Improved Noisy Student Training for Automatic Speech Recognition

Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu and Quoc V. Le

Google Inc., U.S.A.

{danielspark, ngyuzh, jiaye, weihan, chungchengc, boboli, yonghui, qvl}@google.com

Abstract

Recently, a semi-supervised learning method known as "noisy student training" has been shown to improve image classification performance of deep networks significantly. Noisy student training is an iterative self-training method that leverages augmentation to improve network performance. In this work, we adapt and improve noisy student training for automatic speech recognition, employing (adaptive) SpecAugment as the augmentation method. We find effective methods to filter, balance and augment the data generated in between self-training iterations. By doing so, we are able to obtain word error rates (WERs) 4.2%/8.6% on the clean/noisy LibriSpeech test sets by only using the clean 100h subset of LibriSpeech as the supervised set and the rest (860h) as the unlabeled set. Furthermore, we are able to achieve WERs 1.7%/3.4% on the clean/noisy LibriSpeech test sets by using the unlab-60k subset of LibriLight as the unlabeled set for LibriSpeech 960h. We are thus able to improve upon the previous state-of-the-art clean/noisy test WERs achieved on LibriSpeech 100h (4.74%/12.20%) and LibriSpeech (1.9%/4.1%).

Index Terms: speech recognition, semi-supervised learning, data augmentation

1. Introduction

We aim to improve semi-supervised learning in automatic speech recognition (ASR) by adapting a method proven successful for image classification that we refer to as "noisy student training" (NST) [1]. Noisy student training is an iterative self-training method that makes use of unlabeled data to improve accuracy. In noisy student training, a series of models are trained in succession, such that for each model, the preceding model in the series serves as a teacher model on the unlabeled portion of the dataset. The distinguishing feature of noisy student training is the exploitation of augmentation, where the teacher produces quality labels by reading in clean input, while the student is forced to reproduce those labels with heavily augmented input features. To ensure the integrity of the labels provided by the teacher, the data generated by the teacher is typically filtered by removing low confidence examples, and balanced by matching the labeled distribution.

We adapt noisy student training for the purpose of speech recognition by introducing the following measures:

1. We employ (adaptive) SpecAugment [2, 3], an augmentation method for ASR that directly acts on the spectrogram of the input audio, for noisy student training.

2. We use shallow fusion with a language model (LM) [4] on the teacher network to generate better transcripts for the student network to train on.

3. We propose a **normalized filtering score** for transcripts generated by teacher networks given as a function of the fusion score and number of tokens.

4. We use a variant of sub-modular sampling [5] to weigh the utterance-transcript pairs generated by the teacher network to balance the token statistics of the dataset to be passed on to the student.

When the supervised performance of the task is low, we find gradational self-training over NST generations effective:

1. We use **gradational filtering**, where the filtering criterion is sytematically relaxed to grow the semi-supervised dataset as the model performance improves.

2. We introduce **gradational augmentation**, where augmentation strength is increased with NST iterations.

We are able to achieve state-of-the-art performance on two ASR tasks by using noisy student training. On the first task, LibriSpeech 100-860, the clean 100h subset of LibriSpeech [6] is used as the labeled set and the remaining 860h of LibriSpeech is used as the unlabeled set. We view this as a low-supervised-performance task, the baseline word error rates (WERs) being 5.5%/16.9% on the test-clean/test-other sets of LibriSpeech. By employing a noisy student training pipeline with gradational filtering and augmentation, we are able to achieve test-clean/test-other WERs 4.2%/8.6% on this task.

The second task is LibriSpeech-LibriLight, where the entirety of LibriSpeech is used as the labeled set, and the unlab-60k subset of LibriLight [7], a large untranscribed speech dataset based on audio books, is used as the unlabeled set. The baseline model [8] is able to achieve test-clean/test-other WERs 1.9%/4.1% showing high supervised performance. For this set, we obtain WERs 1.7%/3.4% on the test-clean/test-other sets by noisy student training without gradational filtering or augmentation, as we do not find gradational self-training effective.

