A New Training Pipeline for an Improved Neural Transducer

Albert Zeyer\textsuperscript{1,2}, André Merboldt\textsuperscript{1}, Ralf Schlüter\textsuperscript{1,2}, Hermann Ney\textsuperscript{1,2}

\textsuperscript{1}Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52062 Aachen, Germany;
\textsuperscript{2}AppTek GmbH, 52062 Aachen, Germany;
\{zeyer, schlueuter, ney\}@cs.rwth-aachen.de, andre.merboldt@rwth-aachen.de

Abstract

The \textit{RNN transducer} is a promising end-to-end model candidate. We compare the original training criterion with the full marginalization over all alignments, to the commonly used maximum approximation, which simplifies, improves and speeds up our training. We also generalize from the original neural network model and study more powerful models, made possible due to the maximum approximation. We further generalize the output label topology to cover RNN-T, RNA and CTC. We perform several studies among all these aspects, including a study on the effect of external alignments. We find that the transducer model generalizes much better on longer sequences than the attention model. Our final transducer model outperforms our attention model on Switchboard 300h by over 6% relative WER.

Index Terms: RNN-T, RNA, CTC, max. approx.

1. Introduction & Related work

End-to-end models in speech recognition are models with a very simple decoding procedure, and often a simple training pipeline. Usually the model directly outputs characters, subwords or words. One of the earlier end-to-end approaches was \textit{connectionist temporal classification (CTC)} \cite{graves2013speech}. Most prominent is the \textit{encoder-decoder-attention model} which has shown very competitive performance \cite{wang2016sequence, ma2018end, chung2019acl, chung2019asru}. Once the streaming aspect becomes more relevant, or having a monotonicity constraint on the (implicit or explicit) alignment, the global attention model needs to be modified. Several ad-hoc solutions exists with certain shortcomings \cite{graesser1995alignment, bauer1996align}. The \textit{recurrent neural network transducer (RNN-T) model} \cite{graves2006connectionist, graves2012sequence} (or just \textit{transducer model}) is an alternative model where the outputs can be produced in a time-synchronous way, and thus is implicitly monotonic. Because of this property, RNN-T has recently gained interest \cite{wang2016sequence, ma2018end, chung2019acl, chung2019asru}. RNN-T can be seen as a slightly more powerful generalization of CTC.

Several variations of RNN-T exists, such as the recurrent neural aligner (RNA) \cite{schuster2012multilingual}, monotonous RNN-T \cite{schneider2017monotonous} or hybrid autoregressive transducer (HAT) \cite{schneider2018hybrid}. The explicit time-synchronous modeling also makes the alignment of labels explicit, and requires a blank or silence label. The alignment becomes a latent variable. Most existing work keeps the model simple enough such that the marginalization over all possible alignments can be calculated efficiently via the forward-backward algorithm \cite{rabiner1989fundamentals}. In case of RNA, an approximation is introduced.

Initializing the encoder parameters from another model (such as a CTC model) has often been done \cite{chen2016fast, brefeld2017fast, xiong2017multilingual, priegnitz2020almost}. Initializing some of the decoder parameters from a language model is common as well \cite{brefeld2017fast, brefeld2016improved}. Using an external alignment has been studied in \cite{brefeld2017fast, brefeld2016improved}.

Differences in time-synchronous models (such as hybrid hidden Markov model (HMM) - neural network (NN) \cite{priestley1990neural, priestley1995vector}) vs. label synchronous models (such as encoder-decoder-attention and segmental RNN) w.r.t. the alignment behavior are studied in \cite{bethard2015hierarchical}. Time-synchronous decoding is also possible on joint CTC-attention models \cite{zhou2018joint}.

