Improving Attention Based Sequence-to-Sequence Models for End-to-End English Conversational Speech Recognition

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

[1]Tencent AI Lab, Bellevue, USA
[2]Tencent AI Lab, Shenzhen, China

{cweng, jiaacui, vincegswang, joinerwang, czyu, dansu, dyu}@tencent.com

Abstract

In this work, we propose two improvements to attention based sequence-to-sequence models for end-to-end speech recognition systems. For the first improvement, we propose to use an input-feeding architecture which feeds not only the previous context vector but also the previous decoder hidden state information as inputs to the decoder. The second improvement is based on a better hypothesis generation scheme for sequential minimum Bayes risk (MBR) training of sequence-to-sequence models where we introduce softmax smoothing into N-best generation during MBR training. We conduct the experiments on both Switchboard-300hrs and Switchboard+Fisher-2000hrs datasets and observe significant gains from both proposed improvements. Together with other training strategies such as dropout and scheduled sampling, our best model achieved WERs of 8.3%/15.5% on the Switchboard/CallHome subsets of Eval2000 without any external language models which is highly competitive among state-of-the-art English conversational speech recognition systems.

Index Terms: attention based sequence-to-sequence models, end-to-end speech recognition, sequential minimum Bayes risk training, MBR

1. Introduction

The performance of speech recognition has been improved dramatically since deep neural networks (DNNs) were applied to its main components such as acoustic model [1–6], language model [7, 8], pronunciation model [9], etc. Given prior success of applying DNNs to each individual component, there has been growing interest in building an end-to-end speech recognition system, i.e., a consolidated neural framework which subsumes all necessary speech recognition components. Comparing to a conventional hybrid system, such an end-to-end system typically has several advantages including a simpler building process, allowing a joint optimization among components and a compact model size. Current end-to-end speech recognition systems can be categorized into connectionist temporal classification (CTC) based [10–16] and attention based [17–19]. Attention based sequence-to-sequence system was first introduced into speech recognition in [20]. Later on, an attention based system, namely listen, attend and spell (LAS), was examined on a large-scale speech task [21] and more recently it shows a superior performance to a conventional hybrid system [22]. Although an attention based sequence-to-sequence system has matched or outperformed a conventional hybrid system when trained on large-scale datasets [22, 23] and gained its popularity lately, there are few, if any, previous works demonstrating whether it can also achieve a comparable performance to a conventional system on the standard Switchboard dataset, a widely used English conversational speech benchmark.

This work focuses on two improvements to an attention based sequence-to-sequence speech recognition system and demonstrates how it can be trained to perform comparably well to a hybrid system on the Switchboard dataset. For the first improvement, we propose to use the input-feeding architecture [24] which feeds not only the previous context vector but also the previous decoder hidden state information as inputs to facilitate the decoder making current label prediction. The second improvement is based on a better hypothesis generation scheme for sequential minimum Bayes risk (MBR) training of sequence-to-sequence models [25] where we introduce softmax smoothing into N-best generation during MBR training. Together with other training strategies such as dropout and scheduled sampling, our best model achieved WERs of 8.3%/15.5% on the Switchboard/CallHome subsets of Eval2000 without any external language models.

The remainder of the paper is organized as follows. In Section 2, we first elaborate how an attention based end-to-end speech recognition system works and then describe proposed improvements to the system in details. All the experiment details and results are presented in Section 3. We conclude our work in Section 4.

2. Improvements to attention based sequence-to-sequence models

2.1. Attention based sequence-to-sequence models for speech recognition

The architecture of the baseline sequence-to-sequence model adopted in this work is similar to LAS [21] which is depicted in Fig. 1-a. The inputs to the framework are typically several hundred frames of speech features such as log-mel filterbanks or MFCCs extracted from the input speech signal. Given T input speech frames $\mathbf{x} = x_1, x_2, ..., x_T$, the encoder transforms them into $\mathbf{h}^{\text{enc}}$, a sequence of hidden states with the length T which can be treated as a high level representation of the inputs:

$$\begin{aligned}\mathbf{h}^{\text{enc}} &= h_1^{\text{enc}}, h_2^{\text{enc}}, ..., h_t^{\text{enc}}, ..., h_T^{\text{enc}} \\ &= \text{Encoder}(x_1, x_2, ..., x_t, ..., x_T). \end{aligned} \quad (1)$$

