Using Data Augmentation and Consistency Regularization to Improve Semi-supervised Speech Recognition

Ashtosh Sapru

Amazon Alexa, Bangalore, India

sapru@amazon.com

Abstract

State-of-the-art automatic speech recognition (ASR) networks use attention mechanism and optimize transducer loss on labeled acoustic data. Recently, Semi-Supervised Learning (SSL) techniques that leverage large amount of unlabeled data have become an active area of interest to improve the performance of ASR networks. In this paper we approach SSL based on the framework of consistency regularization, where data augmentation transforms are used to make ASR network predictions invariant to perturbations in the acoustic data. To increase data diversity we present a combination technique that randomly fuses multiple waveform and feature transforms. For each unlabeled acoustic waveform, two versions, i.e., a weakly augmented and a strongly augmented version of the unaugmented input are generated. During training, a semi-supervised loss is assigned that enforces consistent outputs between the weak and strong augmentations of the unlabeled input. Moreover, we employ model averaging technique to generate stable outputs over time. We compare and demonstrate the benefits of the proposed approach against standard SSL strategies like iterative self-labeling. We leverage over 100000 hours of unlabeled data to train the ASR network using streaming transducer loss and reach improvements in the range of 8%-12% over self-labeling baseline.

Index Terms: Semi-Supervised Learning, Consistency Regularization, Data Augmentation, Transducer Loss

1. Introduction

State-of-the-art ASR models typically comprise tens to hundreds of millions of model parameters and are expressed as large deep neural networks enhanced with attention mechanism. ASR models inspired from Transformer [1] architecture, such as, Conformers [2] capture the time varying nature of speech using self attention mechanism to express the highly complex mapping between the input acoustic sequence and the output word sequence. These models are trained explicitly in End-to-End (E2E) manner using either transducer loss [3] or auto-regressive loss [4]. Unlike the dependence on external lexicon and finite state transducers in Hybrid ASR techniques [5], E2E-ASR models do not require an external alignment model for training [6]. However, the alignment free nature of these models necessitates substantial amount of labeled data to train these models effectively and to get the best performance out of them [7, 8].

Manually transcribing large amounts of acoustic data for every desired condition is both time consuming and expensive. Moreover, to ensure end user privacy, imposes further restrictions on availability of ever increasing labeled training data. Recently, a lot of interest in ASR has focused on applying SSL [9, 10, 11]. SSL is halfway between supervised and unsupervised learning, but similar to supervised learning, SSL also estimates a mapping from feature space to label space and it leverages large amount of unlabeled data to learn that mapping [12]. A prerequisite for applying SSL techniques is having access to limited but sufficient amount of labeled data.

In self-labeling or self-training based SSL methods, labeled data is used for building initial seed models, which are then used to iteratively decode large quantities of unlabeled data and subsequently reliable hypotheses (machine-labeled data) are selected based on confidence scores for ASR training [13, 14, 15, 16]. Self-labeling can introduce confirmation bias, though data augmentation methods have been applied to mitigate this bias. More recently, self-supervised representation learning techniques that optimize predictive and contrastive loss have been used to improve ASR performance [17, 18, 19, 20].

SSL based on knowledge distillation (KD) [21] from a teacher model to a student model has become popular for ASR model training [10, 22, 23]. In KD, the teacher model is often trained on labeled data alone, and subsequently this model is used to generate pseudo labels on a much larger volume of unlabeled data. Given a well-trained teacher model unlabeled data can significantly improve the performance of the student model. However, the KD approach incurs an additional cost in training and inference due to an auxiliary and much larger teacher model. In parallel with KD, SSL techniques based on consistency regularization (CR) have become popular in image domain [24, 25, 26]. Although relatively less studied, CR using data augmentation has recently been explored for ASR [27, 28, 29].

CR relies on the principle that when a realistic perturbation is applied to a model's input then its prediction should not diverge. The success of CR is therefore related to the quality and diversity of input perturbations. In this work, we have explored multiple waveform and feature transforms to augment acoustic data. In parallel we apply model averaging technique [30] to improve quality of inferred pseudo labels. Inspired by work in image domain [26], we adopt the simple strategy of randomly combining multiple acoustic transforms to generate a strong augmentation of unlabeled audio. Similar to [25], we infer pseudo labels from weak augmentation of data and simultaneously enforce consistency with the corresponding strong augmentation. We use state-of-the-art conformer [31] architecture to validate our approach on both cross entropy (CE) pretrained model and E2E model trained with transducer loss. Compared to [27], this work does not depend on an auxiliary teacher model to enforce consistency. In contrast to [27, 28], we do not limit data augmentation to Spec Augment [32], but amplify acoustic diversity with multiple augmentation techniques. Also unlike [27, 28, 29], this work applies model averaging to stabilize the inferred pseudo labels.

