IMPROVING ATTENTION-BASED END-TO-END ASR SYSTEMS WITH SEQUENCE-BASED LOSS FUNCTIONS

Jia Cui[1]*, Chao Weng*[1]*, Guangsen Wang*[2]*, Jun Wang*[2]*,Peidong Wang*[1,3]*, Chengzhu Yu*[1]*, Dan Su*[2]*, Dong Yu*[1]

[1]Tencent AI Lab, Bellevue, USA
[2]Tencent AI Lab, Shenzhen, China
[3]The Ohio State University, USA

ABSTRACT

Acoustic model and language model (LM) have been two major components in conventional speech recognition systems. They are normally trained independently, but recently there has been a trend to optimize both components simultaneously in a unified end-to-end (E2E) framework. However, the performance gap between the E2E systems and the traditional hybrid systems suggests that some knowledge has not yet been fully utilized in the new framework. An observation is that the current attention-based E2E systems could produce better recognition results when decoded with LMs which are independently trained with the same resource.

In this paper, we focus on how to improve attention-based E2E systems without increasing model complexity or resorting to extra data. A novel training strategy is proposed for multi-task training with the connectionist temporal classification (CTC) loss. The sequence-based minimum Bayes risk (MBR) loss is also investigated. Our experiments on SWB 300hrs showed that both loss functions could significantly improve the baseline model performance. The additional gain from joint-LM decoding remains the same for CTC trained model but is only marginal for MBR trained model. This implies that while CTC loss function is able to capture more acoustic knowledge, MBR loss function exploits more word/character dependency.

Index Terms— speech recognition, end-to-end, multi-task learning, CTC

1. INTRODUCTION

A conventional automatic speech recognition (ASR) system is usually factorized into acoustic model (AM) and language model (LM) based on a probabilistic noisy channel model [1, 2]: $\text{argmax}_w P(W|A) = \text{argmax}_w P(W)P(A|W)$ where A is the acoustic signal and W is the word sequence. Hybrid systems improve recognition performance by enhancing each component with deep learning methods[3, 4, 5, 6]. These systems often adopt expert-crafted feature representations and pronunciation dictionaries, plus iteratively trained decision trees for clustering sub-phone units. The recent end-to-end (E2E) speech recognition systems simplify training pipelines by optimizing $P(W|A)$ directly without modeling expressive intermediate representations. Though E2E systems have achieved promising results approaching to that of hybrid systems in some certain tasks[7, 8, 9], there is still a non-negligible gap in between when large corpora are not available[10, 11]. The performance improvement by decoding with extra LMs trained on the same corpus suggests that certain knowledge has yet to be exploited in E2E systems[12]. There have been some prior research in knowledge exploration in E2E framework [13, 14, 15]. Here, we focus on speech recognition improvement with sequence-based loss functions.

Attention-based E2E model refers to the sequence to sequence model which is widely used for natural language processing tasks[8, 16, 17, 18]. It relies on the encoder-decoder paradigm where the encoder encodes the input sequence and the decoder produces the target sequence. The attention module allows the decoder to focus on specific part of the input sequence at each step. For speech recognition, the encoder is a network which transforms a sequence of acoustic frames into a sequence of hidden representations. The decoder is also a network which takes the current character input and previous predictions as input. The output of the decoder is used as a query to retrieve acoustic context from the hidden representations. The generator combines the decoder output and acoustic context to predict future characters. Minimum cross-entropy (XENT) criterion is then applied to optimize the system. In this framework, the whole utterances of acoustic frames are processed in advance. During training, the attention scheme establishes a soft alignment from the character sequence to the encoded acoustic sequence. To enforce the temporal constraints inherent in speech recognition tasks, various attention mechanisms are proposed [19, 20]. We stay with the content-based attention where acoustic hidden representations are weighted by their relevance to the query vectors.

We propose a new training strategy for integrating CTC loss in the attention-based E2E systems. The CTC loss[21, 7]

has been successfully applied in speech recognition systems, achieving performances comparable to hybrid systems when a large amount of training data (over 100k hours) is available. Conventional CTC models employ only one deep neural network [21, 7, 10], with each input frame being predicted independently. The character dependency is modeled implicitly through the Markov-assumption based loss function. Some followup work have integrated character dependencies directly into the model, such as recurrent neural network transducer (RNN-transducer) [22, 20] and recurrent neural aligner [23, 24]. In this paper, we add CTC loss criterion as an auxiliary training criterion in the attention-based framework. Even though the CTC output is excluded during inference, the new model provides a relative 15% word error rate (WER) deduction on top of the baseline E2E model.

