SELF-ATTENTION NETWORKS FOR CONNECTIONIST TEMPORAL CLASSIFICATION IN SPEECH RECOGNITION

Julian Salazar* Katrin Kirchhoff† Zhiheng Huang*

*Amazon AI †University of Washington {julsal,katrinki,zhiheng}@amazon.com

ABSTRACT

The success of self-attention in NLP has led to recent applications in end-to-end encoder-decoder architectures for speech recognition. Separately, connectionist temporal classification (CTC) has matured as an alignment-free, non-autoregressive approach to sequence transduction, either by itself or in various multitask and decoding frameworks. We propose SAN-CTC, a deep, fully self-attentional network for CTC, and show it is tractable and competitive for end-to-end speech recognition. SAN-CTC trains quickly and outperforms existing CTC models and most encoder-decoder models, with character error rates (CERs) of 4.7% in 1 day on WSJ eval92 and 2.8% in 1 week on LibriSpeech test-clean, with a fixed architecture and one GPU. Similar improvements hold for WERs after LM decoding. We motivate the architecture for speech, evaluate position and downsampling approaches, and explore how label alphabets (character, phoneme, subword) affect attention heads and performance.

Index Terms— speech recognition, connectionist temporal classification, self-attention, multi-head attention, end-to-end

1. INTRODUCTION

Connectionist temporal classification (CTC) [1] has matured as a scalable, end-to-end approach to monotonous sequence transduction tasks like handwriting recognition [2], action labeling [3], and automatic speech recognition (ASR) [1, 4–11], sidestepping the label alignment procedure required by traditional hidden Markov model plus neural network (HMM-NN) approaches [12]. However, the most successful end-to-end approach to general sequence transduction has been the encoder-decoder [13] with attention [14]. Though first used in machine translation, its generality makes it useful to ASR as well [15–20]. However, the lack of enforced monotonicity makes encoder-decoder ASR models difficult to train, often necessitating thousands of hours of data [18], careful learning rate schedules [19], pretraining [20], or auxiliary CTC losses [17–20] to approach state-of-the-art results. The decoders are also typically autoregressive at prediction time [13, 22], restricting inference speed.

Both approaches have conventionally used recurrent layers to model temporal dependencies. As this hinders parallelization, later works proposed partially- or purely-convolutional CTC models [8, 11] and convolution-heavy encoder-decoder models [16] for ASR. However, convolutional models must be significantly deeper to retrieve the same temporal receptive field [23]. Recently, the mechanism of self-attention [24] was proposed, which uses the whole sequence at once to model feature interactions that are arbitrarily distant in time. Its use in both encoder-decoder and feedforward contexts has led to faster training and state-of-the-art results in translation (via the Transformer [23]), sentiment analysis [25], and other tasks. These successes have motivated preliminary work in self-attention for ASR. Time-restricted self-attention was used as a drop-in replacement for individual layers in the state-of-the-art lattice-free MMI model [26], an HMM-NN system. Hybrid self-attention/LSTM encoders were studied in the context of listen-attend-spell (LAS) [27], and the Transformer was directly adapted to speech in [19, 28, 29]: both are encoder-decoder systems.

In this work, we propose and evaluate fully self-attentional networks for CTC (SAN-CTC). We are motivated by practicality: self-attention could be used as a drop-in replacement in existing CTC-like systems, where only attention has been evaluated in the past [30, 31]: unlike encoder-decoder systems, SAN-CTC is able to predict tokens in parallel at inference time; an analysis of SAN-CTC is useful for future state-of-the-art ASR systems, which may equip self-attentive encoders with auxiliary CTC losses [17, 20]. Unlike past works, we do not require convolutional frontends [19] or interleaved recurrences [27] to train self-attention for ASR. In Section 2, we motivate the model and relevant design choices (position, downsampling) for ASR. In Section 3 we validate SAN-CTC on the Wall Street Journal and LibriSpeech datasets by outperforming existing CTC models and most encoder-decoder models in character error rates (CERs), with fewer parameters or less training time. Finally, we train our models with different label alphabets (character, phoneme, subword), use WFST decoding to give word error rates (WERs), and examine the learned attention heads for insights.