The rest of paper is organized as follows. Section 2 presents the approach used in this work for training CE and E2E ASR networks using CR. Section 3 describes the details of E2E network, experimental setup used in this study and results. Finally, Section 4 concludes our work.

2. Consistency Regularization for ASR

In SSL the main assumption behind using unlabeled data to improve classification is that the decision boundary between classes lies in low density regions. Consistency regularization exploits this idea by applying realistic perturbations to the unlabeled input data. If we make the reasonable assumption that unlabeled data is likely to have been sampled from high density regions of input distribution and by SSL assumption is also far from the decision boundary, then by applying a realistic perturbation to the unlabeled data, the model prediction should not diverge.

More formally, we assume a dataset of paired labeled examples $\mathcal{D}_L = \{(X_1, Y_1), ..., (X_L, Y_L)\}$. For any $(X_l, Y_l) \in \mathcal{D}_L$ we maximize the likelihood of ground-truth transcription Y_l given the input utterance X_l and the stochastic model F parameterized by θ. We also assume access to a dataset of unlabeled examples $\mathcal{D}_U = \{X_{L+1}, ..., X_{L+U}\}$. For unlabeled $X_u \in \mathcal{D}_U$ we transform it by applying data augmentation transform $\mathcal{T}$ to create a perturbed version $\widehat{X_u} = \mathcal{T}(X_u)$. In consistency regularization, the unlabeled objective is to minimize the distance D between the two model outputs $D(F(X_u; \theta), F(\widehat{X_u}; \theta))$. Supervised log likelihood loss is combined with unsupervised consistency loss by a weighting factor w to give the joint loss:

$$\mathcal{L} = - \sum_{(X_l, Y_l) \in \mathcal{D}_L} \log P(Y_l | X_l, \theta) + w \sum_{X_u \in \mathcal{D}_U} D(F(X_u), F(\widehat{X_u})) \tag{1}$$

A standard choice of distance function D is Kullback–Leibler (KL) divergence between posterior distributions of unlabeled data and its augmentation.

2.1. Proposed Approach

In this work, E2E ASR model is a sequence transducer consisting of a Conformer encoder, a LSTM prediction network and a joint network. The Conformer encoder F^e encodes at time t, each acoustic feature $\mathbf{x}_t$ into a hidden representation $\mathbf{h}_t$. The prediction network F^p maps a output token into another hidden representation $\mathbf{g}_i$, where i is the index of the output token labels. The joint network F^j fuses information from both speech encoder and prediction network to compute the posterior probability of next token or blank. The E2E transducer loss is defined as sum of posterior probabilities over all consistent input output alignments.

Its natural to consider KL divergence, coming directly from output of F^j, as the distance function in (1). However, prediction network tends to produce spiky posteriors and augmentation of input features, such as time warping, can cause these posteriors to spike at different positions in the posterior lattice. Instead, our approach is based on combining consistency regularization with self labeling in a single framework. Our motivation comes from success of applying similar strategy in computer vision tasks [25]. The key idea is to simultaneously perform both inference and training on unlabeled data. First, we infer pseudo labels on a weakly augmented version of the input unlabeled example $\tilde{X}_u$ as follows:

$$\tilde{Y}_u = \underset{Y_u}{\operatorname{argmax}} \log P(Y_u | \tilde{X}_u, \theta) \tag{2}$$

Similar to (1), we then enforce consistency by maximizing the likelihood with respect to the inferred pseudo label $\tilde{Y}_u$ on a strongly augmented version of the unlabeled example $\widehat{X}_u$. The overall loss is given as:

$$\mathcal{L} = - \sum_{X_l, Y_l} \log P(Y_l | X_l, \theta) - w \sum_{X_u} \log P(\tilde{Y}_u | \widehat{X}_u, \theta) \tag{3}$$