The attention module takes as inputs all encoder hidden states $\mathbf{h}^{\text{enc}}$ and the current decoder hidden state h_i^{dec}. Based on a compatibility score between the current decoder hidden state h_i^{dec} and each encoded hidden state h_t^{enc}, the attention module computes the attention weights, i.e., the alignment between input and output:

$$\begin{aligned} a_{t,i} &= \text{align}(h_i^{\text{dec}}, \mathbf{h}^{\text{enc}}) \\ &= \frac{\exp(\text{score}(h_i^{\text{dec}}, h_t^{\text{enc}}))}{\sum_{t'=1}^{T} \exp(\text{score}(h_i^{\text{dec}}, h_{t'}^{\text{enc}}))}. \end{aligned} \quad (2)$$

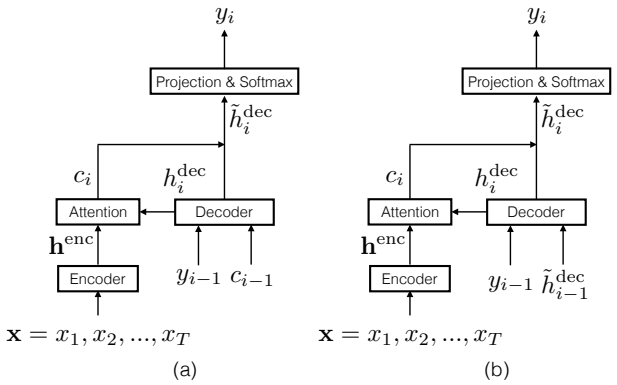

Figure 1: *Comparing baseline (a) to input-feeding (b) architecture: one of the inputs to the decoder, c_{i-1}, is replaced by previous attentional hidden state $\tilde{h}_{i-1}^{\text{dec}}$ in the input-feeding architecture*

Depending on different compatibility score functions in use, the attention modules can be categorized into dot-product, MLP and general [24]. In this work, we adopt MLP based attention, specifically,

$$\text{score}(h_i^{\text{dec}}, h_t^{\text{enc}}) = v_a^{\intercal} \tanh(W_a[h_i^{\text{dec}}; h_t^{\text{enc}}]), \quad (3)$$

where $[a; b]$ denotes the concatenation of two vectors. The output of the attention module is a context vector c_i calculated via a weighted sum of the encoder hidden states which can be interpreted as a summary of all encoder hidden state information used in the current prediction:

$$c_i = \sum_{t=1}^{T} a_{t,i} h_t^{\text{enc}}. \quad (4)$$

The decoder takes the previous embedded label prediction y_{i-1} and context vector c_{i-1} as inputs and outputs the current hidden state h_i^{dec}:

$$h_i^{\text{dec}} = \text{Decoder}(y_{i-1}, c_{i-1}). \quad (5)$$

h_i^{dec} is first used by the attention module to calculate the context vector c_i and then the attentional hidden state $\tilde{h}_i^{\text{dec}}$ is obtained as:

$$\tilde{h}_i^{\text{dec}} = \tanh(W_h[c_i; h_i^{\text{dec}}]). \quad (6)$$

Finally the projection and softmax layer produce the distribution of current label outputs:

$$p(y_i | y_{1:i-1}, \mathbf{x}) = \text{softmax}(W_o \tilde{h}_i^{\text{dec}}). \quad (7)$$

2.2. Input-feeding sequence-to-sequence models

In the baseline architecture, the inputs to the sequence-to-sequence model are y_{i-1} and c_{i-1}. The context vector c_{i-1} summarizes all encoder hidden state information from the last step whereas the embedded prediction y_{i-1} only contains the label information with highest probability from the last step instead of the full decoder attentional hidden state information. To overcome this, we propose to use the input-feeding architecture as in [24] where instead of feeding the previous context vector c_{i-1}, we feed the previous attentional hidden state, $\tilde{h}_{i-1}^{\text{dec}}$, to the decoder as depicted in Fig. 1-b. Therefore, Eq. (5) becomes:

$$h_i^{\text{dec}} = \text{Decoder}(y_{i-1}, \tilde{h}_{i-1}^{\text{dec}}). \quad (8)$$