2. MODEL ARCHITECTURES FOR CTC AND ASR

Consider an input sequence of $T$ feature vectors, viewed as a matrix $X \in \mathbb{R}^{T \times d_F}$. Let $L$ denote the (finite) label alphabet, and denote the output sequence as $y = (y_1, \ldots, y_T) \in L^U$. In ASR, $X$ is the sequence of acoustic frames, $\mathbb{L}$ is the set of graphemes/phonemes/wordpieces, and $y$ is the corresponding ground-truth transcription over $\mathbb{L}$.

For CTC, one assumes $U \leq T$ and defines an intermediate alphabet $L' = \mathbb{L} \cup \{-\}$, where ‘-’ is called the blank symbol. A path is a $T$-length sequence of intermediate labels $\pi = (\pi_1, \ldots, \pi_T) \in \mathbb{L}^T$. Paths are related to output sequences by a many-to-one mapping that collapses repeated labels then removes blank symbols:

$$B : \mathbb{L}^T \rightarrow L^\leq T,$$

e.g., $(a, b, -, - , b, - , a) \mapsto (a, b, b, a)$. (1)

In this way, paths are analogous to framewise alignments in the HMM-NN framework. CTC models the distribution of sequences by marginalizing over all paths corresponding to an output:

$$P(y \mid X) = \sum_{\pi \in B^{-1}(y)} P(\pi \mid X).$$

Finally, CTC models each $P(\pi \mid X)$ non-autoregressively, as a sequence of conditionally-independent outputs:

$$P(\pi \mid X) = \prod_{t=1}^{T} P(\pi_t \mid X).$$
This model assumption means each \( P(\tau_i, t \mid X) \) could be computed in parallel, after which one can do prediction via beam search, or training with gradient descent using the objective \( \mathcal{L}_{\text{CTC}}(X, y) = -\log P(y \mid X) \); the order-monotonicity of \( B \) ensures \( \mathcal{L}_{\text{CTC}} \) can be efficiently evaluated with dynamic programming \( \text{(1|4)} \).

2.1. Recurrent and convolutional models

In practice, one models \( P(\tau_i, t \mid X) \) with a neural network. As inspired by HMMs, the model simplification of conditional independence can be tempered by multiple layers of (recurrent) bidirectional long short-term memory units (BLSTMs) \( \text{[1|4]} \). However, these are computationally expensive (Table \( \text{[1]} \)), leading to simplifications like gated recurrent units (GRUs) \( \text{[8|12]} \), furthermore, the success of the ReLU \( (x) = \max(0, x) \) nonlinearity in preventing vanishing gradients enabled the use of vanilla bidirectional recurrent deep neural networks (BRDNNs) \( \text{[5|6|33]} \) to further reduce operations per layer.

Convolutions over time and/or frequency were first used as initial layers to recurrent neural models, beginning with HMM-NNs \( \text{[44]} \) and later with CTC, where they are viewed as promoting invariance to temporal and spectral translation in ASR \( \text{[8]} \), or image translation in handwriting recognition \( \text{[15]} \); they also serve as a form of dimensionality reduction (Section \( \text{[2.4]} \)). However, these networks were still bottlenecked by the sequentiality of operations at the recurrent layers, leading \( \text{[8]} \) to propose row convolutions for unidirectional RNNs, which had finite lookaheads to enable online processing while having some future context.

This led to convolution-only CTC models for long-range temporal dependencies \( \text{[9|11]} \). However, these models have to be very deep (e.g., 17-19 convolutional layers on LibriSpeech \( \text{[23]} \)) to cover the same context (Table \( \text{[1]} \)). While in theory, a relatively local context could suffice for ASR, this is complicated by alphabets which violate the conditional independence assumption of CTC (e.g., English characters \( \text{[56]} \)). Wide contexts also enable incorporation of noise/speaker contexts, as \( \text{[27]} \) suggest regarding the broad-context attention heads in the first layer of their self-attentional LAS model.