2.1.1. Model Averaging

In (2) and (3), the same model with parameters θ is used to generate predictions for both weak and strong augmentations of data. It is possible that during training the model will make errors in those predictions for some unlabeled batches. Those errors can in turn then get reinforced due to enforced consistency. To overcome this we can use a separate model for inferring the pseudo labels. In this work, we apply the mean teacher method proposed in [30] for generating the pseudo labels $\tilde{Y}_u$. Two identical networks called teacher and student are maintained throughout training. The parameters of the teacher model θ' are defined as the exponential moving average of the parameters of the student model θ. At each training iteration τ those parameters are updated as:

$$\theta'_\tau = \alpha \theta'_{\tau-1} + (1 - \alpha) \theta_\tau \tag{4}$$

Instead of inferring $\tilde{Y}_u$ using student model θ, we now use the mean teacher model θ' at each training step τ to infer $\tilde{Y}_u$:

$$\tilde{Y}_u = \underset{Y_u}{\operatorname{argmax}} \log P(Y_u | \tilde{X}_u, \theta') \tag{5}$$

The overall loss is till defined by (3), the only change is that we infer more stable pseudo labels $\tilde{Y}_u$ using averaged model with parameters θ'.

2.1.2. Data Augmentation

The data augmentation methods considered in this work are summarized now:

- Pitch shift: The pitch of the raw audio waveform is raised or lowered without affecting the tempo. We randomly sample $n \in [-6, 6]$ and randomly shift the pitch by n semitones.
- Background Noise: We mixed background environmental noise from diverse sources such as, cafe, inside pub, music, inside office and other domestic scenarios to the input audio. A random SNR was sampled in range of 0 to 20 dB as target SNR for mixed audio.
- Reverberations: Audio is convolved with a randomly-generated room impulse response (RIR). We sampled RIRs from an inhouse database containing more than a million of those with a T60 cutoff of 400ms.
- SpecAugment [32]: Time frequency masking is applied to the spectrogram of the input audio. Similar to [32], we randomly sample widths of time masks and frequency masks, the number of time masks, and the number of frequency masks.
- Input Mixup [33]: This transform is also applied at feature level by creating a convex combination of a batch of input features with a shuffled version of the same batch. The mixing factor is sampled from a beta distribution with shape α set to 0.3. In contrast to [33], we did not apply mixup to the targets.

2.1.3. *Method*

We perform the training in two stages: in the first stage we add a logit layer on top of the encoder and pretrain the encoder against frame level targets. We follow the approach detailed in [34] to generate frame level targets $E_{l,t}$ for labeled data. For unlabeled data we infer frame level pseudo labels $\tilde{E}_{u,t}$. During this stage, the overall utterance level log likelihood is accumulated across frame level cross entropy loss for both supervised and consistency regularization terms in (3). Subsequently, we seed the second stage with pretrained encoder and compute utterance level E2E transducer loss from the output of joint network for both labeled and unlabeled data. We denote the set of weak augmentations by $\mathcal{A}_W$ and the set of strong augmentations by $\mathcal{A}_S$, the main difference between them is that $\mathcal{A}_W$ and $\mathcal{A}_S$ are parameterized with different parameters. We also assign a fixed selection probability $p_i \in [0, 1]$ for each transform $\mathcal{T}_i$ sampled from $\mathcal{A}_S$. Summary of our approach is describe now:

* Pretraining Stage:

1. Sample $\mathcal{T}_W \sim \mathcal{A}_\mathcal{W}$
2. Compute frame level pseudo labels at time t for feature $X_{u,t}$ as $\tilde{E}_{u,t} = \text{argmax}\, F^e(\mathcal{T}_W(X_{u,t}))$
3. Sample k different augmentations $\mathcal{T}_1, ..., \mathcal{T}_k \sim \mathcal{A}_S$.
4. For $i = 1...k$, apply $\mathcal{T}_i$ such that,

$$\mathcal{T}_i(X) = \begin{cases} X, & \text{if } p_i < q_i \sim U(0,1) \\ \mathcal{T}_i(X), & \text{otherwise} \end{cases}$$

5. Compose strong augmentation $\hat{X}_u = \mathcal{T}_1, ..., \mathcal{T}_k(X_u)$
6. Use cross-entropy loss and, frame level targets $E_{l,t}$ and $\tilde{E}_{u,t}$ in (3), to pretrain the encoder by minimizing the sum of supervised and consistency loss.

* E2E Stage:

7. Apply Step 1. above to get weakly augmented feature $\tilde{X}_u$.
8. Using $\tilde{X}_u$ perform beam search at the output of F^j to find the best pseudo label sequence $\tilde{Y}_u$
9. Apply Step 3. to Step 5. above to get strongly augmented feature $\hat{X}_u$
10. Use E2E transducer loss and, sequence labels Y_l and $\tilde{Y}_u$ in (3), to minimize the sum of supervised and consistency loss.

