TRIGGERED ATTENTION FOR END-TO-END SPEECH RECOGNITION

Niko Moritz, Takaaki Hori, Jonathan Le Roux

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

ABSTRACT

A new system architecture for end-to-end automatic speech recognition (ASR) is proposed that combines the alignment capabilities of the connectionist temporal classification (CTC) approach and the modeling strength of the attention mechanism. The proposed system architecture, named triggered attention (TA), uses a CTC-based classifier to control the activation of an attention-based decoder neural network. This allows for a frame-synchronous decoding scheme with an adjustable look-ahead parameter to control the induced delay and opens the door to streaming recognition with attention-based end-to-end ASR systems. We present ASR results of the TA model on three data sets of different size and language and compare the scores to a well-tuned attention-based end-to-end ASR baseline system, which consumes input frames in the traditional full-sequence manner. The proposed triggered attention (TA) decoder concept achieves similar or better ASR results in all experiments compared to the full-sequence attention model, while also limiting the decoding delay to two look-ahead frames, which in our setup corresponds to an output delay of 80 ms.

Index Terms— Triggered attention, end-to-end automatic speech recognition, connectionist temporal classification, attention mechanism, frame-synchronous decoding

1. INTRODUCTION

End-to-end and sequence-to-sequence neural network models, respectively, have recently gained increased interest and popularity in the automatic speech recognition (ASR) community [1–4]. The output of an end-to-end ASR system is usually a grapheme sequence that can either be single letters or larger units such as word-pieces and entire words [5]. The appeal of end-to-end ASR is that it enables a simplified system architecture compared to traditional ASR systems [6] by being composed of neural network components only and by avoiding the need for language specific linguistic expert knowledge to build such systems. Connectionist temporal classification (CTC) [7] and the attention mechanism [8] are the two most widely used neural network architectures for end-to-end ASR, and attention-based encoder-decoder neural networks have shown the ability to outperform CTC-based neural networks [9, 10]. However, attention-based decoders are per se not well suited to be applied in a streaming fashion, i.e. to compute outputs as audio samples are recorded, since attention weights are typically computed from an input sequence of an entire speech utterance, which is referred to as full-sequence mode. This is because it is unable to align an input and output sequence frame-by-frame, in contrast with CTC. Recently, the neural transducer (NT) concept was proposed [11] that adds a block processing strategy to the attention mechanism by using a fixed number of input frames and by introducing a special symbol to detect the end of an output sequence for each chunk of input frames. Disadvantages of the NT model are that it requires alignment information from an auxiliary ASR system to be trained and parameter initialization from a pre-trained full-sequence model to achieve a high recognition accuracy [12].

In the present paper, we propose the TA system architecture that is designed to utilize the alignment properties of a CTC objective function and the modeling strength of the attention mechanism. This is achieved by using a CTC trained neural network to dynamically partition an input sequence, which is typically pre-partitioned based on speech pauses between utterances, further into smaller subsequences based on its speech content prior to recognition by an attention-based decoder neural network. The TA decoder consists of a trigger model, which spots the time instants of grapheme outputs from an encoder sequence, and an attention-based decoder neural network, whose activation gets controlled by the trigger model. The encoder neural network is shared by the trigger network and the attention mechanism. Attention weights can only see encoder frames preceded by the triggering event plus some look-ahead frames. During training, forced alignment of the CTC output sequence is used to derive the time instants of the triggering. During decoding, uncertainties of the CTC trained trigger model are taken into account to generate alternative trigger sequences and output sequences, respectively. TA inference is conducted in a frame-synchronous decoding manner, which allows the model to be applied in an online ASR system, if an unidirectional encoder neural network is used, which is however not the focus of this work. Note that this work is different from the hybrid CTC/attention end-to-end ASR system proposed in [4], which uses label-synchronous decoding and combines the posterior probabilities of both sequence-to-sequence models to score outputs jointly. Here the posterior probabilities of the CTC model are not combined with the attention model scores, which is the subject of future work. The content-based, dot-product as well as location-aware attention mechanisms are studied along with the TA concept [2, 13]. ASR results of the TA model are compared to a well-optimized baseline attention model, which computes the output by seeing the full input sequence. Three ASR corpora of different size (80 to 960 hours of training data) and language (English and Mandarin Chinese) are used for the evaluation. Our proposed system architecture shows word error rate improvements on all ASR tasks, while also improving online recognition capabilities by frame-synchronous decoding. Particular advantages over the full-sequence model are determined when using the dot-product and content-based attention mechanisms.

