KAIZEN: CONTINUOUSLY IMPROVING TEACHER USING EXPONENTIAL MOVING AVERAGE FOR SEMI-SUPERVISED SPEECH RECOGNITION

Vimal Manohar, Tatiana Likhomanenko, Qiantong Xu, Wei-Ning Hsu,
Ronan Collobert, Yatharth Saraf, Geoffrey Zweig, Abdelrahman Mohamed

Facebook AI

ABSTRACT

In this paper, we introduce the Kaizen framework that uses a continuously improving teacher to generate pseudo-labels for semi-supervised speech recognition (ASR). The proposed approach uses a teacher model which is updated as the exponential moving average (EMA) of the student model parameters. We demonstrate that it is critical for EMA to be accumulated with full-precision floating point. The Kaizen framework can be seen as a continuous version of the iterative pseudo-labeling approach for semi-supervised training. It is applicable for different training criteria, and in this paper we demonstrate its effectiveness for frame-level hybrid hidden Markov model-deep neural network (HMM-DNN) systems as well as sequence-level Connectionist Temporal Classification (CTC) based models. For large scale real-world unsupervised public videos in UK English and Italian languages the proposed approach i) shows more than 10% relative word error rate (WER) reduction over standard teacher-student training; ii) using just 10 hours of supervised data and a large amount of unsupervised data closes the gap to the upper-bound supervised ASR system that uses 650h or 2700h respectively.

Index Terms— speech recognition, semi-supervised training, pseudo-labeling, low-resource, teacher-student

1. INTRODUCTION

Self-training [1, 2, 3] is one of the most widely used approaches for semi-supervised training of automatic speech recognition (ASR). This approach uses an initial model that is called "teacher" or "seed" model to generate labels for unsupervised data. The generated labels are called pseudo-labels (PLs). The labeling can be done at the frame-level, which is usually in the form of soft targets or a distribution as in the case of knowledge distillation [4, 5, 6], or at sequence-level. While there are approaches to use a distribution over sequences [7, 8, 9] for sequence-level distillation, often only the best hypothesis sequence is used as PLs. The unsupervised data with the PLs is combined with the supervised data to train a new model. This approach is also known as "pseudo-labeling" (PL) and serves as the baseline for semi-supervised training methods. This process can be repeated for several "generations" to obtain better models in successive generations [10]. Strong data augmentation while training the student model is shown to improve self-training and helps to avoid local optima [11, 12, 13]. As opposed to changing the teacher model in discrete steps, i.e. after each PL generation, some recent works explored updating the model continuously and using it to generate PLs [13, 12, 14]. In this class of approaches, we propose a new PL framework, called Kaizen. In Kaizen, we propose to use the Exponential Moving Average (EMA) of the student model as the teacher model.

Our research is focused on semi-supervised learning for low-resource scenarios when only 1-10h of supervised data is available. In this paper, we make the following novel contributions: 1) We propose EMA teacher for semi-supervised ASR and empirically show that Kaizen framework in combination with data augmentation stabilizes the training even on large-scale realistic datasets with more than 10k hours of unlabeled data and only 1-10h of labeled data. 2) We analyse the training dynamics with EMA teacher and show that for stable training it is critical for the teacher to be sufficiently far away from the student model: EMA teacher being too close to the student model causes model's collapse and divergence, while being too far leads to slow convergence. 3) Kaizen outperforms a 10h supervised baseline and a single generation of PL by more than 50% and 10% relative WER reduction, respectively, with large scale real-world unsupervised public videos in UK English and Italian languages. 4) EMA teacher combines effectively with slimIPL [14], an alternate approach to stabilize training, and achieves new state-of-the-art results for greedy decoding on LibriSpeech [15] using labeled 10h and unlabeled 54k hours of Libri-Light [16] data.

In Section 2, we compare our proposed work to related works in the literature. In Section 3, we describe the Kaizen framework and the training criteria used. In Section 4, we describe the experimental setup and discuss results. In Section 5, we provide our conclusions and planned future work.

