CAT: A CTC-CRF based ASR Toolkit Bridging the Hybrid and the End-to-end Approaches towards Data Efficiency and Low Latency

Keyu An, Hongyu Xiang, Zhijian Ou†

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, China

aky19@mails.tsinghua.edu.cn, xianghy16@mails.tsinghua.edu.cn, ozj@tsinghua.edu.cn

Abstract

In this paper, we present a new open source toolkit for speech recognition, named CAT (CTC-CRF based ASR Toolkit). CAT inherits the data-efficiency of the hybrid approach and the simplicity of the E2E approach, providing a full-fledged implementation of CTC-CRFs and complete training and testing scripts for a number of English and Chinese benchmarks. Experiments show CAT obtains state-of-the-art results, which are comparable to the fine-tuned hybrid models in Kaldi but with a much simpler training pipeline. Compared to existing non-modularized E2E models, CAT performs better on limited-scale datasets, demonstrating its data efficiency. Furthermore, we propose a new method called contextualized soft forgetting, which enables CAT to do streaming ASR without accuracy degradation. We hope CAT, especially the CTC-CRF based framework and software, will be of broad interest to the community, and can be further explored and improved.

Index Terms: speech recognition, CRF, CTC, end-to-end, data-efficiency

1. Introduction

Deep neural networks (DNNs) of various architectures have become dominantly used in automatic speech recognition (ASR), which roughly can be classified into two approaches - the DNN-HMM hybrid and the end-to-end (E2E) approaches. Initially, the DNN-HMM hybrid approach was adopted [1], which is featured by using the frame-level loss (cross-entropy) to train the DNN to estimate the posterior probabilities of HMM states. A GMM-HMM training is firstly needed to obtain frame-level alignments and then the DNN-HMM is trained. The hybrid approach usually consists of an DNN-HMM based acoustic model (AM), a state-tying decision tree for context-dependent phone modeling, a pronunciation lexicon and a language model (LM), which can be compactly combined into a weighted finite-state transducer (WFST) [2] for efficient decoding.

Recently, the E2E approach has emerged [3, 4, 5, 6], which is characterized by eliminating the construction of GMM-HMMs and phonetic decision-trees, training the DNN from scratch (in single-stage) and, even ambitiously, removing the need for a pronunciation lexicon and training the acoustic and language models jointly rather than separately. The key to achieve this is to define a differentiable sequence-level loss of mapping the acoustic sequence to the label sequence. Three widely-used E2E losses are based on Connectionist Temporal Classification (CTC) [3], RNN-transducer (RNN-T) [5], and attention based encoder-decoder [6] respectively.

When comparing the hybrid and E2E approaches (modularity versus a single neural network, separate optimization versus joint optimization), it is worthwhile to note the pros and cons of each approach. The E2E approach aims to subsume the acoustic, pronunciation, and language models into a single neural network and perform joint optimization. This appealing feature comes at a cost, i.e. the E2E ASR systems are *data hungry*, which require above thousands of hours of labeled speech to be competitive with the hybrid systems [7, 8, 9]. In contrast, the modularity of the hybrid approach permits training the AM and LM independently and on different data sets. A decent acoustic model can be trained with around 100 hours of labeled speech whereas the LM can be trained on text-only data, which is available in vast amounts for many languages. In this sense, modularity promotes *data efficiency*. Due to the lack of modularity, it is difficult for an E2E model to exploit the text-only data, though there are recent efforts to alleviate this drawback [10, 11]. In this paper, we are interested in bridging the hybrid and the E2E approaches, trying to inherit the data-efficiency of the hybrid approach and the simplicity of the E2E approach. A second motivation for such bridging is that low latency ASR has been addressed relatively easier and better in the hybrid approach than in the E2E approach, as will be discussed later in Section 2.

Specifically, we base on the recently developed CTC-CRF approach [12]. Basically, CTC-CRF is a CRF (Conditional Random Field) with CTC topology, which eliminates the conditional independence assumption in CTC and performs significantly better than CTC. It has been shown [12] that CTC-CRF has achieved state-of-the-art benchmarking performance with training data ranging from $\sim$100 to $\sim$1000 hours, while being end-to-end with a simplified pipeline (eliminating GMM-HMMs and phonetic decision-trees, training DNN-based AM in single-stage) and being data-efficient in the sense that cheaply available LMs can be leveraged effectively with or without a pronunciation lexicon.

