INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Reducing Exposure Bias in Training Recurrent Neural Network Transducers

Xiaodong Cui, Brian Kingsbury, George Saon, David Haws, Zoltan Tuske

IBM Research AI
IBM T. J. Watson Research Center, Yorktown Heights, New York, USA
{cuix,bedk,gsaon,dhaws}@us.ibm.com, zoltan.tuske@ibm.com

Abstract

When recurrent neural network transducers (RNNTs) are trained using the typical maximum likelihood criterion, the prediction network is trained only on ground truth label sequences. This leads to a mismatch during inference, known as exposure bias, when the model must deal with label sequences containing errors. In this paper we investigate approaches to reducing exposure bias in training to improve the generalization of RNNT models for automatic speech recognition (ASR). A label-preserving input perturbation to the prediction network is introduced. The input token sequences are perturbed using SwitchOut and scheduled sampling based on an additional token language model. Experiments conducted on the 300-hour Switchboard dataset demonstrate their effectiveness. By reducing the exposure bias, we show that we can further improve the accuracy of a high-performance RNNT ASR model and obtain state-of-the-art results on the 300-hour Switchboard dataset.

Index Terms: automatic speech recognition, end-to-end model, RNN transducer, exposure bias, scheduled sampling

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) systems based on deep neural networks (DNNs) have made great progress in the past decade and have become more and more dominant in modern ASR. Compared to the conventional hybrid DNN ASR systems [1–4] where output units represent context-dependent hidden Markov model (HMMs) states [5], E2E systems directly map an input acoustic sequence to an output text sequence. Therefore, pronunciation, acoustic and language modeling are carried out in the same framework. E2E models also usually require no hard frame level alignment, which greatly simplifies the ASR training pipeline. A broad variety of network architectures have been proposed for E2E systems in literature, notably connectionist temporal classification (CTC) [6], attention-based encoder-decoder (AED) [7–9], recurrent neural network transducer (RNNT) [10–13] and self-attention-based transformer [14].

In recent years, RNNT models have emerged as a promising E2E ASR framework. They achieve competitive performance and in the meantime are streaming friendly. This makes them attractive in real-world deployments and hence they have received increasing attention in the community. In this paper, we are focused on improving RNNT ASR performance by reducing exposure bias during training. Exposure bias, a generic issue in text generation, arises due to the training-generation discrepancy. It happens when training is conducted with ground truth labels, while generation is conducted with errorful label sequences. In RNNT, the RNN prediction network predicts the next token conditioned on a history of previous tokens. In training, the prediction network is always fed with ground truth label sequences. However, errors may occur in decoding. Therefore the prediction of the next token is conditioned on a history contaminated with errors. This mismatch in training and decoding gives rise to exposure bias in RNNT models and hurts their generalization under test conditions.

Scheduled sampling [15] is a representative technique for mitigating exposure bias. It has been used in E2E ASR frameworks such as AED models [16, 17] and self-attention-based transformers [18]. Given the generative nature of the encoder-decoder architecture, scheduled sampling is relatively straightforward to realize in these frameworks. In RNNT, however, the prediction network essentially provides a "soft" token alignment of the acoustic feature sequence to compute the transition probabilities on the time-token lattice. Since it is not a generative architecture, it is not obvious how conventional scheduled sampling can be applied. Few efforts have been published on using scheduled sampling for training RNNT models. Scheduled sampling in [15] can be viewed as a label-preserving perturbation of token history to introduce uncertainty during training. The perturbation is carried out such that the token history on which the next token prediction is conditioned is close to that observed in decoding so that the gap between the two is reduced. In the same light, we investigate in this paper label-preserving perturbation of input token sequences to the prediction network in RNNT to reduce the exposure bias. Specifically, we introduce perturbation based on SwitchOut [19] and perturbation based on scheduled sampling from a token language model (LM). Experiments are carried out on the 300-hour Switchboard (SWB300) dataset. We show that the perturbation of input token sequences is helpful in reducing exposure bias. It can improve over a high-performance RNNT ASR model with state-of-the-art word error rates (WERs).

The remainder of the paper is organized as follows. Section 2 describes the RNNT framework. Section 3 introduces the two perturbation techniques to the input of the RNNT prediction network, namely, Switchout and scheduled sampling based on a token LM. Experimental results on SWB300 are reported in Section 4 followed by a discussion in Section 5. Section 6 concludes the paper with a summary.