2.2. Motivating the self-attention layer

We now replace recurrent and convolutional layers for CTC with self-attention \( \text{[24]} \). Our proposed framework (Figure \( \text{[1|h]} \)) is built around self-attention layers, as used in the Transformer encoder \( \text{[22]} \), previous explorations of self-attention in ASR \( \text{[19|27]} \), and defined in Section \( \text{[2.3]} \). The other stages are downsampling, which reduces input length \( T \) via methods like those in Section \( \text{[2.4]} \) embedding, which learns a \( d_h \)-dim. embedding that also describes token position (Section \( \text{[2.3]} \)) and projection, where each final representation is mapped framewise to logits over the intermediate alphabet \( L' \).

A layer’s structure (Figure \( \text{[1|b]} \)) is composed of two sublayers. The first implements self-attention, where the success of attention in CTC and encoder-decoder models \( \text{[12|31]} \) is parallelized by using each position’s representation to attend to all others, giving a contextualized representation for that position. Hence, the full receptive field is immediately available at the cost of \( O(T^2) \) inner products (Table \( \text{[1]} \)), enabling richer representations in fewer layers.

![Fig. 1: Self-attention and CTC](image)

| Model                  | Operations per layer | Sequential operations | Maximum path length |
|------------------------|----------------------|-----------------------|---------------------|
| Recurrent              | \( O(Td^2) \)        | \( O(T) \)            | \( O(T) \)          |
| Convolutional          | \( O(kTd^2) \)       | \( O(1) \)            | \( O(T/k) \)        |
| Convolutional (strided/dilated/pooled) | \( O(kTd^2) \) | \( O(1) \) | \( O(T/k) \) |
| Self-attention (restricted) | \( O(T^2d) \) | \( O(T) \) | \( O(T/k) \) |

Table 1: Operation complexity of each layer type, based on \( \text{(22)} \). \( T \) is input length, \( d \) is no. of hidden units, and \( k \) is filter/context width.

We also see inspiration from convolutional blocks: residual connections, layer normalization, and tied dense layers with ReLU for representation learning. In particular, multi-head attention is akin to having a number of infinitely-wide filters whose weights adapt to the content (allowing fewer “filters” to suffice). One can also assign interpretations; for example, \( \text{[27]} \) argue their LAS self-attention heads are differentiated phoneme detectors. Further inductive biases like filter widths and causality could be expressed through time-restricted self-attention \( \text{[26]} \) and directed self-attention \( \text{[25]} \), respectively.

2.3. Formulation

Let \( H \in \mathbb{R}^{T \times d_h} \) denote a sublayer’s input. The first sublayer performs multi-head, scaled dot-product, self-attention \( \text{[22]} \). For each head \( i \) of \( n_{head} \), we learn linear maps \( W_Q^{(i)} \in \mathbb{R}^{d_h \times d_h}, W_K^{(i)} \in \mathbb{R}^{d_h \times d_h}, W_V^{(i)} \in \mathbb{R}^{d_h \times d_h} \). Left multiplication by \( H \) gives the queries \( Q^{(i)} \), keys \( K^{(i)} \), and values \( V^{(i)} \) of the \( i \)-th head, which combine to give

\[
\text{HdAtt}^{(i)} = \sigma \left( Q^{(i)} K^{(i)\top} \sqrt{d_k} \right) V^{(i)},
\]

where \( \sigma \) is row-wise softmax. Heads are concatenated along the \( d_h/n_{head} \) axis to give MitHdAtt = \( [\text{HdAtt}^{(1)}, \ldots, \text{HdAtt}^{(n_{head})}] \). The second sublayer is a position-wise feed-forward network \( \text{[22]} \)