In this paper we present CAT (CTC-CRF based ASR Toolkit) towards data-efficient and low-latency E2E ASR, which trains CTC-CRF based AMs in single-stage and uses separate LMs, with or without a pronunciation lexicon. On top of the previous work [12], the new contributions of this work are as follows.

1. CAT releases an full-fledged implementation of CTC-CRFs. A non-trivial issue in training CTC-CRFs is that the gradient is the difference between empirical expectation and model expectation. CAT contains efficient implementations of the forward-backward algorithm for calculating these expectations using CUDA C/C++ interface. CAT adopts PyTorch [13] to build DNNs and do automatic gradient computation, and so inherits the power of PyTorch in handling DNNs. In CAT, we can readily use the PyTorch DistributedDataParallel module to support training over multi-node and multi-GPU hardwares.

2. We add the support of streaming ASR in the toolkit. To this end, we propose a new method called contextualized soft forgetting (CSF), which combines soft forgetting [14] and

† Corresponding author. This work is supported by NSFC 61976122. Code released at https://github.com/thu-spmi/CAT

context-sensitive-chunk [15] in bidirectional LSTM (BLSTM). Extensive experiments show that: (a) CTC-CRF with soft forgetting improves over CTC with soft forgetting significantly and consistently; (b) With contextualized soft forgetting, the chunk BLSTM based CTC-CRF with a latency[1] of 300ms outperforms the whole-utterance BLSTM based CTC-CRF.

3. CAT provides reproducible, complete training and testing scripts for a number of English and Chinese benchmarks, including but not limited to WSJ, Switchboard, Fisher-Switchboard, and AISHELL datasets which are presented in this paper. CAT achieves state-of-the-art ASR performance on these datasets, which are comparable to the LF-MMI [18] results in Kaldi (one of the strongest, fine-tuned hybrid ASR toolkit) but with a much simpler training pipeline. Remarkably, compared to existing non-modularized E2E models, CAT performs better on limited-scale datasets (with ∼100 to ∼2000 hours of training data), demonstrating its data efficiency.

2. Related Work

ASR toolkits. Roughly speaking, there are two approaches to using DNNs in ASR - the DNN-HMM hybrid and the E2E approaches. So does the classification of existing ASR toolkits. For the hybrid approach, Kaldi [19] may be the most widely-used hybrid DNN-HMM based ASR toolkit. In Kaldi, lattice-free maximum-mutual-information (LF-MMI) training needs a multi-stage pipeline consisting of GMM-HMM training and phonetic decision tree construction. There have emerged some E2E ASR toolkits (e.g. ESPnet [20]/ESPRESSO [21], Wav2letter++ [22], and Lingvo [23]), mostly focusing on using attention-based encoder-decoder or hybrid CTC/attention.

EESEN [4] and E2E-LF-MMI [24, 25] seem to bridge the hybrid and the E2E approaches, by using the sequence-level loss (CTC and LF-MMI respectively) to train single-stage AMs and employing WFST based decoding. EESEN is based on CTC, which, different from CTC-CRF, is limited by its conditional independence assumption and weak performance. E2E-LF-MMI [24, 25] was developed with two versions of using mono-phones or bi-phones, and bi-phone E2E-LF-MMI obtains comparable results to hybrid LF-MMI. It is shown in our experiments that mono-phone CTC-CRF performs comparably to bi-phone E2E-LF-MMI but with a simpler pipeline. Bi-phone CTC-CRFs is found to slightly improve over mono-phone CTC-CRFs but will complicate the training pipeline. The differences between E2E-LF-MMI and CTC-CRF are detailed in [12].

Low latency ASR. An important feature for a practical ASR toolkit is its ability to do streaming ASR with low latency. In the hybrid approach, chunk-based schemes have been investigated in BLSTM [15, 26]. Time-delay neural networks (TDNNs) with interleaving LSTM layers (TDNN-LSTM) [27] has been developed in Kaldi to successfully limit the latency while keeping the recognition accuracy. In contrast, it is challenging and more complicated for attention-based encoder-decoders to do streaming ASR, which recently has received increasing studies, such as monotonic chunkwise attention (MoChA) [28], triggered attention [29], or using limited future context in the encoder [17]. RNN-T has some advantage for streaming ASR but is data hungry, requiring large-scale training data to work. The RNN-T result over the Fisher-Swichboard data (2300 hours) [30] is worse than CAT, as shown in Table 4.