2. RNN Transducers

Following the notation in [10], let $\boldsymbol{x} = (x_1, x_2, \cdots, x_T)$ be the input acoustic sequence of length T where $x_t \in \mathcal{X}$ and $\boldsymbol{y} = (y_1, y_2, \cdots, y_U)$ be the output token sequence of length U where $y_u \in \mathcal{Y}$. $\mathcal{X}$ and $\mathcal{Y}$ are input and output spaces, respectively. Define the extended output space

$$\bar{\mathcal{Y}} = \mathcal{Y} \cup \varnothing \tag{1}$$

where $\varnothing$ represents a null output.

RNNT evaluates the conditional distribution of the output

http://dx.doi.org/10.21437/Interspeech.2021-587

sequence given the input sequence

$$\Pr(\boldsymbol{y} \in \mathcal{Y}^* | \boldsymbol{x}) = \sum_{\boldsymbol{a} \in \mathcal{B}^{-1}(\boldsymbol{y})} \Pr(\boldsymbol{a}|\boldsymbol{x}) \quad (2)$$

where $\boldsymbol{a} \in \bar{\mathcal{Y}}^*$ are alignments of acoustic sequence $\boldsymbol{x}$ against the token sequence $\boldsymbol{y}$ with the null output symbol $\varnothing$ and $\mathcal{B} : \bar{\mathcal{Y}}^* \mapsto \mathcal{Y}^*$ is the function that removes null symbols from the alignments.

The acoustic features x_t are embedded in a latent space by a bi-directional long short-term memory (LSTM) network [20], referred to as the transcription network $\mathcal{T}$

$$f_t = \mathcal{T}(\boldsymbol{x}_{1:T}, t). \quad (3)$$

The label tokens y_u are embedded in a latent space by a uni-directional LSTM network, referred to as the prediction network $\mathcal{P}$:

$$g_u = \mathcal{P}(\boldsymbol{y}_{[1:u-1]}, u). \quad (4)$$

Given the acoustic embedding f_t and the label token embedding g_u, the predictive output probability at (t, u) is implemented as

$$p(\cdot|t, u) = \text{softmax}[\mathbf{W}^{\text{out}} \tanh(\mathbf{W}^{\text{enc}} f_t \odot \mathbf{W}^{\text{pred}} g_u + b)]. \quad (5)$$

In Eq. 5, matrices $\mathbf{W}^{\text{enc}}$ and $\mathbf{W}^{\text{pred}}$ are linear transforms that project the acoustic embedding f_t and the label token embedding g_u into the same joint latent space where element-wise multiplication is performed[1] and a hyperbolic tangent ($tanh$) nonlinearity is applied. Finally, it is projected to the output space by a linear transform $\mathbf{W}^{\text{out}}$ and normalized by softmax, producing a predictive probability estimate.

The parameters of RNNT θ are optimized using the maximum likelihood criterion:

$$\theta^* = \underset{\theta}{\text{argmax}} \log \Pr(\boldsymbol{y}|\boldsymbol{x}; \theta). \quad (6)$$

The conditional likelihood in Eq. 2 over all possible alignments can be evaluated using the forward-backward algorithm:

$$\Pr(\boldsymbol{y} \in \mathcal{Y}^* | \boldsymbol{x}) = \sum_{(t,u):t+u=n} \alpha(t, u)\beta(t, u) \quad (7)$$

$\forall n : 1 \leq n \leq U + T$ where $\alpha(t, u)$ and $\beta(t, u)$ are forward and backward variables that can be computed recursively on the lattice.

Optimization is typically based on stochastic gradient descent or its variants with back-propagation. Once the RNNT model is trained, decoding is carried out based on beam search [10, 21].

3. Input Perturbation of the Prediction Network

In standard RNNT training, the prediction network is always fed with the ground truth label token sequences. In Eq. 4, the LSTM $\mathcal{P}$ always uses the previous ground truth token history $\boldsymbol{y}_{[1:u-1]}$ to recurrently encode the current token $\boldsymbol{y}_u$. However, in decoding, the input sequence to the prediction network, $\hat{\boldsymbol{y}} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_U\}$, is composed of token hypotheses. As a result, the prediction network has to encode the current token $\hat{y}_u$ using the history $\hat{\boldsymbol{y}}_{[1:u-1]}$ which may contain errors. This discrepancy between the training and decoding will incur the mismatch of the output probability in Eq. 5 and consequently affect the probability diffusion over the lattice.

