ADVANCING MOMENTUM PSEUDO-LABELING WITH CONFORMER AND INITIALIZATION STRATEGY

Yosuke Higuchi[1,2*], *Niko Moritz*[1], *Jonathan Le Roux*[1], *Takaaki Hori*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), USA [2]Waseda University, Japan

higuchi@pcl.cs.waseda.ac.jp, {leroux, thori}@merl.com

ABSTRACT

Pseudo-labeling (PL), a semi-supervised learning (SSL) method where a seed model performs self-training using pseudo-labels generated from untranscribed speech, has been shown to enhance the performance of end-to-end automatic speech recognition (ASR). Our prior work proposed momentum pseudo-labeling (MPL), which performs PL-based SSL via an interaction between online and offline models, inspired by the mean teacher framework. MPL achieves remarkable results on various semi-supervised settings, showing robustness to variations in the amount of data and domain mismatch severity. However, there is further room for improving the seed model used to initialize the MPL training, as it is in general critical for a PL-based method to start training from high-quality pseudo-labels. To this end, we propose to enhance MPL by (1) introducing the Conformer architecture to boost the overall recognition accuracy and (2) exploiting iterative pseudo-labeling with a language model to improve the seed model before applying MPL. The experimental results demonstrate that the proposed approaches effectively improve MPL performance, outperforming other PL-based methods. We also present in-depth investigations to make our improvements effective, e.g., with regard to batch normalization typically used in Conformer and LM quality.

***Index Terms*—** pseudo-labeling, self-training, semi-supervised learning, end-to-end speech recognition, deep learning

1. INTRODUCTION

Recent progress in automatic speech recognition (ASR) has shifted towards the end-to-end (E2E) framework, which aims to model direct speech-to-text conversion using a single deep neural network [1–3]. With well-established sequence-to-sequence modeling techniques [4–7] and more sophisticated neural network architectures [8, 9], E2E ASR models have shown promising results on various benchmarks [10–12]. However, the performance often depends on the availability of a large quantity of labeled (transcribed) speech data, which is not always feasible with high annotation costs.

To compensate for the limited amount of labeled data, semi-supervised learning (SSL) [13] methods that make use of a large amount of unlabeled data to improve the model performance can be applied. While various efforts have been made to perform SSL in E2E ASR [14–19], a pseudo-labeling (PL) [20] (or self-training [21])-based approach has been attracting increasing attention due to its effectiveness and simplicity [22–29]. In PL, a seed model is first trained on labeled data and used to generate pseudo-labels for unlabeled data. Both the labeled and pseudo-labeled data are then used to train a better-performing model. In our previous work [29], we proposed a PL-based method for E2E ASR, called momentum pseudo-labeling (MPL). MPL trains a pair of online and offline models that interact and learn from each other, inspired by the mean teacher framework [30]. The online model is trained to predict pseudo-labels generated on the fly by the offline model, which maintains an exponential moving average of the online model parameters. Through the interaction between the two models, MPL effectively stabilizes the training with unlabeled data and significantly improves over seed model performance.

One of the crucial factors for making PL-based approaches successful is to avoid generating severely erroneous pseudo-labels, which can lead to limiting the improvement of an E2E ASR model. In a typical SSL setting in ASR, the amount of labeled data is quite small and the quality of pseudo-labels is not necessarily guaranteed. To this end, an external language model (LM) and beam-search decoding are often incorporated into the labeling process [22, 23]. In [25, 26], low-quality pseudo-labels are excluded via confidence-based filtering to promote model training with high-quality labels. And in [27], an N-best list of pseudo-labels is leveraged to incorporate more appropriate supervision from alternative ASR hypotheses.