As CTC loss is imposed on the encoder component, MBR is also investigated on the generator component. Discriminative training has a long history of success in speech recognition systems [25, 26, 27, 28]. Our implementation follows recent [28], where the optimization goal is to reduce the expected WER.

The organization of the rest of this paper is as follows. Section 2 illustrates the attention schemes in our baseline model as well as the joint-LM decoding strategy. Section 3 proposes several multi-task training strategies with CTC loss, followed by Section 4 explaining MBR criterion. Section 6 presents the experimental results including parameter tuning details and model analysis. Finally, Section 7 gives conclusions and future work.

2. ATTENTION-BASED END-TO-END SPEECH RECOGNITION MODEL

Our E2E framework adopts an input-feeding attention-based architecture [18] as shown in Figure(1). The decoder is a uni-directional LSTM, reading both the previous target label $\hat{y}_{i-1}$ (with embedding) and the previous prediction v_{i-1} (before projection). Its output s_i could be regarded as a prediction from the character information. The encoder could be either uni-directional or bi-directional and its output $h_{1:U}$ is produced before any character generated. The acoustic context c_i is extracted by a weighted sum over all $h_{1:U}$ at the request of each decoder output s_i. The final output distribution for y_i is a projection of the concatenation of decoder state s_i and context c_i, as showed in Equation (1-4).

$$s_i = \text{LSTM}([s_{i-1}, [\hat{y}_{i-1}; v_{i-1}]) \tag{1}$$

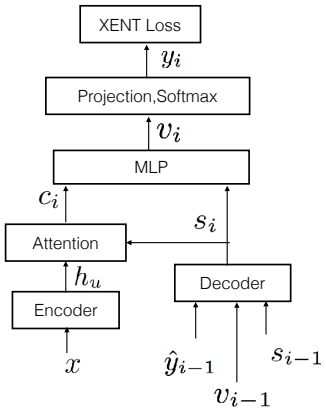
$$c_i = \text{AttentionContext}(s_i, h) \tag{2}$$

$$v_i = \tanh(W_h \cdot [s_i; c_i]) \tag{3}$$

$$P(y_i|x, y_{j<i}) = \text{softmax}(\text{proj}(v_i)) \tag{4}$$

We use content-based general attention as described in [29]. At each decoder step i, the AttentionContext function

Fig. 1. Attention-based E2E ASR System with Input-feeding Decoder



generates scalar energy $e_{i,u}$ by computing content similarity between h_u at each input time step and the linearly transformed s_i. The scalar energy $e_{i,u}$ is converted into a probability distribution $\alpha_{i,u}$ over times. The final context vector c_i is the linear blending of h_u with $\alpha_{i,u}$, as elaborated in Equation (5-7).

$$e_{i,u} = h_u^T W_a s_i \tag{5}$$

$$\alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_u \exp(e_{i,u})} \tag{6}$$

$$c_i = \sum_u \alpha_{i,u} h_u \tag{7}$$

The decoder component in this framework assumes the same structure as a regular recurrent language model, except that the last hidden layer output is concatenated with an acoustic context vector before it is projected and compared with the ground truth. This concatenation masks the predictability of the characters and therefore leaves improvement space for combining an independently trained language model.

There are several ways of combining LM predictions with E2E outputs. Here we adopt the joint-LM approach proposed in [30] as the decoding solution. LM scores are fused into prediction at every time step, which is either a plain character-based LM score or a consolidated word-based LM score upon a word boundary. The probability of predicting character c given the current decoding history g is formulated as:

$$p_{lm}(c|g) = \begin{cases} \frac{p_{wlm}(w_g|\phi_g)}{p_{clm}(w_g|\phi_g)} & \text{if } c \in S, w_g \in \mathcal{V} \\ p_{wlm}(<\text{UNK}>|\phi_g)\tilde{\beta} & \text{if } c \in S, w_g \notin \mathcal{V} \\ p_{clm}(c|g) & \text{otherwise} \end{cases}$$

where wlm denotes a word LM and clm denotes a character LM. w_g is the last word (character sequence between spaces

or <EOS>) of the character sequence g, and ϕ_g is the word-level history, which does not include w_g. If w_g forms a word in vocabulary $\mathcal{V}$, the word LM probability p_{wlm} is added while the accumulated character LM probability is removed. If w_g does not belong to $\mathcal{V}$, the probability of the unknown word is applied and scaled with a constant β. In our experiments, the LM score is interpolated with E2E score by weight 0.1 during decoding.