3. CTC-CRF based ASR

CAT consists of separable AM and LM, which meets our rationale to be data efficient by keeping necessary modularity. In the following we mainly describe our CTC-CRF based AMs. CAT uses SRILM for LM training, and some code from Kaldi and EESEN for data preparation, decoding graph compiling and WFST based decoding. More details can be found in the toolkit.

Consider discriminative training of DNN-based AMs in single-stage based on the loss defined by conditional maximum likelihood [12]:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p_{\boldsymbol{\theta}}(\boldsymbol{l}|\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} \triangleq x_1, \cdots x_T$ is the speech feature sequence and $\boldsymbol{l} \triangleq l_1, \cdots l_L$ is the label (phone, character, word-piece and etc) sequence, and $\boldsymbol{\theta}$ is the model parameter. Note that $\boldsymbol{x}$ and $\boldsymbol{l}$ are in different lengths and usually not aligned. To handle this, a hidden state sequence $\boldsymbol{\pi} \triangleq \pi_1, \cdots \pi_T$ is introduced; state topology refers to the state transition structure in $\boldsymbol{\pi}$, which basically defines a mapping $\mathcal{B} : S_\pi^* \to S_l^*$ that maps a state sequence $\boldsymbol{\pi}$ to a unique label sequence $\boldsymbol{l}$. Here S_l^* denote the set of all sequences over the alphabet S_l of labels, and S_π^* similarly for the alphabet S_π of states. It can be seen that HMM, CTC, and RNN-T implement different topologies. CTC topology defines a mapping that removes consecutive repetitive labels and blanks, with S_π defined by adding a special blank symbol <blk> to S_l. CTC topology is appealing, since it allows a minimum size of S_π and avoids the inclusion of silence symbol, as discussed in [12].

Basically, CTC-CRF is a CRF with CTC topology. The posteriori of $\boldsymbol{l}$ is defined through the posteriori of $\boldsymbol{\pi}$ as follows:

$$p_{\boldsymbol{\theta}}(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|\boldsymbol{x}) \tag{2}$$

And the posteriori of $\boldsymbol{\pi}$ is further defined by a CRF:

$$p_{\boldsymbol{\theta}}(\boldsymbol{\pi}|\boldsymbol{x}) = \frac{\exp(\phi_{\boldsymbol{\theta}}(\boldsymbol{\pi}, \boldsymbol{x}))}{\sum_{\boldsymbol{\pi}'} \exp(\phi_{\boldsymbol{\theta}}(\boldsymbol{\pi}', \boldsymbol{x}))} \tag{3}$$

Here $\phi_{\boldsymbol{\theta}}(\boldsymbol{\pi}, \boldsymbol{x})$ denotes the potential function of the CRF, defined as:

$$\phi_{\boldsymbol{\theta}}(\boldsymbol{\pi}, \boldsymbol{x}) = \log p(\boldsymbol{l}) + \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(\pi_t|\boldsymbol{x})$$

where $\boldsymbol{l} = \mathcal{B}(\boldsymbol{\pi})$. $\sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(\pi_t|\boldsymbol{x})$ defines the node potential, calculated from the bottom DNN. $\log p(\boldsymbol{l})$ defines the edge potential, realized by an n-gram LM of labels and, for reasons to be clear in the following, referred to as the denominator n-gram LM. Remarkably, regular CTC suffers from the conditional independence between the states in $\boldsymbol{\pi}$. In contrast, by incorporating $\log p(\boldsymbol{l})$ into the potential function in CTC-CRF, this drawback is naturally avoided. Combining Eq. (1)-(3) yields the sequence-level loss used in CTC-CRF:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log \frac{\sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} \exp(\phi_{\boldsymbol{\theta}}(\boldsymbol{\pi}, \boldsymbol{x}))}{\sum_{\boldsymbol{\pi}'} \exp(\phi_{\boldsymbol{\theta}}(\boldsymbol{\pi}', \boldsymbol{x}))} \tag{4}$$

The gradient of the above loss involves two gradients calculated from the numerator and denominator respectively, which essentially correspond to the two terms of empirical expectation and model expectation as commonly found in estimating CRFs. Similarly to LF-MMI, both terms can be obtained via the forward-backward algorithm. Specifically, the denominator

[1] We define the latency as in [16, 17], which is the time span corresponding to the right contextual frames. In our experiment, we use 10 right contextual frames by default, and the frames are computed with 10ms shift and 3-fold sampling.