We believe that MPL still has room for further improvement by making the models capable of generating pseudo-labels with higher quality. Thus, in this work, we propose to enhance MPL by (1) introducing the Conformer architecture to boost the overall recognition accuracy, and (2) using iterative pseudo-labeling [23] to transfer LM knowledge into a seed model before performing MPL. The key contributions of this work are summarized as follows. (a) We show that vanilla Conformer suffers from generalizing to unlabeled data, especially when there is a domain mismatch against labeled data. We mitigate this issue by substituting batch normalization with group normalization for the convolution module. (b) We demonstrate that the improved MPL is robust to over-fitting to an LM training text set, which has been reported as problematic for using an LM in PL [26, 28]. We also investigate the importance of LM quality in our framework. (c) We show the proposed approaches effectively enhance MPL, conducting experiments on a variety of SSL scenarios with varying amounts of unlabeled data or domain mismatch.

2. MOMENTUM PSEUDO-LABELING

In this section, we review the MPL method proposed in our prior work [29]. MPL is described in two steps: 1) the supervised training of a seed E2E ASR model, and 2) the MPL-based semi-supervised training of the model using unlabeled data.

2.1. Supervised training of a seed model

E2E ASR is formulated as a sequence mapping problem between a T-length input sequence $X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \dots, T)$ and an L-length output sequence $Y = (y_l \in \mathcal{V} | l = 1, \dots, L)$. Here, $\mathbf{x}_t$ is a D-dimensional acoustic feature at frame t, y_l an output token at position l, and $\mathcal{V}$ a vocabulary. This work focuses on the connectionist temporal classification (CTC)-based E2E ASR model [1, 4], which

[*] Work done during an internship at MERL

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

is less prone to the looping and early stopping issues often caused by autoregressive decoder networks [22, 31]. CTC predicts a frame-level latent sequence $Z = (z_t \in \mathcal{V} \cup \{\epsilon\} | t = 1, \ldots, T)$, which is obtained by augmenting Y with a special blank token ϵ. Based on the conditional independence assumption between token predictions, CTC models the conditional probability $P(Y|X)$ by marginalizing over latent sequences as

$$P(Y|X) \approx \sum_{Z \in \mathcal{B}^{-1}(Y)} \prod_{t=1}^{T} P(z_t|X), \tag{1}$$

where $\mathcal{B}^{-1}(Y)$ returns all possible latent sequences compatible with Y. Given labeled data $\mathcal{D}_{\mathsf{sup}} = \{(X_n, Y_n) | n = 1, \ldots, N\}$, a seed model P_θ with parameters θ is optimized by minimizing the negative log-likelihood of Eq. (1):

$$\mathcal{L}_{\mathsf{sup}}(\theta) = -\log P_\theta(Y_n | A(X_n)), \tag{2}$$

where $A(\cdot)$ indicates SpecAugment [32] for augmenting the input.

2.2. Semi-supervised training with MPL

The goal of semi-supervised training is to exploit unlabeled data $\mathcal{D}_{\mathsf{unsup}} = \{X_m | m = N{+}1, \ldots, N{+}M\}$ for enhancing the seed model trained on labeled data $\mathcal{D}_{\mathsf{sup}}$. MPL performs the training using a pair of *online* and *offline* models that interact and learn from each other. Let P_ξ and P_ϕ be the online and offline models with parameters ξ and ϕ, which are initialized with the seed model parameters θ.

Online model training: Given an unlabeled sample $X \in \mathcal{D}_{\mathsf{unsup}}$, the online model is trained on pseudo-labels $\hat{Y}$ generated on the fly by the offline model:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, P_\phi(Y|X), \tag{3}$$

where argmax is performed by the best path decoding of CTC [4]. With the pseudo-labels generated from Eq. (3), the objective of the online model is defined in the same manner as Eq. (2):

$$\mathcal{L}_{\mathsf{unsup}}(\xi) = -\log P_\xi(\hat{Y}_{N+m} | A(X_{N+m})), \tag{4}$$

where $\mathcal{L}_{\mathsf{unsup}}$ is optimized via a gradient descent optimization. Note that, during the semi-supervised training, we also use labeled data $\mathcal{D}_{\mathsf{sup}}$ and the supervised loss $\mathcal{L}_{\mathsf{sup}}(\xi)$, which helps the online model stabilize and promote learning from unlabeled data with $\mathcal{L}_{\mathsf{unsup}}(\xi)$.