3. MULTI-TASK TRAINING WITH CTC LOSS IN E2E MODELS

The CTC loss function Loss(H, S) in our implementation is defined as a mean value of normalized sequence loss between hypothesis $H(x)$ and the corresponding targets as in Equation 8, where $S = (x, t)$ is the training set containing all pairs of input x and its target t. The probability of generating a hypothesis l (without special "blank" symbols or duplicates) from x is the sum over all possible output paths π corresponding to l as in Equation(9). π is an output sequence which has the same length as the input sequence x. As each frame is predicted independently, the probability of π is simply the multiplication of each frame prediction $z^u_{\pi_u}$.

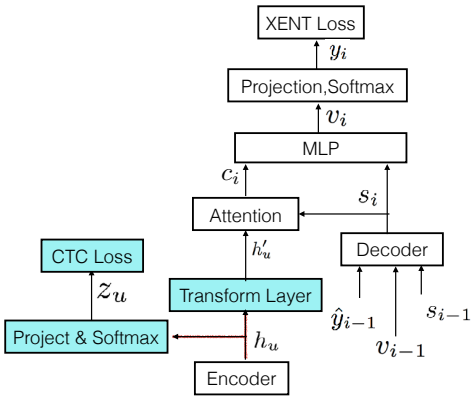
$$\text{Loss}(H, S) = \frac{1}{|S|} \sum_{x,t \in S} \frac{\text{editDistance}(H(x), t)}{|t|} \tag{8}$$

$$p(l|x) = \sum_{\pi \in \mathcal{D}(l)} p(\pi|x) = \sum_{\pi \in \mathcal{D}(l)} \prod_u z^u_{\pi_u} \tag{9}$$

$$z_u = \text{softmax}(\text{proj}(h_u)) \tag{10}$$

In this paper, we project the encoder output to predict the character sequences for the CTC calculation. The projection layer, together with the encoder parameters, are optimized to reduce the CTC loss. Such training process encourages the encoder output to be aligned to a single label, which is either one of the real target symbols or "blank". With a closer inspection of the attention framework, we notice that for CTC loss, each frame is supposed to map into one label while for the original E2E framework, multiple frames are accumulated to be mapped into one label. Accordingly, we propose to insert extra layers between the encoder and attention as shown in Figure (2). The extra layers serve as a transformer to enable better content match between query and context by transforming h_u to a new pattern h'_u. While the CTC loss keeps using the original encoder output h_u, the attention models adopts the transformed h'_u as input. During decoding, the new added projection layer and softmax layer is not involved. If the transform layer assumes the same structure as encoder hidden layers, the structure can be regarded as applying CTC loss at a lower encoder layer.

Fig. 2. Alternate CTC Training with Transform Layers in Attention-based E2E Framework



Inspired by the swapping-output-layer style multi-lingual modeling in speech recognition [31, 32, 33, 34], we also propose to do alternate training instead of using a predetermined weight to interpolate CTC loss and XENT loss. Here are three training strategies investigated in this paper:

1. Pre-training: initialize encoder by training it with CTC loss, then conduct regular XENT training.

2. Joint training: interpolate CTC loss and XENT loss with a predetermined weight.

3. Alternate training: optimize model with CTC loss and XENT loss alternatively in each epoch.

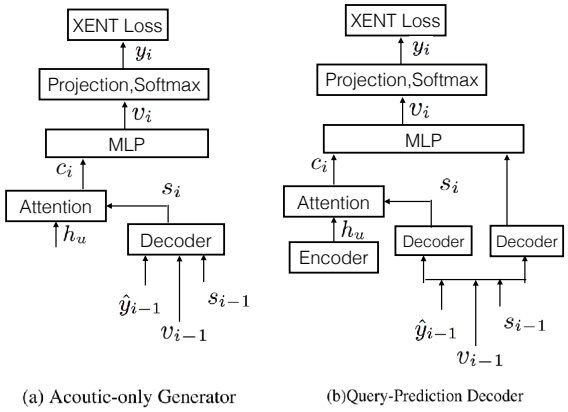
4. MBR TRAINING

Besides the CTC loss which is applied to the encoder, we also investigate MBR loss function applied to the final model output with the recipe from [28]. The optimization goal is to minimize the edit distance of the whole sequence, which is a criterion much closer to WER than XENT. Let y_u be one of the output label sequence produced by input x_u, $R(y_u, y^r_u)$ be the risk of hypothesis y_u compared to the reference y^r_u, the overall expected loss can be written as:

