MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Streaming End-to-End Speech Recognition with Joint CTC-Attention Based Models

Moritz, Niko; Hori, Takaaki; Le Roux, Jonathan

TR2019-159 December 19, 2019

Abstract

In this paper, we present a one-pass decoding algorithm for streaming recognition with joint connectionist temporal classification (CTC) and attention-based end-to-end automatic speech recognition (ASR) models. The decoding scheme is based on a frame-synchronous CTC prefix beam search algorithm and the recently proposed triggered attention concept. To achieve a fully streaming end-to-end ASR system, the CTC-triggered attention decoder is combined with a unidirectional encoder neural network based on parallel time-delayed long short-term memory (PTDLSTM) streams, which has demonstrated superior performance compared to various other streaming encoder architectures in earlier work. A new type of pre-training method is studied to further improve our streaming ASR models by adding residual connections to the encoder neural network and layer-wise removing them during the training process. The proposed joint CTC-triggered attention decoding algorithm, which enables streaming recognition of attention-based ASR systems, achieves similar ASR results compared to offline CTC-attention decoding and significantly better results compared to CTC prefix beam search decoding alone.

IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)

STREAMING END-TO-END SPEECH RECOGNITION WITH JOINT CTC-ATTENTION BASED MODELS

Niko Moritz, Takaaki Hori, Jonathan Le Roux

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

ABSTRACT

In this paper, we present a one-pass decoding algorithm for streaming recognition with joint connectionist temporal classification (CTC) and attention-based end-to-end automatic speech recognition (ASR) models. The decoding scheme is based on a frame-synchronous CTC prefix beam search algorithm and the recently proposed triggered attention concept. To achieve a fully streaming end-to-end ASR system, the CTC-triggered attention decoder is combined with a unidirectional encoder neural network based on parallel time-delayed long short-term memory (PTDLSTM) streams, which has demonstrated superior performance compared to various other streaming encoder architectures in earlier work. A new type of pre-training method is studied to further improve our streaming ASR models by adding residual connections to the encoder neural network and layer-wise removing them during the training process. The proposed joint CTC-triggered attention decoding algorithm, which enables streaming recognition of attention-based ASR systems, achieves similar ASR results compared to offline CTC-attention decoding and significantly better results compared to CTC prefix beam search decoding alone.

Index Terms— Automatic speech recognition, end-to-end, triggered attention, joint CTC-attention, streaming recognition

1. INTRODUCTION

The speech-to-text conversion problem can be viewed as a sequence-to-sequence modeling task without making use of any linguistic prior knowledge. This view has inspired a relatively recent and important research direction in automatic speech recognition (ASR) known as end-to-end ASR, which has the benefit of simplifying the ASR pipeline [1] and enabling better ways for optimization [2]. The prevailing end-to-end ASR models are based on the connectionist temporal classification (CTC) approach [3], the attention mechanism [4], or a combination of both [5]. CTC-based ASR models and their extensions and variants, such as the recurrent neural network (RNN) transducer (RNN-T) [6] and the automatic segmentation criterion (ASG) [7], are well suited for online/streaming recognition, where an ASR output is generated with only little delay after each spoken word. Attention-based encoder-decoder models, however, usually require the full input sequence of an entire speech utterance, since the alignment between input and output sequences is unknown prior to estimation of the attention weight distributions and traditional block-processing could truncate connected information. Therefore, this concept cannot be easily applied in a streaming fashion. Monotonic chunkwise attention (MoChA) [8], the neural transducer [9], windowed attention [10], and triggered attention [11] are methods proposed to enable streaming recognition of attention-based neural network models. The MoChA approach requires that input and output alignments be monotonic and frame-wise analyzes the encoder states to compute a "selection probability" based on which the encoder state sequence is chunked prior to being processed by the attention mechanism [8]. The neural transducer is based on a block-processing scheme for the attention mechanism with a fixed window length and generates zero to many output symbols for each of these blocks [9]. The windowed attention concept uses the current decoder state and two jointly-trained multi-layer perceptrons (MLPs) to predict the window shift and size for selecting a chunk of succeeding encoder states, which are fed to the attention mechanism to generate the next output symbol [10]. The triggered attention concept relies on an auxiliary system, which is typically CTC-based and jointly trained with the attention decoder, to predict an alignment and to trigger the attention decoding process [11].