1.1. Related Work

Self-training, where a teacher model is used to generate labels for the unlabeled set the student can train on, is one of the earliest methods in semi-supervised learning (e.g., [9, 10, 11]). Self-training has a long history of being applied to ASR through numerous studies, e.g., [12, 13, 14, 15, 16, 17, 18, 19]. Noise has been studied and shown to have positive effects in self-training for machine translation and image classification recently [1, 20], directly motivating our work. Gradational growth of the labelled dataset in self-training has been used in [21].

Another direction of semi-supervised learning research is to use consistency-based methods (e.g., [22, 23, 24]). In this approach, a separate task/loss to (pre-)train the acoustic model (or a part thereof) is introduced to help the model learn a good representation of the input, e.g., [25, 26, 27, 28, 29, 30, 31].

Our work builds upon self-training and has three main contributions. First, we make use of new augmentation methods in speech to improve the task performance. Second, we introduce a normalized filtering score for shallow-fused models for filtering generated transcripts. Third, we apply gradational filtering

where the filtering criterion for unlabeled data used for training is systematically relaxed with the self-training iterations.

2. Noisy Student Training for ASR

The NST algorithm we propose assumes a labeled set S, an unlabeled set U and a fixed LM trained on a separate text corpus. Then, NST generates a series of ASR models as follows:

1. Train M_0 on S with SpecAugment. Set $M = M_0$.

2. Fuse M with LM and measure performance.

3. Generate labeled dataset $M(U)$ with fused model.

4. Filter generated data $M(U)$ to obtain $f(M(U))$.

5. Balance filtered data $f(M(U))$ to obtain $b \cdot f(M(U))$.

6. Mix dataset $b \cdot f(M(U))$ and S. Use mixed dataset to train new model M' with SpecAugment.

7. Set $M = M'$ and go to 2.

We denote each cycle of the noisy student training a generation. We denote the filtered and balanced dataset $b \cdot f(M(U))$ the "semi-supervised portion" of the training set. We now elaborate on the details of the presented components.

SpecAugment: We use SpecAugment [2] to augment the input data at each step of NST. We employ adaptive time masking [3], where the size of the time masks scales linearly with the length of the input utterance, depending on the task.

Language Model Fusion: To produce better transcripts for the student, we shallow-fuse [4] the teacher networks with an LM trained on a fixed text corpus. We introduce a coverage penalty term [32] in the fusion score with parameter c for LAS models [33], while we use a constant non-blank reward ρ to shift the logits produced by the LM for RNN-T [34] networks [35, 36]. These parameters are tuned at each generation via grid search to minimize the dev-set WERs. The transcripts of the unlabeled set are generated from the fused model using beam search.

Filtering: Given that the generated transcripts were found based on a fusion score that does not have an interpretation as a log-probability, it is unclear what to use to judge the quality of the transcripts generated by a teacher network.

We introduce a filtering score given as a function of the shallow fusion score $\mathcal{S}$ and the token length ℓ of a transcript generated by the fused teacher model. The normalized filtering score is defined to be

$$s(\mathcal{S}, \ell) = (\mathcal{S} - \mu\ell - \beta)/(\sigma\sqrt{\ell}), \qquad (1)$$

with parameters μ, β and σ fit on the dev-set. μ, β are fit via linear regression on the value pairs $(\ell_i, \mathcal{S}_i)$ of the transcripts generated from the dev-set utterances. σ is obtained by computing the standard deviation of $(\mathcal{S}_i - \mu\ell_i - \beta)/\sqrt{\ell_i}$. These parameters are fit for each new generation of teacher models.