2. TRIGGERED ATTENTION

The TA system architecture is composed of an encoder neural network and the TA decoder, which is illustrated in Fig. 1. The encoder neural network converts an input sequence X of ASR features such as log-mel spectral energies into a T-length encoder state sequence H:

$$H = \text{Encoder}(X). \tag{1}$$

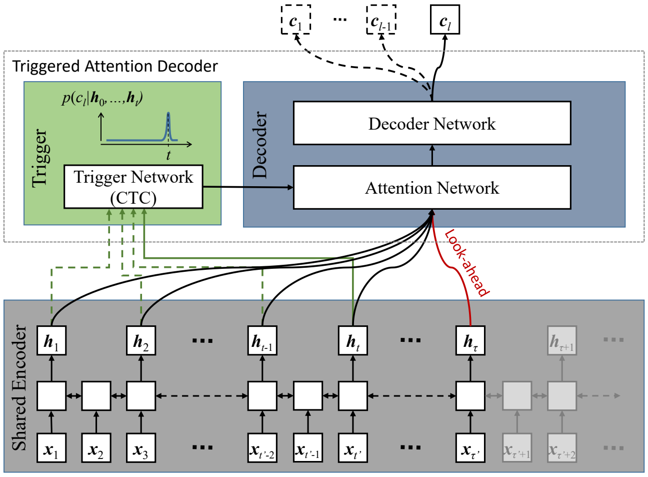

Fig. 1. Triggered attention system architecture. The shared encoder is trained jointly by the CTC and attention model objectives. The CTC-based trigger neural network is solely used to generate frame-level alignments and to trigger the attention-based decoder. The dashed lines to and from the TA decoder indicate the input and output of earlier time steps.

The encoder is based on a convolutional neural network (CNN) with a VGG structure [14, 15], followed by a bidirectional long short-term memory (BLSTM) neural network[1] [16, 17]. With this setup, the encoder output is sub-sampled to a four-times lower frame rate compared to the feature matrix X, which has a sampling rate of 100 Hz. The TA decoder consists of a trigger mechanism based on a CTC model, and an attention-based decoder neural network, which both take the output of the encoder as input.

Let $Z = (z_1, \ldots, z_T)$ denote a framewise CTC sequence of length T, with $z_t \in \mathcal{U} \cup \langle b \rangle$, where $\mathcal{U}$ denotes a set of distinct graphemes that can either be single characters or word-pieces, and $\langle b \rangle$ the blank symbol. Let $C = (c_1, \ldots, c_L)$, with $c_l \in \mathcal{U}$, denote a grapheme sequence of length L, and assume that the sequence Z reduces to C when collapsing repeated labels into single occurrences and removing blank symbols. Following [7], the CTC model probability is classically derived as:

$$p_{\text{ctc}}(C|H) = \sum_Z p(C|Z,H)p(Z|H) \tag{2}$$

$$\approx \sum_Z p(C|Z)p(Z|H) \tag{3}$$

$$= \sum_Z p(Z|C)p(Z|H)\frac{p(C)}{p(Z)}, \tag{4}$$

where $p(Z|C)$ denotes the transition probability and $p(Z|H)$ an acoustic model.