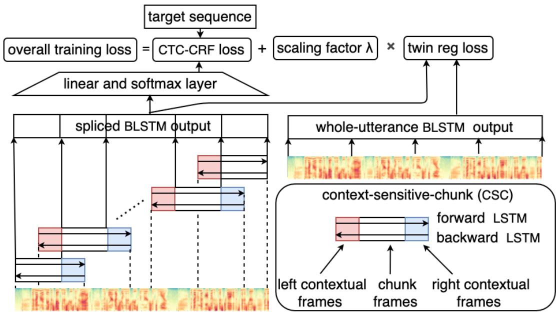

Figure 1: *Contextualized Soft Forgetting for streaming ASR.*

calculation involves running the forward-backward algorithm over the denominator WFST $\mathbf{T}_{\mathbf{den}}$. $\mathbf{T}_{\mathbf{den}}$ is an composition of the CTC topology WFST and the WFST representation of the n-gram LM of labels, which is called the denominator n-gram LM, to be differentiated from the word-level LM in decoding.

4. Contextualized Soft Forgetting towards Streaming ASR

To enable streaming ASR in CAT, we draw inspirations from soft forgetting [14] and context-sensitive-chunk [15] in using BLSTM. With the hypothesis that whole-utterance unrolling of the BLSTM leads to overfitting, soft forgetting, which is developed for CTC-based ASR, consists of three elements. First, the BLSTM network is unrolled over non-overlapping chunks. The hidden and cell states are hence forgotten at chunk boundaries in training. Second, the chunk duration is perturbed across training minibatches, which is called chunk size jitter. Third, the CTC loss is added with a twin regularization term, which is the mean-squared error between the hidden states of a pre-trained fixed whole-utterance BLSTM and the chunk-based BLSTM being currently trained. Since twin regularization promotes some remembering across chunks, this method is called soft forgetting. In streaming recognition, the hidden and cell states of the forward LSTM are copied over from one chunk to the next, and the backward LSTM hidden and cell states are reset to zero.

The idea of context-sensitive-chunk (CSC) is proposed in the BLSTM-HMM hybrid system to reduce the latency from a whole utterance to a chunk. In CSC, a chunk is appended with a fixed number of left and right frames as left and right contexts.

In CAT, we propose to apply soft forgetting to context-sensitive-chunks, which is called contextualized soft forgetting (CSF) as illustrated in Figure 1. First, we split an utterance into non-overlapping chunks. For each chunk, a fixed number of frames to the left and right of the chunk are appended as contextual frames except for the first and last chunk, where we use zeros as the left and right contexts respectively. Thus we form context-sensitive-chunks and run BLSTM over each CSC. The hidden and cell states of the forward and backward LSTM networks are reset to zeros at the left and right boundaries of each CSC in both training and inference. When calculating the sequence-level loss in CTC-CRF, we splice the neural network output from chunks into a sequence again, but excluding the network outputs from contextual frames. A pre-trained fixed whole-utterance BLSTM is used to regularize the hidden states of the CSC-based BLSTM, and the overall training loss is the sum of the CTC-CRF loss and the twin regularization loss with a scaling factor λ. Note that once the CSC-based BLSTM is trained, we can discard the whole-utterance BLSTM and perform inference over testing utterances without it.

5. Experiment Settings

The experiment consists of two parts. In the first part, we introduce the results on several representative benchmarks, including WSJ (80-h), AISHELL (170-h Chinese), Switchboard (260-h) and Fisher-Switchboard (2300-h) (the numbers in the parentheses are the size of training data in hours). The performances over these limited-scale datasets reveal the data efficiency of different ASR models. The second part presents the results for streaming ASR by the proposed contextualized soft forgetting method with ablation study.

It should be noted that the results shown in this paper should not be compared with results obtained with heavy data augmentation (e.g. specAugment [31]), much larger DNNs, and model combination. When compared to results reported from other papers, unless otherwise stated, we cite those results under comparable conditions to the best of our knowledge.

5.1. Setup for benchmarking experiment

We compare CAT with state-of-the-art ASR systems on several benchmarks, as stated above. We apply speed perturbation for 3-fold training data augmentation, except on Fisher-Switchboard. Unless otherwise stated, 40 dimension filter bank with delta and delta-delta features are extracted. The features are normalized via mean subtraction and variance normalization per utterance, and sampled by a factor of 3.