3. Experiments

3.1. Model Architecture

In this work we have implemented a ASR architecture composed of a CNN preprocessor, Conformer encoder, LSTM prediction network and joint network that adds the outputs from the encoder and prediction network. CNN preprocessor produces a time-frequency embedding of input acoustic features and the conformer encoder models the long term temporal dependence in speech using self-attention mechanism. Conformer encoder is stack of 14 conformer blocks, with each block composed of several layers, such as: self-attention, depth-wise convolution, batch normalization and feedforward layers. Throughout this work we have used causal convolutions and self attention limited to left context to achieve streaming behavior. The architectural details of the model are show in Table 1.

Table 1: *Model architecture and setup*

Feature representation	3 * 64 dimensional LFBE Features
Label representation	4000 Word Pieces (Plus Blank Symbol)
Feature Embedding	CNN: Layers = 2, Kernel = 3x3, Stride Layer 1= 2 Stride Layer 2 = 1
Encoder architecture	Conformer Block : Layer = 14, Kernel = 15, Attention Heads = 8, Encoder Dim = 512, FeedForward Dim = 1024
Decoder architecture	LSTM: Unidirectional, Layers = 2, Units = 1024
Labeled data	2000 hours
Unlabeled data	$\sim$ 100000 hours

3.2. Experimental Details

3.2.1. *Data selection*

Experiments were performed using de-identifed data drawn from Alexa family of devices. For our experiments utterances were filtered out using criteria, such as confidence, wakeword and we also cutoff utterances with common lexical content that occurs with high frequency. Almost 100000 hours of unlabeled and de-identifed data from British English locale was sampled using this data selection technique. The labeled data used in all our experiments is an in-house de-identifed British English dataset of 2000 hours. For evaluation we used a test set with 20 hours of speech, containing both clean and noisy far field conditions. We also included a rare words test set, that includes words that occur very infrequently in the training set. We set a cutoff of 10% for word frequency occurrence to compile the rare words test set. We report results as relative Word Error Rate Reduction (WERR) to compare the proposed method against supervised baselines.

3.2.2. *Experimental Setup*

The input log Mel-filterbank energy (LFBE) features are normalized using global mean-variance statistics of the training pool and we employ a frame skipping approach where three consecutive frames are stacked to obtain a 192 dimensional features. We used byte pair encoding algorithm [35] to generate 4000 word-pieces (WP) as subword units. The frame level targets for CE pretraining were generated by force aligning WP boundaries using the approach described in [34].

Distributed training was used to train both cross-entropy (CE) pretrained and E2E models. The networks were initialized with a learning rate of $4e^{-4}$ and Adam optimizer was used as the Stochastic Gradient Descent (SGD) algorithm. The weighting factor w in (3), is set to 1.0 during training, except for a warm-up phase of first 5000 SGD steps where it is set to 0. In the initial CE pretraining phase we added a logit layer to the Conformer encoder. The seed model for E2E stage was then initialized by stripping off the logit layer from the CE trained encoder.

3.3. Results

We use iterative self-labeling as the baseline method for comparison to the proposed approach. We follow a training strategy similar to that proposed in [15]. The seed model for self-labeling is first pretrained using cross-entropy training followed by end-to-end training using transducer loss on labeled data. Following [15], we generate the pseudo labels on the unlabeled data after every 10 epochs and augment the training set with a mix of both labeled and unlabeled data to retrain the self-labeling model. Data augmentation based on SpecAugment (SA) is applied during training of the self-labeling baseline.

We compare the baseline against proposed CR methods detailed in Section 2.1.3. Similar to unsupervised baseline, for all CR models we used SA as the weak augmentation method. The weak augmentation parameters were: number of frequency masks $n_F = 2$, number of time masks $n_T = 1$ and frequency width as 20% of the spectral range. The order of applied augmentation is time-domain transforms techniques; Pitch-Shift, Noise and Reverb; followed by feature augmentation methods. For strongly augmented SA, we set $n_F = 2$ and $n_T = 3$ and increased the width of frequency masks to 25%. In the random combination approach we apply SA along with augmentation techniques detailed in Section 2.1.2. Each augmentation technique is selected with equal probability and composed according to Section 2.1.3