Another important part of end-to-end ASR systems is the encoder neural network architecture. Bidirectional long short-term memory (BLSTM) neural network architectures achieve state-of-the-art results but are unsuitable for streaming applications. Latency-controlled BLSTMs (LCBLSTMs) are an alternative that enables the use of BLSTM-like architectures for streaming recognition by restricting the future context of the backward-directed LSTM to a fixed size [12–15]. LCBLSTMs have shown improved performance compared to LSTMs but at the expense of an increased computational cost due to overlapping chunks for which the backward-directed LSTM output has to be recomputed each time. Recently, a unidirectional neural network architecture based on parallel time-delayed LSTM (PTDLSTM) streams was proposed [16], which demonstrated improved ASR robustness compared to the LCBLSTM architecture of similar latency as well as to

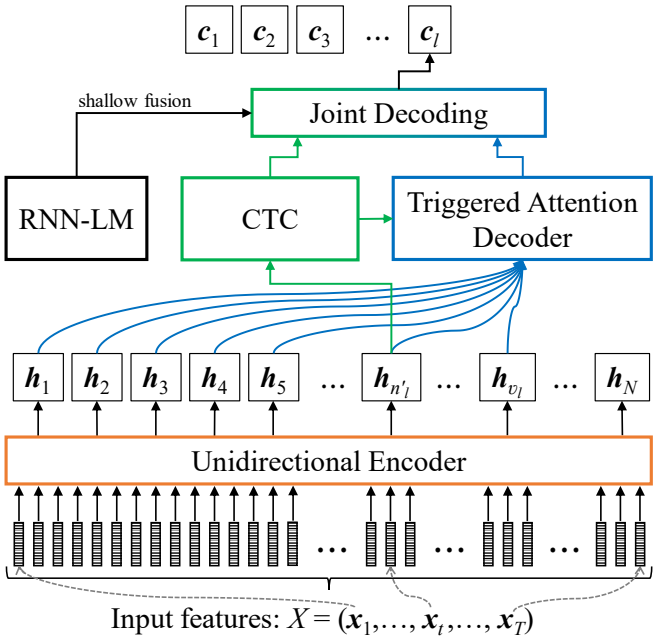

Fig. 1. Joint CTC-triggered attention architecture for streaming recognition.

other streaming architectures such as the time-delayed neural network LSTM (TDNN-LSTM) [17].

In previous work, ASR experiments published on triggered attention were using BLSTM-based encoder architectures, which are not suitable for streaming recognition [11]. Similarly, experiments with the PTDLSTM encoder architecture in [16] were based on offline CTC-attention decoding only [5]. In this work, we combine the triggered attention concept with the unidirectional PTDLSTM encoder architecture to achieve a fully online CTC-attention based end-to-end ASR system. For joint CTC-triggered attention model scoring, a new one-pass decoding algorithm for streaming recognition is proposed, which relies on the CTC prefix beam search algorithm of [18] and the triggered attention concept of [11]. Note that the decoding concept presented in this work is different from [11], since [11] relies on a greedy best path CTC score analysis to trigger the attention decoder, which is unsuitable for joint one-pass CTC-attention scoring. In addition, a new pre-training method is proposed that aims at improving the gradient propagation by adding residual connections to the encoder neural network, which are gradually and permanently removed during a pre-training phase.

2. STREAMING END-TO-END ASR

Figure 1 shows each system module of our proposed end-to-end ASR architecture for streaming recognition. A unidirectional encoder neural network is used to convert a sequence of acoustic features $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$, which here are 80-dimensional log-Mel spectral energies plus one pitch feature and its first and second-order derivatives computed every 10 ms, into the encoder state sequence $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N)$. Details of the encoder neural network are described in Section 2.1. The encoder state sequence H is shared by the CTC module and the triggered attention decoder, which are both described in Section 2.2. The joint decoding process of the CTC output and the triggered attention decoder is described in Section 2.3. Scores of an RNN-based language model (RNN-LM) are incorporated via shallow fusion to derive the final label sequence $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L)$, where in this work a label $\boldsymbol{c}$ can either be a character or a sentence piece [19]. Note that the arrows from the encoder state sequence H to the CTC and triggered attention decoder modules only depict the information flow for decoding output character $\boldsymbol{c}_l$ at time step t and encoder state index n'_l, respectively. Since the triggered attention module is using a fixed number of look-ahead frames, the attention decoder is attending up to encoder state index ν_l, as described in Section 2.2.

2.1. Encoder architecture

The unidirectional encoder neural network of our proposed end-to-end ASR system is based on TDLSTM and PTDLSTM building blocks that are composed together using a deep time-delay architecture as shown in Fig. 2, which generates an overall delay of 25 features frames corresponding to 250 ms. The first neural network layer consists of a TDLSTM building block that takes as input three consecutive feature frames, whereby an output is generated at a three-times lower frame rate, i.e., subsampling by a factor of three is applied. The TDLSTM building block first concatenates (cat) all the input frames before processing them using an LSTM layer followed by a bottleneck feed-forward neural network (BN) and a rectified linear unit (ReLU) non-linearity. The remaining encoder neural network layers, i.e., layers two to five, are based on PTDLSTM building blocks in which each time-delayed input stream is processed by a separate LSTM prior to concatenating LSTM outputs and further processing using a BN layer plus a ReLU non-linearity. Note that in the last encoder layer, i.e., layer five, we are not using any non-linearity function to generate the final encoder states, which are fed to the CTC layer as well as to the attention decoder.