Offline model training: After every update of the online model, the offline model accumulates parameters of the online model via the momentum-based moving average:

$$\phi \leftarrow \alpha\phi + (1 - \alpha)\xi, \tag{5}$$

where $\alpha \in [0, 1]$ is a momentum coefficient. This momentum update makes the offline model evolve more smoothly than the online model, preventing the pseudo-labels from deviating too quickly from the labels initially generated by the seed model. To handle the sensitive tuning of the momentum coefficient α, we follow our prior work and indirectly derive α from a weight $w = \alpha^K$, where K is the number of iterations (i.e., batches) in a training epoch. The weight w can be regarded as the proportion of the seed model retained after a training epoch, and we fix it to 50% (i.e., $w = 0.5$), as it has been shown consistently effective in various semi-supervised settings [29].

3. PROPOSED IMPROVEMENTS FOR MOMENTUM PSEUDO-LABELING

We propose to enhance MPL by (1) introducing the Conformer architecture [9] to improve the overall accuracy, and (2) adopting iterative pseudo-labeling (IPL) [23] to transfer LM knowledge into the seed model. We expect these approaches to promote the MPL training by enabling the models to generate higher-quality pseudo-labels.

Algorithm 1 Momentum pseudo-labeling using iterative pseudo-labeling for transferring LM knowledge into seed model

```
Input:
    $\mathcal{D}_{\mathsf{sup}}$, $\mathcal{D}_{\mathsf{unsup}}$        ▷ labeled and unlabeled data
    $\mathcal{A}$                  ▷ an ASR model architecture
    $\alpha$                  ▷ a momentum coefficient
# 1.Seed model training
Train a seed model $P_\theta$ with architecture $\mathcal{A}$ on $\mathcal{D}_{\mathsf{sup}}$ using Eq. (2)
# 2.Iterative pseudo-labeling
for $i = 1, \ldots, I_{\mathsf{ipl}}$ do
    Generate pseudo-labels $\hat{\mathcal{D}}_{\mathsf{unsup}} = \{(X_m, \hat{Y}_m) | X_m \in \mathcal{D}_{\mathsf{unsup}}\}$,
        using $P_\theta$ and LM with beam-search decoding
    for $e = 1, \ldots, E_{\mathsf{ipl}}$ do
        for all $(X, Y) \in \mathcal{D}_{\mathsf{sup}} \cup \hat{\mathcal{D}}_{\mathsf{unsup}}$ do
            Compute loss $\mathcal{L}$ for $P_\theta(Y|X)$ with Eq. (2)
            Update $\theta$ using $\nabla_\theta \mathcal{L}$
        end for
    end for
end for
# 3.Momentum pseudo-labeling
Initialize an online model $P_\xi$ and an offline model $P_\phi$ with $P_\theta$
for $e = 1, \ldots, E_{\mathsf{mpl}}$ do
    for all $S \in \mathcal{D}_{\mathsf{sup}} \cup \mathcal{D}_{\mathsf{unsup}}$ do
        Obtain $X \sim S$
        Obtain $Y = \begin{cases} Y \sim S & (S \in \mathcal{D}_{\mathsf{sup}}) \\ \hat{Y} \sim P_\phi(Y|X) & (S \in \mathcal{D}_{\mathsf{unsup}}) \end{cases}$
        Compute loss $\mathcal{L}$ for $P_\xi(Y|X)$ with Eq. (2) or (4)
        Update $\xi$ using $\nabla_\xi \mathcal{L}$
        Update $\phi \leftarrow \alpha\phi + (1 - \alpha)\xi$
    end for
end for
return $P_\xi$                  ▷ online model is returned for final evaluation
```

3.1. Conformer for semi-supervised training

Conformer is a variant of Transformer augmented with convolution to increase the capability for capturing local feature patterns [9]. In addition to the multi-head self-attention layer in the Transformer encoder, Conformer introduces a module based on depthwise separable convolution [33]. Unlike Transformer, Conformer employs relative positional encoding and macaron-like feed-forward layers.