2.3. MBR training and softmax smoothing for N-best generation

In this subsection, we first present thorough mathematical details of MBR training for attention based sequence-to-sequence models and then describe the proposed softmax smoothing for N-best generation. Let $\mathbf{y}$ denote the output sequence from the sequence-to-sequence model: $\mathbf{y} = y_1, y_2, y_i, ...y_L$. Given U pairs of the training speech utterance $\mathbf{x}$ and its corresponding reference label sequence $\mathbf{y}^r$, the MBR loss function can be written as:

$$\mathcal{L}_{\text{MBR}}(\mathbf{x}_{1:U}, \mathbf{y}_{1:U}^r) = \sum_{u=1}^{U} \sum_{\mathbf{y}_u} \frac{P(\mathbf{y}_u|\mathbf{x}_u) R(\mathbf{y}_u, \mathbf{y}_u^r)}{\sum_{\mathbf{y}'_u} P(\mathbf{y}'_u|\mathbf{x}_u)}, \quad (9)$$

where $\mathbf{y}_u$ represents one of hypothesized output label sequences corresponds to $\mathbf{x}_u$. $R(\mathbf{y}_u, \mathbf{y}_u^r)$ is the risk function between a hypothesized and reference label sequence, e.g., edit-distance. $P(\mathbf{y}_u|\mathbf{x}_u)$ is the sequence probability given input $\mathbf{x}_u$. According to the chain rule,

$$\begin{aligned} P(\mathbf{y}_u|\mathbf{x}_u) &= P(y_1, y_2, y_i, ..., y_L|\mathbf{x}_u) \\ &= p(y_1|\mathbf{x}_u) p(y_2|y_1, \mathbf{x}_u) \cdots p(y_L|y_{1:L-1}, \mathbf{x}_u) \\ &= \prod_{i=1}^{L} p(y_i|y_{1:i-1}, \mathbf{x}_u). \end{aligned} \quad (10)$$

Note that $p(y_i|y_{1:i-1}, \mathbf{x}_u)$ is exactly the output of sequence-to-sequence model as in Eq. (7). Therefore, to perform MBR training of sequence-to-sequence model, we will need to derive the gradients of MBR loss function w.r.t. $p(y_i|y_{1:i-1}, \mathbf{x}_u)$. For convenience, we use $p(y_i = y)$ as the shorthand for $p(y_i = y|y_{1:i-1}, \mathbf{x}_u)$, i.e., the probability of the model emitting a particular label y at the ith step, $f(\mathbf{y}_u)$ and $g(\mathbf{y}_u)$ as the shorthands for $P(\mathbf{y}_u|\mathbf{x}_u)$ and $R(\mathbf{y}_u, \mathbf{y}_u^r)$ in Eq. (9). Accordingly, we define a hypothesis set $S = \{\mathbf{y}_u | y_i = y\}$ which contains all the hypothesized sequences whose ith label is y. All hypothesized sequences can be divided into two disjoint sets, $\mathbf{y}_u \in S$ and $\mathbf{y}_u \notin S$. The MBR loss function can be rewritten as:

$$\mathcal{L}_{\text{MBR}} = \sum_{u=1}^{U} \frac{\sum_{\mathbf{y}_u \in S} f(\mathbf{y}_u) g(\mathbf{y}_u) + \sum_{\mathbf{y}_u \notin S} f(\mathbf{y}_u) g(\mathbf{y}_u)}{\sum_{\mathbf{y}'_u \in S} f(\mathbf{y}'_u) + \sum_{\mathbf{y}'_u \notin S} f(\mathbf{y}'_u)}. \quad (11)$$