The trigger mechanism computes the trigger instances based on the CTC model by identifying the first frame of each sub-sequence of frames corresponding to the same grapheme label in Z, as illustrated in Fig. 2. We can rewrite the CTC sequence Z using the indices i_l and j_l for the beginning and end of the occurrence of the l-th label c_l in Z, with $i_l \leq j_l < i_{l+1}, \forall l$, and $z_t = c_l$ for all t such that $i_l \leq t \leq j_l$ and $z_t = \langle b \rangle$ for all other indices. The trigger mechanism conducts a mapping from a CTC sequence Z to a trigger event sequence $Z' = (\langle b \rangle^*, c_1, \langle b \rangle^*, c_2, \langle b \rangle^*, \ldots, c_L, \langle b \rangle^*)$ of same length T, in which * denotes zero or more repetitions and where each c_l occurs exactly once at frame i_l.

```
     t  =   1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9
     Z  = (<b>,<b>, d , d ,<b>, o , g , g ,<b>)
 p(Z|H) = (0.9,0.7,0.4,0.7,0.7,0.8,0.9,0.6,0.5)
     Z' = (<b>,<b>, d ,<b>,<b>, o , g ,<b>,<b>)
                    ↑               ↑   ↑
```

Fig. 2. Conversion of the CTC sequence Z into the trigger sequence Z', using an example with the word "dog". The red dashed boxes and the arrows indicate the frame position of a trigger event.

The trigger sequence Z' is input to the attention-based decoder of our TA model:

$$p_{\text{ta}}(C|H) = \sum_{Z'} \prod_{l=1}^{L} p(c_l|c_1, \ldots, c_{l-1}, Z', H) \tag{5}$$

In theory, multiple trigger sequences can be obtained over which we have to marginalize. However, applying the forward-backward algorithm, i.e. Baum-Welch learning, to reestimate parameters of the attention-based decoder, as shown in Eq. (5), requires high computational resources. In addition, it is difficult to handle the backward computation properly, since the attention decoder is conditioned on its previous hidden states, cf. Eq. (10). Hence, we simplify the model by using Viterbi learning instead, considering only the CTC sequence Z^* of highest overall probability computed by forced alignment, and its corresponding trigger sequence Z'^*, leading to the following approximation:

$$p_{\text{ta}}(C|H) \approx \prod_{l=1}^{L} p(c_l|c_1, \ldots, c_{l-1}, Z'^*, H). \tag{6}$$

Alignment information provided by the trigger sequence Z' is used to condition the attention mechanism on past encoder frames only:

$$p_{\text{ta}}(C|H) \approx \prod_{l=1}^{L} p(c_l|c_1, \ldots, c_{l-1}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{\tau_l}), \tag{7}$$

with $\tau_l = t'_l + \varepsilon$, where ε denotes the look-ahead hyperparameter and t'_l the time index of c_l in Z'.

The attention-based decoder model $p(c_l|c_1, ..., c_{l-1}, \boldsymbol{h}_1, ..., \boldsymbol{h}_{\tau_l})$ of Eq. (7) can be written as follows:

$$a_{lt} = \begin{cases} \text{DotProductAttention}(\boldsymbol{q}_{l-1}, \boldsymbol{h}_t) \\ \text{ContentAttention}(\boldsymbol{q}_{l-1}, \boldsymbol{h}_t) \\ \text{LocationAttention}(\{a_{l-1}\}_{t=1}^{\tau_l}, \boldsymbol{q}_{l-1}, \boldsymbol{h}_t) \end{cases} \tag{8}$$

$$\boldsymbol{r}_l = \sum_{t=1}^{\tau_l} a_{lt} \boldsymbol{h}_t \tag{9}$$

$$p(c_l|c_1, ..., c_{l-1}, \boldsymbol{h}_1, ..., \boldsymbol{h}_{\tau_l}) = \text{Decoder}(\boldsymbol{r}_l, \boldsymbol{q}_{l-1}, c_{l-1}) \tag{10}$$