2. Model

Let $x_1^T$ be the input sequence, which is encoded by a bidirectional LSTM \cite{hochreiter1997long} with time downsampling via max-pooling \cite{graves2013speech} and optional local windowed self-attention \cite{vaswani2017attention}. We define a discriminative model

$$p(y_i^N \mid x_1^T) = \sum_{\alpha} p(\alpha \mid \alpha_0, h_i^T),$$

where $\alpha_j \in \Sigma'$ := \{\{b\}\} $\cup$ $\Sigma$, where \{b\} is the \textit{blank label}. The output label topology $T$ over $\Sigma'$ defines the mapping on $t$, and generates the sequence $y_i^N$. More specifically, the topology $T$ defines $\Delta \tau(t, \alpha)$ $\geq$ 0 such that $t_{j+1} = t_j + \Delta \tau(t_j, \alpha_j)$, and $t_s = 1$, $t_f = T$, and $\Delta \tau(t, \alpha)$ $\geq$ 0 such that $t_{j+1} = t_j + \Delta \tau(t_j, \alpha_j)$, and $i_0 = 1$, $i_0 = N$. We study multiple variants of the label topology for $\alpha$. Emitting a \{b\} label will always consume a time frame, and \{b\} will be removed from the final output. We study three variants:

- \textit{CTC topology} \cite{graves2013speech}: Label emits time frame, repeated label will be collapsed. In this case, $U = T$, and $\Delta t \equiv 1$, $t_j = j$, and $\Delta \tau(t_j, \alpha_j) = 1_{\alpha_j \neq (b) \land \alpha_{j-1} \neq (b)}$.
- \textit{RNA topology} \cite{schuster2012multilingual} or \textit{monotonous RNN-T} \cite{schneider2017monotonous}: Label emits time frame, repeats will not be collapsed. $U = T$, and $\Delta t \equiv 1$, $t_j = j$, and $\Delta \tau(t, \alpha) = 1_{\alpha \neq (b)}$.
- \textit{RNN-T topology} \cite{schneider2018hybrid}: Label does not emit time frame, repeats will not be collapsed. $U = N + T$, and $\Delta \tau(t, \alpha) = 1_{t = \text{min}(b), \text{and } \Delta \tau(t, \alpha) = 1_{t \neq (b)}}$.

We generalize from the common RNN-T and RNA model and describe our decoder network in terms of a fast and slow RNN \cite{chung2019acl, ma2018end, chung2019asru}. The fast RNN iterates over $j \in \{1, \ldots, U\}$, while the slow RNN is calculated in certain sub frames $i_j \in \{1, \ldots, N\}$, whenever there was a new generated $y$. We visualize the unrolled decoder in Figure 1. The decoder for given frame $j$ is defined by

$$s_{j}^{\text{fast}} := \text{FastRNN} \left( s_{j-1}^{\text{fast}}, s_{j}^{\text{slow}}, \alpha_{j-1}, h_{t_j} \right),$$

$$s_{j}^{\text{slow}} := \text{SlowRNN} \left( s_{j-1}^{\text{slow}}, \alpha_{j-1}, h_{t_j} \right),$$

$$j' := \min \{ k \mid k \leq j, i_k = i_j \}. \quad \text{(last emit)}$$
Both the SlowRNN and the FastRNN are LSTMs in our baseline. Note that we have $y_{t-1} = \alpha_{t-1}$, and if we remove the dependency on $h$ in SlowRNN, and if we remove FastRNN and just set $s_{t-1}^ \text{fast} = h_{t-1}$, we get the original RNN-T model. To get the probability distribution for $\alpha$ over $\Sigma$, we could use a single softmax, as it was done for the original RNN-T and also RNA. Our baseline splits $\Sigma$ and $h$ explicitly into two separate probability distributions:

$$p(\alpha_{j} = h \mid ...):= \sigma(\text{Readout}(s_{j}^\text{fast}, s_{j}^\text{slow})), \quad p(\alpha_{j} \neq h \mid ...):= \sigma(-\text{Readout}(s_{j}^\text{fast}, s_{j}^{\text{slow}})), \quad q(\alpha_{j} \mid ...):= \text{softmax}(\text{Readout}(s_{j}^\text{fast}, s_{j}^\text{slow})), \quad \alpha_{j} \in \Sigma$$

where Readout is some feed-forward NN. This can also be interpreted as a hierarchical shallow softmax. HAT [25] uses a similar definition. Note that the expressive power is equivalent to the single distribution:

$$q(\alpha_{j} \mid ...):= \frac{p(\alpha_{j} = h \mid ...) \cdot q(\alpha_{j} \mid ...)}{\sum_{\alpha_{j}' \in \Sigma} p(\alpha_{j}' \mid ...) \cdot q(\alpha_{j}' \mid ...)} \cdot \alpha_{j} \in \Sigma$$

$$p(\alpha_{j} \neq h \mid ...):= 1 - p(\alpha_{j} = h \mid ...).$$

2.1. Training

The model will be trained by minimizing the loss

$$L := - \log p(y_{T}^N \mid x_{T}^N) = - \log \sum_{\alpha_{T}^N} p(\alpha_{T}^N \mid x_{T}^N).$$