While Conformer-based models have achieved outstanding ASR performance compared with standard Transformers [34], we empirically observe that Conformer suffers from poor generalization from labeled data to unlabeled data. A similar issue has been reported in other ASR tasks [35–37]. Simply adopting Conformer for MPL makes the training become unstable and diverge easily, especially when a domain mismatch exists between labeled and unlabeled data.

We assume that such a problem comes from unreliable statistics computed and used by batch normalization (BN) [38] in the convolution module. As we suppose the amount of labeled data relatively small (i.e., 100h), the estimated mean and variance of the whole dataset are likely to become less accurate in BN [39]. A simple solution is to increase the mini-batch size. However, we observe that a large mini-batch size degrades the seed model performance, which can lead to degrading the quality of pseudo-labels during the MPL training. Hence, we consider replacing BN with group normalization (GN) [40] in the convolution module, as it has been investigated in [35, 41]. GN divides feature maps into groups and normalizes the features within each group, which makes the training less dependent on the mini-batch size. This is found critical for stabilizing the Conformer-based MPL training, as examined in Sec. 4.2.

3.2. Iterative pseudo-labeling for enhancing seed model

To provide the MPL training with a better model for initializing the online and offline models, we consider enhancing the seed model using (IPL). IPL continuously trains a model with periodic regeneration of pseudo-labels, where an external LM and beam-search decoding are used to generate the labels [23]. While beam-search decoding with an LM plays an important role for generating pseudo-labels with high quality [22, 42], it is computationally intensive for MPL due to the on-the-fly label generation. Hence, we exploit IPL to implicitly transfer LM knowledge to the seed model before applying MPL, providing the MPL training with a better initialization for generating higher-quality pseudo-labels. Moreover, by not using the LM-based pseudo-labels during the MPL traning, we can prevent the model from over-fitting to the LM training text data, which often degrades the generalization capability of the ASR model [26, 28].

Algorithm 1 shows the proposed MPL training with IPL initialization. In the beginning, a seed model is trained using a labeled set as in Sec. 2.1 (line 1–2). Then, the seed model is further trained via IPL with LM and beam-search decoding (line 3–12). Here, we denote I_{ipl} as the number of iterations (pseudo-label updates), and E_{ipl} as the number of epochs trained in each iteration. We refer to standard pseudo-labeling (PL) [22] when $I_{\mathsf{ipl}} = 1$ and IPL [23] when $I_{\mathsf{ipl}} > 1$. Finally, the enhanced seed model is used to initialize the models for MPL (line 13–23). The MPL training lasts E_{mpl} epochs.

In our prior work, we have discussed a little about applying PL as the initialization strategy for MPL [29] and demonstrated its effectiveness. This work extends this early idea by focusing on the better-performing IPL. In Sec. 4.5, we also investigate the influence of the quality of LM used for PL on improving MPL.

4. EXPERIMENTS

4.1. Experimental setting

Data: We conducted experiments using the LibriSpeech (LS) [43] and TEDLIUM3 (TED3) [44] datasets. LS is a corpus of read English speech, containing 960 hours of training data (split into "train-clean-100", "train-clean-360", and "train-other-500"). TED3 is a corpus of English Ted Talks consisting of 450 hours of training data ("train-ted3"). We used the standard development and test sets for each dataset. Kaldi [45] was used to extract 80 mel-scale filterbank coefficients with three-dimensional pitch features. For text tokenization, we used a 1k subword vocabulary, which was constructed from the "train-clean-100" transcriptions using SentencePiece [46].

Semi-supervised settings: After training a seed model on the labeled "train-clean-100" (LS-100) set, we considered three semi-supervised settings with different unlabeled sets: LS-100/LS-360, an in-domain setting with "train-clean-360" (LS-360); LS-100/LS-860, an in-domain setting with "train-{clean-360,other-500}" (LS-860); and LS-100/TED3, an out-of-domain setting with "train-ted3".

ASR model: We used the Conformer architecture [9] implemented in ESPnet [47], which consists of two convolutional neural network layers followed by a stack of 12 self-attention layers. The number of heads H, dimension of a self-attention layer d_{model}, dimension of a feed-forward network d_{ff}, and kernel size K were set to 4, 256, 2048, and 31, respectively. We set the number of groups to 8 for group normalization when used in the convolution module.