Noticing $\frac{\partial \sum_{\mathbf{y}_u \notin S} f(\mathbf{y}_u)}{\partial p(y_i = y)} = 0$, we take the derivative of MBR loss function w.r.t. $\log p(y_i = y)$,

$$\begin{aligned} \frac{\partial \mathcal{L}_{\text{MBR}}}{\partial \log p(y_i = y)} &= \sum_{u=1}^{U} \frac{\partial \mathcal{L}_{\text{MBR}}}{\partial p(y_i = y)} \cdot \frac{\partial p(y_i = y)}{\partial \log p(y_i = y)} \\ &= \sum_{u=1}^{U} \frac{\partial \mathcal{L}_{\text{MBR}}}{\partial p(y_i = y)} \cdot p(y_i = y) \\ &= \sum_{u=1}^{U} \Bigg(\frac{\sum_{\mathbf{y}_u \in S} f(\mathbf{y}_u) g(\mathbf{y}_u)}{\sum_{\mathbf{y}'_u} f(\mathbf{y}'_u)} - \\ &\quad \frac{\sum_{\mathbf{y}_u} f(\mathbf{y}_u) g(\mathbf{y}_u) \sum_{\mathbf{y}_u \in S} f(\mathbf{y}_u)}{[\sum_{\mathbf{y}'_u} f(\mathbf{y}'_u)]^2} \Bigg) \\ &= \sum_{u=1}^{U} \sum_{\mathbf{y}_u \in S} \gamma(\mathbf{y}_u)(g(\mathbf{y}_u) - \overline{R}_u), \end{aligned} \quad (12)$$

where $\gamma(\mathbf{y}_u)$ is the normalized sequence probability, i.e., $\gamma(\mathbf{y}_u) = \frac{f(\mathbf{y}_u)}{\sum_{\mathbf{y}'_u} f(\mathbf{y}'_u)} = \frac{P(\mathbf{y}_u|\mathbf{x}_u)}{\sum_{\mathbf{y}'_u} P(\mathbf{y}'_u|\mathbf{x}_u)}$. And $\overline{R}_u$ is the averaged risk among all the hypothesized sequences for the training utterance u,

$$\overline{R}_u = \frac{\sum_{\mathbf{y}_u} f(\mathbf{y}_u) g(\mathbf{y}_u)}{\sum_{\mathbf{y}'_u} f(\mathbf{y}'_u)} = \sum_{\mathbf{y}_u} \gamma(\mathbf{y}_u) R(\mathbf{y}_u, \mathbf{y}^r_u). \quad (13)$$

Unlike hybrid systems, attention based sequence-to-sequence model makes the label prediction not only conditioned on the acoustic inputs but also previously emitted labels. Using lattices as the hypothesis set will practically complicate the forward computation procedure in MBR training. Therefore N-best obtained via beam-search has been found both efficient and effective for MBR training in [25]. In this work, we use a simple left-to-right beam-search algorithm [20] to generate the hypothesis set. The N-best set for MBR training is obtained by re-scoring the hypothesis set according to [26]:

$$\text{score}(\mathbf{y}, \mathbf{x}) = \log P(\mathbf{y}|\mathbf{x}) / \Big(\frac{(5 + |\mathbf{y}|)^\alpha}{(5+1)^\alpha} \Big). \quad (14)$$

Note that we've also tried the attention coverage penalty [26] for re-scoring but it never worked in our experiment which is in line with what was observed in [27]. A sequence-to-sequence model tends to make over-confident predictions and some approaches such as label smoothing have been proposed to combat this issue [22]. For beam-search, over-confident predictions will lead to too many alike hypothesized sequences among N-best which might prevent the MBR training procedure from seeing a more diverse hypothesis space. To this end, analogous to using a weaker language model for lattice generation in hybrid systems, we introduce softmax smoothing [28] during N-best generation. Specifically, when searching for N-best during MBR training, we modify Eq. (7) as,

$$p(y_i|y_{1:i-1}, \mathbf{x}) = \text{softmax}(\beta\, W_o \tilde{h}^{\text{dec}}_i), \quad \beta < 1, \quad (15)$$

to smooth the label prediction distribution and generate the scores at each step of beam-search.

