STATE-OF-THE-ART SPEECH RECOGNITION WITH SEQUENCE-TO-SEQUENCE MODELS

Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, Michiel Bacchiani

Google, USA

{chungchengc,tsainath,yonghui,prabhavalkar,drpng,zhifengc,anjuli
ronw,kanishkarao,kgonina,ndjaitly,boboli,chorowski,michiel}@google.com

ABSTRACT

Attention-based encoder-decoder architectures such as Listen, Attend, and Spell (LAS), subsume the acoustic, pronunciation and language model components of a traditional automatic speech recognition (ASR) system into a single neural network. In previous work, we have shown that such architectures are comparable to state-of-the-art ASR systems on dictation tasks, but it was not clear if such architectures would be practical for more challenging tasks such as voice search. In this work, we explore a variety of structural and optimization improvements to our LAS model which significantly improve performance. On the structural side, we show that word piece models can be used instead of graphemes. We also introduce a multi-head attention architecture, which offers improvements over the commonly-used single-head attention. On the optimization side, we explore synchronous training, scheduled sampling, label smoothing, and minimum word error rate optimization, which are all shown to improve accuracy. We present results with a unidirectional LSTM encoder for streaming recognition. On a $12,500$ hour voice search task, we find that the proposed changes improve the WER from 9.2% to 5.6%, while the best conventional system achieves 6.7%; on a dictation task our model achieves a WER of 4.1% compared to 5% for the conventional system.

1. INTRODUCTION

Sequence-to-sequence models have been gaining in popularity in the automatic speech recognition (ASR) community as a way of folding separate acoustic, pronunciation and language models (AM, PM, LM) of a conventional ASR system into a single neural network. There have been a variety of sequence-to-sequence models explored in the literature, including Recurrent Neural Network Transducer (RNN-T) [1], Listen, Attend and Spell (LAS) [2], Neural Transducer [3], Monotonic Alignments [4] and Recurrent Neural Aligner (RNA) [5]. While these models have shown promising results, thus far, it is not clear if such approaches would be practical to unseat the current state-of-the-art, HMM-based neural network acoustic models, which are combined with a separate PM and LM in a conventional system. Such sequence-to-sequence models are fully neural, without finite state transducers, a lexicon, or text normalization modules. Training such models is simpler than conventional ASR systems: they do not require bootstrapping from decision trees or time alignments generated from a separate system. To date, however, none of these models has been able to outperform a state-of-the art ASR system on a large vocabulary continuous speech recognition (LVCSR) task. The goal of this paper is to explore various structure and optimization improvements to allow sequence-to-sequence models to significantly outperform a conventional ASR system on a voice search task.

Since previous work showed that LAS offered improvements over other sequence-to-sequence models [6], we focus on improvements to the LAS model in this work. The LAS model is a single neural network that includes an *encoder* which is analogous to a conventional acoustic model, an *attender* that acts as an alignment model, and a *decoder* that is analogous to the language model in a conventional system. We consider both modifications to the model structure, as well as in the optimization process. On the structure side, first, we explore word piece models (WPM) which have been applied to machine translation [7] and more recently to speech in RNN-T [8] and LAS [9]. We compare graphemes and WPM for LAS, and find modest improvement with WPM. Next, we explore incorporating multi-head attention [10], which allows the model to learn to attend to multiple locations of the encoded features. Overall, we get 13% relative improvement in WER with these structure improvements.

On the optimization side, we explore a variety of strategies as well. Conventional ASR systems benefit from discriminative sequence training, which optimizes criteria more closely related to WER [11]. Therefore, in the present work, we explore training our LAS models to minimize the number of expected word errors (MWER) [12], which significantly improves performance. Second, we include scheduled sampling (SS) [13, 2], which feeds the previous label prediction during training rather than ground truth. Third, label smoothing [14] helps to make the model less confident in its predictions, and is a regularization mechanism that has successfully been applied in both vision [14] and speech tasks [15, 16]. Fourth, while many of our models are trained with asynchronous SGD [17], synchronous training has recently been shown to improve neural systems [18]. We find that all four optimization strategies allow for additional 27.5% relative improvement in WER on top of our structure improvements.