The sum $\sum_{\alpha_{T}^N}$ is usually solved via dynamic programming [40] by iterating over $j \in \{1, \ldots, U\}$. When we have a dependence on individual $\alpha_{j}$ in the decoder, it is not possible to calculate the sum $\sum_{\alpha_{j}^N}$ efficiently. It is possible though to do an approximation in the recombination [23]. The simplest approximation is to only use a single item of the sum, specifically $\arg \max \alpha_{j}^N$, which is the well known maximum approximation, which is done via the Viterbi alignment, or some other external alignment, as it is the standard approach for hybrid HMM-NN models [28, 29]. We study two variants here:

- Exact calculation of the sum (when possible).
- Using the maximum approximation, or a variant using some external alignment.

In our case, we always keep the alignment fixed, which is an external alignment. This is in line with most other work which uses the max. approximation [29]. This has the disadvantage that we depend on a good external alignment, which complicates the training pipeline. But it comes with a number of advantages:

- We can train strictly more expressive models, which can potentially be more powerful, where the full sum cannot be calculated efficiently anymore.
- The training itself is simpler and reduces to simple frame-wise cross entropy training. This requires less computation and should be faster.
- It is more flexible, and we can use methods like chunking [29], focal loss [41], label smoothing [42].
- In addition, maybe it is more stable? Or we get faster convergence rate?

We also study the relevance of the type of external alignment. This is a forced alignment on some other unrelated model with the same output label topology. This other model can be a weaker model, only trained with the goal to generate the alignments. We study several variants of models for this task.

2.2. Decoding

We use beam search decoding with a fixed small beam size (12 hypotheses). The hypotheses in the beam are partially finished sequences $\alpha_{j}^N$ of the same length $j$. The pruning is based on the scores $p(\alpha_{j}^N \mid x_{T}^N)$. I.e. the decoding is time-synchronous (in case $U = T$), or synchronous over the axis $\{1, \ldots, U\}$ [22]. This is the same beam search algorithm and implementation as for our attention-based encoder-decoder model [2]. The only difference is that it runs over the axis $\{1, \ldots, U\}$.

As an optimization of the beam search space, we combine multiple hypotheses in the beam when they correspond to the same partial word sequence (after BPE merging), and we take the sum of their scores (which is another approximation, based on the model).

3. Experiments

We use RETURN [43] as the training framework, which builds upon TensorFlow [44]. For the full-sum experiments on the RNN-T label topology, we use warp-transducer\footnote{https://github.com/HawkAaron/warp-transducer}. Our current full-sum implementation on the other label topologies is a pure TensorFlow implementation of the dynamic-programming forward computation. Via auto-diff, this results in the usual forward-backward algorithm. This is reasonably fast, but still slower than a handcrafted pure CUDA implementation, and also slower than the simple CE training. Both training and decoding is done on GPU. We present the training speeds in Table 1. We publish all our code, config and full training pipeline to reproduce our results\footnote{https://github.com/rwh-16/returns-experiments/tree/master/2022-rnn-transducer}.

All the individual studies are performed on the Switchboard 300h English telephone speech corpus [45]. We use SpecAugment [3] as a simple on-the-fly data augmentation.

3.1. Full-sum vs. maximum approximation

We compare full-sum (FS) training vs. Viterbi training in Table 2. We observe that the full-sum training is more unstable, esp. in the beginning of the training, and thus leads to worse performance, within the same amount of training time.

