IMPROVING PSEUDO-LABEL TRAINING FOR END-TO-END SPEECH RECOGNITION USING GRADIENT MASK

Shaoshi Ling, Chen Shen, Meng Cai, Zejun Ma

Bytedance AI Lab
lingshaoshi@bytedance.com

ABSTRACT

In the recent trend of semi-supervised speech recognition, both self-supervised representation learning and pseudo-labeling have shown promising results. In this paper, we propose a novel approach to combine their ideas for end-to-end speech recognition model. Without any extra loss function, we utilize the *Gradient Mask* to optimize the model when training on pseudo-label. This method forces the speech recognition model to predict from the masked input to learn strong acoustic representation and make training robust to label noise. In our semi-supervised experiments, the method can improve the model's performance when training on pseudo-label and our method achieved competitive results comparing with other semi-supervised approaches on the Librispeech 100 hours experiments.

Index Terms— speech recognition, semi-supervised learning, pseudo-labeling, end-to-end model

1. INTRODUCTION

Pseudo-labeling [1, 2, 3, 4, 5, 6, 7, 8, 9] is one of the most popular semi-supervised learning approaches and recently demonstrated its efficacy in automatic speech recognition. In this approach, a smaller labeled set is used to train an initial seed model, which is applied to a larger amount of unlabeled data to generate hypotheses. The unlabeled data with the most reliable hypotheses are added to the training data for re-training. This process can be repeated iteratively to improve the quality of pseudo labels [7]. However, pseudo-label training is sensitive to the quality of the hypotheses. Errors or noise in labels can cause training unstable and resulted in sub-optimal states, especially for end-to-end speech recognition models [6]. Thus, pseudo-label training usually requires careful calibration by confidence measures [6, 8]. But confidence-based data filtering will not always work perfectly since most pseudo-label sequences would contain errors.

Starting from BERT [10], masked prediction has becoming a new principle to solve problems in self-supervised settings in NLP. The core idea of masked prediction is to force the model to learn good high-level representations of unmasked inputs to infer the targets of masked ones correctly. In speech, the approaches sharing the same spirit have been proposed: masked prediction of audio acoustic features [11, 12], masked prediction of quantized acoustic features [13] and masked prediction of unsupervised clusters [14]. Experiments in [14] also showed that computing loss only from the masked regions achieves better performance than all regions.

We draw inspiration from masked prediction and integrate its idea into pseudo-label training. We propose the *Gradient Mask* to improve pseudo-label training in end-to-end speech recognition. In our approach, we first train a seed model to generate pseudo labels and then use the Gradient Mask to train a student model on the pseudo labels. The model only allows gradients corresponding to masked input back-propagate through the model encoder by masking the gradients corresponding to unmask input. The model is trained by jointly minimizing the loss on labeled and pseudo-label data while the Gradient Mask is turned off on labeled data.

Our training method can force the model to learn strong acoustic representation in order to infer from masked input. Moreover, it can also improve pseudo-label training by making the model less affected by label noise. The intuition is that only gradients of the masked part are used when updating the model's parameters, so it can avoid the sudden dramatic change in gradients caused by errors and also alleviate the over-fit to corrupted labels. Our approach is simple and efficient since it doesn't require any extra parameters, extra loss or data filtering steps. We run our experiments using the Transducer model [15]. The experiment showed that our method is robust to label noise and can achieve competitive results comparing with other self/semi-supervised approaches in the Librispeech 100 hours experiments.

2. RELATED WORK

2.1. Combating noisy labels

DNNs are known to be susceptible to noisy labels [16, 17] and the errors in labels could be extremely harmful to models. Beyond conventional data filtering/cleaning techniques, deep learning techniques have recently gained vast interest.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

There are several works to investigate supervised learning under noisy labels [17] in computer vision. However, these models cannot be directly applied to ASR and fewer studies have proposed to combat noise labels for ASR. In [18], the phonetic sequence was inferred from several noisy transcriptions made by non-native transcribers using a misperception model, and then used to train a conventional hybrid ASR model. [19] propose a novel loss function that jointly learns the ASR model and a transcription graph that can search for better transcriptions of the training data.