In this section, we introduce label-preserving perturbation to the input label token sequences to the prediction network. Specifically, we introduce a perturbation $\sigma(\cdot)$

$$\sigma(\boldsymbol{y}) \rightarrow \tilde{\boldsymbol{y}} \quad (8)$$

to inject errors into the ground truth sequence. The resulting token sequence $\tilde{\boldsymbol{y}}$ perturbs the token embedding and also alters the alignment $\boldsymbol{a}$ in Eq. 2

$$\Pr(\boldsymbol{y}|\tilde{\boldsymbol{y}}, \boldsymbol{x}) = \sum_{\boldsymbol{a} \in \mathcal{B}^{-1}(\tilde{\boldsymbol{y}})} \Pr(\boldsymbol{a}|\boldsymbol{x}). \quad (9)$$

In the meantime, since it is label preserving the loss in Eq. 6 is still optimized under the likelihood of ground truth $\boldsymbol{y}$.

In the sequel, we investigate two perturbation strategies, SwitchOut [19] and scheduled sampling [15].

3.1. SwitchOut

Given an input token sequence $\boldsymbol{y}$, SwitchOut randomly corrupts a number of tokens in $\boldsymbol{y}$.

First, the number of tokens to be corrupted, $\hat{n}$, is sampled according to the following distribution

$$p(n) = \frac{e^{-\frac{n}{\tau}}}{\sum_{n'=0}^{|\boldsymbol{y}|} e^{-\frac{n'}{\tau}}} \quad (10)$$

where τ is a temperature parameter controlling the "flatness" of the distribution.

Define a Bernoulli random variable

$$\gamma \sim \text{Bernoulli}(\hat{n}/|\boldsymbol{y}|). \quad (11)$$

which is fixed for each token sequence $\boldsymbol{y}$. For each token y_u in $\boldsymbol{y}$, $u \in \{1, 2, \cdots, U\}$,

$$\tilde{y}_u = \begin{cases} v \in \mathcal{Y}, v \neq y_u, & \gamma = 1 \\ y_u, & \gamma = 0 \end{cases} \quad (12)$$

3.2. Scheduled Sampling

SwitchOut introduces uncertainty to the ground truth sequence in a random fashion, which does not consider the history of tokens when making the perturbation. In decoding, every token is a predictive outcome of the previous history, and the prediction network implicitly learns the dependency similar to a language model. In order for the training to be closer to the decoding, we leverage a token LM to perturb the next token given the history.

First, an LSTM LM $\tilde{p}(z_u|\boldsymbol{z}_{[1:u-1]})$ is trained on the token sequences of the training set. Suppose $\boldsymbol{y}$ is the ground truth token sequence and $\tilde{\boldsymbol{y}}$ the perturbed. Define a Bernoulli random variable

$$\gamma \sim \text{Bernoulli}(p). \quad (13)$$

where p is the teacher forcing probability.

To make the prediction of the next token in the perturbed sequence, it either sticks to the ground truth or samples from the token LM with probability p:

$$\tilde{y}_u = \begin{cases} y_u, & \gamma = 1 \\ z_u \sim \tilde{p}(z_u|\tilde{\boldsymbol{y}}_{[1:u-1]}), & \gamma = 0 \end{cases} \quad (14)$$

[1] Following the implementation in [13] due to observed superior performance on multiple datasets.

When sampling from $\tilde{p}(z_u|\boldsymbol{z}_{[1:u-1]})$, the predicted next token is uniformly selected from the top k token candidates given the history $\tilde{\boldsymbol{y}}_{[1:u-1]}$, where k is a hyper-parameter. When $k = 1$, the most likely token given the history is always chosen.

4. Experiments

4.1. Setup

Dataset The experiments are conducted on the SWB300 dataset [22, 23]. Test sets include a 2.1-hour switchboard (SWB) set and a 1.6-hour call-home (CH) set from the NIST Hub5-2000 test set. The data preparation pipeline follows the Kaldi [24] s5c recipe.

Architecture There are 6 bi-directional LSTM layers in the transcription network with 1,280 cells in each layer (640 cells in each direction). The prediction network is a uni-directional LSTM consisting of a single layer with 1,024 cells. Both the acoustic embedding after the transcription network and the label embedding after the prediction network are projected down to a 256-dimensional latent space where they are combined by element-wise multiplication in the joint network. After a hyperbolic tangent nonlinearity followed by a linear transform, the topmost softmax layer has 46 output units which correspond to 45 characters and the null symbol.

