Advancing sequence-to-sequence based speech recognition

Zoltán Tüske, Kartik Audhkhasi, George Saon
IBM Research AI, Yorktown Heights, USA
zoltan.tuske@ibm.com

Abstract
The paper presents our endeavor to improve state-of-the-art speech recognition results using attention based neural network approaches. Our test focus was LibriSpeech, a well-known, publicly available, large, speech corpus, but the methodologies are clearly applicable to other tasks. After systematic application of standard techniques – sophisticated data augmentation, various dropout schemes, scheduled sampling, warm-restart –, and optimizing search configurations, our model achieves 4.09% and 11.7% word error rate (WER) on the test-clean and test-other sets, without any external language model. A powerful recurrent language model drops the error rate further to 2.7% and 8.2%. Thus, we not only report the lowest sequence-to-sequence model based numbers on this task, but our single system even challenges the best result known in the literature, namely a hybrid model together with recurrent language model rescoring. A simple ROVER combination of several of our attention based systems achieved 2.5% and 7.3% WER on the clean and other test sets.

Index Terms: encoder-decoder, attention, speech recognition, LibriSpeech

1. Introduction
With the widespread use of deep neural networks (NN), end-to-end approaches have rapidly gained popularity in speech recognition community as a result of dramatic simplification of the training and decoding pipelines. Such models can operate on characters, sub-words, or even full-words instead of phonetic sub-states, hence making pronunciation lexicon unnecessary. The model can directly map audio feature sequences (x₁^n) to word sequences (y₁^T).Expressed in a statistical framework, the aim is to learn the conditional posterior distribution as precisely as possible.

\[
p(y^n_T|x^n_1) = \prod_{l=1}^L p(y|y^{l-1}_1, x^n_T) \tag{1}
\]

Depending on the end-to-end approach, the alignment problem, which arises from the different length of the input and target sequences (T vs. L), is handled differently. In the CTC [1], RNN-transducer [2], or standard hybrid based approaches [3, 4], alignment is handled explicitly in the training criterion by introducing hidden variables. In order to keep the input and output streams synchronous, skip and/or repetition is allowed, or a special blank symbol is introduced to equalize the sequence lengths. The other frequently used approach, on which this paper will focus, is attention based encoder-decoder modeling [5, 6, 7, 8]. This model handles the alignment problem internally by squashing the input stream dynamically by a trainable attention mechanism in synchrony with the output sequence. The model is capable to handle problems with non-monotonic alignment exceptionally well, and has become the state-of-the-art approach in e.g. machine translation [9].

Despite the simple left-to-right nature of speech, several attempts have been made to reach the state-of-the-art performance of a hybrid system with attention based speech recognition models [10, 11, 12]. As has been shown, a sufficiently large database might be a key component to achieve this goal [13]. Nevertheless, the usual comparisons consider hybrid results after decoding with count based language model, and often discard the fact that encoder-decoder models learn neural-network-like word-sequence representation internally, which is much more powerful than count based models [14, 15]. Thus, it has still remained an open question whether attention based models can match the performance of a standard hybrid model combined with RNN-based rescoring/decoding.

Looking for the answer of the above question, we carried out several experiments with our attention based speech models on a relatively large, publicly available dataset, LibriSpeech. We present the detailed description of our successful attempt to achieve a new state of the art on this task by encoder-decoder based sequence-to-sequence model.

2. Experimental setup

2.1. Data description
LibriSpeech is a publicly available, read speech corpus [16]. The overall audio data for training sum up to roughly 1000 hours. The development and test sets contain approximately 10 hours of speech. Each set is split in two, the easier part is referred to as “clean” whereas the more challenging half is called “other”. A 200k lexicon with manually and automatically generated pronunciations is also provided. With this lexicon the out-of-vocabulary (OOV) rate is well below 1% on the development and test sets. A text corpus of 803M million running words is also available for language modeling (LM) purposes.

