Semi-Supervised Learning with Data Augmentation for End-to-End ASR

Felix Weninger¹, Franco Mana², Roberto Gemello², Jesús Andrés-Ferrer³, Puming Zhan¹

Nuance Communications, Inc., ¹Burlington, MA, USA, ²Torino, Italy, ³Valencia, Spain
{felix.weninger,franco.mana,roberto.gemello,jesusandres.ferrer,puming.zhan}@nuance.com

Abstract

In this paper, we apply Semi-Supervised Learning (SSL) along with Data Augmentation (DA) for improving the accuracy of End-to-End ASR. We focus on the consistency regularization principle, which has been successfully applied to image classification tasks, and present sequence-to-sequence (seq2seq) versions of the FixMatch and Noisy Student algorithms. Specifically, we generate the pseudo labels for the unlabeled data on-the-fly with a seq2seq model after perturbing the input features with DA. We also propose soft label variants of both algorithms to cope with pseudo label errors, showing further performance improvements. We conduct SSL experiments on a conversational speech data set (doctor-patient conversations) with 1.9kh manually transcribed training data, using only 23% of the original labels (475h labeled data). In the result, the Noisy Student algorithm with soft labels and consistency regularization achieves 10.4% word error rate (WER) reduction when adding 475h of unlabeled data, corresponding to a recovery rate of 92%. Furthermore, when iteratively adding 950h more unlabeled data, our best SSL performance is within 5% WER increase compared to using the full labeled training set (recovery rate: 78%).

Index Terms: automatic speech recognition, semi-supervised learning, data augmentation, sequence-to-sequence, end-to-end

1. Introduction

End-to-end (E2E) systems have become a focus of ASR research in recent years, due to their ability of integrating all components of an ASR system in a single deep neural network (DNN), which greatly simplifies and unifies the training and decoding process [1][5]. The Sequence-to-Sequence (seq2seq) model with attention is one of the model architectures for E2E ASR systems which have shown state-of-the-art performance [6][10]. However, a general observation is that E2E ASR needs large amounts of training data for achieving state-of-the-art performance, especially when no language model trained with external text data is included in the system. Data Augmentation (DA) and semi-supervised learning (SSL) are two approaches that can be used for improving E2E model performance with limited amounts of manually transcribed training data.

DA perturbs (usually randomly) the input data without altering the corresponding labels. This not only increases the variety of the data, but also serves as implicit regularization to avoid overfitting [11]. It has been successfully used for both conventional [12][14] and E2E ASR systems [15][16]. In particular, the SpecAugment approach proposed in [17] has shown impressive improvement for seq2seq based E2E ASR models.

SSL (also called semi-supervised training) aims at leveraging unlabeled data for improving ASR model accuracy. In the self-training paradigm for SSL, a seed model trained with limited amount of labeled data is used to generate transcriptions (pseudo labels) for unlabeled data (cf. [18]). This procedure can be iterated on additional unlabeled data [19]. Another possible implementation of SSL is via teacher-student training, where a ‘student’ model is trained to replicate the outputs of a powerful ‘teacher’ model on the unlabeled data [20].

The central research question of our paper is how to best integrate SSL with DA for training E2E ASR systems. So far, the use of DA with SSL for E2E ASR has been largely limited to a simple cascade of both, i.e., doing pseudo label generation for the unlabeled data and then applying DA [21][22]. In contrast, for image classification, several algorithms have recently been proposed that use DA for teacher-student training [23] and consistency regularization in SSL [24][25]. Consistency regularization stems from the intuition that a small perturbation to an input data sample should not change the output distribution a lot. However, these SSL algorithms were designed only for static classification and need to be modified to support the seq2seq ASR use case.

In this regard, our paper makes the following contributions: First, we modify the Noisy Student [25] and FixMatch [26] algorithms – which have only been applied to image classification so far – for the seq2seq ASR use case. Second, we show performance improvements for both algorithms by using soft labels and consistency training (via SpecAugment and dropout). Finally, we demonstrate additional gains from iterative generation of pseudo labels by exploiting a larger amount of unlabeled data. We show that our proposed methods outperform the simple approach of doing DA after generating pseudo labels.

Relation to prior work: Several studies have recently investigated SSL techniques for E2E ASR, e.g. representation learning [29], the usage of external text data [30], text-to-speech [31], and transcriptions generated by conventional ASR [32]. Regarding self-training for E2E ASR, [33] proposed data filtering and ensemble schemes, generating hard pseudo labels via beam search. In [34], dropout was employed to improve the pseudo label accuracy and confidence measure, due to its well-known model ensemble property. All of these works did not consider the interaction of SSL with DA, which we investigate in our study. Very recently, [35] proposed teacher-student learning with DA for consistency training, without considering the Noisy Student or the FixMatch algorithm or soft labels as in our work.