\[
\text{FFN}(H) = \text{ReLU}(HW_1 + b_1)W_2 + b_2 \text{ where parameters } W_1 \in \mathbb{R}^{d_h \times d_h}, b_1 \in \mathbb{R}^{d_h}, W_2 \in \mathbb{R}^{d_h \times d_h}, b_2 \in \mathbb{R}^{d_h}, \text{are learned},
\]

with the biases \( b_1, b_2 \) broadcasted over all \( T \) positions. This sublayer aggregates the multiple heads at time \( t \) into the attention layer’s final output at \( t \). All together, the layer is given by:

\[
\text{MidLyr}(H) = \text{LN}(\text{MitHdAtt}(H) + H),
\]

\[
\text{SelfAttLyr}(H) = \text{LN}(\text{FFN}(\text{MidLyr}(H)) + \text{MidLyr}(H)).
\]

2.4. Downsampling

In speech, the input length \( T \) of frames can be many times larger than the output length \( U \), in contrast to the roughly word-to-word setting of machine translation. This is especially prohibitive for self-attention in terms of memory: recall that an attention matrix
of dimension \( Q^{(i)} K^{(i)\top} \in \mathbb{R}^{T \times T} \) is created, giving the \( T \) factor in Table 1. A convolutional frontend is a typical downsampling strategy [8, 19]; however, we leave integrating other layer types into SAN-CTC as future work. Instead, we consider three fixed approaches, from least- to most-preserving of the input data: subsampling, which only takes every \( k \)-th frame; pooling, which aggregates every \( k \) consecutive frames via a statistic (average, maximum); reshaping, where one concatenates \( k \) consecutive frames into one [27]. Note that CTC will still require \( U \leq T/k \), however.

2.5. Position

Self-attention is inherently content-based [22], and so one often encodes position into the post-embedding vectors. We use standard trigonometric embeddings, where for \( 0 \leq i \leq d_{\text{emb}}/2 \), we define

\[
\text{PE}(t, 2i) = \sin \frac{t \cdot (2i)}{10000^{2i/d_{\text{emb}}}}, \quad \text{PE}(t, 2i + 1) = \cos \frac{t \cdot (2i + 1)}{10000^{2i/d_{\text{emb}}}}.
\]

for position \( t \). We consider three approaches: content-only [21], which forgoes position encodings; additive [19], which takes \( d_{\text{emb}} = d_h \) and adds the encoding to the embedding; and concatenative, where one takes \( d_{\text{emb}} = 40 \) and concatenates it to the embedding. The latter was found necessary for self-attentional LAS [27], as additive encodings did not give convergence. However, the monotonicity of CTC is a further positional inductive bias, which may enable the success of content-only and additive encodings.

3. EXPERIMENTS

We take \( (n_{\text{layers}}, d_h, n_{\text{heads}}, d_t) = (10, 512, 8, 2048) \), giving \approx 30M parameters. This is on par with models on WSJ (10-30M) [3, 19] and an order of magnitude below models on LibriSpeech (100-250M) [8, 23]. We use MXNet [37] for modeling and Kaldi/EESEN [7, 38] for data preparation and decoding. Our self-attention code is based on GluonNLP’s implementation. At train time, utterances are sorted by length: we exclude those longer than 1800 frames (<1% of each training set). We take a window of 25ms, a hop of 10ms, and concatenate cepstral mean-variance normalized features with temporal downsampling; in later layers, unlike [27] we still see a differentiation of variances. Inspired by [26], we further plot along with wide-context heads; in later layers, unlike [27] we still see a differentiation of variances, which forgoes position encodings; additive encodings did not give convergence. However, the monotonicity of CTC is a further positional inductive bias, which may enable the success of content-only and additive encodings.