3. Experiments

We conduct our experiments on both Switchboard-300hrs and Switchboard+Fisher-2000hrs datasets. For input features, we use 40 dimensional log-mel filterbanks and splice central frame with left 5 plus right 3 frames. The targets of our end-to-end system are a set of 49 characters which contains English letters, numbers, punctuations, special transcribed notations in Switchboard including '[laughter]', '[noise]', '[vocalized-noise]' plus '⟨space⟩', '⟨SOS⟩', '⟨EOS⟩' which are used as the indicators for the word boundary, start and end of the sequence. Stacked bidirectional and two-layer unidirectional LSTMs are used for the encoder and decoder, both with 512 hidden units. MLP based attention module is adopted in our experiment as described in Eq. (2) - (4). Adam algorithm [29] is used as the optimization method for all our experiment where we set $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ as suggested in [29]. For cross-entropy training, we use 0.001 as initial learning rate and halve it once the improvement on the validation set is saturating while for MBR training, we keep using 3×10^{-6}.

3.1. Frame subsampling and data augmentation

During preliminary experiments, we found two practical issues when training attention based sequence-to-sequence models on Switchboard. One issue is that the speech utterances in the dataset are typically much longer than in other tasks like voice search which causes the model training to easily run out of GPU memory. To reduce the length of the input sequences, we subsample the input by a factor of three similar to [22]. In the meantime, to guarantee the training procedure still sees the same number of input frames, we augment the training data three times using different speaking rates and volumes as in [6]. The other issue is even with data augmentation, sequence-to-sequence model tends to overfit severely. To alleviate the issue, we use dropout [30] and set the value to 0.2 throughout all our experiments.

Table 1: *WERs of baseline and input-feeding architectures with different maximum length limits trained on Switchboard-300hrs. See Fig. 1 for details on the two architectures in use.*

Architectures	MaxLen (# frames)	WERs(%)	
		SWB	Total
baseline	900	15.0	20.5
input-feeding	900	15.3	20.8
baseline	600	15.1	20.7
input-feeding	600	14.4	20.2

3.2. Scheduled sampling

We use teacher forcing, i.e., feeding the ground-truth label as the previous label prediction, at early stages of our model training. However it introduces the mismatch between training and inference. To this end, similar to [22], scheduled sampling [31] is adopted. Instead of the ground-truth label, the label predicted by the decoder from the last step is used with certain probability. We launch the scheduled sampling training once the cross-entropy loss improvement on the validation set is saturating.

3.3. Cross-entropy regularization for MBR training

It has been found in [25] that cross-entropy regularization is crucial for MBR training of the sequence-to-sequence models. In this work, the weighted cross-entropy loss of each hypothesis in the N-best list is used to regularize the MBR loss where the weight is the normalized probability $\gamma(\mathbf{y}_u)$. The regularized MBR loss function is:

$$\mathcal{L}'_{\text{MBR}} = \mathcal{L}_{\text{MBR}} + \lambda \sum_{u=1}^{U} \sum_{\mathbf{y}_u} \gamma(\mathbf{y}_u) \mathcal{L}_{\text{xent}}(\mathbf{y}_u, \mathbf{y}^r_u), \quad (16)$$

where λ is the regularization factor which we set to 0.01 throughout all our MBR experiments. Note that as label outputs are not frame-synchronized, the lengths of certain N-best $\mathbf{y}_u$ and the true label $\mathbf{y}^r_u$ can be different. We simply use padding or truncating to make sure the lengths match before computing the cross-entropy loss.

3.4. Experimental results on Switchboard-300hrs

We use pytorch [32] and Kaldi [33] to implement all the models and experiments in this work. First we conduct the experiments to compare baseline and input-feeding architectures as described in Section 2. For both architectures, we use 6-layer bidirectional LSTMs for the encoder and 2-layer unidirectional LSTMs for the decoder. We first sort the training utterances according to their length, group every 16 sequences, i.e., batch-size=16, and then shuffle the groups before training. As we discard the utterances that beyond the maximum length limits, we tried two maximum length limits for both architectures, 600

Table 2: *WERs of input-feeding models with different number of encoder layers trained on Switchboard-300hrs.*

#Enc. Layers	#Parameters	WERs(%)	
		SWB	Total
4	11.48M	16.2	22.0
5	13.08M	15.6	21.2
6	14.66M	14.4	20.2

