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

Chao Weng¹, Jia Cui¹, Guangsen Wang², Jun Wang¹, Chengzhu Yu¹, Dan Su², Dong Yu¹

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

{cweng, jiaacui, vincegwang, 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 model was first introduced into speech recognition in [20]. Later on, an attention based system, namely listen, attend and spell (LAS), was examined into speech recognition in [20]. The inputs to the framework are typically several hundred frames of speech features such as log-mel filterbanks in Fig. 1-a. The inputs to the framework are typically several hundred frames of speech features such as log-mel filterbanks in Fig. 1-a. The attention module takes as inputs all encoder hidden states \( h^{\text{enc}} \), a sequence of hidden states with the length \( T \) which can be treated as a high level representation of the inputs:

\[
h^{\text{enc}} = h_1^{\text{enc}}, h_2^{\text{enc}}, \ldots, h_t^{\text{enc}}, \ldots, h_T^{\text{enc}}
\]

\[
= \text{Encoder}(x_1, x_2, \ldots, x_T).
\]

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

\[
a_{t,i} = \frac{\exp(\text{score}(h_t^{\text{dec}}, h_i^{\text{enc}}))}{\sum_{i'} \exp(\text{score}(h_t^{\text{dec}}, h_{i'}^{\text{enc}}))}.
\]

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.
2.2. 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, $h^\text{enc}_{i-1}$, to the decoder as depicted in Fig. 1-b. Therefore, Eq. (5) becomes:

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

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 $y$ denote the output sequence from the sequence-to-sequence model: $y = y_1, y_2, y_3, ..., y_L$. Given $U$ pairs of the training speech utterance $x$ and its corresponding reference label sequence $y'$, the MBR loss function can be written as:

$$L_{\text{MBR}}(x; y; y') = \sum_{u=1}^{U} \sum_{i=1}^{L} \sum_{y_u \in S} \frac{P(y_u | x_u) R(y_u, y'_i)}{\sum_{y_{u'} \in S} P(y_{u'} | x_u)},$$

where $y_u$ represents one of hypothesized output label sequences corresponds to $x_u$. $R(y_u, y'_i)$ is the risk function between a hypothesized and reference label sequence, e.g., edit-distance. $P(y_u | x_u)$ is the sequence probability given input $x_u$. According to the chain rule,

$$P(y_u | x_u) = P(y_1, y_2, y_3, ..., y_L | x_u) = p(y_1 | x_u) p(y_2 | y_1, x_u) \cdots p(y_L | y_{L-1}, x_u) = \prod_{i=1}^{L} p(y_i | y_{i-1}, x_u).$$

Note that $p(y_i | y_{i-1}, x_u)$ is exactly the output of sequence-to-sequence model as in Eq. (7). To perform MBR training of sequence-to-sequence model, we will need to derive the gradients of MBR loss function w.r.t. $p(y_i | y_{i-1}, x_u)$. For convenience, we use $p(y_i = y)$ as the shorthand for $p(y_i = y | y_{i-1}, x_u)$, i.e., the probability of the model emitting a particular label $y$ at the $i$th step, $f(y_u)$ and $g(y_u)$ as the shorthand for $P(y_u | x_u)$ and $R(y_u, y'_i)$ in Eq. (9). Accordingly, we define a hypothesis set $S = \{y_u | y_u = y\}$ which contains all the hypothesized sequences whose $i$th label is $y$. All hypothesized sequences can be divided into two disjoint sets, $y_u \in S$ and $y_u \notin S$. The MBR loss function can be rewritten as:

$$L_{\text{MBR}} = \sum_{u=1}^{U} \sum_{y_u \in S} f(y_u) g(y_u) + \sum_{y_u \notin S} f(y_u) g(y_u).$$

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

$$p(y_i | y_{i-1}, x) = \text{softmax}(W_0 \tilde{h}^\text{dec}_i).$$

$$\frac{\partial L_{\text{MBR}}}{\partial \log p(y_i = y)} = \sum_{u=1}^{U} \frac{\partial L_{\text{MBR}}}{p(y_i = y)} \cdot \frac{\partial p(y_i = y)}{\partial \log p(y_i = y)} = \sum_{u=1}^{U} \frac{\sum_{y_u \in S} f(y_u) g(y_u)}{\sum_{y_u \notin S} f(y_u)} - \frac{\sum_{y_u \notin S} f(y_u) g(y_u) \sum_{y_{u'} \in S} f(y_{u'})}{[\sum_{y_{u'} \in S} f(y_{u'})]^2}$$

$$= \sum_{y_u \in S} \gamma(y_u) (g(y_u) - \overline{P_u}),$$

where $\gamma(y_u) = g(y_u) / \sum_{y_{u'} \in S} f(y_{u'}).$
where $\gamma(y_u)$ is the normalized sequence probability, i.e.,
$$\gamma(y_u) = \frac{f(y_u)}{\sum_{y_u'} f(y_u')} = \frac{P(y_u|x)}{\sum_{y_u'} P(y_u'|x)},$$
and $\mathbb{R}_u$ is the averaged risk among all the hypothesized sequences for the training utterance $u$,
$$\mathbb{R}_u = \frac{\sum_{y_u} f(y_u) \gamma(y_u)}{\sum_{y_u} f(y_u')} = \sum_{y_u} \gamma(y_u) R(y_u, y_u').$$
(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]:
$$\text{score}(y, x) = \log P(y|x) / \left( \frac{(5 + |y|)^\alpha}{(5 + 1)^\alpha} \right).$$
(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,
$$p(y_i | y_{1:i-1}, x) = \text{softmax}(\beta W h_{i}^{\text{dec}}), \quad \beta < 1,$$
(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 $\times$ 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-300hrs. See Fig. 1 for details on the two architectures in use.

<table>
<thead>
<tr>
<th>Architectures</th>
<th>MaxLen (# frames)</th>
<th>WERs(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>900</td>
<td>13.0</td>
</tr>
<tr>
<td>input-feeding</td>
<td>900</td>
<td>15.3</td>
</tr>
<tr>
<td>baseline</td>
<td>600</td>
<td>15.1</td>
</tr>
<tr>
<td>input-feeding</td>
<td>600</td>
<td>14.4</td>
</tr>
</tbody>
</table>

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.

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 $\gamma(y_u)$. The regularized MBR loss function is:
$$L_{MBR}' = L_{MBR} + \lambda \sum_{u=1}^{U} \sum_{y_u} \gamma(y_u) L_{\text{sent}}(y_u, y'_u),$$
(16)
where $\lambda$ 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 $y_u$ and the true label $y'_u$ 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 exceed the maximum length limits, we try two maximum length limits for both architectures, 600
### 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.

### 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 other training strategies such as dropout and scheduled sampling.

### 5. Acknowledgements

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