The AM network, different from [12], is two blocks of VGG layers followed by a 6-layer BLSTM similar to [32]. We apply 1D max-pooling to the feature maps produced by VGG blocks on the frequency dimension only, since the input features have been sampled in time-domain and we find that max-pooling along the time dimension will deteriorate the performance. The first VGG block has 3 input channels corresponding to spectral features, delta, and delta delta features. The BLSTM has 320 hidden units per direction for WSJ and AISHELL, and 512 for Switchboard and Fisher-Switchboard. The total number of parameters is 16M and 37M respectively, much smaller than most E2E models. Denominator 4-gram LMs are used throughout the experiments. In training, a dropout [33] probability of 50% is applied to the LSTM to prevent overfitting. Following [12], a CTC loss with a weight α is combined with the CRF loss to help convergence. We set $\alpha = 0.01$ by default and find in practice that the smaller α is, the better the final result will be.

5.2. Setup for streaming ASR experiment

To evaluate the effectiveness of contextualized soft forgetting, we first implement soft forgetting with the CTC-CRF loss. For a fair comparison, we adopt the same neural network architecture as in [14], which is a 6-layer BLSTM with 512 hidden units per direction. 40 dimension MFCC with delta and delta-delta are extracted, and the chunk size is set to 40. The whole-utterance BLSTM pre-trained on 260hr Switchboard obtain 14.3% WER on eval2000. For twin regularization, the scaling factor λ is set to 0.005. In contextualized soft forgetting, the chunk size is also 40, with 10 left and 10 right frames appended.

Table 1: *Results over WSJ (80-h training data).*

Model	Unit	LM	dev93	eval92
LF-MMI [25]	mono-phone	4-gram	6.0	3.0
LF-MMI [25]	bi-phone	4-gram	5.3	2.7
E2E-LF-MMI [25]	mono-phone	4-gram	6.3	3.1
E2E-LF-MMI [25]	bi-phone	4-gram	6.0	3.0
EESEN [4]	mono-phone	3-gram	10.87	7.28
ESPnet [20]	mono-char	RNN	12.4	8.9
Wav2letter++ [34]	mono-char	4-gram	9.5	5.6
Wav2letter++ [34]	mono-char	Conv	7.5	4.1
CTC/attention [35]	mono-char	RNN	6.8	4.4
CAT	mono-phone	4-gram	5.7	3.2
CAT	mono-char	4-gram	8.1	5.0

Table 2: *Results over AISHELL (170-h Chinese training data).*

Model	Unit	LM	Test
LF-MMI with i-vector [19]	tri-phone	3-gram	7.43
ESPnet [20]	Chinese char	RNN	8.0
CTC/attention [35]	Chinese char	RNN	6.7
Attention [36]	Chinese char	RNN	18.7
Attention [37]	Chinese char	RNN	8.71
CAT	mono-phone	3-gram	6.34

6. Experimental results

6.1. Results for benchmarking experiment

The WER results on WSJ are shown in Table 1, including two test sets - dev93 and eval92. It can be seen that CTC-CRF, hybrid LF-MMI and E2E-LF-MMI, which all keep modularity, perform comparable, and much better than other E2E models[2].

The CER (Character Error rate) results on AISHELL are shown in Table 2. It can be seen that CTC-CRF obtains state-of-the-art performance on AISHELL dataset - the CER is much better than other E2E models and the hybrid LF-MMI in Kaldi.

The WER results on Switchboard are shown in Table 3. The Eval2000 test set consists of two subsets - Switchboard (SW) and Callhome (CH). It can be seen that compared to bi-phone hybrid LF-MMI and E2E-LF-MMI, mono-phone CTC-CRF performs comparably but with a simpler pipeline. Remarkably, mono-phone CTC-CRF performs significantly better than other E2E models.

The WER results on Fisher-Switchboard are shown in Table 4. The performance of CTC-CRF, with no data augmentation, is on par with state-of-the-art hybrid and E2E models. Summing up the above results, we can see that on the limited-scale datasets (such as 80-h, 170-h, 260-h and 2300-h training data), the modularity of CTC-CRF clearly promotes data efficiency and achieve better results than other data hungry E2E models.