2. Methods

For our E2E ASR models, we use an encoder-decoder architecture with attention as described in [9], which is similar to Listen-Attend-Spell (LAS) [6]. The ASR task is treated as a seq2seq learning problem: The model \( M \) is trained to predict a sequence \( y_j \) of symbols (here, we use sub-word units) from a sequence of acoustic features (usually a spectrogram \( x \)).

The encoder \( e \) creates a hidden representation of the acoustic features. Here, \( e \) is implemented as a stack of convolutional (CNN) layers followed by bidirectional Long Short-Term Memory (BiLSTM) layers. The decoder is similar to an RNN LM that takes into account a context vector \( c_j \). Bahdanau attention [36] is used to focus \( c_j \) on various parts of the encoder output. The
output distribution \( p_j = p(y_j | y_1, \ldots, y_{j-1}, x) = M(x, y_{<j}) \) for the \( j \)-th symbol is computed by

\[
\begin{align*}
    s_j &= f(s_{j-1}, \text{Embedding}(y_{j-1}, c_{j-1})), \\
    c_j &= \text{Attention}(s_j, c(x)), \\
    s_j' &= \text{Dense}(s_j, c_j), \\
    p_j &= g(s_j'),
\end{align*}
\]

where \( g \) is a softmax layer and \( f \) is a stack of LSTM layers.

To perform ASR using the seq2seq model, hypotheses are generated by beam search. The model is conditioned on its previous output \( y_{j-1} \) in Eq. 1. The sequence \( y_1, \ldots, y_j, \ldots \) of output symbols is produced one at a time, until the end-of-sentence symbol is reached.

### 2.1. Supervised and Semi-Supervised Training

For **supervised** training, the following cross-entropy (CE) loss is minimized for a mini-batch \( B \) of training examples:

\[
L_B(\theta) = \frac{1}{|B|} \sum_{i \in B} \sum_{j} L_{CE}(\hat{y}_{i,j}, M_{\theta}(A(x_i)), y_{i,j}^*) \tag{5}
\]

Here, \( y_{i,j}^* \) is the \( j \)-th symbol in the ground truth (GT) transcription of the \( i \)-th utterance in \( B \), represented as a one-hot vector, \( A(x_i) \) denotes the features of the \( i \)-th utterance in \( B \) after data augmentation (cf. Section 2.2), and \( \theta \) are the model parameters.

In **semi-supervised training**, the loss (5) is split into a labeled and an unlabeled part as \( L_B(\theta) = (1/|B|)(L_B(B^L) + L_B(B^U)) \),

\[
\begin{align*}
    L_B(B^L) &= \sum_{i \in B^L} \sum_{j} L_{CE}(\hat{y}_{i,j}, M_{\theta}(A(x_i)), y_{i,j}^*), \\
    L_B(B^U) &= \sum_{i \in B^U} \sum_{j} \mathbb{1}(\hat{y}_{i,j} \geq C) L_{CE}(\hat{y}_{i,j}, M_{\theta}(A(x_i^\epsilon)), \hat{y}_{i,j}),
\end{align*}
\]

where \( B^L \) contains labeled and \( B^U \) contains unlabeled utterances, \( B = B^L \cup B^U \). For the unlabeled loss (7), pseudo labels, i.e., pseudo truth (PT) transcriptions \( \hat{y}_{i,j} \), need to be obtained (cf. Section 2.3.1). Denoting the pseudo label confidence for utterance \( i \) and token \( j \) as \( \hat{p}_{i,j} \), confidence-based filtering is included in the loss (7) by setting a confidence threshold \( C > 0 \).

### 2.2. Data Augmentation (DA)

In our training framework, DA is modeled by a function \( A \), which also depends on the current state of the random number generator. We parameterize the function \( A \) on SpecAugment (SA) [17] with hyperparameters \( F_{\text{max}}, T_{\text{max}}, m_F \) and \( m_T \). The SA algorithm masks (replaces by zeros) up to \( F_{\text{max}} \) contiguous frequency bands and up to \( T_{\text{max}} \) contiguous time frames in the spectrogram \( x \). The starting positions and the actual number of masked rows / columns are sampled from a uniform distribution. This process is repeated \( m_F \) times for masking frequencies and \( m_T \) times for masking time frames. Due to the combinatorial explosion of possible corruptions applied by \( A \) to a single input \( x \), the usage of SA results in a practically infinite amount of training data.