In this work, we are using dot-product, content-based, and location-aware attention, as indicated in Eq. (8). The dot-product as well as the content-based attention mechanism are unaware of the attention weight distribution from a previous time step, and thus may easily be confused by similar input fragments. The location-aware attention mechanism, however, is taking the attention weight distribution

[1] Note that a BLSTM encoder is not suited for online ASR purposes. However, we here focus on improving the decoder part and leave the development of a well-tuned unidirectional encoder to future work.

of the previous time step into account, as indicated by a_{l-1}. For reasons of space, here we skip a more detailed description of all three attention mechanisms and the decoder setup, which are not different from previous work except for the conditioning on past encoder frames through parameter τ, and the interested reader is referred to [2, 4, 13]. The decoder model of Eq. (10), in which the vector $\boldsymbol{q}_{l-1}$ denotes the decoder hidden state of the previous time step, is based on an LSTM neural network.

The CTC model of Eq. (4) and the triggered-attention-based decoder model of Eq. (7) are trained jointly using the multi-objective loss function

$$\mathcal{L} = -\lambda \log p_{\text{ctc}} - (1 - \lambda) \log p_{\text{ta}}, \tag{11}$$

where the tunable parameter λ controls the weighting between the two objective functions p_{ctc} and p_{ta}.

2.1. Decoding

Inference with the TA model is realized in a frame-synchronous decoding manner. A best path search with the CTC-based trigger model is used to generate trigger events. Note that the trigger model is generating a character sequence as well but we do not combine this output with the scores of the attention model, which is subject to future work. However, uncertainties of the CTC-based trigger model are taken into account to generate alternative trigger sequences. This is achieved by tracking the posterior probabilities of the CTC output and whenever the softmax score of an alternative path is higher than a manually chosen threshold, a new trigger event is initiated. We chose a threshold of 0.2 throughout all experiments, which was experimentally determined using the Wall Street Journal (WSJ) data sets but not further optimized for the other ASR tasks. Trigger events of low confidence can be skipped by the attention-based decoder by bypassing the attention decoder states of the previous time step. After the attention model is triggered, the decoding is similar to a conventional beam-search algorithm [4]. Note that the problem of misalignment is almost nonexistent for TA decoding, because we are only attending to past encoder frames (relative to the trigger event) and the length of the output sequence is mostly determined by the trigger model. Thus, penalty factors and other terms that are often used in label-synchronous decoding to supervise the length of the generated output sequence are not required [4]. However, since hypotheses of different length can be in the decoding beam, we require an adjusted pruning concept using sequence-length-normalized scores instead of unnormalized probabilities to determine the best hypotheses that are kept.

3. EXPERIMENTAL SETUP

ASR experiments of this paper are conducted on three different data sets of different size, ranging from 90 to 960 hours of training, and of different language, which is English and Mandarin Chinese. We are using the WSJ corpus of read English newspapers [18], the LibriSpeech corpus, which is based on an open-source English audio books project featuring various recording qualities [20], and the Mandarin telephone speech corpus developed by the Hong Kong University of Science and Technology (HKUST) [19]. Basic information about the corpora are shown in Table 1.

Table 1. ASR corpora information.

WSJ1 (English) [18]	#Utterances	Size [h]
Training	37,416	80
Development (dev93)	503	1.1
Test (eval92)	333	0.7
HKUST (Mandarin) [19]	#Utterances	Size [h]
Training	197,391	174
Development	4,000	4.8
Test	5,413	4.9
LibriSpeech (English) [20]	#Utterances	Size [h]
Training	281,231	960
Development [clean/other]	2,703 / 2,864	5.4 / 5.3
Test [clean/other]	2,620 / 2,939	5.4 / 5.1

Table 2. Experimental hyperparameters.