$$\mathcal{L}_{MBR}(x_{1:U}, y^r_{1:U}) = \sum_{u=1}^{U} \sum_{y_u} \frac{P(y_u|x_u) R(y_u, y^r_u)}{\sum_{y'_u} P(y'_u|x_u)} \tag{11}$$

where the probability of a hypothesis is approximated by its model output probability normalized by the probability mass of the N-best hypothesis. The N-best sentences are generated by the left-to-right beam search [35] with a heuristic rescoring formula in [36]:

Fig. 3. Two Architectures for Investigating Acoustic and Lexical Dependencies



(a) Acoutic-only Generator (b)Query-Prediction Decoder

$$\text{score}(y, x) = \log P(y|x)/\frac{(5+|y|)^\alpha}{(5+1)^\alpha} \quad (12)$$

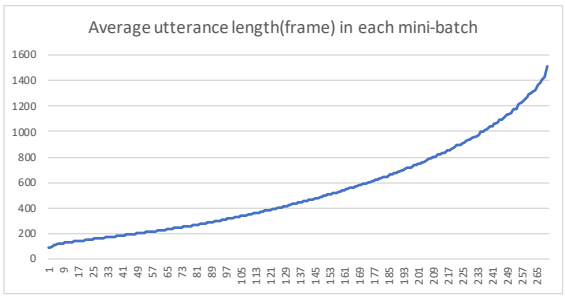
The model is usually pre-trained so that during training, the N-best hypothesis generated by the current model is a reasonable approximation of the real path space. In our implementation of MBR training, softmax smoothing [37] is also adopted during N-best generation to diversify hypothesis space.

5. ACOUSTIC-ONLY GENERATOR AND QUERY-PREDICTION DECODER

To investigate the combination of acoustic information and LM information, we investigate two structure variations as showed in Figure 3. The first one describes an acoustic-only generator where the retrieved acoustic context alone is used for the final output. The second one contains two RNNs in the decoder part, one for generating the query vector used in the attention module and the other for combining the attention output. Different from scheduled sampling, which is aimed to solve the discrepancy between training and inference, these two new structures are investigated for the interaction of acoustic and lexical information.

6. EXPERIMENTS

We have conducted experiments on Switchboard 300-hour set containing 262 hours of segmented speech from the Switchboard-1 audio. 10% of the data is randomly sampled from the training corpus as validation data set. The model performance is reported in terms of WER on Switchboard and call home data set respectively. English characters are used as target labels which include all lower-case letters



plus digits as well as special symbols: & - ' . / and annotations [laughter], [noise], [vocalized-noise], <unk>, <space>, <BOS>, <EOS>. For both word LM and character LM, only words/characters from the training transcriptions are included in the vocabulary. The <space> character is used to segment the output character sequence into word sequences. No other post-processing is used this paper.

The frontend features are extracted with Kaldi toolkit [38] and the models are built with PyTorch [39] based on some implementation from OpenNMT [40] and WarpCTC [41]. The baseline model takes 40-dimensional log-mel filterbank features over 25 ms frames for every 10 ms from the input speech signal. Sentences which are longer than 1800 frames are removed from both training and validation data. 10% of speakers are randomly selected as validation data. The baseline contains 4-layer bidirectional LSTM (BLSTM) with 256 hidden units in each direction. The decoder is a two layer LSTM with 512 hidden units. All parameters are initialized randomly. During inference, beam size 16 and $\alpha = 0.6$ are used.