2.2. Triggered attention training

The triggered attention model is based on an attention-based decoder neural network and an auxiliary CTC objective [11]. The CTC model is trained using the probability function

$$p_{\text{ctc}}(C|H) = \sum_Z p(C|Z,H)p(Z|H) \quad (1)$$

$$\approx \sum_Z p(C|Z)p(Z|H) \quad (2)$$

with the framewise CTC sequence $Z = (z_1, \ldots, z_N)$ of length N and $z_n \in \mathcal{U}$, where $\mathcal{U}$ denotes a set of output labels plus the blank label $\langle b \rangle$ of CTC [3]. Equation (2) is derived by using the assumption that the CTC label model $p(C|Z)$ is

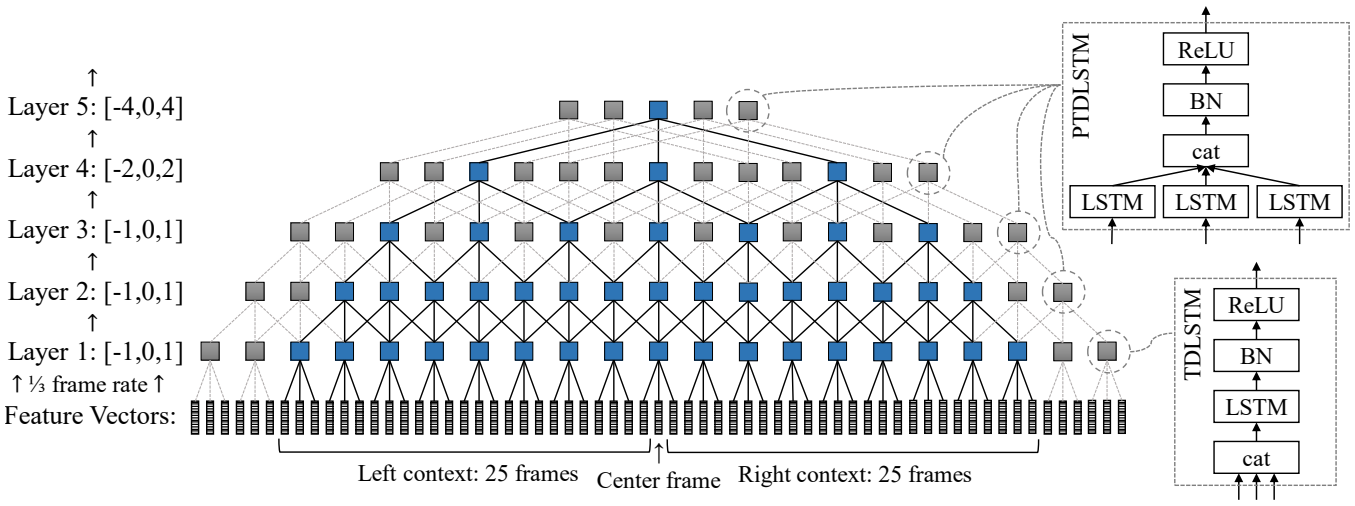

Fig. 2. Unidirectional encoder architecture using the deep time-delay structure shown on the left and the neural network building blocks shown on the right. Each rectangle box of the deep time-delay structure represents either a TDLSTM building block, in Layer 1, or a PTDLSTM building block, in Layers 2-5. The numbers in square brackets denote the frame delays of the input to each layer. The solid black lines and the blue rectangles highlight the path of a single encoder output frame. The dashed lines and gray rectangles denote connections and building blocks to generate past and future encoder output frames.

conditionally independent of the encoder state sequence H, which is classically used to simplify the CTC model [3]. The CTC posterior distribution $p(Z|H)$ is computed by applying a projection layer as well as a softmax function on top of the encoder state sequence H. Training of the triggered attention decoder requires an alignment between the encoder state sequence H and the label sequence C to condition the attention mechanism on past encoder frames only plus a fixed number of look-ahead frames. This information is generated by forced alignment using the CTC output, which provides the CTC label sequence Z^* of overall highest probability [11].

The CTC path of highest probability Z^* is converted into the trigger sequence Z' of same length by replacing label repetitions with the blank symbol $\langle b \rangle$ and only keeping the first occurrence of each label $\boldsymbol{c}_l$ [11]. For example, the sequence $Z^* = (\langle b \rangle, \langle b \rangle, \boldsymbol{c}_1, \boldsymbol{c}_1, \boldsymbol{c}_2, \langle b \rangle, \boldsymbol{c}_3, \boldsymbol{c}_3, \langle b \rangle)$ would be converted to $Z' = (\langle b \rangle, \langle b \rangle, \boldsymbol{c}'_1, \langle b \rangle, \boldsymbol{c}'_2, \langle b \rangle, \boldsymbol{c}'_3, \langle b \rangle, \langle b \rangle)$, where each $\boldsymbol{c}'_l$ denotes a trigger label. The trigger sequence Z' provides the required alignment information to condition the attention mechanism to encoder states preceding the trigger label $\boldsymbol{c}'_l$ plus a fixed number of look-ahead frames ε. To improve generalization, the positions of trigger labels are perturbed by ± 1 frame during training using a uniformly distributed random variable. The triggered attention objective function is defined as