Using this score, we filter the transcript-utterance pairs generated by the trained models in a gradational manner, i.e., we lower the filtering cutoff s_{cutoff} for which only the utterances with scores $s(\mathcal{S}, \ell) > s_{\text{cutoff}}$ are kept, as the self-training cycle is iterated. This strategy is expected to work best when the teacher model becomes markedly better with NST iterations.

Balancing: The distribution of tokens of the transcripts of the generated set $f(M(U))$ can differ significantly from that of the supervised training set distribution. We may choose to weigh the samples in the newly generated dataset to bridge this gap.

We use sub-modular sampling [5] with the "cost-benefit score" to balance the dataset. Sub-modular sampling is a method that samples (with replacement) a set of sentences from a sentence pool so that the token distribution of the sampled set is close to a target distribution. This is done by optimizing the KL divergence between the token distributions of the sampled set and the target distribution in a greedy way, where sentences are collected by batches of size B. Each batch is chosen from the pool of sentences by selecting the top-B sentences in terms of cost-benefit, i.e., the decrease in KL divergence by adding that sentence to the current set of sampled transcripts, divided by its number of tokens. After a batch of samples have been collected and added to the sampled set, the KL divergence between the token distribution of assembled sentences and the target distribution is computed, and the process is iterated.

We impose additional constraints on this algorithm by capping the sampling multiplicity of a sentence by 2 and requiring the total number of tokens in the sampled set to be bounded below by the total number of tokens in the supervised set. Furthermore, we take the batch size to be a tenth of the filtered dataset. We optimize the KL divergence with these restrictions to obtain the balanced set $b \cdot f(M(U))$ from $f(M(U))$.

Mixing: We combine $b \cdot f(M(U))$ with the supervised dataset S for training using two different methods of mixing. In batch-wise mixing, we fix the ratio between supervised and semi-supervised samples in each training batch. In non-batch-wise mixing, data is uniformly sampled from both datasets.

3. Experiments

3.1. LibriSpeech 100-860

LibriSpeech 100-860 is a semi-supervised task where the clean 100h subset of LibriSpeech [6] is taken to be the supervised set, while the remaining 860h of audio is taken to be the unlabeled set. The unlabeled audio consists of 360h of clean data and 500h of noisy data. We tokenize the transcripts using a WPM model [37] with vocabulary size 16k constructed from the clean 100h subset transcripts. We use 80-dimensional filter bank coefficients with delta and delta-delta acceleration as the input features. We use the model denoted LAS-6-1280 in [2] (see also [38]) as our acoustic model, which is a Listen, Attend and Spell network [33] with a bi-directional LSTM encoder.

We train 6 generations of models numbered 0 to 5, where we count the baseline model trained with the supervised set as the zeroth generation. Each generation is trained with peak Adam learning rate of 0.001 and batch size of 512 on 32 Google Cloud TPU chips for 10 days. The checkpoint to be used for LM fusion is chosen based on its dev-set WER. The choices made for the components of NST are as follows:

SpecAugment: For generation 0, we use two frequency masks with mask parameter (F) 27, two time masks with mask parameter (T) 40, and time warping with warp parameter (W) 40 [2]. At generations 2 and 4, we re-tuned the time mask parameter T on a proxy task (with model LAS-4-1024 and schedule LB from [2]). The time masking parameter used for generations 0 and 1, 2 and 3, 4 and 5 were set to 40, 80 and 100, respectively.

LM: We use a 3-layer LSTM language model with width 4096 trained on the LibriSpeech LM corpus tokenized with the 16k WPM. The LM has word-level perplexity 63.9 on the dev-set.

Filtering: We apply gradational filtering from generation 1 to 5 with cutoffs, 1, 0.5, 0, -1 and $-\infty$. We apply filtering separately to the clean-360h and other-500h audio, i.e., the scoring coefficients are fit separately on dev-clean and dev-other, and applied to clean-360h and other-500h respectively.

Other: Neither balancing or batch-wise mixing is used.