2. RELATED WORK

EMA has been used previously for semi/self-supervised training. Temporal Ensembling [17] uses EMA on network predictions while in this work we apply it on the network parameters. Mean Teacher [18] uses EMA on parameters and consistency cost for image recognition tasks. In this work, we generalize EMA teacher to use with sequence-level loss like Connectionist Temporal Classification (CTC) [19] and on ASR tasks. BYOL [20] showed that EMA teacher can be used for self-supervised learning without using negative examples while our work focuses on semi-supervised learning. Multiple iterations of PL along with strong data augmentation are shown to be superior to the single generation of PL [11, 12]. Work [13] extends this to continuously train a single model and use the latest model state to generate new PLs. In [14], this approach was found to be unstable and prone to divergence. slimIPL algorithm [14] gets around this via a dynamic cache containing PLs generated from the older model states. Our proposed Kaizen framework is an alternative way of stabilizing the training when using a continuously updating teacher via EMA with a sufficiently large discount factor. Independently, a concurrent work [21] proposed to use EMA teacher with CTC criterion and applied it to semi-supervised ASR with 100s of hours of supervised data. In this work, we focus on the low resource scenario of 1-10hr of supervised data, and understanding the training dynamics with EMA teacher in this setting.

3. METHOD

3.1. Kaizen: continuously improving teacher

The Kaizen framework consists of a pair of models – the teacher model and the student model – that are trained simultaneously. The student model is trained using standard gradient-based optimization. Let its parameters be θ_t after t updates. The teacher model parameters ξ_t are updated every Δ steps as the EMA of the student model parameters:

$$\xi_t = (1-\alpha)\xi_{t-\Delta} + \alpha\theta_t, \qquad t = n\Delta,\, n \in \mathbb{Z}^+, \quad (1)$$

where α is a discount factor. A higher α discounts the older student models' parameters and gives more weight to the more recent student models' parameters.

In Kaizen framework, training progresses in 2 stages – *burn-in* and continuous PL. In the *burn-in* stage, the student model is trained using PLs from a previous seed model. In the continuous PL stage, the student model is trained using the PLs generated by the continuously updating teacher model.

The continuous PL stage can be described using a block diagram, Figure 1. The audio features x from an utterance in the unsupervised dataset is fed through both teacher and student neural network models. For the student model, the data is augmented on-the-fly using data augmentation approaches like SpecAugment [22]. The student network hidden activations are also randomly dropped using dropout [23], while dropout is not applied on the teacher network. The resultant

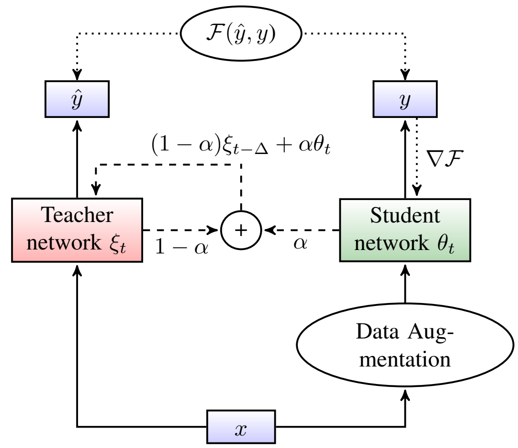

Fig. 1. Block diagram of the Kaizen framework.

outputs from the teacher and student models, $\hat{y}$ and y respectively, are used to compute the loss $\mathcal{F}(\hat{y}, y)$. The gradients are backpropagated through the student network to update its parameters θ_t. The gradients are not backpropagated through the teacher model, which instead is updated as EMA of the student model parameters with Eq. (1). Supervised data can be used along with unsupervised data during training.

3.2. Exponential Moving Average (EMA)

EMA is more commonly described by $\lambda = 1 - \alpha$, a decay factor. However, we find the discount factor α to be more intuitive to quantify the "distance" between the student model, which is also referred to as the online model, and its slow moving average (teacher model). The EMA model parameters can also be unrolled as a summation over student models after different number of updates, with each student model θ_i contributing with a weight w_i to the summation:

$$\begin{aligned}\xi_t &= \sum_{i \le t} w_i \theta_i \\ &\triangleq \alpha\theta_t + (1-\alpha)\alpha\theta_{t-\Delta} + \cdots + (1-\alpha)^n \alpha\theta_{t-n\Delta} + \ldots\end{aligned}$$

Another useful quantity is the half-life τ which is defined as:

$$w_{t-\tau} = \frac{w_t}{2} \qquad \text{or} \qquad \tau = -\Delta \frac{\ln 2}{\ln(1-\alpha)} \qquad (2)$$

Table 1 shows the half-lives for some α and Δ values.