2.2. Attention model
Our attention based encoder-decoder model is summarized in Eq. 2 and Fig. 1.

\[
p(y^n_T|y^{l-1}_1, x^n_T) = \text{softmax} (W (c_l + d_l) + b_w)
\]

\[
c_l = \sum_{n=1}^N \alpha_{l,n} h_n(x^n_T) \tag{2}
\]

\[
\alpha_{l,n} = \text{softmax}_n \left( v^T \text{ReLU} (d_l + h_n + \bar{\alpha}_{l-1,n}) + b_n \right)
\]

After encoding the input sequence x^n_1 into h^n_1 with a deep, pyramidal, bidirectional long-short term memory (LSTM) network [17, 18, 10], h^n_1 is processed by the decoder network (Fig. 1). At the lth step, the posterior vector for the output symbols is generated from the transformed decoder state d_l and context-vector c_l. The context vector corresponds to the attention (\(\alpha\)) based linear combination of the encoded input vector sequence h^n_1. Our decoder network uses the location aware attention mechanism of [19]. And the previous attention is fed back after convolving it with a set of 1-dimensional kernel functions (A) along the time axis. The energy function of the attention calculation is performed by a single-layer rectified-linear-unit (ReLU) feed-forward neural network [20]. W, A (matri-
Figure 1: The attention based decoder network used in the experiments.

At recognition time, the following decision is made to find the optimal sequence using an external LM:

$$
g_{y_{1:L}}^L = \arg \max \_{y_{1:L}} \sum_{l=1}^{L} \log p(y_{l}|y_{l-1}, x_T^{l-1}) + \lambda_{LM} \sum_{l=1}^{L} \log p(y_{l}|y_{l-1}) + \lambda_{ten}L + \lambda_{ct} \sum_{n=1}^{N} \mathbb{I}_{(a \geq \tau)} \left( \max_{l} \alpha_{l,n} \right)
$$

A simple model combination of the sequence-to-sequence and LM in the first and second term is often referred to as shallow-fusion [21]. The third term denotes the length normalization [22], which helps promoting longer hypotheses. The last component corresponds to a modified version of the coverage term of [23, 24], where \( I \) is the indicator function, and \( \tau \) is a tunable parameter. Together with the max operation it forces the search to prefer paths with sharp and non-overlapping attention distributions. We found that a cumulative term might fire unnecessarily and boost weak hypothesis with unfocused attention. Eq. 3 was evaluated by beam search. The coverage term was updated at each step keeping track of the necessary statistics. Length normalization was applied when the hypothesis reached sentence end, before expanding the n-best list. The input and output of the encoder-decoder models are not handled synchronously, therefore special care should be taken to avoid running the hypophere indefinitely. We used a single attention mechanism, and \( d_t \) is calculated by a unidirectional LSTM NN.

Table 1: LSTM language models for LibriSpeech, perplexity (PPL) measured on the combined set of dev-clean and -other.

<table>
<thead>
<tr>
<th>unit</th>
<th>vocab.</th>
<th>#param.</th>
<th>PPL</th>
<th>PPL_word</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>29</td>
<td>54.8M</td>
<td>2.44</td>
<td>90.9</td>
</tr>
<tr>
<td>BPE</td>
<td>100</td>
<td>156</td>
<td>122.1M</td>
<td>3.47</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>1056</td>
<td>35.4M</td>
<td>6.08</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>10056</td>
<td>61.3M</td>
<td>8.91</td>
</tr>
<tr>
<td></td>
<td>30k</td>
<td>29998</td>
<td>74.0M</td>
<td>9.43</td>
</tr>
<tr>
<td>word</td>
<td>89115</td>
<td>111.9M</td>
<td>1.4*</td>
<td></td>
</tr>
</tbody>
</table>