6.1. Model Parameter Tuning and Data Perturbation

Hyper-parameter tuning is crucial for building an effective deep neural network model, which can change model performance dramatically. Our experiments started with a simple setup: stochastic gradient descent (SGD) optimizer, batch_size=8, random sampled mini-batch and stride=3 (taking one out of 3 frames but each feature expanded to 8 frames). The learning rate decays to half if there is not enough improvement on the validation loss. The initial WER was around 25%. Switching optimizer to AdaDelta [42] or Adam [43] with gradient clipping=50 reduces WER by absolute 3.5%. Dropout=0.2 and stride=2 (two successive frames are stacked together and every other frame is dropped) yielded another 1% and 0.4% improvement respectively. Moreover, as other researchers have also observed [10], sorting the training data by sentence length in ascending order provide better model performance than the reversed or randomized order. Figure 6.1 presents the average utterance length (frame) in each mini-batch sampled by every 20 mini-batches.

We take a controlled re-ordering strategy: first sorting all utterances in ascending order, then at epoch, randomize utterances within every 30 mini-batches. This strategy provides different mini-batches for each epoch, leading to a model with

improved WER 18.2%. All these experiments are on SWB test data (Hub'00).

Table 1. Model Performance with Parameter Tuning

Models	WER
Baseline	25.0
Ada-delta	22.5
+Dropout	21.1
+controlled re-ordering	18.2
+augmented data + 5-layer encoder	15.0

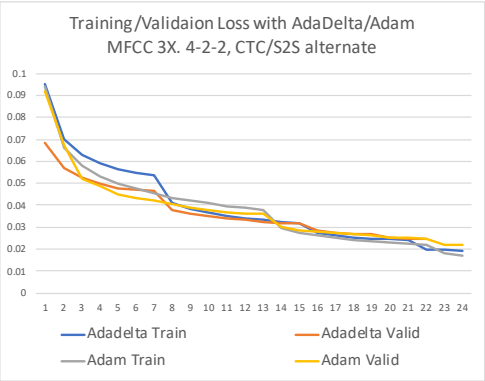
Data augmentation has shown to be beneficial for speech recognition. For this work, we change the speed of the audio signal, producing three versions of the original signal with speed factors of 0.9, 1.0 and 1.1 as in [44, 45]. Considering that tripling the training data prolongs training time dramatically, for each epoch, we only draw one random copy of perturbed samples for each utterance. The model therefore sees different data sets in different epochs, and eventually sees all perturbed data after a certain number of epochs. Such change enables earlier annealing therefore saves training time without performance loss. With augmented data, we could also split training and validation randomly so most of utterances have at least one copy in the training data. The threshold for terminating the training process then needs to be adjusted accordingly to avoid over-fitting. With data augmentation and by increasing encoder model size from 4-layer to 5-layer, the WER of newly trained model is improved from 18.2% to **15.0%**.

Some variations yield trivial performance change, such as batch sizes, feature selections (filter bank or Mel Frequency Cepstral Coefficient (MFCC)) and choices between Adam and AdaDelta Figure 4 characterizes the convergence properties of Adam and AdaDelta optimization with 3X augmented MFCC features on our best setup. For Adam, the starting learning rate is 0.01 with β_1= 0.9, β_2 = 0.999, $\epsilon = 10^{-8}$ as suggested in [43]. For AdaDelta, the starting learning rate is 0.5 with ρ=0.9, ϵ=1e-06. Even though Adam gives worse training and validation loss at the very beginning, it eventually catches up and both model converge at the same WER. A similar pattern is also observed with 1X sampled data sets.

6.2. Alternate Multi-task Training with CTC Loss

For CTC loss, three training strategies presented in Section 2 have been explored: pre-training, joint training and alternate training. The architecture is shown in Figure 2 where two layers of BLSTM are used as transforming layers. In the joint training setup, CTC loss and XENT loss are interpolated with ratio 1 : 9 whereas in alternate training, the model balances the contributions of the two loss functions automatically. In both cases, the XENT targets stay the same. The experimental results are shown in Table 2 with WER from both raw decoding and joint decoding. The pre-training gives

Fig. 4. Compare Adam and AdaDelta with 3X training data



a mild 0.4% absolute gain while joint training yields 1%. The alternate CTC training improves the model with the largest gain 2%, a relative 15% WER deduction. Adding more layers does not guarantee better performance: A structure for CTC and E2E decoder having their own individual BLSTM layers (alternate-T) leads to the same performance as that in joint training. With joint-LM decoding, the alternate-CTC model achieves WER **12.3%**.