λ	α	Δ	τ
0.99	0.01	1	69
0.999	0.001	1	693
0.9999	0.0001	1	6931
0.999	0.001	10	6928
0.975	0.0025	10	2769

Table 1. Half-lives for common values of α and Δ.

A larger α or equivalently a small half-life results in the teacher model being "too close" to the student model. This can encourage the model to produce targets that are easier for the model to predict and also leads to the model "collapse" so that the model starts to predict just silence or CTC `<blank>` token. For $\alpha = 1$ and $\Delta = 1$ this is consistent[1] with the observed divergence in [14] but is in contrast to the [13] where the authors were able to train the model successfully. However, our setup is significantly different from the setup in [13] because we have only 1-10 hours of supervised data.

A smaller α gives smaller weight for the recent student model which results in more stable training. However, the teacher model is more static and this can lead to worse performance. We find that for better performance and stable training the half-life should be at least 1000 or higher.

In one extreme of $\alpha = 0$, the teacher model is not updated at all. This is equivalent to the single-stage PL a.k.a. teacher-student training. In the other extreme of $\alpha = 1$, the teacher model is replaced with the student model every Δ updates. This is equivalent to iterative PL (IPL) [12]. Kaizen thus provides a generalized framework for semi-supervised training that encompasses both single-stage PL as well as IPL.

3.3. Training criteria

The Kaizen framework can be used with different training criteria and modeling paradigms. In this paper, we investigate two modeling paradigms.

Hybrid HMM-DNN Hybrid hidden Markov model – deep neural network (HMM-DNN) is the simplest paradigm where the neural network predicts context-dependent character (chenone) units [24] at the frame-level. Here, we train the student network to minimize the Kullback-Leibler divergence [25] between the teacher network's chenone posterior distribution $\hat{y}$ and student network's chenone posterior distribution y, $\mathcal{F}(\hat{y}, y; x) = \mathcal{D}(\hat{y} \mid\mid y)$. This is similar to the case of standard teacher-student training. In our work, we take the top-k posteriors from the teacher network to get at least 0.99 probability mass as done in [26].

CTC In this paradigm, the neural network is trained with the sequence-level criterion of CTC. We train the student network by minimizing the conditional probability of the token sequence $\hat{y}$ predicted by the teacher network, $\mathcal{F}(\hat{y}, y; x) = -\log p_\theta(\hat{y} \mid x)$. In this work, we use greedy decoding as in [13, 14] where the sequence $\hat{y}$ is obtained by de-duplicating the output label sequence of the teacher model and removing the `<blank>` labels. Alternatively, a beam-search decoding can be used to obtain $\hat{y}$. However, this is computationally more expensive and we did not try it in the current work.

3.4. Half-precision floating-point (fp16) training

When the models are in full-precision floating point (fp32) representation, the Kaizen framework is straightforward. However, when the models are trained with half-precision floating point (fp16) for efficiency, we found that it is critical that the EMA parameters are accumulated in fp32. This results in an extra copy of EMA parameters in fp32. Without this, there is a significant degradation relative to full fp32 training, and for some parameter settings it does no better than single generation of PL. This shows that high precision is essential to capture the small changes in the EMA model.

Note that the additional fp32 copy is only used for the EMA update step. After the update step of EMA parameters, it can be cast back to fp16 so that the forward pass through the teacher network is in fp16. This allows using 1.5 times larger batch size compared to fp32 without any loss in accuracy.

[1] [14] regularizes training via the dynamic cache while we do it via EMA.

4. EXPERIMENTS

4.1. Data Preparation

Public videos For training data, we use de-identified public videos with no personally identifiable information (PII) in UK English and Italian languages. In this paper, we simulate a low-resource scenario by limiting to a subset of 10h of supervised data, and a more extreme scenario with just 1h of supervised data in UK English. For both these languages, we use a much larger amount of unsupervised data consisting of 75k hours for UK English and 50k hours for Italian. As an upper-bound experiment, we compare with a supervised-only setting where we have 650h for UK English and 3,700h for Italian. The supervised data is augmented 3x with speed perturbation [27]. For evaluation, we use a 23h test set for UK English and 3 test sets for Italian – clean, noisy and extreme – which contain 24h, 24h and 45h of data, respectively. We use a separate 14h development set for hyper-parameter tuning for both UK English and Italian. For UK English, we use transcripts corresponding to 650h plus an additional 13k hours of generic English video transcripts for n-gram language model (LM) training. For Italian, we use the transcripts from the same 3.7k hours for n-gram LM training.