For the maximum approximation, which is standard cross entropy, we use a fixed external alignment. We study the influence of the external alignment in Table 3. We see that a CTC model can be used to generate an alignment, but we also see that
Table 1: On Switchboard 300h. For each model, label topology, loss (full-sum (FS) or max. approx. (equivalent to cross entropy (CE))), and loss implementation (pure TensorFlow (TF), or CUDA), we compare the training time on a single GTX 1080 Ti GPU. This measures the whole training, not just the loss calculation.

| Model | Label Topology | Loss | Loss Impl. | # params [M] | time / epoch [min] |
|-------|----------------|------|------------|--------------|-------------------|
| Transd. | RNA | FS | TF | 147 | 306 |
| | CTC | CUDA | 219 | 326 |
| | RNN-T | CE | TF | 160 | 333 |
| Attention | – | – | – | 162 | 19.8 |

Table 2: On Switchboard 300h, transducer model, without external LM. Comparison of full-sum (FS) training and using the maximum approximation via an external Viterbi (Vit.) alignment, which is the standard cross entropy training. All models are trained for 25 epochs and share the same network topology, which has no label feedback to allow full-sum training. The models trained using external alignments both use CTC alignments, where label repetition is enabled for CTC-Vit and disabled for RNA-Vit.

| Label Topology | Training Criterion | WER [%] | SWB | CH | Σ | Σ |
|----------------|--------------------|---------|-----|----|----|----|
| RNA | FS | 11.5 | 23.4 | 17.5 | 16.5 |
| | Vit. | 10.1 | 20.4 | 15.2 | 14.8 |
| CTC | FS | 15.0 | 24.6 | 19.8 | 20.1 |
| | Vit. | 10.5 | 20.6 | 15.6 | 15.3 |
| RNN-T | FS | 11.6 | 22.3 | 17.0 | 16.4 |

Table 3: On Switchboard 300h, WER on Hub5’00. Comparing alignments (specifically the models used to get the alignments). The baselines are the transducer B1 and B2 models (see Section 3.2 for details), without external LM, with randomly initialized parameters, trained for 25 epochs. The model to calculate the Viterbi alignment was also trained always for 25 epochs.

| Alignment | WER [%] |
|-----------|---------|
| B1 | B2 |
| CTC-align 4l | 14.7 | 14.3 |
| CTC-align 6l | 14.7 | 14.5 |
| CTC-align 6l with prior (non-peaky) | 15.4 | 14.9 |
| CTC-align 6l, less training | 14.6 | 14.6 |
| Att. + CTC-align | 14.4 | 14.2 |
| Transducer-align | 14.2 | 14.1 |

Table 4: On Switchboard 300h, WER on Hub5’00. Ablations and variations. Using transducer baselines B1 and B2 (see Section 3.2 for details), without external LM. B2 is exactly the B1 baseline without local windowed self-attention.

| Variant | WER [%] |
|---------|---------|
| B1 | B2 |
| Baseline | 14.7 | 14.5 |
| No chunked training | 16.3 | 15.7 |
| No switchout | 15.0 | 14.5 |
| SlowRNN always updated (not slow) | 14.8 | 14.8 |
| No SlowRNN | 14.8 | 14.7 |
| No attention | 14.5 | * |
| FastRNN dim 128 → 512 | 14.3 | 14.7 |
| No label feedback | 14.9 | 14.5 |
| No encoder feedback to SlowRNN | 14.9 | 14.5 |
| No separate blank sigmoid | 14.9 | 14.9 |

other models produce better alignments for our purpose. Specifically, using a transducer model (trained from scratch with full-sum) to generate the alignment seems to work best. This is as expected, as this is the most consistent setup.

3.2. Ablations and variations

Along our research on training transducer models, we came up with many variants, until we eventually ended up with the baselines B1 and B2. Both transducer baselines use the CTC label topology with a separate sigmoid for the blank label (similar as in [25]). We use CE training using the fixed alignment CTC-align 6l (as in Table 3). Sequences of the training data are cut into chunks [29]. We use focal loss [41], an additional auxiliary CTC loss on the encoder (for regularization [2, 46]), dropout [47], dropconnect [48] (weight dropout) for the FastRNN, and switchout [49] (randomly switch labels for label feedback). B1 uses local windowed self-attention, while B2 does not, which is the only difference between B1 and B2. Some of these tricks were copied from our hybrid HMM-NN model [29]. Based on these baselines, we want to see the effect of individual aspects of the model or training. We summarized the variations and ablations in Table 4. We see that chunked training greatly helped for CE training, which is consistent with the literature [50]. We find that the results about label feedback are not conclusive. Having the separate sigmoid for blank seems to help.