3.1. Wall Street Journal (WSJ)

We train both character- and phoneme-label systems on the 80-hour WSJ training set to evaluate our architectural choices. Similar to [17, 19], we use 40-dim. mel-scale filter banks and hence 120-dim. features. We warmup for 8000 steps, use a dropout of 0.2, and switch schedules at epoch 40. For the WSJ dataset, we compare with similar MLE-trained, end-to-end, open-vocabulary systems in Table 2. We get an eval92 CER of 4.7%, outdoing all previous CTC-like results except 4.6% with a trainable frontend [40]. We use the provided extended 3-gram LM to retrieve WERs. For phoneme training, our labels come from the CMU pronunciation lexicon [Table 5]. These models train in one day (Tesla V100), comparable to the Speech Transformer [19]; however, SAN-CTC gives further benefits at inference time as token predictions are generated in parallel.

We also evaluate design choices in Table 4. Here, we consider the effects of downsampling and position encoding on accuracy for our fixed training regime. We see that unlike self-attentional LAS [27], SAN-CTC works respectfully even with no position encoding; in fact, the contribution of position is relatively minor (compare with [21], where location in an encoder-decoder system improved CER by 3% absolute). Lossy downsampling appears to preserve performance in CER while degrading WER (as information about frame transitions is lost). We believe these observations align with the monotonicity and independence assumptions of CTC.

Inspired by [27], we plot the standard deviation of attention weights for each head as training progresses; see Figure 2 for details. In the first layers, we similarly observe a differentiation of variances, along with wide-context heads; in later layers, unlike [27] we still see mild differentiation of variances. Inspired by [29], we further plot the attention weights relative to the current time position (here, per head). Character labels gave forward- and backward-attending heads (incidentally, averaging these would retrieve the bimodal distribution

\[ 1 \text{Rescaling so that these differences also have var. } \approx 1 \text{ helped WSJ training.} \]
Fig. 2: Average attention weights over WSJ’s test set. The left half is a character model; the right half is a phoneme model. Each curve corresponds to a head, at layers 1 and 10, of representative SAN-CTC models (reshape + additive embedding). The first row charts standard deviation over time; the second row charts the attention magnitudes relative to position, where 0 represents attending to the same position.

| Downsampling    | Position embedding | dev93  |
|-----------------|--------------------|--------|
| reshape content-only |                   | 7.62   |
| reshape additive     |                   | 7.10   |
| reshape concat.      |                   | 7.10   |
| pooling (max)       | additive          | 7.15   |
| pooling (avg)       | additive          | 6.82   |
| subsample          | additive none     | none   |

Table 4: SAN-CTC character model on WSJ with modifications.

in [26]) at all layers. This suggests a gradual expansion of context over depth, as is often engineered in convolutional CTC. This also suggests possibly using fewer heads, directed self-attention [25], and restricted contexts for faster training (Table 1). Phoneme labels gave a sharp backward-attending head and more diffuse heads. We believe this to be a symptom of English characters being more context-dependent than phonemes (for example, emitting ‘tt’ requires looking ahead, as ‘–’ must occur between two runs of ‘t’ tokens).

3.2. LibriSpeech

We give the first large-scale demonstration of a fully self-attentional ASR model using the LibriSpeech ASR corpus [42], an English corpus produced from audio books giving 960 hours of training data. We use 13-dim. mel-freq. cepstral coeffs. and hence 39-dim. features. We double the warmup period, use a dropout of 0.1, and switch schedules at epoch 30. Using character labels, we attained a test-clean CER of 2.8%, outdoing all previous end-to-end results except OCD training [41].