LibriSpeech We also perform experiments using Libri-Light data [16]: 10h labeled Libri-Light subset and 54k hours of unlabeled audio plus LibriSpeech [15] itself without labels. The standard LibriSpeech validation sets (*dev-clean* and *dev-other*) are used to tune all hyper-parameters, as well as to select the best models. Test sets (*test-clean* and *test-other*) are used only to report final word error rate (WER). All features are normalized to have zero mean and unit variance per input sequence before feeding them into the acoustic model. We report not only WER without an LM, but also WER obtained by a one-pass beam-search decoder [28] leveraging a 4-gram word-level LM [29] and further rescoring the beam of hypothesis with a strong Transformer LM [30], following the procedure from [30]. The LMs here were trained on the official LibriSpeech language modeling data [15].

To prepare public videos we use a time-delay neural network, bi-directional long short-term memory [31, 32]

(TDNN-BLSTM) model trained using flatstart lattice-free maximum mutual information (LF-MMI) [33, 34] on the 10h supervised data. We refer to this as the alignment model and it was used to align and segment the labeled data into 10s segments for training. The unsupervised data was pre-processed using a voice activity detection model to select only speech segments of 45s maximum duration. These segments were then decoded using the alignment model to produce machine generated transcriptions which are used as PLs for *burn-in* stage of training. The same hybrid TDNN-BLSTM LF-MMI model is trained on 10h labeled Libri-Light data and is used to generate *burn-in* PLs with the 4-gram $GB \setminus LV \setminus LS$ LM [12] for all LibriSpeech training data.

4.2. Model details

The input features to all the models are 80 dimensional Mel-scale log filterbank coefficients computed every 10ms over 25ms windows. Spectral masking (both frequency and time) is applied on-the-fly using SpecAugment except for the teacher model in Kaizen framework. The TDNN-BLSTM alignment model has 2 BLSTM [35] layers with 640 hidden units in each recurrence direction and 3 TDNN layers [36, 37] with 640 hidden units interleaved between input and first BLSTM layer, and between the 2 BLSTM layers. The modeling units are context-dependent bi-character units, each modeled with a 1-state HMM topology with state-tying done using context-dependency tree built using purely the text transcripts (no alignments) and silence inserted randomly between words as done in [38, 34].

Public videos We investigate two modeling paradigms – hybrid HMM-DNN and CTC. For hybrid HMM-DNN paradigm, the modeling units are context-dependent tri-character units, each modeled with a 1-state HMM topology with state-tying done using a context-dependency tree build using statistics from the frame-level character alignments produced by the alignment model. For CTC paradigm, we use sentence-piece [39] units. For both these paradigms, we use a 80M parameter neural network with a 2 VGG layers [40] followed by 12 Transformer blocks (768 hidden units, 8 heads) [41] following [42]. Each VGG layer sub-samples by 2 in the time-axes using max-pooling [43], resulting in the model that outputs at a rate of 25Hz (40ms time step).

LibriSpeech Here we investigate only CTC paradigm using English alphabet letters as modeling units. The neural network strictly follows [14]: a 1-D convolution with kernel size 7 and stride 3 followed by 36 4-head Transformer blocks (768 hidden units, 4 heads), resulting in a 270M parameter model that outputs at a rate of 33.(3)Hz (30ms time step).

4.3. Training details

For 1h/10h of public videos data, the hybrid TDNN-BLSTM LF-MMI trained alignment model has lower WER than the 12-layer Transformer model trained using either cross-entropy (CE) or CTC losses. For example, the dev results in Table 2 for 10h supervised with CTC paradigm is significantly worse than the hybrid model. Thus the alignment model also serves as the supervised baseline. For LibriSpeech a hybrid model also outperforms a CTC model on 10h of supervised data, however the gap is small, see Table 5.