Acoustic features The acoustic features are 40-dimensional logMel features after conversation side based mean and variance normalization plus their first and second order derivatives. The features are extracted every 10ms, but every two adjacent frames are concatenated, which amounts to a frame downsampling from 10ms to 20ms. This gives rise to the final 240-dimensional acoustic input to the transcription network.

Regularization The 300-hour training data is first augmented by speed and tempo perturbation [25] to produce additional 4 replicas of the training data. On top of that, two additional data augmentation techniques are conducted on the fly when loading the training data. One is sequence noise injection [26], where a training utterance is artificially corrupted by adding a randomly selected downscaled training utterance from the training set, and the other is SpecAugment [27], where the spectrum of a training utterance is randomly masked in blocks in both the time and frequency domains. The training is also regularized by dropout with a dropout rate of 0.25 for the LSTM and 0.05 for the embedding. In addition, DropConnect [28] is applied with a rate of 0.25, which randomly zeros out elements of the LSTM hidden-to-hidden transition matrices. The transcription network is initialized by an LSTM model of the same configuration trained with CTC, following [29].

Optimizer and Training Schedule The RNNT is trained using `AdamW` [30], a version of the `Adam` optimizer [31] that adds decoupled weight decay. A long warmup and long hold training schedule is employed. The learning rate starts at 0.0002 in the first epoch and then linearly scales up to 0.002 in the first 10 epochs. It holds for another 6 epochs before being annealed by $\frac{1}{\sqrt{2}}$ every epoch after the 17$^{\text{th}}$ epoch. The training finishes after 30 epochs. In each epoch, the training utterances are provided in a sorted order, starting with short utterances. This amounts to a curriculum learning scheme that stabilizes the training early on with short utterances and then gradually increases the learning difficulty to longer utterances. The batch size is 250 which is uniformly distributed to 5 v100 GPUs (50 per each GPU). Gradients are aggregated using `all-reduce`.

Decoding Inference uses alignment-length synchronous decoding [21], which only allows hypotheses with the same alignment length in the beam for the beam search. In all test cases, we measure the WERs with and without an external LM. When decoding with an external LM, the following density ratio LM fusion [32] is used:

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}\in\mathcal{Y}^*}{\operatorname{argmax}} \{\log \Pr(\boldsymbol{y}|\boldsymbol{x}) - \mu \log \Pr^{\text{src}}(\boldsymbol{y}) + \lambda \log \Pr^{\text{ext}}(\boldsymbol{y}) + \rho|\boldsymbol{y}|\} \quad (15)$$

where μ and λ are the weights to balance the contribution of the source and external LMs, respectively, and ρ for the label length reward $|\boldsymbol{y}|$. The external LM is trained on a target domain corpus (Fisher and Switchboard) and the source LM is trained only on the training transcripts.

Most of the above experimental settings and the training recipe follow those used in [13].

4.2. Results

SwitchOut Table 1 shows the WERs of the RNNT baseline and RNNT after prediction network input perturbation using SwitchOut. The WERs for the RNNT baseline are 11.6% on average for SWB and CH without an external LM and 10.2% on average with the external LM using density ratio LM fusion. WERs for SwitchOut perturbation under various temperatures, τ, are presented, among which $\tau = 0.1$ gives the best performance. Under this condition, Switchout perturbation gives 0.2% absolute improvement over the baseline in both test conditions.

Table 1: *WERs of baseline and SwitchOut perturbation under various temperatures τ.*

	w/o LM			w/ LM		
	swb	ch	avg	swb	ch	avg
baseline	**7.7**	**15.5**	**11.6**	**6.5**	**13.9**	**10.2**
$\tau = 0.1$	**7.5**	**15.3**	**11.4**	**6.4**	**13.5**	**10.0**
$\tau = 0.2$	7.3	15.5	11.4	6.2	13.8	10.0
$\tau = 0.3$	7.5	15.5	11.5	6.3	13.8	10.1
$\tau = 0.5$	7.5	15.6	11.6	6.4	13.9	10.2

Scheduled Sampling Table 2 shows the WERs of the RNNT baseline, which is the same as Table 1, and RNNT under scheduled sampling with various configurations. The token LM is a single layer LSTM trained on the character sequences of the training transcripts. There are two hyper-parameters under investigation in the experiments. One is the number of candidates (top$k$) from which the next token is sampled from the token LM. The other hyper-parameter is the teacher forcing probability p. Experiments are carried out by varying the top$k$ and teacher forcing probability. It can be observed from the table that scheduled sampling is helpful to improve the recognition performance. The best result is given by top$k = 3$ and $p = 0.9$ where the average WER is 11.2% without the external LM and 9.7% with the external LM. This amounts to 0.4% and 0.5% absolute improvement under the two test conditions, respectively. It also outperforms the best result under SwitchOut in Table 1 by 0.2% absolute without using the external LM and 0.3% absolute when using it. However, it is worth noting that scheduled sampling is more computationally demanding, as the sampling depends on the token history.