Training configuration: We basically followed our prior work [29]. The seed model was trained for 150 epochs using the Adam optimizer [48], and Noam learning rate scheduling [49]. The semi-supervised training was done up to 200 epochs, where the gradient-based optimization was done by using the Adam optimizer. IPL was performed by iterating PL for the maximum of 8 times ($I_{\mathsf{ipl}} \leq 8$), where the model was trained for 25 epochs ($E_{\mathsf{ipl}} = 25$) in each iteration. Note that, after each iteration, we averaged model parameters over the last 5 checkpoints to stabilize the pseudo-label generation. We set $w = 0.5$ for MPL training, following our prior work [29].

Decoding configuration: For evaluation, a final model was obtained by averaging model parameters over 10 checkpoints with the best validation performance. We trained an LM consisting of 4 long short-term memory (LSTM) layers with 2048 units, using the LS-100 transcriptions combined with the external text data provided by LibriSpeech [43]. For decoding with the LM, we adopted a frame-synchronous CTC prefix beam search algorithm [50, 51], where we used a beam-size of 20, a score-based pruning threshold of 14.0, an LM weight of 1.0, and an insertion bonus factor of 2.0. For decoding without an LM, we performed the best path decoding of CTC [4].

Table 1. Validation WER [%] for seed models trained on labeled LS-100. For the Conformer-based models, we explored different normalization methods for the convolution module.

		LibriSpeech		TED3
Model	**Norm. type**	dev-clean	dev-other	Dev
Transformer	–	12.2	30.0	31.2
Conformer	Batch	8.6	23.1	27.3
	Instance	8.9	23.5	27.1
	Group	**8.4**	**22.5**	**26.4**
	Layer	**8.4**	22.9	26.9

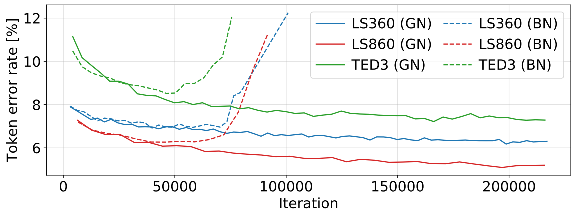

Fig. 1. Validation token error rate [%] of MPL training using Conformer with batch (dotted line) or group (solid line) normalization.

4.2. Effectiveness of adopting Conformer for MPL

In Table 1, we compare the word error rate (WER) of seed models trained with the Transformer (Trf) or the Conformer (Cfm) architecture. For Cfm-based models, we investigated different normalization methods for the convolution module, including {batch [38], instance [52], group [40], layer [53]} normalization ({BN, IN, GN, LN}). Note that IN and LN are the same as GN with group sizes 1 and 256 ($= d_{\mathsf{model}}$), respectively. Comparing the two architectures, the Cfm-based models significantly improved over the Trf-based model. Within the Cfm-based models, GN resulted in the best performance on both LS and TED3, and the 100-hour training data seemed to be too small to take advantage of BN. As normalizing across feature maps (i.e., IN, GN, and LN) achieved better performance than BN on the out-of-domain TED3 set, it indicates that BN led to lower generalization capability with unreliable statistics. Note that in [41], BN achieved better performance than the other normalization methods when another depthwise separable convolution-based ASR model is trained on the full 960-hour set of LS.

Figure 1 shows learning curves from MPL training using Cfm with BN or GN. In all semi-supervised settings, BN caused the training to become unstable. Especially in the out-of-domain setting with TED3, the model diverged more quickly than in the other settings. In contrast, GN successfully stabilized the MPL training with Cfm.