| Model                              | Tok. | test-clean | test-other |
|------------------------------------|------|------------|------------|
| CTC/ASG (Wav2Letter) [9]           | chr. | 6.9        | 7.2        |
| CTC (DS1-like) [33, 43]            | chr. | 6.5        | —          |
| Enc-Dec (4-4) [44]                 | chr. | 6.5        | 18.1       |
| Enc-Dec (6-1) [45]                 | chr. | 4.5        | 11.6       |
| CTC (DS2-like) [6, 12]             | chr. | 5.7        | 15.2       |
| Enc-Dec+CTC (6-1, pretr.) [20]     | 10k  | 4.8        | 15.3       |
| CTC/ASG (Gated CNN) [23]           | chr. | 4.8        | 14.5       |
| Enc-Dec (2.6-1) [41]               | 10k  | 2.9        | 8.4        |
| CTC (SAN), reshape, additive       | chr. | 3.2        | 5.2        |
| + label smoothing, $\lambda = 0.05$ | chr. | 3.5        | 11.3       |
| CTC (SAN), reshape, concat.        | chr. | 2.8        | 4.8        |

Table 5: End-to-end, MLE-based, open-vocab. models trained on LibriSpeech. Only WERs incorporating the 4-gram LM are listed.

older GPU (GTX 1080 Ti) to do its ~12.5 full passes over the data.

Finally, we trained the same model with BPE subwords as CTC targets, to get more context-independent units [36]. We did 300 merge operations (10k was unstable) and attained a CER of 7.4%. This gave a WER of 8.7% with no LM (compare with Table 5’s LM-based entries), and 5.2% with a subword WFST of the LM. We still observed attention heads in both directions in the first layer, suggesting our subwords were still more context-dependent than phonemes.

4. CONCLUSION

We introduced SAN-CTC, a novel framework which integrates a fully self-attentional network with a connectionist temporal classification loss. We addressed the challenges of adapting self-attention to CTC and to speech recognition, showing that SAN-CTC is competitive with or outperforms existing end-to-end models on WSJ and LibriSpeech. Future avenues of work include multitasking SAN-CTC with other decoders or objectives, and streamlining network structure via directed or restricted attention.

https://github.com/facebookresearch/wav2letter/issues/11

https://github.com/rwth-i6/returnn-experiments/tree/master/2018-asr-attention/librispeech/full-setup-attention
5. REFERENCES

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist tem-
poral classification: labelling unsegmented sequence data with recurrent neural
networks,” in Proc. Int. Conf. Mach. Learning (ICML). ACM, 2006, pp. 369–376.

[2] A. Graves and J. Schmidhuber, “Offline handwriting recognition with multi-di-
dimensional recurrent neural networks,” in Proc. Adv. Neural Inform. Process. Sys-
tem (NIPS), 2009, pp. 545–552.

[3] D.-A. Huang, L. Fei-Fei, and J.C. Niebles, “Connectionist temporal modeling for
weakened supervised action labeling,” in Proc. Eur. Conf. Comput. Vision (ECCV),
2016, pp. 137–153.

[4] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent
neural networks,” in Proc. Int. Conf. Mach. Learning (ICML). ACM, 2014, pp.
1764–1772.

[5] A.Y. Hannun, A.L. Maas, D. Jurafsky, and A.Y. Ng, “First-pass large vocabulary
continuous speech recognition using bi-directional recurrent DNNs,” CoRR, vol.
abs/1408.2873, 2014.

[6] A.L. Maas, Z. Xie, D. Jurafsky, and A.Y. Ng, “Lexicon-free convolutional speech
recognition with neural networks,” in Proc. Conf. North Amer. Ch. Assoc. Comput.
Linguistics Human Lang. Technol. (NAACL-HLT), 2015, pp. 345–354.

[7] Y. Miao, M. Goyalwadi, and F. Metze, “EESEEN: End-to-end speech recogni-
tion using deep RNN models and WST-based decoding,” in Proc. IEEE Autom.
Speech Recognition Understanding Workshop (ASRU). IEEE, 2015, pp. 167–174.

[8] D. Amodei, S. Ananthanarayan, R. Anubhai, B. Bai, E. Battenberg, C. Case, J.
Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., “Deep Speech 2: End-to-end
speech recognition in English and Mandarin,” in Proc. Int. Conf. Mach. Learning
(ICML). ACM, 2016, pp. 173–182.