Input with i-vector We also train RNNT models using input with i-vectors to further improve the baseline. The input features are a concatenation of the previous 240-dimensional features and speaker-dependent 100-dimensional i-vectors [33, 34]. Therefore, the input dimensionality in this case is 340. We reduce the

Table 2: *WERs of baseline and scheduled sampling under various top k candidates and teacher forcing probability p.*

	w/o LM			w/ LM		
	swb	ch	avg	swb	ch	avg
baseline	**7.7**	**15.5**	**11.6**	**6.5**	**13.9**	**10.2**
top$k=1$, $p=0.9$	7.3	15.4	11.4	6.3	13.7	10.0
top$k=2$, $p=0.9$	7.4	15.2	11.3	6.4	13.7	10.1
top$k=3$, $p=0.9$	**7.2**	**15.2**	**11.2**	**6.1**	**13.3**	**9.7**
top$k=5$, $p=0.9$	7.3	15.2	11.3	6.2	13.5	9.9
top$k=1$, $p=0.8$	7.5	15.3	11.4	6.3	13.6	10.0
top$k=2$, $p=0.8$	7.4	15.3	11.4	6.3	13.5	9.9

hidden cells of the LSTM in the prediction network from 1,024 to 768. Table 3 presents the WERs of the baseline RNNT with i-vectors and the WERs under scheduled sampling with various top$k$ and teacher forcing probabilities. The WERs of the RNNT baseline with i-vector are 11.3% on average without using the external LM and 9.7% with the external LM. This matches the best performance reported in [13], which is a state-of-art result on SWB300. As can be seen from the table, with the help of scheduled sampling, the best result is obtained when top$k=4$ and $p=0.8$: 11.0% on average without the external LM and 9.6% with the external LM. We also investigate an annealing strategy where scheduled sampling is used up to 25 epochs, after which it is lifted and only ground truth label sequences are used. The learning rate for the 26$^{\text{th}}$ epoch is set to 1.2e-4 and annealed by $\frac{1}{\sqrt{2}}$ for each epoch afterwards. This can further improve the WERs when no external LM is used ($11.0\% \rightarrow 10.9\%$), but with the external LM used the WERs stay the same (9.6%).

Table 3: *WERs of baseline and scheduled sampling under various top k candidates and teacher forcing probability. Input acoustic features include i-vector.*

	w/o LM			w/ LM		
	swb	ch	avg	swb	ch	avg
baseline	**7.3**	**15.3**	**11.3**	**6.0**	**13.4**	**9.7**
top$k=1$, $p=0.9$	7.3	15.2	11.3	6.2	13.4	9.8
top$k=3$, $p=0.9$	7.3	15.1	11.2	6.3	13.4	9.9
top$k=5$, $p=0.9$	7.1	15.1	11.1	6.1	13.3	9.7
top$k=1$, $p=0.8$	7.4	15.0	11.2	6.1	13.5	9.8
top$k=2$, $p=0.8$	7.4	15.0	11.2	6.2	13.4	9.8
top$k=3$, $p=0.8$	7.1	15.0	11.1	6.0	13.2	9.6
+ anneal	7.0	14.7	10.9	6.0	13.2	9.6
top$k=4$, $p=0.8$	**7.0**	**15.0**	**11.0**	**6.0**	**13.2**	**9.6**
+ anneal	**6.9**	**14.9**	**10.9**	**5.9**	**13.2**	**9.6**
top$k=5$, $p=0.8$	6.9	15.4	11.2	6.0	13.4	9.7

5. Discussion

Both SwitchOut and scheduled sampling are shown to improve the performance of RNNT models. While SwitchOut perturbs the label sequence by uniformly sampling tokens from $\mathcal{Y}$ to replace the original token, scheduled sampling samples the next predicted token using a token LM given the observed history. Hence, scheduled sampling outperforms SwitchOut in this case to reduce exposure bias. This superior performance, however, comes with a higher computational cost in the training as one has to recurrently generate the history with perturbed tokens for each label sequence. Once the model is trained, the inference time of both strategies is no different from the RNNT baseline.

