34th Conf. Neural Information Processing Systems* (*NeurIPS 2020*), Vancouver, Canada, 2020, pp. 12449–12460.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, LlionJones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st Conf. Neural Information Processing Systems* (*NIPS 2017*), Long Beach, CA, USA, 2017, pp. 5998–6008, 2017.

[12] Y. Zhang, J. Qin, D. S. Park, W. Han, C. C. Chiu, R. Pang, Q. V. Le, and Y. Wu, Pushing the limits of semi-supervised learning for automatic speech recognition, arXiv preprint arXiv: 2010.10504, 2020.

[13] A. Gulati, J. Qin, C. C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al., Conformer: Convolution-augmented Transformer for speech recognition, in *Proc. 21st Annu. Conf. Int. Speech Communication Association* (*INTERSPEECH 2020*), Shanghai, China, 2020, pp. 5036–5040.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks, in *Proc. 23rd Int. Conf. Machine learning*, Pittsburgh, PA, USA, 2006, pp. 369–376.

[15] H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, Exploring pre-training with alignments for RNN transducer based end-to-end speech recognition, in *Proc. 2020 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), Barcelona, Spain, 2020, pp. 7079–7083.

[16] R. Kumar, S. Srivastava, and J. R. P. Gupta, Comparative study of neural networks for control of nonlinear dynamical systems with Lyapunov stability-based adaptive learning rates, *Arab. J. Sci. Eng.*, vol. 43, no. 6, pp. 2971–2993, 2018.

[17] R. Kumar, Double internal loop higher-order recurrent neural network-based adaptive control of the nonlinear dynamical system, *Soft Comput.*, vol. 27, no. 22, pp. 17313–17331, 2023.

[18] R. Kumar, S. Srivastava, and A. Mohindru, Lyapunov stability-Dynamic Back Propagation-based comparative study of different types of functional link neural networks for the identification of nonlinear systems, *Soft Comput.*, vol. 24, no. 7, pp. 5463–5482, 2020.

[19] R. Kumar, Recurrent context layered radial basis function neural network for the identification of nonlinear dynamical systems, *Neurocomputing*, vol. 580, pp. 127524, 2024.

[20] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss, in *Proc. 2020 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), Barcelona, Spain, 2020, pp. 7829–7833.

[21] Y. Song, D. Jiang, X. Zhao, Q. Xu, R. C. W. Wong, L. Fan, and Q. Yang, L2RS: A learning-to-rescore mechanism for automatic speech recognition, arXiv preprint arXiv: 1910.11496, 2019.

[22] B. Zhang, D. Wu, Z. Yao, X. Wang, F. Yu, C. Yang, L. Guo, Y. Hu, L. Xie, and X. Lei, Unified streaming and non-streaming two-pass end-to-end model for speech recognition, arXiv preprint arXiv: 2012.05481, 2020.

[23] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit, in *Proc. 22nd Annu. Conf. Int. Speech Communication Association* (*INTERSPEECH 2021*), Brno, Czech Republic, 2021, pp. 4054–4058.

[24] D. Jiang, C. Tan, J. Peng, C. Chen, X. Wu, W. Zhao, Y. Song, Y. Tong, C. Liu, Q. Xu et al., A GDPR-compliant ecosystem for speech recognition with transfer, federated, and evolutionary learning, *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 3, pp. 1–19, 2021.

[25] Y. Song, X. Huang, X. Zhao, D. Jiang, and R. C. W. Wong, Multimodal N-best list rescoring with weakly supervised pre-training in hybrid speech recognition, in *Proc. 2021 IEEE Int. Conf. Data Mining* (*ICDM*), Auckland, New Zealand, 2021, pp. 1336–1341.

[26] C. Tan, D. Jiang, J. Peng, X. Wu, Q. Xu, and Q. Yang, A *de novo* divide-and-merge paradigm for acoustic model optimization in automatic speech recognition, in *Proc. 29th Int. Joint Conf. Artificial Intelligence*, Yokohama, Japan, 2020, pp. 3709–3715.

[27] C. Tan, D. Jiang, H. Mo, J. Peng, Y. Tong, W. Zhao, C. Chen, R. Lian, Y. Song, and Q. Xu, Federated acoustic model optimization for automatic speech recognition, in *Proc. 25th Int. Conf. Database Systems for Advanced Applications*, Jeju, Republic of Korea, 2020, pp. 771–774.

[28] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, Learning the speech front-end with raw waveform CLDNNs, in *Proc. 16th Annu. Conf. Int. Speech Communication Association*, Dresden, Germany, 2015, pp. 1–5.

[29] S. W. Fu, Y. Tsao, X. Lu, and H. Kawai, Raw waveform-based speech enhancement by fully convolutional networks, in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conf.* (*APSIPA ASC*), Kuala Lumpur, Malaysia, 2017, pp. 6–12.

[30] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, End-to-end speech recognition from the raw waveform, arXiv preprint arXiv: 1806.07098, 2018.