The performance of the fused models through the generations are plotted in figure 1. Our best trained model is the generation-4 model. We compare the performance of our baseline model, as well as the generation-4 model before and after LM fusion, against other results in the literature in table 1.

Table 1: *LibriSpeech 100h WERs (%).*

Method	Dev		Test	
	clean	other	clean	other
Supervised				
Lüscher et al., (2019) [39]	5.0	19.5	5.8	18.6
Kahn et al., (2019) [16]	7.78	28.15	8.06	30.44
Hsu et al., (2019) [19]	14.00	37.02	14.85	39.95
Ling et al., (2019) [31]			6.10	17.43
Semi-supervised (w/ LibriSpeech 860h)				
Kahn et al., (2019) [16]	5.41	18.95	5.79	20.11
Hsu et al., (2019) [19]	5.39	14.89	5.78	16.27
Ling et al., (2019) [31]			4.74	12.20
This Work				
Baseline (LAS + SpecAugment)	5.3	16.5	5.5	16.9
+ NST before LM Fusion	4.3	9.7	4.5	9.5
+ NST with LM Fusion	**3.9**	**8.8**	**4.2**	**8.6**

3.2. LibriSpeech-LibriLight

The entire LibriSpeech [6] training set is used as the supervised training set of this task, while the "unlab-60k" subset of LibriLight [7], an unlabeled audio dataset derived from audio books, is used as the unlabeled set. We do not use the "duplicate" set, nor any supervised subset of LibriLight. We segment the audio using a maximum expected segmentation length of 36 seconds, but filter further down to choose segments that have less than 22 seconds of audio. This results in producing approximately a million utterances. The transcripts of the supervised set are tokenized using a WPM model with vocabulary size 1k constructed from the LibriSpeech training set transcripts.
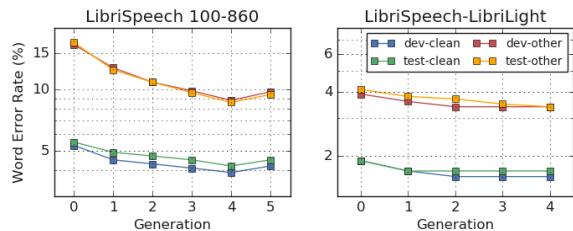

Figure 1: *WERs (%) plotted against training generation.*

We use 80-dimensional filter bank coefficients of the utterances as single-channel input features. We use a family of ContextNets [8], which are RNN-T models [34] with deep CNN encoders, for this task. We use CN-w to denote a model with a 23-convolutional-block encoder and 2-layer decoder with cell size 1280 and width scaling factor w studied in [8].

Five generations of models numbered 0 to 4, are trained, where the baseline model is taken to be the generation-zero model. The baseline ContextNet model has the same encoder as CN-2, but has a one-layer RNN decoder with dimension 640. Meanwhile, CN-w with w=1.25, 1.75, 2.25 and 2.5 have been set to be the ASR model from generation 1 through 4. Each generation is trained on 128 Google Cloud TPU chips for 1-3 days. The checkpoint of the model to be used for LM fusion is chosen based on the dev-set WER. The choices for the NST pipeline have been set as follows:

SpecAugment: We use SpecAugment with adaptive time mask size [2, 3]. We use two frequency masks with mask parameter

(F) 27, and ten time masks with maximum time-mask ratio (p_S) 0.05, where the maximum-size of the time mask is set to p_S times the length of the utterance. Time warping is not used.

LM: We use a 3-layer LSTM LM with width 4096 trained on the LibriSpeech LM corpus with the LibriSpeech 960h transcripts added, tokenized with the 1k WPM. The LM has word-level perplexity 68.3 on the dev-set transcripts.

Other: We do not apply filtering, while balancing is applied with the parameters detailed in section 2. We use batch-wise mixing, and gradually increase the portion of the semi-supervised utterances—the ratio between the supervised and semi-supervised utterances are raised from 4:6 at generations 1, 2 to 3:7 at generation 3 to 2:8 at generation 4.