Finally, we incorporate a language model to rescore N-best lists in the second pass, which results in a further 3.4% relative improvement in WER. Taken together, the improvements in model structure and optimization, along with second-pass rescoring, allow us to improve a single-head attention, grapheme LAS system, from a WER of 9.2% to a WER of 5.6% on a voice search task. This provides a 16% relative reduction in WER compared to a strong conventional model baseline which achieves a WER of 6.7%. We also observe a similar

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

trend on a dictation task.

In Sections 2.2.1 and 2.4, we show how language models can be integrated. Section 2.2.2 further extends the model to multi-head attention. We explore discriminative training in Sections 2.3.1 and 2.3.2, and synchronous training regimes in Section 2.3.3. We use unidirectional encoders for low-latency streaming decoding.

2. SYSTEM OVERVIEW

In this section, we detail various structure and optimization improvements to the basic LAS model.

2.1. Basic LAS Model

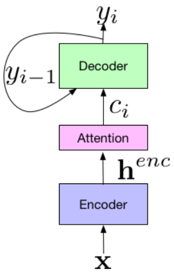

Fig. 1: Components of the LAS end-to-end model.

The basic LAS model, used for experiments in this work, consists of 3 modules as shown in Figure 1. The *listener* encoder module, which is similar to a standard acoustic model, takes the input features, $\mathbf{x}$, and maps them to a higher-level feature representation, $\mathbf{h}^{enc}$. The output of the encoder is passed to an *attender*, which determines which encoder features in $\mathbf{h}^{enc}$ should be attended to in order to predict the next output symbol, y_i, similar to a dynamic time warping (DTW) alignment module. Finally, the output of the attention module is passed to the *speller* (i.e., decoder), which takes the attention context, c_i, generated from the attender, as well as an embedding of the previous prediction, y_{i-1}, in order to produce a probability distribution, $P(y_i|y_{i-1},\ldots,y_0,\mathbf{x})$, over the current sub-word unit, y_i, given the previous units, $\{y_{i-1},\ldots,y_0\}$, and input, $\mathbf{x}$.

2.2. Structure Improvements

2.2.1. *Wordpiece models*

Traditionally, sequence-to-sequence models have used graphemes (characters) as output units, as this folds the AM, PM and LM into one neural network, and side-steps the problem of out-of-vocabulary words [2]. Alternatively, one could use longer units such as word pieces or shorter units such as context-independent phonemes [19]. One of the disadvantages of using phonemes is that it requires having an additional PM and LM, and was not found to improve over graphemes in our experiments [19].

Our motivation for looking at word piece models (WPM) is a follows. Typically, word-level LMs have a much lower perplexity compared to grapheme-level LMs [20]. Thus, we feel that modeling word pieces allows for a much stronger decoder LM compared to graphemes. In addition, modeling longer units improves the effective memory of the decoder LSTMs, and allows the model to potentially memorize pronunciations for frequently occurring words. Furthermore, longer units require fewer decoding steps; this speeds up inference in these models significantly. Finally, WPMs have shown good performance for other sequence-to-sequence models such as RNN-T [8].

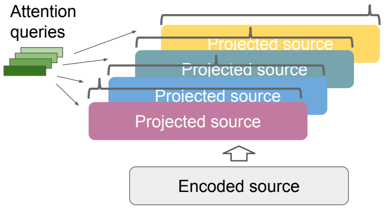

Fig. 2: Multi-headed attention mechanism.

The word piece models [21] used in this paper are sub-word units, ranging from graphemes all the way up to entire words. Thus, there are no out-of-vocabulary words with word piece models. The word piece models are trained to maximize the language model likelihood over the training set. As in [7], the word pieces are "position-dependent", in that a special word separator marker is used to denote word boundaries. Words are segmented deterministically and independent of context, using a greedy algorithm.