For the semi-supervised experiments we use only unsupervised data for both *burn-in* and continuous PL stages, having in total 150k/200k updates for public videos and 500k for LibriSpeech. During the *burn-in* updates (25k for public videos and 80k for LibriSpeech), we use the PLs produced by the baseline model. For public videos, EMA is accumulated only after 15k updates. This was found to not affect the performance on Librispeech; hence on Librispeech EMA was accumulated from the beginning. After the *burn-in* updates, we switch to using PLs from continuously updated teacher model (continuous PL stage). We follow the continuous PL stage with a fine-tuning stage where the final student model[2] is fine-tuned on the supervised data only.

We use the Adam [44] and Adagrad [45] optimizer with mixed-precision [46] training and gradient norm clipping at 10 and 1 for public videos and LibriSpeech, respectively. For the supervised LF-MMI baseline, we use a learning rate that rises from 1.25e-6 to 1.25e-4 in 500 updates and then reduces by a factor of 0.5 when the valid loss improvement is less than 1e-4 relative. We use distributed data-parallel training with batch of 40min of audio distributed across 4 GPUs. For semi-supervised experiments with public videos, the total batch is 17.1h distributed across 64 GPUs and learning rate rises linearly for 7.5k updates to 1.5e-4 and decreases linearly to 0. For LibriSpeech, the total batch is 0.9h distributed across 16 GPUs and learning rate rises linearly for 64k updates to 0.03 and then is decayed by 2 once valid WER reaches the plateau. For fine-tuning stage, a learning rate rises linearly for 500 updates and decreases linearly until 10k updates and the number of GPUs is reduced to 2-4.

4.4. Results

4.4.1. Public videos

Tables 2, 3 and 4 show WER results (including LM decoding) comparing standard PL, Kaizen and IPL on 10h UK English, 1h UK English and 10h Italian public videos setups. For Kaizen, we used $\alpha = 0.0025, \Delta = 10$. For IPL, we used $\Delta = 1000$ for UK English and $\Delta = 2000$ for Italian. We also use a hybrid model trained on 10h or 1h of supervised data as the baseline. The WER reductions (WERR) are reported relative to this baseline for all the models. We also report performance of an upper-bound model that is trained on all the

[2]The final EMA teacher model is only slightly better than the final student model (0.2% absolute WER difference). Due to the small performance difference and for consistency with experiments not using EMA, we use the student model as the final model for fine-tuning.

supervised data that we have access to, i.e. 650h on UK English and 2.7k hours on Italian. On UK English setups, we show WER on dev and test sets. On Italian setup, we show WER on 3 test sets – clean, noisy and extreme.

Model	Paradigm	dev	test	WERR
10h sup	Hybrid	53.9	51.1	
10h sup	CTC	74.4	-	
650h sup	Hybrid	23.3	22.4	56.3
PL	Hybrid	30.2	29.8	41.7
Kaizen	Hybrid	27.3	26.8	47.6
PL	CTC	26.2	25.5	50.2
Kaizen	CTC	23.2	22.7	**55.5**
IPL	CTC	23.9	23.4	54.2

Table 2. WERs on 10h UK English setup with 75k hours of unsupervised data.

Model	Paradigm	dev	test	WERR
1h sup	Hybrid	81.1	79.9	
650h sup	Hybrid	23.3	22.4	72.0
PL	Hybrid	64.6	62.3	22.0
Kaizen	Hybrid	53.4	53.0	33.7
PL	CTC	55.3	54.8	31.4
Kaizen	CTC	37.2	35.3	**55.8**
IPL	CTC	37.9	36.7	54.1

Table 3. WERs on 1hr UK English setup with 75k hours of unsupervised data.

Model	clean	noisy	extreme	WERR
10h sup Hybrid	39.7	43.9	60.4	
2700h sup	9.3	11.8	17.2	73.8
PL	13.2	17.2	26.3	61.4
Kaizen	11.5	14.6	21.8	**67.2**
IPL	11.5	14.6	21.8	**67.2**

Table 4. WERs on 10hr Italian setup with 50k hours of unsupervised data. If not stated all models are CTC-based.

We can see from the results that Kaizen outperforms PL by more than 10% relative on all the setups for both hybrid HMM-DNN and CTC paradigms while Kaizen is similar to or slightly better than IPL. On both UK English and Italian languages, using just 10h of supervised data and a large amount of unsupervised data we close the gap to the upper-bound ASR system that uses 650h or 2.7k hours, respectively.