We have plotted the generational performance of the fused models in figure 1. Our best trained model is the generation-4 model whose performance is presented in table 2.

Table 2: *LibriSpeech 960h WERs (%).*

Method	Dev		Test	
	clean	other	clean	other
Supervised				
Synnaeve et al., (2019) [17]	2.10	4.79	2.33	5.17
Zhang et al., (2020) [36]			2.0	4.6
Han et al., (2020) [8] †	1.9	3.9	1.9	4.1
Semi-supervised				
Synnaeve et al., (2019) [17]	2.00	3.65	2.09	4.11
This Work (with baseline †)				
ContextNet + NST before LM Fusion	**1.6**	3.7	**1.7**	3.7
ContextNet + NST after LM Fusion	**1.6**	**3.4**	**1.7**	**3.4**

4. Discussion

4.1. LibriSpeech 100-860

Gradational Filtration: As noted in section 3.1, for LibriSpeech 100-860, we gradually relax the filtering criterion for the semi-supervised dataset based on their normalized filtering score. Recall that we generate transcripts for the dev-clean and dev-other sets using the generation 0 to 4 LibriSpeech 100-860 models to fit the filtering score parameters. We plot the portion of utterances and their tokens in these datasets that are above a given filtering score in figure 2. The thick dotted line is the cumulative density of the standard normal distribution we have plotted as a reference, while the ten solid curves come from transcripts generated at each generation from the two dev-sets. The distribution of utterances are remarkably stable throughout generations, the solid curves virtually lying on top of each other.
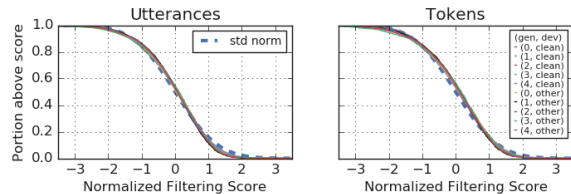

Figure 2: *Portion of utterances/tokens above filtering score. Solid curves are indexed by generation and filtered dev-set.*

We have plotted the WER of utterances above a given filtering score in figure 3 for each generation. The score reasonably ranks the quality of the generated dev-set transcripts, except at generation 4, where the model generates a blank transcript to a long utterance, which makes the cumulative WER become large at high scores. We have marked the score used for filtering

at each generation. The filtered dev-set WERs are comparable throughout the generations, while the size of the filtered dataset grows. This is the intended effect of gradational filtration.
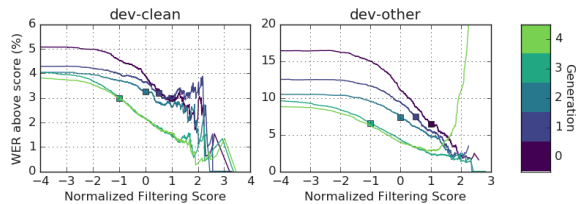

Figure 3: *WER above filtering score for LibriSpeech 100-860 for dev-clean and dev-other with generation 0-4 models.*

We find that slow evolution improves performance in the range we have explored. We have run an experiment where we evolve the model "faster," by training two generations in addition to the baseline, using filtering cutoffs 0 and $-\infty$. We plot the evolution of dev-set WERs against the semi-supervised dataset size in figure 4. Models in the 6-gen pipeline outperforms those in the 3-gen pipeline at comparable dataset sizes.
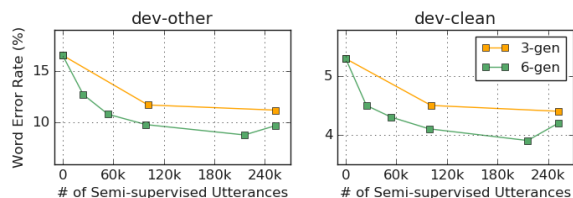