*A2W, 0.7% of words were deleted

2.3. Training setup

Unless otherwise noted, our encoder-decoder model uses the following default settings resulted from a series of initial experiments. The input audio files are speed, tempo, and/or volume perturbed with 75% probability [25]. The encoder consists of 6 bidirectional LSTM layers with 512 nodes per layer and per direction. Its input is 50-dimensional speaker independent, utterance-level mean-normalized Gammatone (GT) features [26]. We also tested 40-dimensional MFCC features, which were speaker level mean-and-variance normalized. Following [27], with 40% probability a randomly chosen utterance was also injected as a sequence noise, and it was mixed up with 0.3 weight with the original feature stream at spectral level. In the first layers of the encoder, the frame rate was reduced by a factor of two per layer until the desired ratio was achieved [10, 28]. As a default setting \( T = 4N \), and the LSTM output is sub-sampled by max-pooling. Furthermore, the forward and backward directions of the LSTM outputs are summed up after every layer. We also apply 20% dropout rate to the LSTM outputs, and 20% drop-connect to the hidden-to-hidden matrices [29, 30]. The final dimension of the encoder output is 256, enforced by a linear bottleneck. In the decoder, the dropout probability is 5% for the embedding and 10% for the decoder LSTM layer. The output and the embedding layers model characters, byte-pair encoding (BPE) based sub-word units, or full-words [31]. According to Eq. 2, \( d_t, h_n, c_t, \) and \( \bar{c}_t \) have the same dimension, thus the LSTM and embedding layers in the decoder also have 256 nodes. Similarly, the attention output is filtered by 256 kernels (A) with kernel size 5. The models were trained to minimize the cross-entropy loss. We used scheduled sampling, forcing the teacher labels in 80% of the time [32]. Recordings of 24 randomly chosen speakers (9.6 hours) were selected for cross-validation. The randomly initialized models were trained from scratch for 50 epochs with variable batch size of up to 256 utterances. Depending on maximum length, the number of sequences in a batch was limited by GPU memory. In the first 10 epochs, sequences were presented in increasing length order to the network. Furthermore, the cross-entropy objective was complemented with CTC loss for 10 epochs, then linearly decayed in 5 epochs [33]. After 30 epochs, the learning rate was annealed by a factor of \( 1/\sqrt{2} \) in 20 steps. The final models were obtained after an additional 50-epoch training using warm-restart (SGDR) [34], resetting merely the learning rate to its initial value of 0.01. We always used Nesterov momentum set to 0.9 [35].

The LSTM LM and the BPE rules were trained on the merged set of the transcription of the acoustic data and the stand-alone text database (822M running words). Segmenting the LM data to characters resulted in 4.3B running tokens. All sub-word unit (including character) based attention systems and LMs used a special token to denote word end. The NNLMs have an embedding size of 512, two LSTM layers with 2048 nodes, and a 128-dimensional bottleneck layer before the softmax output. Similar to [36], drop-connect with 10% ratio was applied to the embedding and also to the hidden-to-hidden transformation matrices of the LSTM layers, and 10% of the outputs of these layers were also dropped out. The initial learning rate and Nesterov momentum were the same as above. After 10 epochs of training, the learning rate was annealed by a factor of \( 1/\sqrt{2} \) in 5 steps. It should be noted that the full-word system’s vocabulary was limited to 89k, the total number of unique words in the acoustic data. Therefore, during training the OOV words were simply deleted (1.2% of the data). The direct acoustic-to-word (A2W) model also imposed this limited vocabulary constraint in recognition time. Table 1 shows the model perplexities (PPL).
measured on the joint dev-clean and dev-other set. In order to compare the various models, we also present perplexities normalized by the number of running words. As can be seen, character and BPE-100 (indicating 100 replacement rules) models are competitive with BPE-10k models, despite modeling much longer sequences. Training a 4-layer LSTM for BPE-100 led to a 9% relative improvement in PPL.

All models were trained with PyTorch [37]. The decoding hyper-parameters \( \beta_{\text{seq}}, \beta_{\text{LM}}, \lambda_{\text{LM}}, \lambda_{\text{len}}, \lambda_{\text{act}}, \tau, \gamma \) were optimized on dev-clean using 1-dimensional grid search iteratively. During recognition, the beam size was 30 without LM, and it was set to 80 with LM.

### 3. Experimental results

**3.1. Optimizing dropout, effect of warm-restart, sequence-noise and perturbation using MFCC or GT features**

In the first set of experiments, we trained the models only for 50 epochs without warm-restart, and focused on optimizing our model at the input-level. As Table 2 shows, data perturbation turned out to be more important than sequence-noise injection, nevertheless sequence-noise consistently improved the results further. Regardless of the choice of features, we ended up roughly at the same performance. We also conducted several experiments to find the optimal dropout and drop-connect values around 0.2. As can also be seen in Table 2, additional 50 epochs of training with warm-restart (SGDR) improved the model significantly. We note that this system already outperforms the lowest stand-alone sequence-to-sequence numbers published on this task [38].

**3.2. Interaction of scheduled sampling and search configurations**

In these experiments we gradually switched off the scheduled sampling in training, the coverage term, and the prefix-tree search constraint in Eq. 3. According to [32, 38], serious word error rate (WER) degradation can be observed without scheduled sampling. Indeed, we obtained the worst results with 100% teacher forcing. However, vocabulary constraints and coverage stabilized the search, and the effect of scheduled sampling faded (Table 3). Nevertheless, the best results on the development set were obtained when all three techniques were applied. We also note, irrespective of scheduled sampling, without the coverage term we observed large, over 10% relative WER variation depending on whether the search hyper-parameters were optimized on clean or other set. The variation reduced to below 0.5% with coverage term and vocabulary constraint, which clearly indicates a much more stable search configuration.