3.3. Output label topology

We compare the output label topology in Table 2. We find that the RNN-T topology seems to perform best, followed by RNA, and CTC is worse. We note that this result is inconsistent to earlier results, where CTC looked better than RNA. However, when we repeat the RNA vs. CTC comparison on the B2 model with CE training, we also see that RNA performs better (14.2% vs 14.5% WER on Hub5’00). For simplicity, we did not follow the RNN-T topology further in this work. Also, because of our earlier results, we focused more on the CTC topology.

3.4. Importing existing parameters

For faster and easier convergence, it can be helpful to import existing model parameters into our RNA model. If we do not use the full-sum training, we anyway make use of an external alignment, which comes from some other model, so it might make sense to reuse these parameters. We collect our results in Table 5. We see that the CTC model parameters (of the same model which was used to create the alignment) seem to be suboptimal, and training from scratch performs better. The encoder of an attention-based encoder-decoder model seems to be very helpful. Importing the model itself, i.e. effectively training twice as long, helps just as much. However, we can also use the attention-based encoder-decoder model with an additional CTC layer on-top of the encoder to generate the alignments, and then get rid of the pure CTC model. We also tried to initialize the SlowRNN with the parameters of a LM but this had no effect.

3.5. Beam search decoding

We study different beam sizes, and compare the attention model and our RNA model. For RNA, we also implemented a variation of the beam search where hypotheses corresponding to the same word sequence (i.e. after collapsing label repetitions, re-
moving blank, and BPE merging) were recombined together by taking their sum (in log space) and only the best hypothesis survives. The results are in Table 6. In all cases, the WER seems to saturate for beam size $\geq 8$.

### 3.6. Generalization on longer sequences

It is known that global attention models do not generalize well to longer sequences than seen during training [51–53]. Expanding the attention process itself has problems with this. The alignment process in our time-synchronous models is more explicit, and at least this aspect should have no problems in generalizing to any sequence length. To analyze that, we take the Switchboard eval corpora and concatenate every $C$ consecutive sequences within a recording. By increasing $C$, we also increase the average sequence length. We show the results for different models in Table 7. We report the WER on RT03S to minimize over-fitting effects. We can see that both the attention model and the transducer model degrade with longer sequences, but the attention model performs much worse, i.e., the transducer model has generalized much better. The remaining small degradation of the transducer could also be explained by unusual sentence boundaries. We note that this small degradation looks better than reported in the literature (relatively) [52–54].

### 3.7. Overall performance

We compare our final transducer model to our attention model, and to other results from the literature in Table 8. We observe that many other works train for much longer, and there seems to be a correlation between training time and WER. We found that our final transducer model is better than our attention model, although it needs the preprocessing step to get an alignment.

### 4. Conclusions

We found that the maximum approximation for transducer models greatly simplifies, speeds up and improves our transducer training, and allows for a wide range of further tricks. We gain interesting insights regarding model behaviour in decoding. Finally, we achieve good results compared to the literature within much less training time. Our final transducer model is better than our attention model, and also generalizes much better on longer sequences.

### 5. Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 694537, project “SEQCLAS”) and from a Google Focused Award. The work reflects only the authors’ views and none of the funding parties is responsible for any use that may be made of the information it contains.

### 6. References

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in IJCL. ACM, 2006, pp. 369–376.

[2] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, “Improved training of end-to-end attention models for speech recognition,” in Inter-speech, Hyderabad, India, Sep. 2018.

[3] 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 Proc. Inter-speech 2019, 2019, pp. 2613–2617.

[4] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, “A comparison of Transformer and LSTM encoder decoder models for ASR,” in ASRU, Sentosa, Singapore, Dec. 2019, pp. 8–15.

[5] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, “Single headed attention based sequence-to-sequence model for state-of-

---

**Table 5:** On Switchboard 300h, WER on Hub5’00. Varying the imported model params, for transducer baseline models B1 and B2, with CTC topology, trained with CE using a fixed external alignment (CTC-align 6l), without external LM. Trained for 25 epochs, with randomly initialized parameters, except of the imported ones. The imported models themselves are also trained for 25 epochs.

| Model | Merge | WER [%] |
|-------|-------|---------|
|       |       | B1      |
|       |       | B2      |
| None  |       | 14.7    |
| CTC as encoder |       | 15.4 |
| Att. encoder |       | 14.2 |
| Transducer (itself) |       | 13.7 |