WSJ model parameters	
Encoder type	VGG + BLSTMP
# BLSTMP cells / projection units / layers	320 / 320 / 6
# decoder LSTM cells / layers	300 / 1
HKUST model parameters	
Encoder type	VGG + BLSTM
# BLSTM cells / projection units / layers	1024 / 1024 / 3
# decoder LSTM cells / layers	1024 / 2
LibriSpeech model parameters	
Encoder type	VGG + BLSTM
# BLSTM cells / projection units / layers	1024 / 1024 / 5
# decoder LSTM cells / layers	1024 / 2
Common training parameters	
Optimization	AdaDelta
Adadelta ρ	0.95
Adadelta ϵ / ϵ decaying factor	10^{-8} / 10^{-2}
Maximum epoch	15 (WSJ, HKUST) 10 (LibriSpeech)
λ	0.2 (WSJ) 0.5 (HKUST, LibriSpeech)

The TA model and the full-sequence attention-based baseline system are both trained using a multi-objective loss function, since it has been shown that CTC guides the attention model to find better temporal alignments during training [4]. Note that the baseline system of this work does not use the CTC model for inference in contrast to the hybrid CTC/attention system of [4,15], because our TA architecture does not combine CTC and attention model scores as well. Specific model and training parameters are summarized in Table 2. Two encoder settings are used here, which are both composed of a VGG-based deep CNN component plus a deep BLSTM or a deep BLSTM interleaved with projection layers (BLSTMP) [15, 21]. The number of output units of the WSJ and HKUST end-to-end systems amount to 50 (number of English characters in WSJ) and 3653 (number of Mandarin characters in HKUST), respectively. The LibriSpeech ASR system is using 5000 sentence-piece units as an output, which are derived by the sentence-piece tokenizer proposed by [22]. In this work, a RNN-based language model (LM) is applied to the output of an end-to-end ASR system, whenever indicated. A word-based LM of 65k words applied to the WSJ test data [23] and a character-based LM is applied for the HKUST and LibriSpeech data sets [15].

4. RESULTS

The evaluation of the proposed TA model for different look-ahead parameter settings ε using the WSJ data set is illustrated in Fig. 3. It can be seen that the dot-product and content-based attention mechanisms require a smaller look-ahead to achieve low character error rates (CERs) in the TA system architecture, by which they obtain a higher guidance by the trigger instance, whereas the location-aware attention mechanism prefers larger look-ahead values. Some fluctu-

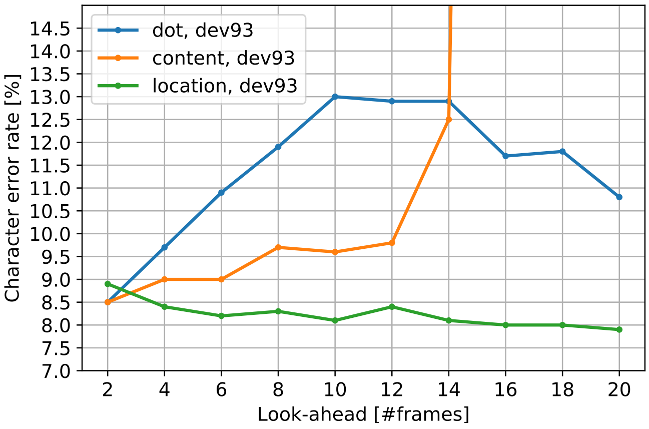

Fig. 3. Evaluation of the TA look-ahead hyperparameter ε for the dot-product, content-based, and location-aware attention mechanisms with the WSJ data set.

Table 3. Character error rates (CERs) and word error rates (WERs) of the WSJ and HKUST ASR tasks. Different system settings of the TA and the full-sequence attention model are compared. "dot", "cont", and "loc" denote the dot-product, content-based, and location-aware attention mechanisms. LM indicates the use of a RNN-based language model via shallow fusion. LM weights are optimized based on the development data for each individual system setting.