$$p_{\text{ta}}(C|H) = \prod_{l=1}^{L} p(\boldsymbol{c}_l|\boldsymbol{c}_{1:l-1}, \boldsymbol{h}_{1:\nu_l}) \tag{3}$$

with $\nu_l = n'_l + \varepsilon$, where n'_l corresponds to the frame index of $\boldsymbol{c}'_l$ in Z', $\boldsymbol{c}_{1:l-1} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{l-1})$, and $\boldsymbol{h}_{1:\nu_l} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{\nu_l})$. The term $p(\boldsymbol{c}_l|\boldsymbol{c}_{1:l-1}, \boldsymbol{h}_{1:\nu_l})$ represents the attention decoder model, which can be written as

$$p(\boldsymbol{c}_l|\boldsymbol{c}_{1:l-1}, \boldsymbol{h}_{1:\nu_l}) = \text{Decoder}(\boldsymbol{r}_l, \boldsymbol{q}_{l-1}, \boldsymbol{c}_{l-1}), \tag{4}$$

$$\boldsymbol{r}_l = \sum_{n=1}^{\nu_l} a_{ln} \boldsymbol{h}_n, \tag{5}$$

$$a_{ln} = \text{AddAttention}(\boldsymbol{q}_{l-1}, \boldsymbol{h}_n). \tag{6}$$

In this work, the Decoder$(\cdot)$ of Eq. (4) is modelled by one or more LSTM layers with $\boldsymbol{q}_{l-1}$ representing the hidden LSTM states of the previous decoding step $l-1$, which are used to initialize LSTM states for the current processing step l. The context vector $\boldsymbol{r}_l$ is derived by the weighted sum over the encoder state sequence $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{\nu_l})$ as in Eq. (5), whereby the weights a_{ln} are computed using the additive attention mechanism as shown in Eq. (6) [4].

The CTC model of Eq. (2) and the triggered attention model of Eq. (3) are trained jointly using the multi-objective loss function

$$\mathcal{L} = -\lambda \log p_{\text{ctc}} - (1-\lambda) \log p_{\text{ta}}, \tag{7}$$

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

2.3. Joint CTC-triggered attention decoding

The proposed one-pass CTC-triggered attention decoding algorithm is based on the frame-synchronous prefix beam search algorithm of [18], which is extended here by integrating the triggered attention decoder. The usage of the triggered attention model in this algorithm is "time-dependent," i.e., in contrast to incorporating language model scores, for example, it matters at what time frame attention scores are computed,

Algorithm 1 Joint CTC-triggered attention decoding

1: **procedure** DECODE(H, p_{ctc}, α_0, α, β, K, P, θ_1, θ_2)
2: $\ell \leftarrow (\langle \text{sos} \rangle,)$
3: $\Omega \leftarrow \{\ell\}$, $\Omega_{\text{att}} \leftarrow \{\ell\}$
4: $p_{\text{nb}}(\ell) \leftarrow 0$, $p_{\text{b}}(\ell) \leftarrow 1$
5: $p_{\text{att}}(\ell) \leftarrow 1$
6: $\Sigma_\ell \leftarrow$ initial attention decoder states
7: **for** $n = 1, \ldots, N$ **do**
8: $\Omega_{\text{ctc}}, p_{\text{nb}}, p_{\text{b}} \leftarrow$ CTCPREFIX$(p_{\text{ctc}}(n), \Omega, p_{\text{nb}}, p_{\text{b}})$
9: **for** ℓ **in** Ω_{ctc} **do**
10: $p_{\text{prfx}}(\ell) \leftarrow p_{\text{nb}}(\ell) + p_{\text{b}}(\ell)$
11: $\widehat{p}_{\text{prfx}}(\ell) \leftarrow \log p_{\text{prfx}}(\ell) + \alpha_0 p_{\text{LM}}(\ell) + \beta|\ell|$
12: $\widehat{\Omega} \leftarrow$ PRUNE$(\Omega_{\text{ctc}}, \widehat{p}_{\text{prfx}}, K, \theta_1)$
13: **for** ℓ **in** $\widehat{\Omega}$ **do** ▷ Delete old prefixes in Ω_{att}
14: **if** ℓ **in** Ω_{att} **and** DCOND$(\ell, \widehat{\Omega}, p_{\text{ctc}})$ **then**
15: delete ℓ in Ω_{att}
16: **for** ℓ **in** $\widehat{\Omega}$ **do** ▷ Compute attention scores
17: **if** ℓ **not in** Ω_{att} **and** ACOND$(\ell, \widehat{\Omega}, p_{\text{ctc}})$ **then**
18: $p_{\text{att}}(\ell), \Sigma_\ell \leftarrow$ ATTDEC$(\boldsymbol{h}_{1:n+\varepsilon}, \ell, \Sigma_{\ell_{:-1}})$
19: add ℓ to Ω_{att}
20: **for** ℓ **in** $\widehat{\Omega}$ **do** ▷ Compute joint scores
21: $\widehat{\ell} \leftarrow \ell$ **if** ℓ **in** Ω_{att} **else** $\ell_{:-1}$
22: $p \leftarrow \lambda \log p_{\text{prfx}}(\ell) + (1-\lambda) \log p_{\text{att}}(\widehat{\ell})$
23: $p_{\text{joint}}(\ell) \leftarrow p + \alpha p_{\text{LM}}(\ell) + \beta|\ell|$
24: $\Omega \leftarrow$ MAX$(\widehat{\Omega}, p_{\text{joint}}, P)$
25: $\widehat{\Omega} \leftarrow$ PRUNE$(\widehat{\Omega}, \widehat{p}_{\text{prfx}}, P, \theta_2)$
26: $\Omega \leftarrow \Omega + \widehat{\Omega}$
27: **return** MAX$(\widehat{\Omega}, p_{\text{joint}}, 1)$