Table 3: *WERs of input-feeding models trained with Teacher-forcing vs. Scheduled Sampling on Switchboard-300hrs*

Models	Sampling Probability	WERs(%)	
		SWB	Total
Teacher forcing	0	14.4	20.2
+Sampling	0.3	13.5	19.2
+Sampling	0.4	13.3	19.0
+Sampling	0.5	13.5	19.1

and 900 frames which translate to 18s and 27s in speech duration. As shown in Table 1, it seems the input-feeding models benefit more from limiting the training utterances to a reasonable length and the best model achieves 14.4% on the Eval2000-Switchboard, an absolute WER improvement of 0.6% over the best baseline architecture. We then study the effects of number of encoder layers by fixing the maximum length to 600. As shown in Table 2, by increasing the encoder depth, an absolute WER reduction of 1.8% is achieved by using 6 encoder layers than using only 4. Based on the best teacher forcing trained cross-entropy model, we perform scheduled sampling with probabilities from 0.3 to 0.5. As shown in Table 3, scheduled sampling trained model with sampling probability 0.4 achieves 13.3% on the Eval2000-Switchboard, i.e., an absolute WER improvement of 1.1% over the best teacher forcing trained model.

On top of the best scheduled sampling trained cross-entropy model, we perform MBR training using character level, word level edit-distance with and without softmax smoothing during N-best generation. Both batch-size and beam-size are set to 4 throughout all our MBR experiments. As shown in Table 4, all three MBR training setups outperform baseline cross-entropy model significantly whereas the word level edit-distance based MBR training is superior to the character level setup since it matches the WER metric better. With softmax smoothing, the model improves further by 0.3% and 0.5% on the Switchboard subset and full Eval2000 set respectively. Overall we obtain another 1.1% absolute improvement over the best scheduled sampling trained model from MBR training.

3.5. Experimental results on Switchboard+Fisher-2000hrs

We then follow the same training steps as in the Switchboard-300hrs experiments but use the full Switchboard+Fisher-2000hrs dataset as training data. All the results are summarized in Table 5 and for better comparisons with previous published systems, we also list the results for the CallHome subset of Eval2000. Finally, we compare our best models built on Switchboard-300hrs and Switchboard+Fisher-2000hrs with previous published systems in Table 6 and Table 7 respectively which show our end-to-end systems are highly competitive among state-of-the-art English conversational speech recognition systems.

Table 4: *WERs of MBR trained input-feeding models with character level, word level edit-distance without and with softmax smoothing trained on Switchboard-300hrs.*

Models	Method	WERs(%)	
		SWB	Total
Sampling	NA	13.3	19.0
+MBR	character level	12.8	18.4
+MBR	word level	12.5	18.3
+MBR	word level + softmax smoothing	12.2	17.8

Table 5: *WERs of baseline, input-feeding, scheduled sampling and MBR models trained on Switchboard+Fisher-2000hrs.*

Models	WERs(%)		
	SWB	CH	Total
baseline	10.8	19.6	15.2
input-feeding	9.3	17.3	13.3
+Sampling	8.5	16.6	12.6
+Sampling+MBR	8.3	16.1	12.2
+Sampling+MBR w. softmax smoothing	8.3	15.5	11.9

Table 6: *Comparing our best model to other hybrid and end-to-end systems built on the Switchboard-300hrs.*

	Systems	WERs(%)	
		SWB	CH
Hybrid	DNN+fMLLR+sMBR+Fisher Trigram [5]	12.6	24.1
	BLSTM+MMI+Ngram [34]	12.3	-
	BLSTM+ivec.+Ngram [35]	11.1	20.9
	BLSTM+fMLLR+ivec.+Ngram [36]	10.8	-
	BLSTM+ivec.+LFMMI+Fisher 4gram [6]	9.6	19.3
End-to-end	Attention Seq2Seq + Trigram [37]	25.8	46.0
	BRNN Grapheme CTC + Ngram [11]	20.0	31.8
	Acoustics-to-Word + noLM [15]	14.6	23.6
	Iterated CTC + Word RNN LM [12]	14.0	25.3
	Attention Seq2Seq + noLM (current)	12.2	23.3