[31] M. W. Y. Lam, J. Wang, C. Weng, D. Su, and D. Yu, Raw waveform encoder with multi-scale globally attentive locally recurrent networks for end-to-end speech recognition, arXiv preprint arXiv: 2106.04275, 2021.

[32] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, Rnn-transducer with stateless prediction network, in *Proc. 2020 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*). Barcelona, Spain, 2020, pp. 7049–7053.

[33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, Librispeech: An ASR corpus based on public domain audio books, in *Proc. 2015 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), South Brisbane, Australia, 2015, pp. 5206–5210.

[34] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline, in *Proc. 2017 20th Conf. Oriental Chapter Int. Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment* (*O-COCOSDA*), Seoul, Republic of Korea, 2017, pp. 1–5.

[35] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, NV, USA, 2016, pp. 770–778.

[36] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv: 1802.05365, 2018.

[37] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, Improving language understanding by generative pre-training, https://openai.com/index/language-unsupervised, 2018.

[38] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, XLNet: Generalized autoregressive pretraining for language understanding, in *Proc. 33rd Conf. Neural Information Processing Systems* (*NeurIPS 2019*), Vancouver, Canada, 2019, pp. 5753–5763.

[39] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, et al., RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv: 1907.11692, 2019.

[40] D. Jiang, C. Zhang, and Y. Song, Probabilistic Topic Models: Foundation and Application, Singapore: Springer Nature, 2023.

[41] Y. Li, D. Jiang, R. Lian, X. Wu, C. Tan, Y. Xu, and Z. Su, Heterogeneous latent topic discovery for semantic text mining, *IEEE Trans. Knowl. Data Eng.* vol. 35, no. 1, pp. 533–544, 2023.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[43] T. Lin, M. Maire, S. Belongie, J. Hays, PietroPerona, D. Ramanan, P. Dollár, and C. L. Zitnick, Microsoft COCO: Common objects in context, in *Proc. 13th European Conf. Computer Vision* (*ECCV*), Zurich, Switzerland, 2014, pp. 740–755.

[44] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 652–663, 2017.

[45] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, 3D human pose estimation in video with temporal convolutions and semi-supervised training, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, CA, USA, 2019, pp. 7753–7762.

[46] M. Ravanelli and Y. Bengio, Learning speaker representations with mutual information, arXiv preprint arXiv: 1812.00271, 2018.

[47] C. Chen, D. Jiang, J. Peng, R. Lian, Y. Li, C. Zhang, L. Chen, and L. Fan, Scalable identity-oriented speech retrieval, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 3261–3265, 2023.

[48] V. Mitra, V. Kowtha, H. Y. S. Chien, E. Azemi, and C. Avendano, Pre-trained model representations and their robustness against noise

for speech emotion analysis, in *Proc. 2023 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), Rhodes Island, Greece, 2023, pp. 1–5.

[49] B. Han, Z. Lv, A. Jiang, W. Huang, Z. Chen, Y. Deng, J. Ding, C. Lu, W. Q. Zhang, P. Fan, et al., Exploring large scale pre-trained models for robust machine anomalous sound detection, in Proc. *2024 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), Seoul, Republic of Korea, 2024, pp. 1326–1330.

[50] G. Synnaeve and E. Dupoux, A temporal coherence loss function for learning unsupervised acoustic embeddings, *Procedia Comput. Sci.*, vol. 81, pp. 95–100, 2016.

[51] A. van den Oord, Y. Li, O. Vinyals, P. de Haan, and S. Löwe, Representation learning with contrastive predictive coding, arXiv preprint arXiv: 1807.03748, 2018.

[52] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, Transfer learning for speech recognition on a budget, arXiv preprint arXiv: 1706.00290, 2017.

[53] P. H. Le-Khac, G. Healy, and A. F. Smeaton, Contrastive representation learning: A framework and review, *IEEE Access*, vol. 8, pp. 193907–193934, 2020.

[54] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, End-to-end continuous speech recognition using attention-based recurrent NN: First results, arXiv preprint arXiv: 1412.1602, 2014.

[55] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[56] D. Hau and K. Chen, Exploring hierarchical speech representations with a deep convolutional neural network, in *Proc. UK Annu. Workshop on Computational Intelligence* (*UKCI'11*), Manchester, UK, 2021.

[57] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, Convolutional neural networks for speech recognition, *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, 2014.

[58] D. Hendrycks and K. Gimpel, Gaussian error linear units (GELUs), arXiv preprint arXiv: 1606.08415, 2016.

[59] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, On using monolingual corpora in neural machine translation, arXiv preprint arXiv: 1503.03535, 2015.

[60] 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, arXiv preprint arXiv: 1904.08779, 2019.

[61] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv: 1711.05101, 2017.

[62] A. Paszke, S. Gross, F. Massa, A. Lerer, JamesBradbury, G. Chanan, T. Killeen, Z. Lin, NataliaGimelshein, L. Antiga, et al., PyTorch: An imperative style, high performance deep learning library, in *Proc. 33rd Conf. Neural Information Processing Systems* (*NeurIPS 2019*), Vancouver, Canada, 2019, pp. 8026–8037.

[63] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations*, virtual, 2020, pp. 38–45.