The RNNT baseline with i-vector in Table 3 with 6.0/13.4/9.7 WERs is one of the best single ASR systems with state-of-the-art performance [13]. With scheduled sampling, the WERs can be further improved to 5.9/13.2/9.6, which gives one of the best results on the SWB300 dataset. Table 4 summarizes the performance of best single models reported in the literature.

Table 4: *WERs of single models reported in literature on SWB300.*

system	model	swb	ch	avg
Park et al. / 2019 [27]	AED	6.8	14.1	10.5
Irie et al. / 2019 [35]	Hybrid	6.7	12.9	9.8
Tuske et al. / 2020 [17]	AED	6.4	12.5	9.5
Saon et al. / 2021 [13]	RNNT	6.3	13.1	9.7
This work	RNNT	5.9	13.2	9.6

By comparing Table 2 and Table 3, one can observe that an RNNT trained with scheduled sampling without using i-vector in the input (9.7%) matches the performance of an RNNT baseline using i-vector in the input (9.7%). Scheduled sampling can further improve the latter (9.6%). This observation is helpful, as computing i-vectors is computationally expensive and needs sufficient data from a single speaker. Therefore, in some real-world applications, including i-vectors in input acoustic features is either computationally inefficient or technically infeasible, especially when ASR is carried out in a streaming mode.

6. Summary

In this paper we introduced label-preserving perturbation to the input to the prediction network of an RNNT to reduce exposure bias. Two perturbation strategies were investigated, SwitchOut and scheduled sampling. SwitchOut randomly corrupts the ground truth label token sequence, while scheduled sampling draws the next token based on an additional token LM given the history. Both strategies have been shown to be helpful in reducing the WERs of a high-performance RNNT ASR model. Input perturbation based on scheduled sampling obtains one of best results on the SWB300 dataset.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Maganize*, pp. 82–97, November 2012.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing,*, vol. 20, no. 1, pp. 30–42, 2012.

[3] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[4] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Interspeech*, 2017, pp. 132–136.

[5] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Kluwer Academic Publishers, 1993.

[6] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 35th*

International Conference on Machine Learning (ICML), 2006, pp. 369–376.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[8] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4945–4949.

[9] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 577–585.

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

[11] A. Graves and A.-r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.

[12] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, S. Yuan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. y. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6381–6385.

[13] G. Saon, Z. Tueske, D. Bolanos, and B. Kingsbury, "Advancing RNN transducer technology for speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhit, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6000–6010.

[15] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 1171–1179.

[16] C.-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 *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4774–4778.

[17] Z. Tuske, G. Saon, K. Audhkhasi, and B. Kingsbury, "Single headed attention based sequence-to-sequence model for state-of-the-art results on switchboard," in *Interspeech*, 2020, pp. 551–555.

[18] P. Zhou, R. Fan, W. Chen, and J. Jia, "Improving generalization of transformer for speech recognition with parallel schedule sampling and relative positional embedding," *arXiv preprint arXiv:1911.00203*, 2019.

[19] X. Wang, H. Pham, Z. Dai, and G. Neubig, "SwitchOut: an efficient data augmentation algorithm for neural machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 856–861.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] G. Saon, Z. Tuske, and K. Audhkhasi, "Alignment-length synchronous decoding for RNN transducer," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7804–7808.

[22] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1992, pp. 517–520.

[23] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generation of speech-to-text," in *Proceedings of ICLRE*, 2004, pp. 69–71.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[25] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015, pp. 3586–3589.

[26] G. Saon, Z. Tuske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6261–6265.

[27] 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, pp. 2613–2617.

[28] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2013, pp. 1058–1066.

[29] K. Audhkhasi, G. Saon, Z. Tuske, B. Kingsbury, and M. Picheny, "Forget a bit to learn better: Soft forgetting for CTC-based automatic speech recognition," in *Interspeech*, 2019, pp. 2618–2622.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[31] D. P. Kingma and J. L. Ba, "ADAM: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[32] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 434–441.

[33] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing,*, vol. 19, no. 4, pp. 788–798, 2011.

[34] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using I-vectors," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013, pp. 55–59.

[35] K. Irie, A. Zeyer, R. Schluter, , and H. Ney, "Training language models for long-span cross-sentence evaluation," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 419–426.