The pretrained supervised Conformer encoder is used as seed to initialized the supervised baseline system trained with transducer loss. Similarly, pretrained CR model is used as seed to initialize E2E training with unlabeled data. Spec Augment is applied to supervised model during training. During the pretraining phase for random augmentation (RA) the probability of selecting any base transform is set to 0.25. However, we amplify strong augmentation by increasing probability of selection for each transform to 0.5 during the E2E stage. We give equal weight to both labeleded and unlabeled loss and set the weighting factor w to 1.0 In Table 2 we present the results of CE pretraining the conformer encoder. The second and third columns of the table lists augmentation techniques applied to labeled and unlabeled data. At the CE stage, we compare all the models against supervised baseline trained with SA (first row). Applying SA for both labeled and unlabeled data in model (Vanilla CR) we see improvement in performance due to CR. Finally, the system trained by randomly combining multiple random augmentation techniques (Random CR) shows the best relative improvement on the evaluation test set. We present the results of training the model with E2E transducer loss in Table 3. All the systems are compared against self-labeling baseline. Unsurprisingly, we observe that the supervised system trained using labeled data alone, degrades by almost $7\% - 20\%$ compared to baseline. We also observe significant difference in performance of the supervised model due to effects of CE pretraining.

Table 2: *WER reduction for CE pretrained models on* 100000 *hours of unlabeled audio and* 2000 *hours of labeled audio. Compared to baseline negative WERR means degradation in performance and Positive WERR means improvement.*

Method	Labeled Aug.	Unlabeled Strong Aug.	Test WERR(%)
Supervised Baseline	SA	-	0.00
Vanilla CR	SA	SA	6.53
Random CR	SA	RA	**8.60**

Table 3: *WER reduction for models E2E (transducer loss) trained on* 100000 *hours of unlabeled audio and* 2000 *hours of labeled audio. Transforms applied in training include: 1.) No Augmentation (NA); 2.) Spec Augment (SA); 3.) Randomly Combined Augmentation (RA). Were applicable only SA was applied as weak augmentation. Random-MA applies model averaging. All the models were CE pretrained, except those indicated by (NP).*

Method	Labeled Aug.	Unlabeled Strong Aug.	Test WERR(%)	Rare Words WERR(%)
Self-labeling	SA	-	0.00	0.00
Supervised (NP)	SA	-	-28.27	-46.90
Supervised	SA	SA	-7.22	-19.86
Vanilla CR	SA	SA	4.11	0.77
Random CR	SA	RA	8.53	8.26
Random-MA CR	SA	RA	**9.16**	**12.32**

Table 4: *Comparing the difference in performance due to different distance measures in CR: 1.) transducer loss computed from Pseudo Labels (PL), 2.) L2 distance and 3.) Cosine distance.*

Method	Transducer (PL)	MSE	Cosine
Test (WERR %)	0.0	-6.29	-2.89
Rare Words (WERR %)	0.0	-8.89	-5.03

There is significant drop in performance of unpretrained supervised model compared to its pretrained counterpart. Comparing baseline against Vanilla CR based on applying only SA as the data augmentation method shows improvements of 4.77% on the head test set and comparable performance on rare words test set. By amplifying acoustic diversity using the proposed random augmentation technique (Random CR) we improve significantly by almost 8% on both the test sets. Finally Random-MA trained by applying random augmentation on the input side and model averaged mean teacher for pseudo label generation shows the best overall performance.

In Table 4, we investigate the difference in performance of CR when instead of inferring the pseudo labels and then applying transducer loss, we use alternative distance measure D in equation (1). The labeled data is still optimized with transduer loss, while L2 distance or cosine distance are computed between the outputs of encoder when the proposed weak and strong data augmentation is applied to unlabeled data. We observe significant drop in performance for both of L2 and cosine distance measures, indicating that pseudo labels are more effective output representations for applying CR to E2E networks.

4. Conclusions

In this work we investigated the impact of consistency regularization for improving E2E ASR models. In particular, we demonstrated that stable pseudo labels inferred during training themselves are effective in enforcing the consistency between different augmented views of the acoustic data. This makes our approach directly applicable to streaming ASR models trained with transducer loss. Furthermore, we explored multiple waveform and feature augmentations and proposed a method based on random combination of these augmentations. We established that by leveraging unlabeled data at scale, proposed method outperforms both self-labeled baseline, as well as, CR method based on SpecAugment. We demonstrated these improvements during both CE pretraining stage and E2E transducer loss training stage, obtaining WERR improvements in the range of $8 - 12\%$.

5. References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” in *Interspeech 2020*. ISCA, 2020, pp. 5036–5040.

[3] Y. Zhu, P. Haghani, A. Tripathi, B. Ramabhadran, B. Farris, H. Xu, H. Lu, H. Sak, I. Leal, N. Gaur, P. J. Moreno, and Q. Zhang, “Multilingual speech recognition with self-attention structured parameterization,” in *Interspeech 2020*. ISCA, 2020, pp. 4741–4745.