**Table 6:** On Switchboard 300h, WER on RT03S. The transducer uses the CTC-label topology. Without external LM. Comparison of performance on different beam sizes. We optionally recombine hypotheses in the beam corresponding to the same word sequence (after collapsing repetitions, removing blank, and BPE merging).

| Model | Merge | WER [%] |
|-------|-------|---------|
| Att.  | no    |         |
| Transd. | yes  |         |

**Table 7:** On Switchboard 300h, WER on RT03S. The transducer uses the CTC output topology. Without external LM. Comparison of performance on varying sequence lengths, by concatenating every $C$ consecutive sequences.

| $C$ | Seq. length [secs] | WER [%] |
|-----|---------------------|---------|
|     | mean±std            | min-max |
| 1   | 2.71±2.38           | 0.17–34.59 |
| 2   | 7.83±4.96           | 0.42–63.70 |
| 4   | 17.74±9.25          | 0.42–91.62 |
| 10  | 45.57±19.93         | 0.74–126.32 |
| 20  | 86.37±39.58         | 0.74–194.10 |
| 30  | 122.10±57.28        | 1.17–297.15 |
| 100 | 290.58±53.50        | 8.34–309.14 |

**Table 8:** On Switchboard 300h, comparing the final results our our transducer model to our attention model, and to other attention models from the literature. One big difference in varying results is the different amount of training time, which we state as number of epochs.

**Work** | **Label Type** | **#Ep** | **LM** | **WER [%]** | **SWB** | **CH** | **Σ** | **Hubs** | **Σ** | **RT** | **Σ** |
|--------|--------------|---------|--------|-------------|--------|-------|------|---------|------|-------|------|
| [55]   | Phone        | 4.5k    | 13     | yes         | 9.6    | 18.5  | 14.0 | 14.1    |      |       |      |
| [4]    | BPE          | 1k      | 33     | no          | 10.1   | 20.6  | 15.4 | 14.7    |      |       |      |
| [56]   | BPE          | 4k      | 50     | yes         | 8.8    | 17.2  | 13.0 |         |      |       |      |
| [57]   | BPE          | 2k      | 100    | yes         | 9.0    | 18.1  | 13.6 |         |      |       |      |
| [58]   | BPE          | 500     | 150    | yes         | 7.9    | 10.1  | 14.5 |         |      |       |      |
| [5]    | BPE          | 600     | 250    | no          | 7.6    | 14.6  |      |         |      |       |      |
| [3]    | WPM          | 1k      | 760    |            | 7.2    | 14.6  |      |         |      |       |      |

**Ours**

| Model | Merge | WER [%] |
|-------|-------|---------|
| Att.  | BPE   | 25      |
| Transd. | 50    |

This work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 694537, project “SEQCLAS”) and from a Google Focused Award. The work reflects only the authors’ views and none of the funding parties is responsible for any use that may be made of the information it contains.
A. Tripathi, H. Lu, H. Sak, and H. Soltau, "Monotonic recursive speech recognition," in \textit{Interspeech}, 2017.

H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural network transducer and decoding strategies," in \textit{Proc. of Interspeech}, 2013.

R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and H. Ney, "Return as a generic flexible neural toolkit with application to translation and speech recognition," in \textit{Annual Meeting of the Assoc. for Computational Linguistics}, Melbourne, Australia, Jul. 2018.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," \textit{The Journal of Machine Learning Research}, vol. 15, no. 1, pp. 1929–1958, 2014.

W. Chen, Y. Wu, R. Pang, T. N. Sainath, P. Nguyen, L. Cao, and Y. Wu, "Analysis of RNN-T models fail to generalize to out-of-domain audio: Causes and solutions," Preprint \texttt{arXiv:2005.03271}, 2020.

C.-C. Chiu, W. Han, Y. Zhang, R. Pang, S. Krishchenko, P. Nguyen, A. Narayanan, H. Liu, S. Zhang, A. Kannan, R. Prabhavalkar, Z. Chen, T. N. Sainath, and Y. Wu, "A comparison of end-to-end models for long-form speech recognition," in \textit{ASRU}, 2019, pp. 889–896.