2.2.2. *Multi-headed attention*

Multi-head attention (MHA) was first explored in [10] for machine translation, and we extend this work to explore the value of MHA for speech. Specifically, as shown in Figure 2, MHA extends the conventional attention mechanism to have multiple heads, where each head can generate a different attention distribution. This allows each head to have a different role on attending the encoder output, which we hypothesize makes it easier for the decoder to learn to retrieve information from the encoder. In the conventional, single-headed architecture, the model relies more on the encoder to provide clearer signals about the utterances so that the decoder can pickup the information with attention. We observed that MHA tends to allocate one of the attention head to the beginning of the utterance which contains mostly background noise, and thus we hypothesize MHA better distinguish speech from noise when the encoded representation is less ideal, for example, in degraded acoustic conditions.

2.3. Optimization improvements

2.3.1. *Minimum Word Error Rate (MWER) Training*

Conventional ASR systems are often trained to optimize a sequence-level criterion (e.g., state-level minimum Bayes risk (sMBR) [11]) in addition to CE or CTC training. Although the loss function that we optimize for the attention-based systems is *a sequence-level loss function*, it is not closely related to the metric that we actually care about, namely word error rate. There have been a variety of methods explored in the literature to address this issue in the context of sequence-to-sequence models [22, 23, 5, 12]. In this work, we focus on the minimum expected word error rate (MWER) training that we proposed in [12].

In the MWER strategy, our objective is to minimize the expected number of word errors. The loss functon is given by Equation 1, where $\mathcal{W}(\mathbf{y},\mathbf{y}^*)$ denotes the number of word errors in the hypothesis, $\mathbf{y}$, compared to the ground-truth label sequence, $\mathbf{y}^*$. This first term is interpolated with the standard cross-entropy based loss, which we find is important in order to stabilize training [12, 24].

$$\mathcal{L}_{\text{MWER}} = \mathbb{E}_{P(\mathbf{y}|\mathbf{x})}[\mathcal{W}(\mathbf{y},\mathbf{y}^*)] + \lambda\mathcal{L}_{\text{CE}} \qquad (1)$$

The expectation in Equation 1 can be approximated either via sampling [5] or by restricting the summation to an N-best list of decoded hypotheses as is commonly used for sequence training [11]; the latter is found to be more effective in our experiments [12]. We denote by $\text{NBest}(\mathbf{x}, N) = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$, the set of N-best hypotheses computed using beam-search decoding [25] for the input utterance $\mathbf{x}$. The loss function can then be approximated as shown in Equation 2, which weights the *normalized* word errors from each hypothesis by the probability $\widehat{P}(\mathbf{y}_i|\mathbf{x})$ concentrated on it:

$$\mathcal{L}_{\text{MWER}}^{\text{N-best}} = \frac{1}{N} \sum_{y_i \in \text{NBest}(\mathbf{x}, N)} [\mathcal{W}(\mathbf{y_i}, \mathbf{y}^*) - \widehat{\mathcal{W}}]\widehat{P}(\mathbf{y}_i|\mathbf{x}) + \lambda \mathcal{L}_{\text{CE}} \quad (2)$$

where, $\widehat{P}(\mathbf{y}_i|\mathbf{x}) = \frac{P(\mathbf{y}_i|\mathbf{x})}{\sum_{\mathbf{y}_i \in \text{NBest}(\mathbf{x}, N)} P(\mathbf{y}_i|\mathbf{x})}$, represents the distribution re-normalized over just the N-best hypotheses, and $\widehat{\mathcal{W}}$ is the average number of word errors over all hypotheses in the N-best list. Subtracting $\widehat{\mathcal{W}}$ is applied as a form of variance reduction since it does not affect the gradient [12].

2.3.2. *Scheduled Sampling*

We explore scheduled sampling [13] for training the decoder. Feeding the ground-truth label as the previous prediction (so-called *teacher forcing*) helps the decoder to learn quickly at the beginning, but introduces a mismatch between training and inference. The scheduled sampling process, on the other hand, samples from the probability distribution of the previous prediction (i.e., from the softmax output) and then uses the resulting token to feed as the previous token when predicting the next label. This process helps reduce the gap between training and inference behavior. Our training process uses teacher forcing at the beginning of training steps, and as training proceeds, we linearly ramp up the probability of sampling from the model's prediction to 0.4 at the specified step, which we then keep constant until the end of training. The step at which we ramp up the probability to 0.4 is set to 1 million steps and $100,000$ steps for asynchronous and synchronous training respectively (See section 2.3.3).