4.3. Results on in-domain setting

Table 2 lists results on the in-domain LS settings in terms of the WER and WER recovery rate (WRR) [54]. The topline results were obtained via fully supervised training on each setting. Looking at the MPL results (A1,B1), MPL led to a substantial improvement over the seed model (L0), effectively learning from unlabeled data using Cfm with GN. These Cfm results significantly outperformed those of prior Trf-based MPL [29] (A0,B0 vs. A1,B1). With pseudo-labels generated using the LM, PL [22] and IPL [23] achieved lower

Table 2. Word error rate (WER) [%] and WER recovery rate (WRR) [%] on in-domain LibriSpeech (LS) settings. The results are divided into two sections: whether the LM with beam-search decoding was applied in the final evaluation or not. † indicates trained for 100 epochs.

			Decoding without LM						Decoding with LM					
			Dev WER		**Test WER**		**Test WRR**		**Dev WER**		**Test WER**		**Test WRR**	
Setting	**Method**	**Init.**	clean	other	clean	other	clean	other	clean	other	clean	other	clean	other
LS-100	L0 seed (Cfm)	–	8.4	22.5	8.6	23.3	0.0	0.0	5.2	15.2	5.5	16.0	0.0	0.0
LS-100 / LS-360	A0 MPL (Trf)	seed (Trf)	8.7	21.4	9.0	21.7	–	–	4.8	13.0	5.1	13.1	–	–
	A1 MPL	L0	6.1	16.0	6.6	15.8	52.3	76.4	4.5	11.2	4.7	**11.1**	34.7	71.7
	A2 PL	L0	5.7	15.9	6.1	15.8	64.6	76.0	4.3	11.4	4.5	11.8	40.6	62.2
	A3 IPL	L0	**5.4**	15.1	5.7	15.3	73.3	81.5	4.2	11.5	4.5	11.7	42.2	62.5
	A4 MPL†	A2@ep100	5.7	15.5	6.1	15.6	64.8	77.8	4.2	11.1	4.5	11.3	44.0	69.3
	A5 MPL†	A3@ep100	5.5	**15.0**	**5.6**	**15.1**	**75.1**	**83.3**	**4.1**	**10.8**	**4.3**	**11.1**	**51.4**	**72.7**
	A6 topline	L0	4.1	13.6	4.7	13.4	100.0	100.0	2.9	9.4	3.2	9.2	100.0	100.0
LS-100 / LS-860	B0 MPL	seed (Trf)	8.1	16.5	8.3	16.8	–	–	4.6	9.7	4.8	10.1	–	–
	B1 MPL	L0	5.7	12.2	6.2	12.2	48.1	76.4	4.1	8.5	4.4	8.7	36.5	74.6
	B2 PL	L0	5.4	13.9	5.7	14.2	57.8	62.3	4.0	10.5	4.2	10.7	43.0	53.7
	B3 IPL	L0	**4.7**	11.5	**5.0**	11.7	**71.0**	79.3	4.1	9.7	4.4	10.2	36.2	58.9
	B4 MPL†	B2@ep100	5.1	12.1	5.3	12.4	64.0	75.1	3.7	8.4	3.9	8.8	51.0	73.3
	B5 MPL†	B3@ep100	**4.7**	**11.0**	**5.0**	**11.1**	70.0	**83.9**	**3.6**	**7.8**	**3.8**	**8.2**	**54.0**	**79.6**
	B6 topline	L0	3.3	9.0	3.5	8.7	100.0	100.0	2.4	6.1	2.5	6.2	100.0	100.0

Table 3. WER [%] and WRR [%] on out-domain TEDLIUM3 (TED3) setting.

			Decoding without LM			Decoding with LM		
Setting	**Method**	**Init.**	**Dev WER**	**Test WER**	**Test WRR**	**Dev WER**	**Test WER**	**Test WRR**
LS-100	L0 seed (Cfm)	–	26.4	26.5	0.0	21.3	21.1	0.0
LS-100 / TED3	C0 MPL (Trf)	seed (Trf)	18.4	17.0	–	14.9	13.3	–
	C1 MPL	L0	15.1	13.9	81.0	12.7	**11.6**	**77.3**
	C2 IPL	L0	16.8	16.8	62.2	16.6	16.9	34.2
	C3 MPL†	C2@ep100	**14.6**	**13.8**	**81.1**	**12.4**	12.0	73.8
	C4 topline	L0	10.4	10.9	100.0	8.6	8.8	100.0