since the attention mechanism is conditioned on $n+\varepsilon$ encoder states only, as described in Section 2.2.

Algorithm 1 shows the procedure of the proposed frame-synchronous joint CTC-attention decoding. In line 3, the joint hypothesis set Ω and the attention hypothesis set Ω_{att} are initialized with the prefix sequence $\ell = (\langle \text{sos} \rangle,)$, where the symbol $\langle \text{sos} \rangle$ denotes the start of sentence label. The CTC prefix beam search algorithm of [18] maintains two separate probabilities for a prefix ending in blank p_{b} and not ending in blank p_{nb}, which we implemented as associative arrays that are initialized in line 4. The attention model scores p_{att} are initialized in line 5 and the initial attention decoder states Σ_ℓ, which include the initial decoder LSTM states $\boldsymbol{q}_0$ as well as the initial output label $\boldsymbol{c}_0 \widehat{=} \langle \text{sos} \rangle$, are defined in line 6.

Lines 7 to 26 show the frame-by-frame processing of the CTC posterior probability sequence p_{ctc} and the encoder state sequence H, where $p_{\text{ctc}}(n, c)$ denotes the CTC posterior probability for label c at frame n. The function CTCPREFIX of line 8, which follows the CTC prefix beam search algorithm described in [20], extends the set of prefixes Ω using the CTC posterior probabilities p_{ctc} of the current time step n and returns the separate CTC prefix scores p_{nb} and p_{b} as well as the newly proposed set of prefixes Ω_{ctc}. In addition, we are using a local pruning threshold of 0.0001 at this point to ignore labels of lower CTC probability. Note that no language model or any other pruning technique is used inside CTCPREFIX, they will be incorporated in the following steps. The prefix probabilities p_{prfx} and scores $\widehat{p}_{\text{prfx}}$ are computed in lines 10 and 11, respectively, where p_{LM} represents the log probability of the language model (LM) and $|\ell|$ denotes the length of prefix sequence ℓ without counting the start of sentence label $\langle \text{sos} \rangle$. The function PRUNE of line 12 prunes a set of input prefixes Ω_{in} in two ways as follows:

procedure PRUNE(Ω_{in}, $\widehat{p}_{\text{prfx}}$, L, θ)
$\Omega_{\text{in}} \leftarrow$ MAX$(\Omega_{\text{in}}, \widehat{p}_{\text{prfx}}, L)$, $\Omega_{\text{out}} \leftarrow \{\}$
for ℓ **in** Ω_{in} **do**
if $\max(\widehat{p}_{\text{prfx}}) - \theta < \widehat{p}_{\text{prfx}}(\ell)$ **then**
add ℓ to Ω_{out}
return Ω_{out}

First, the function MAX reduces Ω_{in} (=Ω_{ctc}) to the L (=K) most probable prefixes based on $\widehat{p}_{\text{prfx}}$, and then every prefix whose score $\widehat{p}_{\text{prfx}}$ deviates more than beam width θ (=θ_1) from the maximum prefix score $\max(\widehat{p}_{\text{prfx}})$ is ignored. The new set of pruned CTC prefixes Ω_{out} is returned by PRUNE and stored in $\widehat{\Omega}$. Lines 13 to 15 of Algorithm 1 are used to delete prefixes in Ω_{att} that satisfy a delete condition DCOND:

procedure DCOND(ℓ, $\widehat{\Omega}$, p_{ctc})
$c \leftarrow \ell_{-1}$ ▷ last element of ℓ
$n_{\text{prev}} \leftarrow$ time index of ℓ when added to $\widehat{\Omega}$
if $|\ell| > 1$ **and** $n - n_{\text{prev}} > 2$ **and** $p_{\text{ctc}}(n, c) > 0.01 > \max(p_{\text{ctc}}(n_{\text{prev}}, c), p_{\text{ctc}}(n_{\text{prev}} + 1, c))$ **then**
return true
else
return false