4.4.2. LibriSpeech

Table 5 shows WER results comparing Kaizen and other methods. We use a hybrid model and a CTC model trained on 10h of supervised data only as the baseline. Kaizen ($\alpha = 10^{-4}, \Delta = 1$) demonstrates similar performance to slimIPL [14] (cache $= 1000$, $p = 0.1$), while a combination of slimIPL and Kaizen together ($\alpha = 10^{-3}, \Delta = 1,$ cache $=$ 1000, $p = 0.1$) achieves better performance than individual approaches. Moreover, the combination achieves a new state-of-the-art result for the greedy decoding and almost closes the gap with state-of-the-art self-supervised methods [47, 48] for decoding with a language model.

Model	LM	dev clean	dev other	test clean	test other
10h sup Hybrid	4-gram	15.9	37.2	16.6	38.2
	$GB \setminus LV \setminus LS$	15.1	36.3	15.9	37.1
10h sup [14]	4-gram	18.8	39.3	19.6	39.7
w2v 2.0 [47]	-	6.3	9.8	6.3	10.0
	Transformer	2.4	4.8	2.6	4.9
HUBERT [48]	-	6.8	9.6	6.7	9.9
	Transformer	2.2	4.3	2.4	4.6
slimIPL	-	5.5	9.4	5.6	9.9
	Transformer	2.6	5.4	3.2	6.1
Kaizen	-	5.4	9.5	5.5	10.1
	Transformer	2.5	5.3	3.0	6.0
Kaizen+slimIPL	-	5.1	8.2	5.1	8.8
	Transformer	2.4	4.9	2.9	5.5

Table 5. LibriSpeech WERs for supervised baselines and different semi/self-supervised methods trained on Libri-Light, 10h labeled and 54k hours unlabeled data. If not stated all models are CTC-based.

4.4.3. Effect of EMA parameters

In this section, we study the effect of two EMA parameters – the discount factor α and update frequency Δ. We do this investigation on the UK English videos dataset in the hybrid HMM-DNN paradigm. The stability of training depends on the distance between teacher and student models, which for Kaizen is quantified using half-life, Eq. (2).

The plots in Figures 2, 3 and 4 show the WER on UK English 10 hours supervised setup (fine-tuning stage is not included) as a function of number of training hours for various training runs. For each training run, the point where the model switches to using the continuously generated PLs is marked with a solid circle.

Figure 2 shows various training runs with $\Delta = 1$ and $\alpha \in \{0.1, 0.01, 0.001, 0.0001\}$. We see that the model diverges very quickly when $\alpha = 0.1$ ($\tau = 7$). The model training gets more stable progressively as we increase the α value towards the most stable 0.0001 ($\tau = 6931$).

Figure 3 demonstrates the effect of EMA update frequency Δ. For $\alpha = 0.001$, there is divergence with $\Delta = 1$ ($\tau = 693$), but the training is stable and WER improves continuously with $\Delta = 10$ ($\tau = 6928$). For higher value of α like 0.1 or 0.25 where half-life is less than 10 if $\Delta = 1$, the training diverges almost immediately as seen for $\alpha = 0.1, \Delta = 1$.

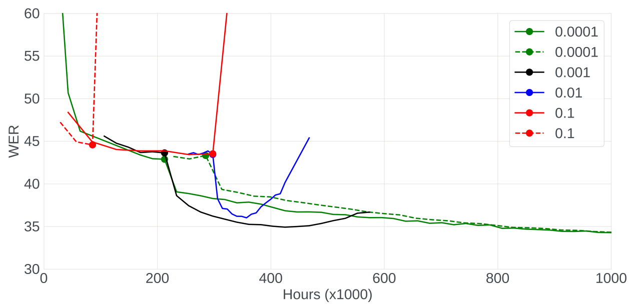

Fig. 2. Effect of EMA discount factor α.