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

[5] H. Sak, A. Senior, K. Rao, O. İrsoy, A. Graves, F. Beaufays, and J. Schalkwyk, “Learning acoustic frame labeling for speech recognition with recurrent neural networks,” in *ICASSP 2015*, 2015, pp. 4280–4284.

[6] A. Graves, “Sequence transduction with recurrent neural networks,” *ICML Workshop*, vol. abs/1211.3711, 2012.

[7] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” in *ICASSP 2018*. IEEE, 2018, pp. 4774–4778.

[8] W. Chan, D. S. Park, C. Lee, Y. Zhang, Q. V. Le, and M. Norouzi, “Speechstew: Simply mix all available speech recognition data to train one large neural network,” *CoRR*, vol. abs/2104.02133, 2021.

[9] Y. Huang, Y. Wang, and Y. Gong, “Semi-supervised training in deep learning acoustic model,” in *Interspeech*, 2016, pp. 133 615–133 627.

[10] S. H. Krishnan Parthasarathi and N. Strom, “Lessons from building acoustic models with a million hours of speech,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6670–6674.

[11] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition.” *CoRR*, vol. abs/2010.10504, 2020.

[12] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[13] T. Kemp and A. Waibel, “Unsupervised training of a speech recognizer: Recent experiments,” in *in Proc. Eurospeech*, 1999, pp. 2725–2728.

[14] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, “Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning,” in *AAAI, 2021*, 2021, pp. 6912–6920.

[15] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” in *Interspeech 2020*, 2020, pp. 1006–1010.

[16] Y. Chen, W. Wang, and C. Wang, “Semi-supervised asr by end-to-end self-training,” in *Interspeech 2020*. ISCA, 2020.

[17] A. Baevski and A. Mohamed, “Effectiveness of self-supervised pre-training for asr,” in *ICASSP 2020*, 2020.

[18] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” in *Interspeech 2019*, 2019.

[19] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[20] W. Huang, Z. Zhang, Y. T. Yeung, X. Jiang, and Q. Liu, “SPIRAL: Self-supervised perturbation-invariant representation learning for speech pre-training,” in *International Conference on Learning Representations*, 2022.

[21] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[22] P. Swarup, D. Chakrabarty, A. Sapru, H. Tulsiani, H. Arsikere, and S. Garimella, “Efficient large scale semi-supervised learning for CTC based acoustic models,” in *IEEE Spoken Language Technology Workshop, SLT 2021*. IEEE, 2021, pp. 148–155.

[23] G. Kurata and G. Saon, “Knowledge distillation from offline to streaming RNN transducer for end-to-end speech recognition,” in *Interspeech 2020*. ISCA, 2020, pp. 2117–2121.

[24] Q. Xie, Z. Dai, E. H. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” in *NeurIPS 2020*, 2020.

[25] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. D. Cubuk, A. Kurakin, and C. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” in *NeurIPS 2020*, 2020.

[26] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in *NeurIPS 2020*, 2020.

[27] R. Masumura, M. Ihori, A. Takashima, T. Moriya, A. Ando, and Y. Shinohara, “Sequence-level consistency training for semi-supervised end-to-end automatic speech recognition,” in *ICASSP 2020*, 2020, pp. 7054–7058.

[28] F. Weninger, F. Mana, R. Gemello, J. Andrés-Ferrer, and P. Zhan, “Semi-supervised learning with data augmentation for end-to-end asr,” in *Interspeech 2020*, 2020, pp. 2802–2806.

[29] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, and P. J. Moreno, “SCADA: stochastic, consistent and adversarial data augmentation to improve ASR,” in *Interspeech 2020*. ISCA, 2020, pp. 2832–2836.

[30] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in Neural Information Processing Systems, 2017*, 2017, pp. 1195–1204.

[31] B. Li, A. Gulati, J. Yu, T. Sainath, and et. al., “A better and faster end-to-end model for streaming asr,” in *ICASSP 2021*, 2021.

[32] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech 2019*. ISCA, 2019, pp. 2613–2617.

[33] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *ICLR 2018*. OpenReview.net, 2018.

[34] H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, “Exploring pre-training with alignments for RNN transducer based end-to-end speech recognition,” in *ICASSP 2020*, 2020.

[35] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, and et al, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *CoRR*, vol. abs/1609.08144, 2016.