2.3.3. *Asynchronous and Synchronous Training*

We compare both asynchronous [17] and synchronous training [18]. As shown in [18], synchronous training can potentially provide faster convergence rates and better model quality, but also requires more effort in order to stabilize network training. Both approaches have a high gradient variance at the beginning of the training when using multiple replicas [17], and we explore different techniques to reduce this variance. In asynchronous training we use replica ramp up: that is, the system will not start all training replicas at once, but instead start them gradually. In synchronous training we use two techniques: learning rate ramp up and a gradient norm tracker. The learning rate ramp up starts with the learning rate at 0 and gradually increases the learning rate, providing a similar effect to replica ramp up. The gradient norm tracker keeps track of the moving average of the gradient norm, and discards gradients with significantly higher variance than the moving average. Both approaches are crucial for making synchronous training stable.

2.3.4. *Label smoothing*

Label smoothing [14, 16] is a regularization mechanism to prevent the model from making over-confident predictions. It encourages the model to have higher entropy at its prediction, and therefore makes the model more adaptable. We followed the same design as [14] by smoothing the ground-truth label distribution with a uniform distribution over all labels.

2.4. Second-Pass Rescoring

While the LAS decoder topology is that of neural language model (LM), it can function as a language model; but it is only exposed to training transcripts. An external LM, on the other hand, can leverage large amounts of additional data for which we only have text (no audio). To address the potentially weak LM learned by the decoder, we incorporate an external LM during inference only.

The external LM is a large 5-gram LM trained on text data from a variety of domains. Since domains have different predictive value for our LVCSR task, domain-specific LMs are first trained, then combined together using Bayesian-interpolation [26].

We incorporate the LM in the second-pass by means of log-linear interpolation. In particular, given the N-best hypotheses produced by the LAS model via beam search, we determine the final transcript $\mathbf{y}^*$ as:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \log P(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y}) + \gamma \text{len}(\mathbf{y}) \quad (3)$$

where, P_{LM} is provided by the LM, $\text{len}(\mathbf{y})$ is the number of words in $\mathbf{y}$, and λ and γ are tuned on a development set. Using this criterion, transcripts which have a low language model probability will be demoted in the final ranked list. Additionally, the the last term addresses the common observation that the incorporation of an LM leads to a higher rate of deletions.

3. EXPERIMENTAL DETAILS

Our experiments are conducted on a ∼12,500 hour training set consisting of 15 million English utterances. The training utterances are anonymized and hand-transcribed, and are representative of Google's voice search traffic. This data set is created by artificially corrupting clean utterances using a room simulator, adding varying degrees of noise and reverberation such that the overall SNR is between 0dB and 30dB, with an average SNR of 12dB. The noise sources are from YouTube and daily life noisy environmental recordings. We report results on a set of ∼14.8K utterances extracted from Google traffic, and also evaluate the resulting model, trained with only voice search data, on a set of 15.7K dictation utterances that have longer sentences than the voice search utterances.

All experiments use 80-dimensional log-Mel features, computed with a 25ms window and shifted every 10ms. Similar to [27, 28], at the current frame, t, these features are stacked with 3 frames to the left and downsampled to a 30ms frame rate. The encoder network architecture consists of 5 long short-term memory [29] (LSTM) layers. We explored both unidirectional [30] and bidirectional LSTMs, where the unidirectional LSTMs have $1,400$ hidden units and bidirectional LSTMs have $1,024$ hidden units in each direction ($2,048$ per layer). Unless otherwise stated, experiments are reported with unidirectional encoders. Additive attention [31] is used for both single-headed and multi-headed attention experiments. All multi-headed attention experiments use 4 heads. The decoder network is a 2-layer LSTM with $1,024$ hidden units per layer.

All neural networks are trained with the cross-entropy criterion (which is used to initialize MWER training) and are trained using TensorFlow [32].

4. RESULTS

4.1. Structure Improvements

Our first set of experiments explore different structure improvements to the LAS model. Table 1 compares performance for LAS models given graphemes (E1) and WPM (E2). The table indicates that WPM perform slightly better than graphemes. This is consistent with the finding in [8] that WPM provides a stronger decoder LM compared to graphemes, resulting in roughly a 2% relative improvement in WER (WERR).