In DCOND, n_{prev} represents the time index at which the triggered attention model score of prefix ℓ in Ω_{att} was computed. DCOND returns "true" if the triggered attention score of prefix ℓ was computed more than 2 frames ago and if the CTC scores $p_{\text{ctc}}(n_{\text{prev}}, c)$ as well as $p_{\text{ctc}}(n_{\text{prev}} + 1, c)$ are below the manually chosen threshold of 0.01 while $p_{\text{ctc}}(n, c)$ is above that threshold, else it returns "false". Such deletion of "old" prefixes from Ω_{att} aims at recomputing triggered attention scores of CTC prefixes that emerged early but not with the best possible score and associated time index.

Triggered attention model scores are computed from line 16 to line 19 for every prefix in $\widehat{\Omega}$ that is not in Ω_{att} and satisfies the add conditions ACOND:

procedure ACOND(ℓ, $\widehat{\Omega}$, p_{ctc})
$c \leftarrow \ell_{-1}$ ▷ last element of ℓ
if $p_{\text{ctc}}(n, c) > \max(p_{\text{ctc}}(n+1, c), p_{\text{ctc}}(n+2, c))$ **or** any($|\widetilde{\ell}| > |\ell| + 1$ **and** $\widetilde{\ell}$ starts with ℓ **for** $\widetilde{\ell}$ **in** $\widehat{\Omega}$) **then**
return true
else
return false

ACOND returns "true" if the CTC probability p_{ctc} of the last label c in prefix sequence ℓ at the current time frame n is

larger than the probability for two next time frames $n+1$ and $n+2$. It also returns "true" if there exists a prefix sequence in $\widehat{\Omega}$, which starts with ℓ and whose length is strictly larger than $|\ell|+1$, i.e., if ℓ has a successor in Ω_{ctc} with at least two extra labels. This condition avoids that the generation of attention scores for newly added labels to a prefix sequence is lacking behind the CTC prefix score generation by more than one label. The function ATTDEC computes the triggered attention scores p_{att} for prefix ℓ using the encoder state sequence $\boldsymbol{h}_{1:n+\varepsilon}$, where ε corresponds to the look-ahead parameter, and the attention decoder states $\Sigma_{\ell_{:-1}}$ of the previous prefix sequence $\ell_{:-1}$, i.e., of ℓ excluding the last element. The joint CTC-triggered attention scores p_{joint} are computed in lines 20 to 23 of Algorithm 1. Due to the delete and add conditions discussed above, the computation of triggered attention scores for a CTC prefix sequence in $\widehat{\Omega}$ may be delayed by one label. Hence, line 21 checks if the triggered attention score for prefix ℓ exists, otherwise the parent prefix sequence $\ell_{:-1}$ and associated score are used. In line 24, P prefixes of highest joint probability are selected and stored in Ω for further processing. These prefixes are augmented with the P most probable and further pruned CTC prefixes $\widehat{\Omega}$ using beam width θ_2 as shown in line 25 and 26. Finally, the joint CTC-triggered attention decoding function DECODE returns the prefix sequence of highest joint probability p_{joint} as shown in line 27.

2.4. Pre-training

In end-to-end ASR, neural network models are typically composed of complex and deep architectures, where gradient descent-based training with randomly initialized neural network weights may be suboptimal, due to the vanishing gradient problem. Pre-training methods, such as the (greedy) layer-wise pre-training, are often used to overcome these issues, and their effectiveness for training of attention-based ASR models has been studied in previous work [21]. In this work, we propose a new pre-training method by equipping a neural network with residual connections that are gradually removed every k-th training epoch, with $k \in \mathbb{R}_{>0}$, to enable an improved gradient propagation for the initial training epochs. We apply such residual pre-training to our encoder neural networks by adding residual connections from layer 1 to layer 3, layer 2 to layer 4, layer 3 to layer 5, etc. These artificially added connections are permanently removed one by one every k-th training epoch, beginning at the first layer of the network, with k here set to 1, 1.5, or 2, depending on the dataset and ASR system configuration. We found that ASR results degrade if these residual connections are not removed during training.

3. EXPERIMENTAL SETUP

Three different ASR data sets are used for the experiments reported in this paper. We are using the Wall Street Journal (WSJ) corpus of read English newspapers [22], the LibriSpeech corpus of read English audio books [23], which is based on the open-source project LibriVox, and the Mandarin telephone speech corpus of the Hong Kong University of Science and Technology (HKUST) [24]. The WSJ corpus has approximately 80 hours of training data, 1.1 hours of development data, and 0.7 hours of test data. The LibriSpeech corpus is divided into 960 hours of training data, 10.7 hours of development data, and 10.5 hours of test data, whereby the development and test data sets are both further split into "clean" and "other" conditions based on the speech quality, which was assessed using an ASR system [23]. The HKUST corpus comprises approximately 174 hours of training data as well as 4.8 hours of development test data and 4.9 hours of evaluation test data.