2.2. Joint training with self-supervised and ASR tasks

The idea of self-supervised learning [11, 12, 13, 14, 20, 21, 22, 23] is to learn speech representations that are useful for ASR. By first pre-training on a large amount of unlabeled data using a proxy task, the model can be fine-tuned on labeled data and achieved impressive results. This process is a two-stage process as it requires running separate pre-training and fine-tuning. Joint training with speech recognition and self-supervised representation [24, 25, 26, 27] is the line of work to simplify this process and is the closest to our method. Those methods typically have two training objectives: one is for the ASR task on the labeled data while the other is to train self-supervised representation (e.g. masked feature prediction [24]), on the unlabeled data. Our method is much simpler and uses only one loss on both label and unlabeled data (pseudo-label data).

3. METHOD

In speech recognition, an E2E model predicts the conditional distribution $P(Y|X)$ of token sequences $Y = [y_1, ..., y_U]$ given a speech-feature sequence $X = [x_1, ..., x_T]$ as the input, where $y \in V$ and x_t is acoustic feature vectors at time t. V is the set of all possible output tokens. We will explain and show our method in transducer model [15], but it can be perfectly adapted to other end-to-end ASR models (e.g. CTC [28], seq2seq [29]) as well.

3.1. Transducer model

Transducer model [15] consists of encoder, prediction network and joint network. The encoder f_{enc} encode the inputs X to higher-level representation $h_{enc} = [h_1^{enc}, ..., h_T^{enc}]$.

The prediction network takes embedding vectors of previous non-blank labels $[y_1, ..., y_{u-1}]$ as input to produce its output h_u^{pred} at step u. Then the logits over vocabulary at frame t and step u can be computed by the joint network:

$$h_{t,u}^{joint} = f_{joint}(h_t^{enc}, h_u^{pred}) \tag{1}$$

The probability distribution over vocabulary at frame t and step u is calculated using a soft-max layer. With forward-backward algorithm, the sum probability $P(Y|X)$ of all alignment paths is adopted as the objective function.

3.2. Gradient mask

For sequence $X = [x_1, ..., x_T]$ which has pseudo labels $Y' = [y'_1, ..., y'_{u-1}]$. The objective is to enable model to predict the labels from the masked features. In another word, f_{enc} is trained to be a strong acoustic representation model which can benefit the ASR tasks.

Before feeding features to the encoder, we randomly generated a sequence $mask = [m_1, ..., m_T]$ representing the mask positions for the input sequence X. Specifically, m_t is 1 if features are masked at time t, otherwise m_t is 0. The features, for example x_t, are masked by replacing it with a learnt mask embedding emb. Then the encoder f_{enc} encode this mask sequence as :

$$h^{enc} = f_{enc}((\sim mask) * X + mask * emb) \tag{2}$$

Our mask strategy is the same as [23], where we randomly sample without replacement a certain proportion p of all time steps to be starting indices and then mask the subsequent m consecutive time steps from every sampled index with overlap spans.

When the gradient is back-propagated to the encoder, we masked the gradients corresponding to the non-masked inputs using the $mask$ sequence:

$$grad_{h^{enc}} = (\sim mask) * grad_{h^{enc}} \tag{3}$$

The prediction network takes the pseudo labels sequence as the input. The joint network then produces output as in (1). When we do back-propagation, we also block the gradient flow into the predictor network. Our intuition is that the predictor network is more likely to suffer from over-fitting and we stop the gradients to protect it from corrupted labels. This process can be expressed in the following functions:

$$h_{t,u}^{joint} = f_{joint}(h_t^{enc}, sg(h_u^{pred})) \tag{4}$$

where $sg(x) \equiv x$, $\frac{d}{dx}sg(x) \equiv 0$ is the stop gradient operator. The objective function is still the same transducer loss where we trying to minimize $P(Y'|X)$ of all alignment paths.

3.3. Training procedure

The whole training process is similar to the standard pseudo-labeling approach. Let $L = \{x_i, y_i\}$ be a labeled dataset and $U = \{x_j\}$ be a large unlabeled dataset. We first train a seed acoustic model M on the labeled dataset L. We use this seed acoustic model M to generate pseudo-labeled on dataset $U = \{x_j, y'_j\}$ and we then combine it with all the label data in L to form new dataset U' = $U \cup L$.