System Settings		WSJ: CER [%]		WSJ: WER [%]		HKUST [%]	
Model	LM	dev93	eval92	dev93	eval92	dev	test
Attention(dot)	✗	10.2	8.0	23.2	19.1	40.8	36.8
Attention(cont)	✗	9.0	6.9	22.3	17.8	35.3	32.9
Attention(loc)	✗	8.7	8.1	21.4	18.6	36.0	33.4
TA(dot, $\varepsilon = 2$)	✗	8.5	6.6	23.0	18.8	**33.7**	32.2
TA(cont, $\varepsilon = 2$)	✗	8.5	6.6	22.1	18.3	33.9	32.0
TA(loc, $\varepsilon = 20$)	✗	**7.9**	**6.1**	**20.6**	**16.2**	34.4	**31.9**
Attention(dot)	✓	7.8	5.3	15.8	11.3	39.0	36.2
Attention(cont)	✓	6.9	4.7	14.9	10.4	33.7	31.4
Attention(loc)	✓	6.9	4.2	14.8	10.1	33.1	31.6
TA(dot, $\varepsilon = 2$)	✓	**6.1**	4.4	13.1	9.6	32.4	31.2
TA(cont, $\varepsilon = 2$)	✓	6.5	4.5	13.8	9.9	**31.8**	**30.5**
TA(loc, $\varepsilon = 20$)	✓	6.6	**4.0**	**12.7**	**7.7**	33.5	31.4

ations in the results can be observed, which might be explained by the relatively small data size of WSJ. On average, the location-aware attention performs the best on this data set but requires a much larger look-ahead value compared to the other two attention mechanisms. For all the following experiments, we are using 2 look-ahead frames for the dot-product and content-based TA mechanisms and 20 look-ahead frames for the location-aware TA model, since these settings resulted in the lowest CERs.

Table 3 shows the CERs and word error rates (WERs) of the proposed TA system architecture with frame-synchronous decoding, in comparison to a full-sequence based attention model with label-synchronous decoding [4]. The TA-based system attains consistently lower error rates than the full-sequence based attention model. For example, on the test sets of WSJ and HKUST and without the use of an external language model, the location-aware TA system achieves 2.0% and 1.5%, respectively, lower CERs than the location-aware full-sequence system. Including the use of the RNN-based language model, error rate differences between both decoder types become smaller. However, the TA system architecture still outperforms the equivalent full-sequence based system irrespective of the used attention mechanism.

Table 4. Character error rates (CER) and word error rates (WER) of the LibriSpeech ASR task.

System Settings		CER [%]		WER [%]	
Model	LM	dev [clean/other]	test [clean/other]	dev [clean/other]	test [clean/other]
Attention(dot)	✗	14.8 / 29.3	16.0 / 30.5	12.4 / 25.2	13.9 / 26.3
Attention(cont)	✗	9.6 / 24.4	8.8 / 23.8	7.4 / 21.3	7.5 / 20.6
Attention(loc)	✗	**7.1** / 22.1	**7.3** / 23.0	**5.8** / 19.2	**6.1** / 20.0
TA(dot, $\varepsilon = 2$)	✗	10.3 / 23.2	10.2 / 23.9	9.2 / 21.0	9.3 / 21.6
TA(cont, $\varepsilon = 2$)	✗	8.2 / **20.3**	8.1 / **21.3**	7.4 / **18.4**	7.4 / **19.2**
TA(loc, $\varepsilon = 20$)	✗	8.0 / 20.7	8.1 / 22.0	7.3 / 19.1	7.4 / 20.0
Attention(dot)	✓	12.6 / 28.3	14.7 / 29.6	10.1 / 22.9	12.5 / 24.3
Attention(cont)	✓	9.8 / 21.8	9.0 / 21.0	7.4 / 18.0	7.8 / 17.0
Attention(loc)	✓	**6.6** / 19.2	**6.7** / 20.0	**5.3 / 15.4**	**5.4 / 16.1**
TA(dot, $\varepsilon = 2$)	✓	9.2 / 21.3	9.1 / 22.5	7.8 / 18.7	8.0 / 19.8
TA(cont, $\varepsilon = 2$)	✓	6.9 / **18.3**	**6.7 / 19.3**	5.8 / 15.8	5.7 / 16.7
TA(loc, $\varepsilon = 20$)	✓	7.1 / 19.1	7.2 / 20.5	6.2 / 17.0	6.3 / 18.3