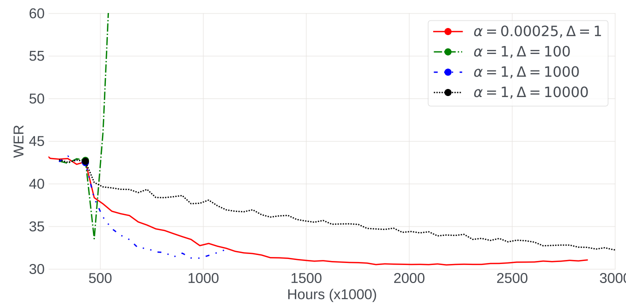

Fig. 4. Comparing IPL and Kaizen.

But even with such α, the training is stable if Δ is increased to 1000 as seen for $\alpha = 0.25, \Delta = 1000$ ($\tau = 2409$). We find that the post-fine-tuning model performance is also similar for similar half-life values.

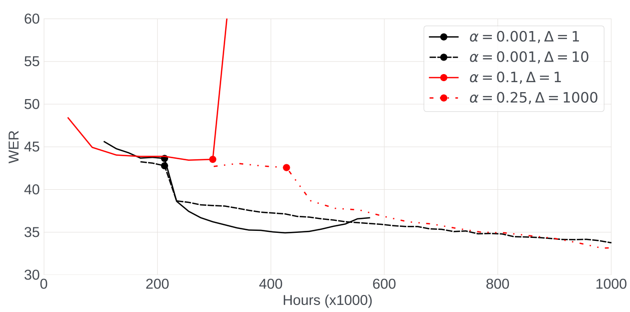

Fig. 3. Effect of EMA update frequency Δ.

Figure 4 compares the basic Kaizen case of $\alpha = 0.00025, \Delta = 1$ with IPL ($\alpha = 1, \Delta \in \{100, 1000, 10000\}$). We see that with $\Delta = 100$, IPL diverges very soon after switching to using continuously generated PLs. Increasing Δ stabilizes it as seen with $\Delta = 1000$ where the divergence happens only after training on 1M hours. Using a much larger Δ value of 10000 (for batch size of 17.1h and dataset of 75k hours, this is 171k hours = 2.8 epochs), the model trains stably but improves more slowly. Using typical Kaizen parameters of $\alpha = 0.00025, \Delta = 1$ ($\tau = 2772$), the training is stable while also showing better WER after 2M hours.

These results show that the model training is not stable unless the distance between teacher and student models is sufficiently large (half-life of more than 2000). Smaller distances i.e. smaller half-lives lead to "collapse" and WER degrades rapidly. In particular, we find that for updating the model continuously $\Delta = 1$ as in [13] requires a small EMA discount factor to discount most recent student models. We also tried to mix-in some supervised data such that 10% of data in each epoch is supervised. This did not help stability. We hypothesize that this is partly due to our supervised dataset being very small in the order of 1-10h. Further experiments with larger supervised datasets are needed in the future to confirm this.

5. CONCLUSIONS AND FUTURE WORK

We introduce the Kaizen framework for semi-supervised training that uses a continuously improving teacher model to generate pseudo-labels. The teacher model is updated as the exponential moving average of the student model. The proposed framework is shown as a generalization of PL and IPL. We analyzed the effect of the EMA parameters and showed that the distance between the teacher and student models is the key for effective and stable training. A small EMA half-life leads to collapse of the model and poor performance, while too large half-life leads to slow improvement. We showed that the proposed approach gives more than 10% WERR over standard teacher-student training and performs comparatively to IPL on public videos dataset in UK English and Italian languages. We also demonstrated that Kaizen can be combined with slimIPL to achieve new state-of-the-art result for the greedy decoding and further close the gap with state-of-the-art self-supervised methods for decoding with an LM on LibriSpeech with 10h of labeled and 54k unlabeled data.

5.1. Future Work

This work has explored Kaizen for Hybrid HMM-DNN and CTC based models, and we plan to explore this further for sequence-to-sequence models like RNN-T. Preliminary experiments also show that scheduling EMA parameters is promising. Using larger discount factor in the beginning of training allows the teacher to forget the history and benefit from the fast improving student in the beginning. The discount factor can be later reduced to make the training more stable. While current work has shown applicability in low-resource scenario, we plan to further expand it to higher resource settings that have 100s to 1000s of hours of supervised data. The proposed approach naturally fits into online training of ASR models.

6. REFERENCES

[1] George Zavaliagkos and Thomas Colthurst, "Utilizing untranscribed training data to improve performance," in

DARPA Broadcast News Transcription and Understanding Workshop, 1998.