Figure 4: *WER (%) vs semi-supervised dataset size for models in 6 and 3 generation noisy student training pipelines.*

Augmentation: The increase of time mask size used for training at generations 2 and 4 is based on proxy task evaluation. In the proxy task, LAS-4-1024 models [2, 38], which have about 1/3 of the parameters of LAS-6-1280, are trained for 100k steps with the combined training set at generations 2 and 4 with varying time mask size. The time mask size has been chosen based on the final WERs tabulated in table 3, where we prioritized performance on the clean set. The performances are measured without LM fusion. We find that a stronger augmentation is favored, as the dataset grows with gradational filtering.

Table 3: *Proxy task WER (%) for LibriSpeech 100-860.*

Time Mask Size (T)		40	60	80	100	120
Gen 2	dev-clean	5.7	5.6	5.5	9.2	-
	dev-other	15.0	14.5	14.7	17.8	-
Gen 4	dev-clean	-	4.8	5.0	4.7	4.7
	dev-other	-	11.7	11.7	11.4	11.7

Mixing and Balancing: We conduct control experiments with batch-wise mixing and balancing for LibriSpeech 100-860. We have experimented with four combinations of balancing and 1:1 batch-wise mixing for the LibriSpeech 100-860 task, the results of which we present in figure 5. Balancing cuts down the semi-supervised dataset size at each generation to about a half, diminishing the performance of the trained network.

4.2. LibriSpeech-LibriLight

We present control experiments for the LibriSpeech-LibriLight task. A slightly inferior baseline with model CN-1.25 with
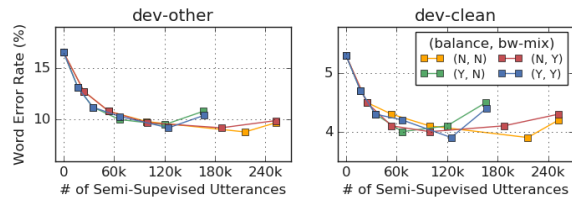

Figure 5: *WERs plotted against the size of the semi-supervised dataset for combinations of balancing and batch-wise mixing.*

dev-clean/other WERs 1.8%/4.2% and test-clean/other WERs 1.9%/4.4% has been used for these experiments.

Balancing and Filtering: Combinations of filtering with score 0 (which filters $\sim 50\%$ of the utterances) and balancing have been tested on the first generation of NST. The pre-fusion WERs are presented in table 4. In contrast with LibriSpeech 100-860, the reduction in the semi-supervised set size by balancing does not seem to outweigh the benefits of making the token distribution closer to that of the supervised set, due to the size of the LibriLight dataset. We have chosen to use balancing, but not to filter in our pipeline based on these experiments.

Table 4: *WERs (%) for balancing and filtering configurations.*

(Bal, Filt)	(N, N)	(N, Y)	(Y, N)	(Y, Y)
dev-clean	1.8	1.8	1.7	1.8
dev-other	4.5	4.4	4.3	4.3

Gradational Self-Training: We find that gradational filtration and augmentation are not as effective for this task as it was for LibriSpeech 100-860. We compare the performance of a network trained with gradational filtration over 5 generations with filtering scores 1, 0.5, 0, -1 and $-\infty$ and one trained on the entire LibriLight dataset without filtering. We do not see the benefit of gradational filtration, as both models, after LM fusion have dev-clean/dev-other WERs 1.7%/3.7%, respectively. An explanation of this result is that the baseline model for these tasks is already extremely good, and the quality of filtered transcripts is not drastically better than the unfiltered transcripts. For example, the WER for the transcripts with score > 1 on the dev-set is 1.5%, against 3.0% of the unfiltered transcripts at generation 0. This should be compared to the $\sim 10\%$ difference in WER between the transcripts with score > 1 against the unfiltered transcripts generated from the dev-other utterances at generation 0 for LibriSpeech 100-860 (gen-0 curve in figure 3).