WERs on the "clean" sets than those obtained from MPL, and IPL resulted in better performance than MPL on the "other" sets as well (⋆2,⋆3 vs. ⋆1). However, when decoded with the LM, the performance gain was larger for MPL with a slight decrease in WRRs, and MPL achieved much lower WERs on the "other" sets. PL and IPL, in contrast, had smaller improvement with degraded WRRs, which indicates PL and IPL are fitted to LM knowledge and have less variations in the hypotheses during the beam-search decoding. ⋆4 and ⋆5 show results for the proposed MPL training using the seed model enhanced by PL and IPL, respectively. Note that we performed PL or IPL for 100 epochs and MPL for another 100 epochs to match the total training epochs of the other methods. The initialization strategy provided MPL with distinct improvements, pushing the limit of the other methods (⋆4,⋆5 vs. ⋆1,⋆2,⋆3). With the IPL-based initialization, MPL achieved the best overall performance on both of the settings with different amounts of unlabeled data (A5,B5). Moreover, when decoded with the LM, the improved MPL retained higher WRRs than IPL (⋆3 vs. ⋆5), maintaining the advantage of MPL and making the model less dependent on the LM knowledege.

4.4. Results on out-of-domain setting

Table 3 shows MPL results on the TED3 setting. Cfm with GN significantly improved MPL over the seed model and Trf-based MPL (C1 vs. L0,C0), successfully stabilizing training on the out-of-domain data. IPL led to a decent improvement over the seed model, but the gain was more substantial for MPL (C1 vs. C2). As there is a domain mismatch between the LM training text and the actual transcriptions of TED3, IPL was less effective at learning from the out-of-domain unlabeled data. Moreover, IPL had little gain from decoding with the LM, indicating the model was prone to over-fit to the LM knowledge. By using IPL to enhance the seed model, MPL further reduced WERs (C1 vs. C3). However, the improvement was much smaller than those observed in the in-domain settings, and the standard MPL performed sufficiently well by decoding with the LM.

Table 4. WER for MPL initialized by PL with different LM quality.

		Small LM		Large LM	
Setting	**Test data**	**PL**	**→ MPL**	**PL**	**→ MPL**
LS-100 / LS-360	test-clean	6.3	6.2	6.2	6.1
	test-other	16.8	15.9	16.4	15.6
LS-100 / LS-860	test-clean	6.2	5.7	5.7	5.3
	test-other	15.3	12.9	14.5	12.4

4.5. Does better language model lead to better MPL results?

In Table 4, we study the importance of LM quality used in PL to improve MPL performance. We focus on in-domain settings (A4,B4 in Table 2), where the initialization strategy was especially effective. We compare a small LM (1-layer LSTM) and large LM (4-layer LSTM), validation perplexities of which were 20.9 and 14.3, respectively. PL was evaluated at epoch 100, which is then used to initialize MPL. As a result, the large LM led to better PL performance and, accordingly, improved MPL with better pseudo-label generation.

5. CONCLUSIONS

We proposed several improvements to momentum pseudo-labeling (MPL) for semi-supervised ASR. Experimental results on various semi-supervised settings demonstrated the effectiveness of the enhanced MPL, showing clear improvements over our prior results and other PL-based methods. Moreover, we investigated and shared the key components to make the proposed approaches effective, including normalization method for Conformer and the quality of LM for generating pseudo-labels. Future work should consider evaluating MPL on lower-resource scenarios (e.g., 10h of labeled data [55]).

6. REFERENCES

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014.

[2] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho *et al.*, "Attention-based models for speech recognition," in *Proc. NeurIPS*, 2015.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

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

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, 2014.

[7] D. Bahdanau *et al.*, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2014.

[8] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, 2018.

[9] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar *et al.*, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*, 2020.

[10] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[11] C. Lüscher, E. Beck, K. Irie, M. Kitza *et al.*, "RWTH ASR systems for LibriSpeech: Hybrid vs attention," in *Proc. Interspeech*, 2019.