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

[3] Jacob Kahn, Ann Lee, and Awni Hannun, "Self-training for end-to-end speech recognition," in *ICASSP*, 2020.

[4] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD*, 2006.

[5] Lei Jimmy Ba and Rich Caruana, "Do deep nets really need to be deep?," in *NIPS*, Cambridge, MA, USA, 2014, MIT Press.

[6] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[7] Jeremy HM Wong and Mark Gales, "Sequence student-teacher training of deep neural networks," in *Interspeech*, 2016.

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

[9] Wei-Ning Hsu, Ann Lee, Gabriel Synnaeve, and Awni Hannun, "Semi-supervised speech recognition via local prior matching," *arXiv preprint arXiv:2002.10336*, 2020.

[10] Frank Wessel and Hermann Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 23–31, 2004.

[11] 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," in *Interspeech*, 2020.

[12] Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert, "Iterative pseudo-labeling for speech recognition," in *Interspeech*, 2020.

[13] Yang Chen, Weiran Wang, and Chao Wang, "Semi-supervised ASR by End-to-end Self-training," in *Interspeech*, 2020.

[14] Tatiana Likhomanenko, Qiantong Xu, Jacob Kahn, Gabriel Synnaeve, and Ronan Collobert, "slimipl: Language-model-free iterative pseudo-labeling," in *Interspeech*, 2021.

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

[16] Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, et al., "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP*, 2020.

[17] Samuli Laine and Timo Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[18] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[19] Alex Graves, Santiago Fernández, and Faustino Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[20] Jean-Bastien Grill et al., "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 21271–21284.

[21] Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, and Takaaki Hori, "Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition," 2021.

[22] 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.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] Duc Le, Xiaohui Zhang, Weiyi Zheng, Christian Fügen, Geoffrey Zweig, and Michael L Seltzer, "From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition," *ASRU*, 2019.

[25] Solomon Kullback and Richard A Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[26] Kritika Singh, Vimal Manohar, et al., "Large scale weakly and semi-supervised learning for low-resource video asr," in *Interspeech*, 2020.

[27] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition.," in *Interspeech*, 2015.

[28] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[29] Tatiana Likhomanenko, Gabriel Synnaeve, and Ronan Collobert, "Who needs words? lexicon-free speech recognition," *Proc. Interspeech 2019*, pp. 3915–3919, 2019.

[30] Gabriel Synnaeve et al., "End-to-end asr: from supervised to semi-supervised learning with modern architectures," in *SAS workshop ICML*, 2020.

[31] Gaofeng Cheng, Vijayaditya Peddinti, Daniel Povey, Vimal Manohar, Sanjeev Khudanpur, and Yonghong Yan, "An exploration of dropout with lstms.," in *Interspeech*, 2017, pp. 1586–1590.

[32] Vijayaditya Peddinti et al., *Low latency modeling of temporal contexts for speech recognition*, Ph.D. thesis, Johns Hopkins University, 2017.

[33] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Interspeech*, 2016.

[34] Xiaohui Zhang, Vimal Manohar, David Zhang, et al., "On lattice-free boosted MMI training of HMM and CTC-based full-context ASR models," *arXiv preprint arXiv:2107.04154*, 2021.

[35] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] Kevin J Lang, Alex H Waibel, and Geoffrey E Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural networks*, vol. 3, no. 1, pp. 23–43, 1990.

[37] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015.

[38] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "End-to-end Speech Recognition Using Lattice-free MMI," in *Interspeech*, 2018, pp. 12–16.

[39] Taku Kudo and John Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *EMNLP*, 2018.

[40] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[42] Yongqiang Wang, Abdelrahman Mohamed, Duc Le, Chunxi Liu, Alex Xiao, et al., "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP*, 2020.

[43] Kouichi Yamaguchi, Kenji Sakamoto, Toshio Akabane, and Yoshiji Fujimoto, "A neural network for speaker-independent isolated word recognition," in *First International Conference on Spoken Language Processing*, 1990.

[44] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *ICLR*, 2014.

[45] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[46] Sharan Narang, Gregory Diamos, Erich Elsen, Paulius Micikevicius, Jonah Alben, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al., "Mixed precision training," in *Int. Conf. on Learning Representation*, 2017.

[47] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[48] 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.