6.2. Results for streaming ASR experiment

First, we introduce different elements of soft forgetting [14] in steps to show their impact on WERs and also compare CTC and CTC-CRF. For this purpose, we follow [14] to report the non-streaming recognition results, as shown in Table 5. We start from training the basic chunk-based BLSTM networks with a fixed chunk size. It can be seen that CTC-CRF improves over CTC significantly under all experiment settings.

Then we examine the streaming recognition. It can be seen from Table 6 that CTC-CRFs trained with CSF improve significantly over CTC-CRFs with SF, and obtain comparable result with state-of-the-art TDNN-LSTM based hybrid model [27]. Remarkably, the CSF based streaming CTC-CRF (14.1%) even outperforms the whole-utterance CTC-CRF (14.3%), presumably because CSF alleviates overfitting in addition to realizing streaming ASR. This is in contrast to streaming ASR results by other E2E models, where streaming E2E models can hardly outperform their whole-utterance models [16, 17, 41].

Table 3: *Results over Switchboard (260-h training data). The numbers in parentheses denote the results after rescoring with RNN-LMs. Results in square brackets denote the weighted average over SW and CH based on our calculation when not reported in the original paper. "No LM" denotes not using shallow fusion with external LMs.*

Model	Unit	LM	SW	CH	Eval2000
LF-MMI [25]	mono-phone	4-gram	10.7	20.3	[15.5]
LF-MMI [25]	bi-phone	4-gram	9.5 (8.3)	18.6 (17.1)	[14.1 (12.7)]
E2E-LF-MMI [25]	mono-phone	4-gram	11.0	20.7	[15.9]
E2E-LF-MMI [25]	bi-phone	4-gram	9.8 (8.5)	19.3 (17.4)	[14.6 (13.0)]
EESEN [4]	mono-phone	3-gram	14.8	26.0	20.4
Attention [38]	subword	No LM	13.5	27.1	20.3
Attention [39]	subword	RNN	11.8	25.7	18.1
LAS [31]	subword	RNN	10.9	19.4	[15.2]
CTC/attention [35]	BPE	RNN	9.0	18.1	[13.6]
CAT	mono-phone	4-gram	9.8 (8.8)	18.8 (17.4)	14.3 (13.1)

Table 4: *Results over Fisher-Switchboard (2300-h training data). Notations are the same as in Table 3.*

Model	Unit	LM	SW	CH	Eval2000
LF-MMI [25]	bi-phone	4-gram	8.4 (7.5)	15.1 (14.3)	[11.8 (10.9)]
E2E-LF-MMI [25]	bi-phone	4-gram	8.6 (7.6)	15.4 (14.5)	[12.0 (11.1)]
E2E-LF-MMI [25]	mono-phone	4-gram	8.9	16.8	[12.9]
RNN-T [30]	char	4-gram	8.1	17.5	[12.8]
Attention [40]	char	No LM	8.3	15.5	[11.9]
CAT	mono-phone	4-gram	7.9 (7.3)	16.0 (15.0)	12.0 (11.2)

Table 5: *Non-streaming recognition results for CTC and CTC-CRF, both trained with soft forgetting over (260-h) Switchboard. Non-streaming recognition means that the hidden and cell states of the forward and backward LSTMs are copied across chunk boundaries. Notations the same as in Table 3.*

Loss	Model	SW	CH	Eval2000
CTC	chunk-based [14]	12.7	22.5	[17.6]
	+ chunk size jitter [14]	12.1	21.5	[16.8]
	+ twin reg [14]	11.1	19.7	[15.4]
CTC-CRF	chunk-based	11.1	19.6	15.4
	+ chunk size jitter	10.5	18.8	14.7
	+ twin reg	10.0	18.8	14.4

Table 6: *Streaming recognition results of CTC-CRFs, trained with Soft Forgetting (SF) and Contextualized Soft Forgetting (CSF) over (260-h) Switchboard.*

Method	Model	SW	CH	Eval2000
SF	chunk-based w/o future context	11.0	20.4	15.7
	+ chunk size jitter	11.0	19.7	15.4
	+ twin reg	10.8	19.7	15.3
CSF	chunk-based with context	10.7	20.0	15.4
	+ chunk size jitter	10.4	19.5	15.0
	+ twin reg	9.7	18.4	14.1
Results from literature	online-enabled BLSTM [26]	11.6	23.0	17.3
	TDNN-D [27]	9.6	19.9	14.8
	TDNN-LSTM-D [27]	9.0	-	13.9

7. Conclusion

This paper introduces an open source ASR toolkit - CAT, with the main features of data efficiency, simple pipeline, streaming ASR and superior results. we propose a new method called contextualized soft forgetting, which enables CAT to do streaming ASR without accuracy degradation. We hope CAT, especially the CTC-CRF based framework and software, will be of broad interest to the community, and can be further explored and improved, e.g. exploring different DNN architectures, different topologies of CRFs, and the application in more ASR tasks.