Table 2. Performance with Different Multi-task Training

Models	SWB		CallHome	
	w/o LM	w/ LM	w/o LM	w/ LM
Baseline	15.0	14.3	25.1	24.5
+CTC pre-train	14.6	13.8	25.0	24.5
+CTC joint	14.0	13.3	24.9	24.4
+ alternate-T	14.1	13.5	24.6	24.0
+ alternate	13.0	**12.3**	23.8	23.3

6.3. Structure Variation For Generator

In this subsection, the decoder and generator structures are altered to investigate the information flow in the E2E framework. Will the acoustic information be modeled more efficiently without interference of the character sequence information? With the acoustic-only generator structure, the generator relies only on the retrieved acoustic context to predict future characters. This modification degrades the model performance as expected with WER 14.5% (Table 3) . Joint-LM decoding provides 0.6% gain, almost the same as before, indicating that this new model does not exploit more acoustic information as expected.

The other alteration is a twin RNN structure for decoder: one for producing queries as attention module input and the other for character prediction. The idea is to encourage a better character prediction by relaxing the bonding between the acoustic context and the character prediction. The experimental results present a significant XENT loss deduction

from 0.03 to 0.014 but the WER unexpectedly rises to 14.3%. Moreover, the joint-LM decoding provides no further gain for this model. A possible explanation is that though the character prediction is improved, the acoustic context quality decreases by using a less constrained query.

Table 3. WER on SWB with alternate-CTC and Various Generators

Models	XENT	WER	
		w/o LM	w/ LM
regular generator	0.025	13.0	12.3
acoustic-only	0.03	14.5	13.9
twin-RNN decoder	0.014	14.3	14.3

6.4. Joint Multi-task Training with MBR Loss

Instead of modifying model structures to enhance character sequence dependency, we could resort to additional optimization targets. The MBR loss is successfully applied in [46] to improve overall recognition performance. The model is firstly trained with only XENT loss, plus scheduled sampling and softmax smoothing, yielding WER 13.3%. Joint-MBR training improves WER to 12.8% with character-level edit distance and 12.2% with word level edit distance. We then use joint-LM decoding on these models to see if more lexical information could reduce WER further.

Table 4. Performance of MaxMargin on SWB

Models	Method	w/o LM	w/ LM
XENT	softmax smoothing scheduled sample	13.3	12.8
+MBR	character level	12.8	12.4
+MBR	word level+ ss	12.2	12.0

6.5. Analysis of Joint-LM Decoding with Enhanced Attention Models

Figure 5 demonstrates multi-task trained attention models and their performance with (red) and without (blue) joint-LM decoding. The model starts from the baseline model without data augmentation where the joint-LM yields absolute 1.5% WER reduction. Adding data augmentation improves model dramatically and the gain from LM reduced to 0.7. That might be because the acoustic context is more generalized given perturbed audio signals. Alternate-CTC training improves recognition performance significantly while the LM still produce a decent gain, indicating that the CTC-loss helps rather on the acoustic modeling than on the character prediction. That the MBR model receives little benefit from joint-LM decoding suggests that lexical information could be partially recovered by joint training with word-based minimum WER criterion.

Fig. 5. Effect of joint-LM decoding on different models

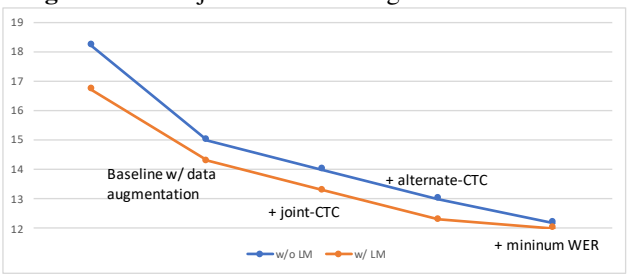


Table 5 shows that our results are competitive with other CTC or attention-based systems on the same data set.