In the LibriSpeech ASR experiments, the results of which are shown in Table 4, the content-based TA model consistently achieves lower error rates than the location-aware TA system, which indicates that the attention mechanism can leverage the trigger mechanism to compensate for missing attention weight information from previous time steps. Without the use of the RNN-based LM, the TA-based system considerably outperforms the full-sequence based attention model on the more difficult *other* test condition, while results are only slightly worse for the *clean* test data. This difference becomes smaller when using the RNN-based LM but still holds for the CER results, whereas the WER results of the full-sequence model show slight advantages over the TA system.

5. DISCUSSION AND FUTURE WORK

In this paper, we proposed the triggered attention (TA) system architecture for attention-based end-to-end ASR systems to enable decoding in a frame-synchronous and streaming fashion, respectively. Three different attention mechanisms are investigated in the TA system architecture and ASR experiments are conducted using the WSJ, HKUST, and LibriSpeech data sets. The results show that the trigger mechanism of the TA model implicitly provides location information to the attention mechanism, so that WER differences between location-aware attention, which makes use of the attention weight distribution computed in a previous time step, and content-based as well as dot-product attention are reduced. On average, the content-based TA system achieved the lowest WERs and even outperformed a strong full-sequence based location-aware attention model, except for the clean test condition of the LibriSpeech data set, where the full-sequence attention model has shown slightly lower error rates. In addition to the ability to improve the recognition accuracy, the TA system architecture enables streaming recognition by limiting the input to the attention mechanism to causal encoder frames plus two look-ahead frames, which here corresponds to a decoding delay of 80 ms.

Future work will focus on improving ASR results of the TA architecture further by joining the posterior probabilities of the attention-based decoder and the CTC-based trigger module and by using a multi-head attention mechanism. In addition, the encoder neural network will be changed to an unidirectional setting to further optimize the streaming nature of the proposed end-to-end ASR system.

6. REFERENCES

[1] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015.

[2] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015.

[3] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE ICASSP*, 2016.

[4] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *J. Sel. Topics Signal Processing*, vol. 11, no. 8, 2017.

[5] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[6] A. L. Maas, P. Qi, Z. Xie, A. Y. Hannun, C. T. Lengerich, D. Jurafsky, and A. Y. Ng, "Building DNN acoustic models for large vocabulary speech recognition," *Computer Speech & Language*, vol. 41, 2017.

[7] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, vol. 148, 2006.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[9] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. ISCA Interspeech*, 2017.

[10] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Proc. ISCA Interspeech*, 2018.

[11] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Proc. NIPS*, 2016.

[12] T. N. Sainath, C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, "Improving the performance of online neural transducer models," *CoRR*, vol. abs/1712.01807, 2017.

[13] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, vol. abs/1508.04025, 2015.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[15] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. ISCA INTERSPEECH*, 2017.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[17] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Proc. International Conference on Artificial Neural Networks: Formal Models and Their Applications (ICANN)*, 2005.

[18] L. D. Consortium, "CSR-II (wsj1) complete," *Linguistic Data Consortium, Philadelphia*, vol. LDC94S13A, 1994.

[19] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, "HKUST/MTS: A very large scale mandarin telephone speech corpus," in *Proc. ISCSLP*, vol. 4274, 2006.

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

[21] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. ISCA Interspeech*, 2014.

[22] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *CoRR*, vol. abs/1808.06226, 2018.

[23] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based RNN language models," *CoRR*, vol. abs/1808.02608, 2018.