### 2.3. Pseudo Labeling

We obtain pseudo labels for SSL based on PT transcriptions of the unlabeled data. These are generated ‘offline’ prior to training using beam search with a seq2seq model. Data augmentation is turned off during this offline PT generation phase. Since the PT generation is done only once, it is reasonable to use a large beam size \( W \) to improve the PT quality (here, we set \( W = 16 \)).

Furthermore, we apply a heuristic loop filtering technique to the PT utterances similar to the one from [22]. This helps us avoid reinforcing the well-known ‘looping’ problem, where the seq2seq model keeps repeating the same n-gram.

Semi-supervised training can be performed by directly using the PT transcriptions as pseudo labels \( \hat{y}_{i,j} \) in Eq. 7. Alternatively, we can dynamically update the PT transcriptions in the training process as in the FixMatch or Noisy Student approaches.

#### 2.3.1. DA for Consistency Training (FixMatch)

The FixMatch algorithm is a self-training method that was proposed in [28] for image classification. The method implements consistency training by applying two kinds of data augmentation to the input \( x \) in both the unlabeled loss and the pseudo label generation, thus encouraging the outputs to be consistent for both augmented inputs. Here, we present an extension of FixMatch to seq2seq models. Specifically, we generate pseudo labels \( \hat{y}_{i,j} \) for unlabeled training examples \( x_i^\epsilon \) as:

\[
\hat{y}_{i,j} = M_{\theta}(A_w(A(x_i^\epsilon)), \hat{y}_{i,j}),
\]

where \( A_w \) is a weak augmentation function. We then use \( \hat{y}_{i,j} \) in place of the PT label \( y_{i,j} \) in Eq. 7.

Figure 1 depicts our FixMatch algorithm for seq2seq ASR.

As argued by [28], choosing a weak augmentation \( A_w \) for pseudo label generation instead of the ‘strong’ augmentation \( A \) in the unlabeled loss ensures reasonable accuracy of the pseudo labels and improves the convergence. Moreover, our experiments show that applying the confidence threshold \( C \) leads to gradual annealing of the supervised training signal, as the model becomes more and more confident and hence includes more and more unlabeled data as the training proceeds. Finally, by applying dropout in all forward passes of the model (including the pseudo label generation), we add further regularization to the training.

Besides extending the algorithm to seq2seq, we make two further modifications. First, we also investigate using soft labels, in order to make the training more robust against pseudo label errors in case of ambiguous instances. Second, while [28] applied weak augmentation on labeled data, we found strong augmentation to perform better in practice.

#### 2.3.2. Teacher-Student Training with DA (Noisy Student)

The Noisy Student algorithm [23] is another recent SSL algorithm proposed for image classification, combining teacher-
student training with DA. The main difference to FixMatch is
that pseudo labels are generated by a pretrained teacher model
\(M_0^\ast\) with ‘frozen’ parameters \(\theta^\ast\), instead of the current model
parameters \(\theta\):
\[
\hat{y}_{i,j} = M_0^\ast(x_i^\ast, \hat{y}_{i,<j}).
\]
In our experiments, the teacher model is of the same topology as
the student model. Figure 2 depicts our Noisy Student algorithm
for ASR.

Since both hard and soft teacher labels (distillation) were
considered in [25], we also explored both for the E2E ASR task.
The hard label variant is equivalent to using the PT transcriptions
in the unlabeled loss \(\lambda_{rp}\), i.e. \(\hat{y}_{i,j} = \hat{y}_{i,j}\). For the soft label
variant, we (re-)compute soft labels on-the-fly via a forward
pass on the teacher model. This is done to avoid storing full
posteriors, which is hard for large label spaces and PT data sets.
The on-the-fly label computation also allows us to include dy-
namic modifications of the input and the model. Specifically, we
explore consistency training for the Noisy Student algorithm by
(weakly) augmenting the input for pseudo label generation as in
FixMatch \(\text{Eq. }[8]\), or running the teacher model with dropout in-
cluded. Our Noisy Student framework also subsumes generating
hard labels with consistency training, similar to \[24\].

2.3.3. Iterative SSL

We also investigate iterative SSL, i.e., several rounds of PT gen-
eration by the E2E model via beam search, in two ways. First,
we perform several iterations of the Noisy Student algorithm
similar to \[25\]. Second, we also compare to an iterative self-
training algorithm with DA inspired by \[24\], that regenerates PT
transcription for each mini-batch based on the current model pa-
rameters. However, while \[24\] used greedy search, we found this
to yield insufficient PT quality and thus adopted a lightweight
beam search \((W = 4)\).