Second, we compare the performance of MHA with WPM, as shown by experiment E3 in the table. We see MHA provides around a 11.1% improvement. This indicates that having the model focus on multiple points of attention in the input signal, which is similar in spirit to having a language model passed from the encoder, helps significantly. Since models with MHA and WPM perform best, we explore the proposed optimization methods on top of this model in the rest of the paper.

Exp-ID	Model	WER	WERR
E1	Grapheme	9.2	-
E2	WPM	9.0	2.2%
E3	+ MHA	8.0	11.1%

Table 1: Impact of word piece models and multi-head attention.

4.2. Optimization improvements

We explore the performance of various optimization improvements discussed in Section 2.3. Table 2 shows that including synchronous training (E4) on top of the WPM+MHA model provides a 3.8% improvement. Furthermore, including scheduled sampling (E5) gives an additional 7.8% relative improvement in WER; label smoothing gives an additional 5.6% relative improvement. Finally, MWER training provides 13.4%. Overall, the gain from optimizations is around 27.5%, moving the WER from 8.0% to 5.8%.

We see that synchronous training, in our configuration, yields a better converged optimum at similar wall clock time. Interestingly, while scheduled sampling and minimum word error rate are both discriminative methods, we have observed that their combination continues to yield additive improvements. Finally, regularization with label smoothing, even with large amounts of data, is proven to be beneficial.

Exp-ID	Model	WER	WERR
E2	WPM	9.0	-
E3	+ MHA	8.0	11.1%
E4	+ Sync	7.7	3.8%
E5	+ SS	7.1	7.8%
E6	+ LS	6.7	5.6%
E7	+ MWER	5.8	13.4%

Table 2: Sync training, scheduled sampling (SS), label smoothing (LS) and minimum word error rate (MWER) training improvements.

4.3. Incorporating Second-Pass Rescoring

Next, we incorporate second-pass rescoring into our model. As can be see in Table 3, second-pass rescoring improves the WER by 3.4%, from 5.8% to 5.6%.

4.4. Unidirectional vs. Bidirectional Encoders

Now that we have established the improvements from structure, optimization and LM strategies, in this section we compare the gains on a unidirectional and bidirectional systems. Table 4 shows that the proposed changes give a 37.8% relative reduction in WER for a unidirectional system, while a slightly smaller improvement of 28.4% for a bidirectional system. This illustrates that most proposed methods offer improvements independent of model topology.

Exp-ID	Model	WER
E7	WPM + MHA + Sync + SS + LS + MWER	5.8
E8	+ LM	**5.6**

Table 3: In second pass rescoring, the log-linear combination with a larger LM results in a 0.2% WER improvement.

Exp-ID	Model	Unidi	Bidi
E2	WPM	9.0	7.4
E8	WPM + all	**5.6**	**5.3**
WERR	-	37.8%	28.4%

Table 4: Both unidirectional and bidirectional models benefit from cumulative improvements.

4.5. Comparison with the Conventional System

Finally, we compare the proposed LAS model in E8 to a state-of-the-art, discriminatively sequence-trained low frame rate (LFR) system [28] in terms of WER. Table 5 shows the proposed sequence-to-sequence model (E8) offers a 16% and 18% relative improvement in WER over our production system (E9) on voice search (VS) and dictation (D) task respectively. Furthermore, comparing the size of the first-pass models, the LAS model is around 18 times smaller than the conventional model. It is important to note that the second pass model is 80 GB and still dominates model size.

Exp-ID	Model	VS/D	1st pass Model Size
E8	Proposed	**5.6/4.1**	**0.4 GB**
E9	Conventional LFR system	6.7/5.0	0.1 GB (AM) + 2.2 GB (PM) + 4.9 GB (LM) = 7.2GB

Table 5: Resulting WER on voice search (VS)/dictation (D). The improved LAS outperforms the conventional LFR system while being more compact. Both models use second-pass rescoring.