[12] S. Karita, N. Chen, T. Hayashi, T. Hori *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, 2019.

[13] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, 2009.

[14] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. ASRU*, 2017.

[15] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang *et al.*, "Cycle-consistency training for end-to-end speech recognition," in *Proc. ICASSP*, 2019.

[16] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *Proc. ICASSP*, 2020.

[17] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.

[18] Y. Zhang, J. Qin, D. S. Park, W. Han *et al.*, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[19] M. K. Baskar, L. Burget, S. Watanabe, R. F. Astudillo *et al.*, "EAT: Enhanced ASR-TTS for self-supervised speech recognition," in *Proc. ICASSP*, 2021.

[20] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. ICML*, 2013.

[21] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Trans. Inf. Theory*, vol. 11, no. 3, 1965.

[22] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *Proc. ICASSP*, 2020.

[23] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun *et al.*, "Iterative pseudo-labeling for speech recognition," in *Proc. Interspeech*, 2020.

[24] Y. Chen, W. Wang, and C. Wang, "Semi-supervised ASR by end-to-end self-training," in *Proc. Interspeech*, 2020.

[25] D. S. Park, Y. Zhang, Y. Jia, W. Han *et al.*, "Improved noisy student training for automatic speech recognition," in *Proc. Interspeech*, 2020.

[26] S. Khurana, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *Proc. ICASSP*, 2021.

[27] N. Moritz, T. Hori, and J. Le Roux, "Semi-supervised speech recognition via graph-based temporal classification," in *Proc. ICASSP*, 2021.

[28] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve *et al.*, "slimIPL: Language-model-free iterative pseudo-labeling," in *Proc. Interspeech*, 2021.

[29] Y. Higuchi, N. Moritz, J. Le Roux, and T. Hori, "Momentum pseudo-labeling for semi-supervised speech recognition," in *Proc. Interspeech*, 2021.

[30] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. NeurIPS*, 2017.

[31] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[32] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019.

[33] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. CVPR*, 2017.

[34] P. Guo, F. Boyer, X. Chang, T. Hayashi *et al.*, "Recent developments on ESPnet toolkit boosted by Conformer," in *Proc. ICASSP*, 2021.

[35] B. Li, A. Gulati, J. Yu, T. N. Sainath *et al.*, "A better and faster end-to-end model for streaming ASR," in *Proc. ICASSP*, 2021.

[36] Y. C. Liu, E. Han, C. Lee, and A. Stolcke, "End-to-end neural diarization: From Transformer to Conformer," in *Proc. Interspeech*, 2021.

[37] J. Kim, J. Lee, and Y. Lee, "Generalizing RNN-transducer to out-domain audio via sparse self-attention layers," *arXiv preprint arXiv:2108.10752*, 2021.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[39] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Proc. NeurIPS*, 2017.

[40] Y. Wu and K. He, "Group normalization," in *Proc. ECCV*, 2018.

[41] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang *et al.*, "QuartzNet: Deep automatic speech recognition with 1D time-channel separable convolutions," in *Proc. ICASSP*, 2020.

[42] E. Wallington, B. Kershenbaum, O. Klejch, and P. Bell, "On the learning dynamics of semi-supervised training for ASR," in *Proc. Interspeech*, 2021.

[43] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[44] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko *et al.*, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Proc. SPECOM*, 2018.

[45] D. Povey, A. Ghoshal, G. Boulianne, L. Burget *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[46] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proc. ACL*, 2018.

[47] S. Watanabe, T. Hori, S. Karita, T. Hayashi *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[50] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs," *arXiv preprint arXiv:1408.2873*, 2014.

[51] N. Moritz, T. Hori, and J. Le Roux, "Streaming end-to-end speech recognition with joint CTC-attention based models," in *Proc. ASRU*, 2019.

[52] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[53] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[54] J. Ma and R. Schwartz, "Unsupervised versus supervised training of acoustic models," in *Proc. Interspeech*, 2008.

[55] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov *et al.*, "Libri-light: A benchmark for ASR with limited or no supervision," in *Proc. ICASSP*, 2020.