5. Conclusion

We have adapted and improved noisy student training for ASR. We have employed SpecAugment, language model fusion and sub-modular sampling into the noisy student training pipeline to adapt it for speech recognition. We have also introduced a normalized filtering score which we use for gradational self-training. The gradational methods prove to be beneficial for the low-supervised-performance task LibriSpeech 100-860, while it does not show much effect for the high-supervised-performance task LibriSpeech-LibriLight. The various components introduced in this work were mixed and matched to improve the state-of-the-art performance of both tasks studied.

6. References

[1] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020.

[2] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.

[3] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Specaugment on large scale datasets," in *arxiv*, 2019.

[4] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," in *arxiv*, 2015.

[5] Y. Shinohara, "A submodular optimization approach to sentence set selection," in *ICASSP*, 2014.

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

[7] J. Kahn, M. Rivire, W. Zheng, E. Kharitonov, Q. Xu, P. Mazar, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *arxiv*, 2019.

[8] W. Han, Y. Zhang, J. Yu, Z. Zhang, J. Qin, A. Gulati, C. cheng Chiu, R. Pang, and Y. Wu, "ContextNet: Convolutional Neural Networks with Larger Context for Speech Recognition," in *arXiv*, 2020.

[9] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[10] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd annual meeting of the association for computational linguistics*, 1995, pp. 189–196.

[11] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 2003, pp. 105–112.

[12] G. Zavaliagkos and T. Colthurst, "Utilizing untranscribed training data to improve performance," in *DARPA Broadcast News Transcription and Understanding Workshop, Landsdowne*, 1998, pp. 301–305.

[13] S. Novotney and R. Schwartz, "Analysis of low-resource acoustic model self-training," in *Interspeech*, 2009.

[14] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *ICASSP*, 2013.

[15] B. Li, T. N. Sainath, R. Pang, and Z. Wu, "Semi-supervised training for end-to-end models via weak distillation," in *ICASSP*, 2019.

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

[17] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," in *arXiv*, 2019.

[18] S. H. K. Parthasarathi and N. Strom, "Lessons from building acoustic models with a million hours of speech," in *ICASSP*, 2019.

[19] W.-N. Hsu, A. Lee, G. Synnaeve, and A. Hannun, "Self-supervised speech recognition via local prior matching," in *arXiv*, 2020.

[20] J. He, J. Gu, J. Shen, and M. Ranzato, "Revisiting self-training for neural sequence generation," in *ICLR*, 2020.

[21] L. Lamel, J. luc Gauvain, and G. Adda, "Lightly supervised acoustic model training," in *Proc. ISCA ITRW ASR2000*, 2000, pp. 150–154.

[22] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," in *arXiv*, 2019.

[23] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *ICCV*, 2019.

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

[25] W.-N. Hsu and J. Glass, "Extracting domain invariant features by unsupervised learning for robust automatic speech recognition," in *ICASSP*, 2018.

[26] Y.-A. Chung and J. Glass, "Speech2vec: A sequenceto-sequence framework for learning word embeddings from speech," in *Interspeech*, 2018.

[27] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Interspeech*, 2019.

[28] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," in *arXiv*, 2019.

[29] S. Schneider, A. Baevski, R. Collober, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019.

[30] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *arXiv*, 2019.

[31] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *arXiv*, 2019.

[32] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Interspeech*, 2017.

[33] 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.

[34] A. Graves, "Sequence transduction with recurrent neural networks," in *arXiv*, 2012.

[35] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Interspeech*, 2015.

[36] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *arXiv*, 2020.

[37] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *ICASSP*, 2012.

[38] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "Model Unit Exploration for Sequence-to-Sequence Speech Recognition," in *arXiv*, 2019.

[39] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for librispeech: Hybrid vs attention - w/o data augmentation," in *Interspeech*, 2019.