Table 7: *Comparing our best model to other hybrid and end-to-end systems built on the Switchboard+Fisher-2000hrs.*

	Systems	WERs(%)	
		SWB	CH
Hybrid	BLSTM+ivec.+LFMMI+Fisher 4gram [6]	8.5	15.3
	ResNet+ivec.+LFMMI+Fisher 4gram [38]	8.6	15.2
	LACE+ivec.+LFMMI+Fisher 4gram [38]	8.5	15.2
	ResNet+ivec+ST+Fisher 4gram [39]	8.3	14.9
End-to-end	Iterated CTC + Word RNN LM [12]	10.2	17.7
	Acoustics-to-Word + noLM [15]	8.8	13.9
	Attention Seq2Seq + noLM [27]	8.6	17.8
	Attention Seq2Seq + noLM (current)	8.3	15.5

4. Conclusions

In this work, we proposed two improvements to attention based sequence-to-sequence models for end-to-end speech recognition systems on the standard Switchboard-300hrs task. Together with other training strategies such as dropout and scheduled sampling, our best model achieved WERs of 8.3%/15.5% on the Switchboard/CallHome subsets of Eval2000 without any external language models. Future work includes extending our work to Mandarin speech tasks and improving attention models with only text training data.

5. Acknowledgements

We would like to thank Rohit Prabhavalkar from Google for discussions on MBR training of sequence-to-sequence models.

6. References

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," vol. 20, January 2012, pp. 30–42.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, and T. Sainath, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012.

[3] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1533–1545, October 2014.

[4] T. N. Sainath, O. Vinyals, A. W. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *ICASSP 2015,*, 2015, pp. 4580–4584.

[5] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH 2013,*, 2013, pp. 2345–2349.

[6] 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,*, 2016, pp. 2751–2755.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[8] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010,*, 2010, pp. 1045–1048.

[9] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *ICASSP 2015,*, 2015, pp. 4225–4229.

[10] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.

[11] 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.

[12] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *ICASSP 2017*, 2017, pp. 4805–4809.

[13] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *CoRR*, vol. abs/1610.09975, 2016.

[14] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," in *Interspeech 2017*, 2017, pp. 959–963.

[15] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," *CoRR*, vol. abs/1712.03133, 2017.

[16] J. Li, G. Ye, R. Zhao, J. Droppo, and Y. Gong, "Acoustic-to-word model without OOV," *CoRR*, vol. abs/1711.10136, 2017.

[17] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *EMNLP2014*, Oct. 2014, pp. 1724–1734.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS2014*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv e-prints, Presented at ICLR 2015*, vol. abs/1409.0473, Sep. 2014.

[20] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *CoRR*, vol. abs/1506.07503, 2015.

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

[22] 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," *CoRR*, vol. abs/1712.17169, 2017.

[23] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *in Proc. Interspeech 17*, 2017.

[24] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP2015*, September 2015, pp. 1412–1421.

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

[26] Y. Wu and M. S. et. al, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[27] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, D. Seetapun, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," *CoRR*, vol. abs/1707.07413, 2017.

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

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[31] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *CoRR*, vol. abs/1506.03099, 2015.

[32] 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," 2017.

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

[34] S. Zhang, H. Jiang, S. Xiong, S. Wei, and L.-R. Dai, "Compact feedforward sequential memory networks for large vocabulary continuous speech recognition," in *Interspeech 2016*, 2016, pp. 3389–3393.

[35] I. Medennikov, A. Prudnikov, and A. Zatvornitskiy, "Improving english conversational telephone speech recognition," in *Interspeech 2016*, 2016, pp. 2–6. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-473

[36] G. Saon, T. Sercu, S. J. Rennie, and H. J. Kuo, "The IBM 2016 english conversational telephone speech recognition system," *CoRR*, vol. abs/1604.08242, 2016.

[37] 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*, 2016.

[38] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *CoRR*, vol. abs/1610.05256, 2016.

[39] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," *CoRR*, vol. abs/1703.02136, 2017.