The next step is to train a student model using both the datasets L and U'. The model is trained by alternately minimizing the losses on L and U'. When updating the model

Method	Model Size	Criterion	LM	G/TPU-days	dev		test	
					clean	other	clean	other
NST [8]	360M	S2S	lstm	1600	3.9	8.8	4.2	8.6
w2v2-base[23]	95M	CTC	None	102.4	6.1	13.5	6.1	13.3
w2v2-large[23]	317M	CTC	None	294.4	4.6	9.3	4.7	9.0
IPL [7]	322M	CTC	None	192	5.5	9.3	6.0	10.3
slimIPL [9]	322M	CTC	None	83.2	3.7	7.3	3.8	7.5
NST-iter1	118M	transducer	None	14	5.3	12.7	5.4	12.9
GM-iter1	118M	transducer	None	14	4.8	11.1	4.9	11.2
GM-iter5	118M	transducer	None	54	4.1	8.8	4.3	8.8

Table 1. Semi-supervised LibriSpeech results using 100 hours as labeled data and 860 hours as unlabled data. Our experiments are in the lower part of the table.

parameters using a minibatch from the pseudo-labels dataset U', we apply the gradient mask method as described in 3.2 on the model. While on a minibatch from the labeled dataset, we do parameters update in the standard way for transducer in 3.1. This process is repeated until convergence of the word error rate on the validation dataset. Since the loss function is the same for both datasets, we only use one momentum optimizer and the same learning rates for simplicity. The ratio of minibatch from L to minibatch from U' is a hyper-paramtete to be tuned.

Method	LM	dev		test	
		clean	other	clean	other
Hybrid [30]	4-gram	5.0	19.5	5.8	18.6
LAS [8]	lstm	5.3	16.5	5.5	16.9
CTC [9]	None	6.2	16.8	6.2	16.8
Transducer	None	6.3	16.8	6.4	16.7

Table 2. WER on the Librispeech 100 hours for supervised system

4. EXPERIMENTS

4.1. Data

We conducted our experiments on the LibriSpeech [31] datasets. The labeled dataset is a 100 hours subset (train-clean-100) of Librispeech, and the remaining 860 hours (train-clean-360, train-other-500) is the unlabeled dataset. During training, samples in the dataset that are longer than 20 seconds are filtered out. The performance of the trained model is validated on the dev-clean and dev-other of Librispeech and tested on the test-clean/other. We did not use any extra text or LM information for any of our experiments.

We use around 5k subword [32] units as our prediction targets. We extracted 80-channel filterbank features computed from a 25ms window with a stride of 10ms. When training on labeled data, we use speed perturbation and SpecAugment [33, 34] with mask parameter (F = 27), and ten time masks with maximum time-mask ratio (pS = 0.05), where the maximum-size of the time mask is set to pS times the length of the utterance.

4.2. Setup

The filterbank features are first passed into 2 blocks of 2d-conv layers, time reduction layers are added after each block to down-sample the frame rate to 4 before passing into the encoder. The encoder model consists of 17 layers of conformer block, where we set the model dimension to 512, the inner dimension in feed forward layer to 2048, with 8 attention heads, 32 kernal size in convolution block, with the same setting as Conformer-L [34]. We use LSTM as our predictor and the LSTM predictor contains 1 layer with 640 units and a projection layer with 640 units. The Transducer's joint network is a simple feed-forward layer. The total number of parameters is about 130M. Our model is implemented in Pytorch and we optimized our model using Adam. We use this same model in all of our experiments.

For the 100 hours seed model, we first train the GMM-based model in Kaldi [35] to obtain the alignment results on the 100 hours subset, and we use the frame-wise phoneme label to pre-train the encoder. Then we use the pre-trained encoder to initialize our transducer model [36]. For training the transducer model, we use learning rate warm-up for the first 10k updates to a peak of 1e-4, and hold for 60k steps, then linearly decayed it. We grouped the input sequences by length with a batch size of 10k frames per GPU, and trained the models on 4 GPUs for 160k steps in total.

For training the student model, The mask is applied between 2d-conv layers and encoder layers. The mask p is set to 0.065 and m is set to 3 (equal to 12 frames or 0.12 second). This masking schema is similar to [23], and it resulted in around half of frames being masked. We set the ratio of minibatch from labeled data to pseudo-label data to 1:9 (same as the ratio between labeled and unlabeled data) and it produces an ASR model with the best performance. We used learning rate warm-up for the first 10k updates to a peak of

2e-4, and hold for 80k, then linearly decayed it. We grouped the input sequences by length with a batch size of 10k frames per GPU, and trained the models on 8 GPUs for 180k steps.