[2] Note that E2E models which use neural network based LMs via shallow fusion, are not directly compared to models using only n-gram LMs; they may be compared to models with RNN-LM rescoring.

8. References

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

[2] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 559–584.

[3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[4] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. ASRU*, 2015, pp. 167–174.

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

[6] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[7] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for librispeech: Hybrid vs attention," in *Proc. Interspeech*, 2019, pp. 231–235.

[8] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018, pp. 4774–4778.

[9] Z. Tüske, K. Audhkhasi, and G. Saon, "Advancing sequence-to-sequence based speech recognition," in *Proc. Interspeech*, 2019, pp. 3780–3784.

[10] S. Toshniwal, A. Kannan, and *et al*, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *Proc. SLT*, 2018, pp. 369–375.

[11] V. T. Pham, F. H. Xu, and *et al*, "Independent language modeling architecture for end-to-end ASR," *arXiv preprint arXiv:1912.00863*, 2019.

[12] H. Xiang and Z. Ou, "CRF-based single-stage acoustic modeling with CTC topology," in *Proc. ICASSP*, 2019, pp. 5676–5680.

[13] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

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

[15] C. Kai and H. Qiang, "Training deep bidirectional LSTM acoustic model for lvcsr by a context-sensitive-chunk BPTT approach," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 7, pp. 1185–1193, 2016.

[16] L. Dong, F. Wang, and B. Xu, "Self-attention aligner: A latency-control end-to-end model for ASR using self-attention network and chunk-hopping," 2019.

[17] H. Miao, G. Cheng, and *et al*, "Transformer-based online CTC/attention end-to-end speech recognition architecture," in *Proc. ICASSP*, 2020, pp. 6084–6088.

[18] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. Interspeech*, 2016, pp. 2751–2755.

[19] D. Povey, A. Ghoshal, and *et al*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1764–1772.

[20] S. Watanabe, T. Hori, and *et al*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[21] Y. Wang, T. Chen, and *et al*, "Espresso: A fast end-to-end neural speech recognition toolkit," *arXiv preprint arXiv:1909.08723*, 2019.

[22] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "wav2letter++: The fastest open-source speech recognition system," *arXiv preprint arXiv:1609.03193*, 2018.

[23] J. Shen, P. Nguyen, and *et al*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *arXiv preprint arXiv:1902.08295*, 2019.

[24] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using Lattice-free MMI." in *Proc. Interspeech*, 2018, pp. 12–16.

[25] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "Flat-start single-stage discriminatively trained HMM-based models for ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 1949–1961, 2018.

[26] A. Zeyer, R. Schlüter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," in *Proc. Interspeech*, 2016, pp. 3424–3428.

[27] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and LSTMs," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[28] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382v2*, 2018.

[29] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. ICASSP*, 2019, pp. 5666–5670.

[30] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*, 2017, pp. 206–213.

[31] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779v3*, 2019.

[32] 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. Interspeech*, 2017, pp. 949–953.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[34] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv preprint arXiv:1812.06864*, 2018.

[35] S. Karita, N. Chen, and *et al*, "A comparative study on transformer vs RNN in speech applications," in *Proc. ASRU*, 2019, pp. 449–456.

[36] M. Li, M. Liu, and H. Masanori, "End-to-end speech recognition with adaptive computation steps," in *Proc. ICASSP*, 2019, pp. 6246–6250.

[37] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Component fusion: Learning replaceable language model component for end-to-end speech recognition system," in *Proc. ICASSP*, 2019, pp. 5361–5635.

[38] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Proc. ACL, System Demonstrations*, 2018, pp. 128–133.

[39] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Proc. Interspeech*, 2018, pp. 7–11.

[40] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," in *Proc. Interspeech*, 2018, pp. 761–765.

[41] N. Moritz, T. Hori, and J. Le Roux, "Streaming automatic speech recognition with the transformer model," in *Proc. ICASSP*, 2020, pp. 6074–6078.