Table 1. Experimental hyperparameters.

WSJ model parameters	
# trainable encoder parameters	18M
Size of projection layer	320
# decoder LSTM cells / layers	300 / 1
triggered attention look-ahead ε	2
HKUST model parameters	
# trainable encoder parameters	80M
Size of projection layer	1024
# decoder LSTM cells / layers	1024 / 2
triggered attention look-ahead ε	4
LibriSpeech model parameters	
# trainable encoder parameters	115M
Size of projection layer	1024
# decoder LSTM cells / layers	1024 / 2
triggered attention look-ahead ε	8
Common training parameters	
Optimization	AdaDelta
Adadelta ρ	0.95
Adadelta ϵ / ϵ decaying factor	10^{-8} / 10^{-2}
Maximum # epochs	15 (WSJ, HKUST)
	10 (LibriSpeech)
λ	0.2 (WSJ)
	0.5 (HKUST, LibriSpeech)
Joint decoding parameters	
LM weight α / CTC weight λ	1.0 / 0.3 (WSJ)
	0.3 / 0.6 (HKUST)
	0.5 / 0.5 (LibriSpeech)
Pruning parameter K / P / θ_1 / θ_2	200 / 50 / 22 / 12 (WSJ)
	200 / 30 / 10 / 4 (HKUST)
	200 / 50 / 10 / 4 (LibriSpeech)
CTC prefix search parameters	
LM weight α_0	1.6 (WSJ)
	0.6 (HKUST, LibriSpeech)
Insertion bonus weight β	1.5

Specific model, training, and decoding parameters are summarized in Table 1. The number of trainable encoder parameters is similar for the BLSTM and PTDLSTM neural network architectures shown in the results section. The number of output targets of the WSJ and HKUST-based end-to-end ASR systems amount to 50 (number of English characters in

Table 2. Word error rates [%] for the WSJ and HKUST tasks using different decoding algorithms and encoder setups trained with and without residual pre-training. TA denotes triggered attention. The PTDLSTM encoder combined with CTC prefix beam search or CTC-TA decoding, results of which are highlighted in green color, represent fully streaming end-to-end ASR systems, while other system configurations are not.

	CTC-attention				CTC prefix search				CTC-TA (proposed)			
	WSJ		HKUST		WSJ		HKUST		WSJ		HKUST	
Encoder	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
BLSTM	7.9	4.7	29.9	28.9	9.8	6.9	31.0	30.0	8.7	5.9	30.3	29.3
+pre-train	7.8	5.1	29.7	28.7	10.8	6.7	31.0	29.8	8.8	5.9	30.2	29.2
PTDLSTM	8.0	5.4	31.4	30.1	11.5	7.6	33.0	31.5	9.1	6.4	31.5	30.1
+pre-train	8.4	4.9	31.4	29.9	11.4	7.7	32.8	31.3	9.0	6.5	31.4	30.1

Table 3. Word error rates [%] of different encoder architectures and decoding methods for the LibriSpeech recognition task.

	CTC-attention				CTC prefix search				CTC-TA (proposed)			
	clean		other		clean		other		clean		other	
Encoder	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
BLSTM	4.7	4.9	14.1	15.2	5.3	5.5	15.2	16.3	4.9	5.0	14.3	15.6
+pre-train	4.6	4.7	14.1	15.1	5.1	5.2	15.3	16.2	4.8	5.0	14.2	15.2
PTDLSTM	5.6	5.7	16.2	16.9	6.4	6.4	17.5	18.5	6.0	6.0	16.4	17.4
+pre-train	5.6	5.7	15.9	16.7	6.3	6.5	17.3	18.2	5.7	5.9	16.1	16.8

WSJ) and 3653 (number of Mandarin characters in HKUST), respectively. The LibriSpeech ASR system uses 5000 word pieces as output targets [19]. In this work, an RNN-based language model is applied via shallow fusion. A word-based RNN-LM of 65k words is applied for the WSJ experiments [25], a character-based RNN-LM for the HKUST data set, and a word piece-based RNN-LM for the LibriSpeech ASR task [26]. The location-aware attention mechanism is applied for the full-sequence based CTC-attention ASR system [5, 11, 27], whereas additive attention is used by the CTC-triggered attention model.