[9] R. Collobert, C. Puhrsch, and G. Synnaeve, “Wav2Letter: an end-to-end ConvNet-
based speech recognition system,” CoRR, vol. abs/1609.03193, 2016.

[10] T. Zenkel, R. Sanabria, F. Metze, and A. Waibel, “End-to-end speech recognition
networks,” in Proc. Conf. North Amer. Ch. Assoc. Comput. Linguistics Human
Lang. Technol. (NAACL-HLT), 2015, pp. 345–354.

[11] Z. Xie, Z. Sun, L. Jin, Z. Feng, and S. Zhang, “Fully convolutional network for
automatic speech recognition,” CoRR, vol. abs/1412.5567, 2014.

[12] T.N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-
term memory, fully connected deep neural networks,” in Proc. IEEE Int. Conf.
Automatic Speech Signal Process. (ICASSP). IEEE, 2015, pp. 4580–4584.

[13] A.Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger,
S. Satheesh, S. Sengupta, A. Coates, et al., “Deep Speech: Scaling up end-to-end
speech recognition,” CoRR, vol. abs/1412.5567, 2014.

[14] A. Graves, N. Jaitly, and J. Argelaguet, “Towards end-to-end speech recognition
with the Transformer in Mandarin Chinese,” in Proc. Ann. Conf. Int. Speech Commu-
nication Assoc. (INTERSPEECH), 2018.

[15] S. Zhou, S. Xu, and B. Xu, “Multilingual end-to-end speech recognition with
a single Transformer on low-resource languages,” CoRR, vol. abs/1806.05059,
2018.

[16] S. Zhou, S. Xu, and B. Xu, “Sentence-based sequence-to-sequence speech
recognition with the Transformer in Mandarin Chinese,” in Proc. Ann. Conf.
Int. Speech Communication Assoc. (INTERSPEECH), 2018.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: An ASR
corpus based on public domain audio books,” in Proc. IEEE Int. Conf. Automatic
Speech Signal Process. (ICASSP). IEEE, 2015, pp. 4580–4584.

[18] Mozilla Foundation, “A journey to <10 word error rate,” https://hacks.mozilla.org/2017/11/a-journey-to-10-word-error-rate/, 2017.

[19] T. Zenkel, R. Sanabria, F. Metze, and A. Waibel, “Subword and crossword units
for CTC acoustic models,” in Proc. Ann. Conf. Int. Speech Communication Assoc.
(INTER SPEECH), 2018, pp. 396–400.

[20] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang,
and Z. Zhang, “MXNet: A flexible and efficient machine learning library for
heterogeneous distributed systems,” CoRR, vol. abs/1512.01274, 2015.

[21] S. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Deguise, “End-to-
end speech recognition from the raw waveform,” in Proc. Ann. Conf. Int. Speech
Communication Assoc. (INTERSPEECH), 2018, pp. 781–785.

[22] S. Sabour, W. Chan, and M. Norouzi, “Optimal completion distillation for se-
quence learning,” CoRR, vol. abs/1810.01398, 2018.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: An ASR
corpus based on public domain audio books,” in Proc. IEEE Int. Conf. Automatic
Speech Signal Process. (ICASSP). IEEE, 2015, pp. 5206–5210.

[24] Mozilla Foundation, “A journey to <10 word error rate,” https://hacks.mozilla.org/2017/11/a-journey-to-10-word-error-rate/, 2017.

[25] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, “A time-restricted
self-attention layer for ASR,” in Proc. IEEE Int. Conf. Acoustics Speech Signal
Process. (ICASSP). IEEE, 2018.

[26] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, “Self-attentional
acoustic models,” in Proc. Ann. Conf. Int. Speech Communication Assoc. (IN-
TERSPEECH), 2018.

[27] J. Cheng, L. Dong, and M. Lapata, “Long-short term memory-networks for
machine reading,” in Proc. Conf. Empirical Methods Natural Lang. Process.
(EMNLP), 2016, pp. 551–561.