### 3.3. Effect of modeling units, model size, sub-sampling rate, and language model

Table 4 and 5 show the results of varying modeling units, the sub-sampling rate in the encoder, and the effect of an external language model. As can be seen, the use of an LM resulted in a 30% relative WER improvement. Optimizing the sub-sampling rate for each representation unit, we found that, in contrast to other papers, larger BPE units are not beneficial if external language model is also used [11, 39]. Without an external language model (best results of each BPE system are indicated by italic font), however, large units (BPE-10k) achieved the best result on the more difficult test-other set. We explain these mixed results by the following. Clean sets can reasonably be recognized with small or even zero language model weight, thus the easier language modeling offered by larger units is not beneficial. In contrast, “other” sets are acoustically more challenging, and larger modeling units lead to better language modeling by the decoder, which helps reducing the confusion. Similarly, an external LM helps a great deal reducing the WER. As Table 1 shows, dedicated language models are roughly in the same ballpark; and word constraints further reduce the perplexity gap between those models. Thus, finer grade modeling of speech (less sub-sampling, and smaller units) is a crucial factor recognizing speech robustly, especially if only 1000 hours of training data is available. The best results were obtained by the BPE-100 system, it achieved 2.96% and 8.48% WER on test-clean and test-other. We point out that these results are even better than the best hybrid system with lattice rescoring [40].

Aiming at improving the sequence-to-sequence model results further, we also carried out experiments adding more and wider layers to the encoder and/or decoder. In Table 5, we present the best numbers for each modeling unit. We were also curious if more parameters could compensate for the stronger sub-sampling. Comparing the best WERs without the external LM in Table 5 (indicated by the italic font) with the corresponding number in Table 4 significant improvement was measured (e.g. 12.31 \( \rightarrow \) 11.72% on test other). However, the increased number of parameters could not outperform the best number in Table 5 when external LM was also used.
3.5. Comparison with others

Table 7 compares our best systems to other state-of-the-art results published in the literature. As can be seen, our single system outperforms the previous best by a large margin, 29% relative improvement on test-clean and 7% relative gain on test-other. We also ROVER-combined several of our attention based systems ([42]), which differ only in modeling unit, input features, and model size. Our combination results are significantly better than the combined output of multiple hybrid systems [40].

<table>
<thead>
<tr>
<th>model</th>
<th>ref.</th>
<th>LM</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>hybrid</td>
<td>[43]</td>
<td>4gram</td>
<td>3.3</td>
<td>8.7</td>
</tr>
<tr>
<td></td>
<td>[40]</td>
<td>RNN</td>
<td>3.12</td>
<td>8.28</td>
</tr>
<tr>
<td>wav2letter</td>
<td>[44]</td>
<td>conv</td>
<td>3.54</td>
<td>11.52</td>
</tr>
<tr>
<td></td>
<td>[41]</td>
<td>LSTM</td>
<td>3.54</td>
<td>11.32</td>
</tr>
<tr>
<td></td>
<td>[39]</td>
<td>LSTM</td>
<td>3.54</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>[38]</td>
<td>none</td>
<td>3.61</td>
<td>11.28</td>
</tr>
<tr>
<td>ours</td>
<td>[40]</td>
<td>LSTM</td>
<td>2.65</td>
<td>7.35</td>
</tr>
<tr>
<td>8 sys. comb.</td>
<td></td>
<td>none</td>
<td>2.68</td>
<td>7.56</td>
</tr>
<tr>
<td></td>
<td>ours</td>
<td>LSTM</td>
<td>2.23</td>
<td>6.63</td>
</tr>
</tbody>
</table>

4. Conclusions

By choosing training configurations and search parameters optimally for sequence-to-sequence models, we demonstrated that attention-based encoder-decoder models are truly able to challenge the current state of the art, the hybrid model with neural network based rescoring. With a set of simple techniques, we could set up a new baseline on LibriSpeech (2.54%, 7.98% on test-clean and test-other), which significantly outperforms all previous single system results, including hybrid ones. To reach this level of performance, we found smaller modeling units and moderate application of frame-rate reduction crucial. Combination results also indicated the true potential of encoder-decoder models: a straightforward variation in model size, representation unit, and input features led to further improvement. Thus, we believe, in the future even a single system should be able to reach that performance level.
5. References