C.-C. Chiu, A. Narayanan, W. Han, R. Prabhavalkar, Y. Zhang, N. Jaitly, R. Pang, T. N. Sainath, P. Nguyen, L. Cao, and Y. Wu, "RNN-T models fail to generalize to out-of-domain audio: Causes and solutions," Preprint \texttt{arXiv:2005.03271}, 2020.

M. Zeinaldeen, A. Zeyer, R. Schlüter, and H. Ney, "Return as the principle of history compression," \textit{Neural Computation}, Sep. 2015, pp. 1929–1958. [Online]. Available: \url{http://proceedings.mlr.press/v28/zeinaldeen15.html}

X. Wang, H. Pham, Z. Dai, and G. Neubig, "SwitchOut: an efficient data augmentation algorithm for neural machine translation," in \textit{EMNLP}, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 856–861. [Online]. Available: \url{https://www.aclweb.org/anthology/D18-1100}

A. Zeyer, E. Beck, R. Schlüter, and H. Ney, "In the context of generalized full-svm training," in \textit{Interspeech}, Stockholm, Sweden, Aug. 2017, pp. 944–948.

J. Rosendahl, V. K. Tran, Y. Wang, and H. Ney, "Analysis of positional encodings for neural machine translation," in \textit{IWSLT}, Hong Kong, China, Nov. 2017.

A. Narayanan, R. Prabhavalkar, C.-C. Chiu, D. Rybach, T. N. Sainath, and T. Strohman, "Recognizing long-form speech using deep streaming end-to-end models," in \textit{ASRU}, 2019.

C.-C. Chiu, W. Han, Y. Zhang, R. Pang, S. Krishchenko, P. Nguyen, A. Narayanan, H. Liu, S. Zhang, A. Kannan, R. Prabhavalkar, Z. Chen, T. N. Sainath, and Y. Wu, "A comparison of end-to-end models for long-form speech recognition," in \textit{ASRU}, 2019, pp. 889–896.

C.-C. Chiu, A. Narayanan, W. Han, R. Prabhavalkar, Y. Zhang, N. Jaitly, R. Pang, T. N. Sainath, P. Nguyen, L. Cao, and Y. Wu, "RNN-T models fail to generalize to out-of-domain audio: Causes and solutions," Preprint \texttt{arXiv:2005.03271}, 2020.

M. Zeinaldeen, A. Zeyer, R. Schlüter, and H. Ney, "Layer-normalized lstm for hybrid-bnn and end-to-end asr," in \textit{ICASSP}, Barcelona, Spain, May 2016.

T.-S. Nguyen, S. Stueker, J. Niehues, and A. Waibel, "Improving convolutional neural networks for automatic speech recognition with global context," Preprint \texttt{arXiv:2003.03191}, 2020.

G. Saon, Z. Tüske, and K. Audhkhasi, "Alignment-length synchronous encoding for RNN transducer," in \textit{ICASSP}, 2020, pp. 7804–7808.

H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence tasks," \textit{Proc. of Interspeech}, 2017.

A. Tripathi, H. Lu, H. Sak, and S. Kumar, "Transformer transducer: A streamable speech recognition model with Transformer encoders and RNN-T loss," in \textit{ICASSP}, IEEE, 2020, pp. 7829–7833.

H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, "Exploring pre-training with alignments for RNN transducer based speech recognition," in \textit{ICASSP}, IEEE, 2020, pp. 6069–6073.

Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer aligner: A streamable speech recognition model with Transformer encoders and RNN-T loss," in \textit{ICASSP}, IEEE, 2020, pp. 7049–7053.

M. Ghodsi, X. Liu, J. Arief, R. Cabrera, and E. Weinstein, "RNN-transducer with stateless prediction network," in \textit{ICASSP}, 2020, pp. 7977–7981.

W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "ContextNet: Improving convolutional neural networks for automatic speech recognition with global context," Preprint \texttt{arXiv:2005.03191}, 2020.

B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, "Towards fast and accurate streaming end-to-end ASR," in \textit{ICASSP}, 2020, pp. 6069–6073.

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 \textit{ICASSP}, IEEE, 2020, pp. 7829–7833.

H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, "Exploring pre-training with alignments for RNN transducer based speech recognition," in \textit{ICASSP}, IEEE, 2020, pp. 7049–7053.

W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "ContextNet: Improving convolutional neural networks for automatic speech recognition with global context," Preprint \texttt{arXiv:2005.03191}, 2020.