5. CONCLUSION

We designed an attention-based model for sequence-to-sequence speech recognition. The model integrates acoustic, pronunciation, and language models into a single neural network, and does not require a lexicon or a separate text normalization component. We explored various structure and optimization mechanisms for improving the model. Cumulatively, structure improvements (WPM, MHA) yielded an 11% improvement in WER, while optimization improvements (MWER, SS, LS and synchronous training) yielded a further 27.5% improvement, and the language model rescoring yielded another 3.4% improvement. Applied on a Google Voice Search task, we achieve a WER of 5.6%, while a hybrid HMM-LSTM system achieves 6.7% WER. Tested the same models on a dictation task, our model achieves 4.1% and the hybrid system achieves 5% WER. We note however, that the unidirectional LAS system has the limitation that the entire utterance must be seen by the encoder, before any labels can be decoded (although, we encode the utterance in a streaming fashion). Therefore, an important next step is to revise this model with an streaming attention-based model, such as Neural Transducer [33].

6. REFERENCES

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

[2] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[3] N. Jaitly, D. Sussillo, Q. V. Le, O. Vinyals, I. Sutskever, and S. Bengio, "An Online Sequence-to-sequence Model Using Partial Conditioning," in *Proc. NIPS*, 2016.

[4] C. Raffel, M. Luong, P. J. Liu, R.J. Weiss, and D. Eck, "Online and Linear-Time Attention by Enforcing Monotonic Alignments," in *Proc. ICML*, 2017.

[5] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent Neural Aligner: An Encoder-Decoder Neural Network Model for Sequence to Sequence Mapping," in *Proc. Interspeech*, 2017.

[6] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A Comparison of Sequence-to-sequence Models for Speech Recognition," in *Proc. Interspeech*, 2017.

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

[8] K. Rao, R. Prabhavalkar, and H. Sak, "Exploring Architectures, Data and Units for Streaming End-to-End Speech Recognition with RNN-Transducer," in *Proc. ASRU*, 2017.

[9] William Chan, Yu Zhang, Quoc Le, and Navdeep Jaitly, "Latent Sequence Decompositions," in *ICLR*, 2017.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, 2017.

[11] B. Kingsbury, "Lattice-Based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling," in *Proc. ICASSP*, 2009.

[12] 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 *Proc. ICASSP (accepted)*, 2018.

[13] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks," in *Proc. NIPS*, 2015, pp. 1171–1179.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," in *Proc. NIPS*, 2015.

[16] J. K. Chorowski and N. Jaitly, "Towards Better Decoding and Language Model Integration in Sequence to Sequence Models," in *Proc. Interspeech*, 2017.

[17] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A.Y. Ng, "Large Scale Distributed Deep Networks," in *Proc. NIPS*, 2012.

[18] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: training imagenet in 1 hour," *CoRR*, vol. abs/1706.02677, 2017.

[19] T.N. Sainath, R. Prabhavalkar, S. Kumar, S. Lee, D. Rybach, A. Kannan, V. Schogol, P. Nguyen, B. Li, Y. Wu, Z. Chen, and C. Chiu, "No need for a lexicon? evaluating the value of the pronunciation lexica in end-to-end models," in *Proc. ICASSP (accepted)*, 2018.

[20] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An Analysis of Incorporating an External Language Model into a Sequence-to-Sequence Model," in *Proc. ICASSP (accepted)*, 2018.

[21] M. Schuster and K. Nakajima, "Japanese and Korean voice search," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.

[22] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-End Attention-based Large Vocabulary Speech Recognition," in *Proc. ICASSP*, 2016.

[23] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *Proc. ICLR*, 2016.

[24] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*. IEEE, 2002, vol. 1, pp. I–105.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014.

[26] C. Allauzen and M. Riley, "Bayesian language model interpolation for mobile speech input," *Interspeech*, 2011.

[27] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition," in *Proc. Interspeech*, 2015.

[28] G. Pundak and T. N. Sainath, "Lower Frame Rate Neural Network Acoustic Models," in *Proc. Interspeech*, 2016.

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[30] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov 1997.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proc. ICLR*, 2015.

[32] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Available online: http://download.tensorflow.org/paper/whitepaper2015.pdf, 2015.

[33] T. N. Sainath, C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, , and Z. Chen, "Improving the Performance of Online Neural Transducer models," in *Proc. ICASSP (accepted)*, 2018.