4.3. Results

4.3.1. Supervised baseline

Table 2 shows the results of our seed model and the comparison with the Librispeech 100 hours supervised model from other papers. We use this seed model to generate the first version of the pseudo-label. The resulted 860 hours pseudo-label have WER around 9.

4.3.2. Semi-supervised experiments

Table 1 shows the results from semi-supervised experiments. *NST-iter1* is the results of the experiment where we simply mixed the pseudo-label data and labeled data to form the new training dataset, and we train the student model using this dataset. This process is a simplified version of noisy student training since we did not do any filtering, LM fusion, or data selection [6, 8].

GM-Iter1 and *GM-Iter5* is the model using gradient mask method. For *GM-Iter1* in the table are the results from the student model directly trained from the pseudo labels generated by the seed model. Our proposed approach significantly outperforms the 100 hours supervised baselines in table 2 and also the noisy student training baseline. For *GM-Iter5*, we iterate the pseudo labeling process 5 times. In particular, the model of *GM-Iter5* achieved highly competitive performance, with a WER of 4.1/8.8 for dev-clean/dev-other and 4.3/8.9 for test-clean/test-other. It is worth noting that our method is highly efficient. We use much fewer computing resources or a much smaller model size compared with other approaches in table 1.

4.4. Ablation study and analysis

4.4.1. Pseudo-labeling iterations

To study the performance from different pseudo-labeling iterations. Table 3 shows the WER on test-clean/other of each training iterations using the gradient mask method. The results are in table 3. We stopped this process after the 5th iteration since the improvement is already minimum at the iter5.

interations	test clean	test other
100h seed	6.2	16.8
iter1	4.9	11.2
iter2	4.6	9.7
iter3	4.4	9.2
iter4	4.3	8.9
iter5	4.3	8.8

Table 3. Ablation on each pseudo-labeling iterations

4.4.2. Gradient mask on labels of different qualities

We conduct an ablation study to investigate the effect of the gradient mask on labels of different qualities. We run the experiments with and without the gradient mask method on those labels. The training is the same as the standard transducer training when we do not use the gradient mask. The training data includes 860 hours pseudo-label data and the 100h labeled data. Pseudo(WER-9) is the pseudo-label generated by the 100h seed model which has around WER 9. Pseudo(WER-15) is generated by the same supervised system but from an early epoch that has WER around 15. Pseudo(WER-5) is generated by the student model from the 3rd iteration. And Pseudo(WER-2) is generated by an intermediate model trained on 960 hours labeled data.

When pseudo-label contains a lot of errors (WER 15), simply adding pseudo label will cause the model performance to degrade comparing with 100h baseline in table 2. Even when we have high-quality pseudo-label (WER 5), the noise in labels still hurts the model performance. On the other hand, the gradient mask method can be robust to bad quality labels and work well consistently on the label of different quality. We found that the worse pseudo-label's quality, the better performance we can obtain using the gradient mask method comparing with the standard training. The standard training will perform comparably to the gradient mask method when pseudo-label has WER around 2, and it would perform better when we use the ground truth reference labels.

data	dev-other	
	gm	w/o gm
Reference	7.5	6.7
Pseudo(WER-2)	7.8	7.6
Pseudo(WER-5)	8.8	9.4
Pseudo(WER-9)	11.1	12.7
Pseudo(WER-15)	14.2	18.9

Table 4. With and without gradient mask on different pseudo-label

5. CONCLUSION

In this paper, we present the Gradient Mask method, a simple and efficient method to improved pseudo-label training for end-to-end speech recognition. Our method can force the model to learn acoustic representation and also be robust to errors in labels. This method can be used to combat label noise in pseudo-label training. In semi-supervised experiments, our method achieved much better performance than the conventional pseudo label training approach and performed comparably to the SOTA approach while being much computation-efficient. Future work includes exploring the extension to other end-to-end ASR systems like LAS and other sequence to sequence tasks like machine translation.

6. REFERENCES

[1] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *ICASSP*. IEEE, 2013, pp. 6704–6708.