2.4. Decoding

The decoding loss is extended with several terms as to avoid the
bias of seq2seq models to both deletions and insertions.
Specifically, we maximize the following score via histogram
pruning target synchronous beam search:

\[
\log p(y|x) + \lambda_{cov} \text{cov}(x, y) + \lambda_{wip}|y| + \left(\frac{5 + |y|}{5 + 1}\right) \lambda_{wip}
\]

where \(p(y|x)\) is the model probability; \(\text{cov}(x, y)\) is a coverage
score \(\text{(see Eq. } (11)\) \[36\]) weighted by \(\lambda_{cov}\) which adds a bonus

---

**Figure 2:** Sequence-to-sequence Noisy Student algorithm.

**Figure 3:** FixMatch behavior over time for various confidence
thresholds \(C\).
1.9 kh data, we achieve 13.79 % WER, which is the lower bound on the WER attainable with SSL.

2.2 % WERR from using consistency regularization (2.8 % WERR). The best result (15.02 % WER) is obtained by combining soft labels and consistency training. The best Noisy Student setup performs similar to the best FixMatch setup, and outperforms the iterative self-training similar to [24] by 2 %.

We also assess the WER recovery rate (WRR) of our SSL algorithms as defined in [22], which is the ratio of performance gain achieved by adding unlabeled data vs. the gain from adding the same amount of labeled data (in this case, training with 950 h labeled data). The simplest variant, which uses one-shot hard label PT generation (15.60 % WER), achieves 61.6 % WRR. The other SSL methods in Table 2, which all use on-the-fly PT label generation, can improve on this baseline. In particular, our variant using soft labels along with consistency training obtains 92.1 % WRR.

While we found only minor performance gains from multiple iterations of the Noisy Student algorithm on the same unlabeled data, we obtained additional improvements when including additional unlabeled data. For a total of 1.4 kh PT data in this case, the PT labels \( \hat{y}_{i,j} \) were generated using a model trained on 475 h GT and 475 h PT using the Noisy Student algorithm. Results are shown in Table 3. With soft PT labels and consistency training, we achieve a 4 % improvement over the previous best result.

5. Conclusions
In this paper, we presented a comprehensive study on SSL strategies for end-to-end ASR. We investigated the FixMatch and Noisy Student algorithms for ASR and demonstrated improvements from using soft labels and consistency training. We believe that this is due to their ability to reduce error reinforcement in SSL. In the result, the performance of SSL can approach the one of supervised learning with similar amounts of data.

In future work, we will extend our investigation to different DA schemes beyond SpecAugment (e.g. [15, 19]). We will also look at iteratively including more and more unlabeled data while increasing the model size as in [25].

Table 1: FixMatch results using 475 h labeled data (GT), 475 h unlabeled data (PT) and 200 k training steps.

| \( C \) | PT labels | PT noise | Init | WER |
|-------|-----------|----------|------|-----|
| –     | –         | –        | random | 16.77 |
| Semi-supervised, FixMatch | – | – | random | 15.97 |
| 0.5 | hard dropout | random | 16.13 |
| 0.5 | + SA (\( F = 5 \)) | random | 16.34 |
| 0.5 | soft dropout | random | 15.79 |
| 0.0 | soft dropout | 475 h GT | 15.22 |

Table 2: Comparison of FixMatch and Noisy Student methods, using 475 h of labeled (GT) and 475 h of unlabeled data (PT), and 70 training epochs. SA: SpecAugment. Initialization: random or from the supervised training on 475 h GT.

| PT labels | PT noise | Init | WER | WRR |
|-----------|----------|------|-----|-----|
| Semi-supervised, Noisy Student | – | – | 15.00 | 61.6 |
| hard | none | random | 15.60 | 79.5 |
| soft | none | random | 15.86 | 82.1 |
| soft | SA (\( F = 5 \)) | random | 15.02 | 92.1 |
| Semi-supervised, FixMatch | – | – | 15.04 | 91.1 |
| hard | none | 475 h GT | 15.34 | 75.3 |
| soft | dropout | 475 h GT | 14.87 | 100.0 |

Table 3: Results with iterative PT generation using a total of 475 h labeled and 1.4 kh unlabeled data.

| PT labels | PT noise | Init | WER | WRR |
|-----------|----------|------|-----|-----|
| Semi-supervised, Noisy Student | – | – | 14.88 | 63.4 |
| soft | none | random | 14.52 | 75.5 |
| soft | SA (\( F = 5 \)) | random | 14.44 | 78.2 |
| Supervised (1.9 kh), oracle performance | – | – | 13.79 | 100.0 |