Table 5. Compare with other E2E speech recognition systems

Systems	SWB	CH
Attention Seq2Seq + Trigram[17]	25.8	46.0
BRNN Grapheme CTC + Ngram [47]	20.0	31.8
BLSTM Phoneme CTC + Fisher LM [48]	14.8	n/a
Acoustic-to-Word + noLM [49]	14.5	25.1
Iterated CTC + RNN WLM [50]	14.0	25.3
Attention + BPE LSTM-LM [11]	11.5	25.7
Attention + MBR [28]	12.2	23.3
Att w/ CTC loss + LSTM-LM (this)	12.3	23.3
Attn + MBR + LSTM-LM (this)	12.0	23.1

7. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated two sequence-based loss functions CTC and MBR in attention-based speech recognition systems. We proposed a new training strategy for CTC which yielded a relative 15% WER reduction, leading to a competitive recognition performance on SWB300hs data set.

While the lexical dependency could be reinforced by decoding with an extra LM, we are more interested in whether such knowledge could be encoded into the E2E model itself without increasing model complexity. The experimental results suggested that training with additional CTC loss on the encoder component could improve acoustic modeling, while imposing MBR criterion on the outputs could enhance lexical dependency.

We notice that different training strategies could make a significant difference in the impact of auxiliary loss functions. In the future, we will continue exploring various advanced loss functions and their corresponding training strategies. Investigating the limit of the model capacity trained with multiple tasks could also be an interesting direction.

8. REFERENCES

[1] F. Jelinek, *Statistical methods for speech recognition*, MIT press, 1997.

[2] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE, vol. 77, no. 2*, p. 257?286, 1989.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, 2012.

[4] L. Deng, G. E. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *ICASSP*, 2013.

[5] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The microsoft 2016 conversational speech recognition system," in *ICASSP*, 2017.

[6] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Interspeech*, 2017.

[7] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," in *arXiv preprint arXiv:1610.09975*, 2016.

[8] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," in *CoRR*, 2015.

[9] C.C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP*, 2018.

[10] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *ICASSP*, 2018.

[11] A. Zeyer, K. Irie1, R. Schluter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Interspeech*, 2018.

[12] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *https://arxiv.org/abs/1609.06773*, 2017.

[13] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *CoRR, vol. abs/1708.06426*, 2017.

[14] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," in *interspeech2017*, 2017.

[15] Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *TASLP*, vol. 26, pp. 184196, 2018.

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

[17] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *ICASSP*, 2016.

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

[19] D. Bahdanau, J. Chorowski, D. Serdyuk, P.Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP*, 2016.

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

[21] A. Graves, S. Fernndez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[22] A. Graves, "Sequence transduction with recurrent neural networks," in *ICML*, 2012.

[23] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Interspeech*, 2017.

[24] H. Liu, Z. Zhu, X. Li, and S. Satheesh, "Gram-ctc: Automatic unit selection and target decomposition for sequence labelling," in *ICML*, 2017.

[25] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University Engineering Department, 2003.

[26] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *ICASSP*, 2009.

[27] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *ICASSP*, 2018.

[28] 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 *Interspeech*, 2018.

[29] M.T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015.

[30] T. Hori, S. Watanabe, and J.R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *ASRU*, 2017.

[31] A. Ghoshal, P. Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *ICASSP*, 2013.

[32] J.T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013.

[33] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual mrasta features for low-resource keyword search and speech recognition systems," in *ICASSP*, 2014.

[34] J. Cui, B. Kingsbury, B. Ramabhadran, and et. al, "Multilingual representations for low resource speech recognition and keyword search," in *ASRU*, 2015.

[35] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS, CoRR, vol. abs/1506.07503*, 2014.

[36] Y. Wu and et. al M. Schuster, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. 1609.08144, 2016.

[37] C. Shan, J. Zhang, Y.Wang, and L. Xie, "Attention-based end-to-end speech recognition in mandarin," *CoRR*, vol. abs/1707.07167, 2017.

[38] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *ASRU*, 2011.

[39] 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*, 2017.

[40] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017.

[41] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *ICML*, 2016, pp. 173–182.

[42] M. D. Zeiler, "Adadelta: An adaptive learning rate method," in *arXiv:1212.5701*, 2012.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.

[44] 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 *Interspeech*, 2016.

[45] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audioaugmentation for speech recognition," in *INTERSPEECH*, 2015.

[46] 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 *Interspeech 2018*, 2018.

[47] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[48] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. H. Waibel, "An empirical exploration of ctc acoustic models," in *ICASSP*, 2016.

[49] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and Nahamoo D, "Direct acoustics-to-word models for english conversational speech recognition," in *Interspeech*, 2017.

[50] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *ICASSP*, 2017.