[2] Yan Huang, Yongqiang Wang, and Yifan Gong, "Semi-supervised training in deep learning acoustic model," in *Interspeech*, 2016, pp. 3848–3852.

[3] Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur, "Semi-supervised training of acoustic models using lattice-free mmi," in *ICASSP*. IEEE, 2018, pp. 4844–4848.

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

[5] Sree Hari Krishnan Parthasarathi and Nikko Strom, "Lessons from building acoustic models with a million hours of speech," in *ICASSP*. IEEE, 2019, pp. 6670–6674.

[6] Jacob Kahn, Ann Lee, and Awni Hannun, "Self-training for end-to-end speech recognition," in *ICASSP*. IEEE, 2020, pp. 7084–7088.

[7] Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert, "Iterative pseudo-labeling for speech recognition," *arXiv preprint arXiv:2005.09267*, 2020.

[8] Daniel S Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V Le, "Improved noisy student training for automatic speech recognition," *arXiv preprint arXiv:2005.09629*, 2020.

[9] Tatiana Likhomanenko, Qiantong Xu, Jacob Kahn, Gabriel Synnaeve, and Ronan Collobert, "slimipl: Language-model-free iterative pseudo-labeling," *arXiv preprint arXiv:2010.11524*, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[11] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP*. IEEE, 2020, pp. 6419–6423.

[12] Andy T Liu, Shang-Wen Li, and Hung-yi Lee, "TERA: Self-supervised Learning of Transformer Encoder Representation for Speech," *arXiv preprint arXiv:2007.06028*, 2020.

[13] Shaoshi Ling and Yuzong Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[14] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[15] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[16] Benoît Frénay and Michel Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[17] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee, "Learning from noisy labels with deep neural networks: A survey," *arXiv preprint arXiv:2007.08199*, 2020.

[18] Mark A Hasegawa-Johnson, Preethi Jyothi, Daniel McCloy, Majid Mirbagheri, Giovanni M Di Liberto, Amit Das, Bradley Ekin, Chunxi Liu, Vimal Manohar, Hao Tang, et al., "Asr for under-resourced languages from probabilistic transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 50–63, 2016.

[19] Adrien Dufraux, Emmanuel Vincent, Awni Hannun, Armelle Brun, and Matthijs Douze, "Lead2gold: Towards exploiting the full potential of noisy transcriptions for speech recognition," in *ASRU*. IEEE, 2019, pp. 78–85.

[20] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019, pp. 3465–3469.

[21] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," in *Interspeech*, 2019, pp. 146–150.

[22] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *ICASSP*. IEEE, 2020, pp. 6429–6433.

[23] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[24] Shaoshi Ling, Julian Salazar, Yuzong Liu, Katrin Kirchhoff, and AWS Amazon, "Bertphone: Phonetically-aware encoder representations for utterance-level speaker and language recognition," in *Proc. Odyssey*, 2020, pp. 9–16.

[25] Chaitanya Talnikar, Tatiana Likhomanenko, Ronan Collobert, and Gabriel Synnaeve, "Joint masked cpc and ctc training for asr," in *ICASSP*. IEEE, 2021, pp. 3045–3049.

[26] Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang, "Unispeech: Unified speech representation learning with labeled and unlabeled data," *arXiv preprint arXiv:2101.07597*, 2021.

[27] Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Yao Qian, Kenichi Kumatani, and Furu Wei, "Unispeech at scale: An empirical study of pre-training method on large-scale speech recognition dataset," *arXiv preprint arXiv:2107.05233*, 2021.

[28] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[29] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*. IEEE, 2016, pp. 4960–4964.

[30] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "Rwth asr systems for librispeech: Hybrid vs attention–w/o data augmentation," *arXiv preprint arXiv:1905.03072*, 2019.

[31] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[32] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[33] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019, pp. 2613–2617.

[34] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[35] Daniel Povey, Arnab Ghoshal, and Gilles Boulianne, "The Kaldi speech recognition toolkit," in *ASRU*. IEEE Signal Processing Society, 2011.

[36] Hu Hu, Rui Zhao, Jinyu Li, Liang Lu, and Yifan Gong, "Exploring pre-training with alignments for rnn transducer based end-to-end speech recognition," in *ICASSP*. IEEE, 2020, pp. 7079–7083.