4. RESULTS

ASR results of various end-to-end ASR systems for the WSJ, HKUST, and LibriSpeech recognition tasks are shown in Tables 2 and 3. Word error rates (WERs) of three different decoding algorithms are compared: the "offline" full-sequence based CTC-attention decoding method of [5], the CTC prefix beam search decoding algorithm of [18], and the CTC-triggered attention decoding algorithm of Section 2.3. In addition, results of the PTDLSTM encoder architecture are compared to a five layer BLSTM of similar size with and without using the residual pre-training method as described in Section 2.4. For the WSJ task, which is the smallest data set of our ASR experiments, the CTC prefix beam search decoding WERs of the dev and test data as well as of all four encoder configurations are on average 2.5% worse compared to the CTC-attention decoding results. WSJ results of the CTC-triggered attention decoder are on average 1.5% better than WERs obtained by CTC prefix beam search decoding and about 1% worse compared to full-sequence CTC-attention decoding, which may be explained by the fact that CTC-triggered attention decoding relies on the CTC prefix beam search algorithm, which did not perform well for WSJ. However, the results of HKUST and LibriSpeech demonstrate that our proposed CTC-triggered attention decoding algorithm is competitive to full-sequence based CTC-attention decoding, if the CTC model is trained more robustly as well. For the HKUST benchmark and the PTDLSTM encoder setup, CTC-attention and CTC-triggered attention decoding results are virtually the same, whereas CTC prefix beam search decoding WERs are about 1.4% worse compared to the CTC-triggered attention decoding results. For the LibriSpeech ASR task, the streaming CTC-triggered attention decoding results of the PTDLSTM encoder architecture with residual pre-training are on average only 0.15% worse compared to the full-sequence based CTC-attention decoding results and on average 0.95% better than the CTC prefix beam search decoding results.

The results shown in Tables 2 and 3 demonstrate that, on average, residual pre-training helped to improve WERs of our end-to-end ASR models for the HKUST and LibriSpeech experiments but not for the WSJ ASR task.

5. CONCLUSIONS

In this paper, we presented an end-to-end ASR system for streaming recognition based on joint CTC-attention models. A new frame-synchronous one-pass decoding algorithm for joint scoring of CTC and attention models is proposed, which relies on a CTC prefix beam search algorithm coupled with the triggered attention concept. The encoder neural network of our streaming ASR system is based on the unidirectional parallel time-delay LSTM (PTDLSTM) architecture. We investigated residual pre-training to further improve robustness of our end-to-end ASR models, which is based on equipping the encoder neural network with residual connections that are gradually and permanently removed during the training process. For the LibriSpeech and HKUST ASR tasks, our proposed CTC-triggered attention decoding algorithm performs almost equally well compared to the full-sequence based CTC-attention decoding algorithm, while the algorithmic delay is limited to 490 ms and 370 ms, respectively. This delay stems from the encoder (250 ms) and the look-ahead parameter of the decoder setup (8 encoder frames of 30 ms frame rate for LibriSpeech and 4 for HKUST). WSJ-based ASR results indicated that CTC-triggered attention decoding is more dependent on a well-trained CTC model than full-sequence based CTC-attention decoding. However, for all tested ASR tasks, our CTC-triggered attention decoding algorithm demonstrated significant improvements compared to CTC prefix beam search decoding alone.

6. REFERENCES

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, Jun. 2014.

[2] 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, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.

[3] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, vol. 148, Jun. 2006.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:abs/1409.0473*, 2014.

[5] 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.

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

[7] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:abs/1609.03193*, 2016.

[8] C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *Proc. International Conference on Learning Representations (ICLR)*, Apr. 2018.

[9] 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*, Dec. 2016.

[10] S. Zhang, E. Loweimi, P. Bell, and S. Renals, "Windowed attention mechanisms for speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.

[11] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.

[12] A. Mohamed, F. Seide, D. Yu, J. Droppo, A. Stoicke, G. Zweig, and G. Penn, "Deep bi-directional recurrent networks over spectral windows," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015.

[13] K. Chen and Q. Huo, "Training deep bidirectional LSTM acoustic model for LVCSR by a context-sensitive-chunk BPTT approach," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 7, 2016.

[14] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. R. Glass, "Highway long short-term memory RNNS for distant speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016.

[15] A. Zeyer, R. Schlüter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," in *Proc. ISCA Interspeech*, Sep. 2016.

[16] N. Moritz, T. Hori, and J. Le Roux, "Unidirectional neural network architectures for end-to-end automatic speech recognition," in *Proc. ISCA Interspeech*, Sep. 2019.

[17] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Process. Lett.*, vol. 25, no. 3, 2018.

[18] A. L. Maas, A. Y. Hannun, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[19] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:abs/1808.06226*, 2018.

[20] 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.

[21] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A comprehensive analysis on attention models," in *Interpretability and Robustness in Audio, Speech, and Language (IRASL) Workshop, NeurIPS*, Dec. 2018.

[22] "CSR-II (WSJ1) complete," vol. LDC94S13A. Philadelphia: Linguistic Data Consortium, 1994.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.

[24] 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.

[25] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based RNN language models," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec. 2018.

[26] 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*, Aug. 2017.

[27] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:abs/1508.04025*, 2015.