INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

A Comparison of Sequence-to-Sequence Models for Speech Recognition

Rohit Prabhavalkar[1], *Kanishka Rao*[1], *Tara N. Sainath*[1], *Bo Li*[1], *Leif Johnson*[1], *Navdeep Jaitly*[2†]

[1]Google Inc., U.S.A
[2]NVIDIA, U.S.A.

{prabhavalkar,kanishkarao,tsainath,boboli,leif}@google.com, njaitly@nvidia.com

Abstract

In this work, we conduct a detailed evaluation of various all-neural, end-to-end trained, sequence-to-sequence models applied to the task of speech recognition. Notably, each of these systems directly predicts graphemes in the written domain, without using an external pronunciation lexicon, or a separate language model. We examine several sequence-to-sequence models including connectionist temporal classification (CTC), the recurrent neural network (RNN) transducer, an attention-based model, and a model which augments the RNN transducer with an attention mechanism.

We find that the sequence-to-sequence models are competitive with traditional state-of-the-art approaches on dictation test sets, although the baseline, which uses a separate pronunciation and language model, outperforms these models on voice-search test sets.

Index Terms: sequence-to-sequence models, attention models, end-to-end models, RNN transducer

1. Introduction

Most state-of-the-art automatic speech recognition (ASR) systems are comprised of separate acoustic, pronunciation, and language modeling components that are trained independently [1, 2]. The acoustic model is typically trained to recognize context-dependent (CD) states or phonemes, by bootstrapping from an existing model which is used for alignment. The pronunciation model, curated by expert linguists, maps the sequences of phonemes produced by the acoustic model into word sequences. Word sequences are scored using language models trained on large amounts of text data, which estimate probabilities of word sequences.

There has been growing interest in building end-to-end trained systems that directly map the input acoustic speech signal to grapheme or word sequences [3, 4, 5, 6, 7, 8, 9, 10, 11, 12].[1] In such sequence-to-sequence models, the acoustic, pronunciation, and language modeling components are trained jointly in a single system. Since these models directly predict graphemes or words, the process of decoding utterances is greatly simplified.

The connectionist temporal classification (CTC) criterion [13] has been used to train end-to-end systems that directly predict grapheme sequences [5, 12]. In recent work, Soltau et al. [8] train a CTC-based model with word output targets, which was shown to outperform a state-of-the-art CD-phoneme baseline on a YouTube video captioning task. The basic CTC model was extended by Graves [3] to include a separate recurrent language model component, in a model referred to as the recurrent neural network (RNN) transducer. Although this model has shown promising results [4] on TIMIT [14] phone recognition tasks, to the best of our knowledge, it has not been evaluated on a large vocabulary continuous speech recognition (LVCSR) task.

Attention-based models have become increasingly popular [6, 7, 9, 10, 11]. These models consist of an *encoder* network, which maps the input acoustics into a higher-level representation, and an *attention-based decoder* that predicts the next output symbol conditioned on the full sequence of previous predictions. Although these models have shown promising results, when evaluated without language models (LMs), performance has typically been found to be much worse than the baseline, especially on more challenging tasks: For example, the system presented in [6] has a $\sim$76% relative degredation in word error rate (WER) over the baseline on a voice-search task; Lu et al. [11] find that WERs are double those of a baseline system on the Switchboard task [15].

In this work, we explore various sequence-to-sequence approches on an LVCSR task. We compare a CTC-trained system which directly outputs grapheme sequences, the RNN transducer, attention-based models, and a novel model that augments the RNN transducer with attention. In contrast to most previous works, our models are trained on a very large amount of transcribed acoustic data, totalling $\sim$12,500 hours. We find that when trained on such large amounts of training data, sequence-to-sequence approaches are competitive with a strong state-of-the-art baseline system on dictation test sets. When evaluated on voice-search test sets, which exhibit more variability and have higher perplexity than dictation test sets, we find that the sequence-to-sequence approaches are still 13–35% worse than the baseline, but the gap between our models and the baseline is significantly lower than has been reported in previous work (e.g., [6]). Furthermore, we find that these systems are capable of implicitly learning the mapping from *spoken* to *written* forms (e.g., "one hundred dollars" to "$100"), which is typically accomplished in the baseline system through special rules defined for this purpose.

The organization of the rest of the paper is as follows: In Section 2, we review the various sequence-to-sequence modeling approaches that are compared in this work. We describe our experimental setup and discuss our results in Section 3 and Section 4, respectively. We analyze our results in Section 4.1, before concluding in Section 5.

2. Sequence-to-Sequence Models

In this section, we describe the various sequence-to-sequence modeling approaches that are compared in this work. We as-

[†]Work performed while at Google.

[1]In this work, we refer to a model as *sequence-to-sequence* if it directly maps a sequence of input acoustic features into a sequence of graphemes or words. By *end-to-end* trained, we mean that the system is trained to optimize criteria that are related to the final evaluation metric that we are interested in (typically, word error rate). We would, therefore, consider a CTC-trained grapheme model to be an end-to-end trained, sequence-to-sequence model.

http://dx.doi.org/10.21437/Interspeech.2017-233

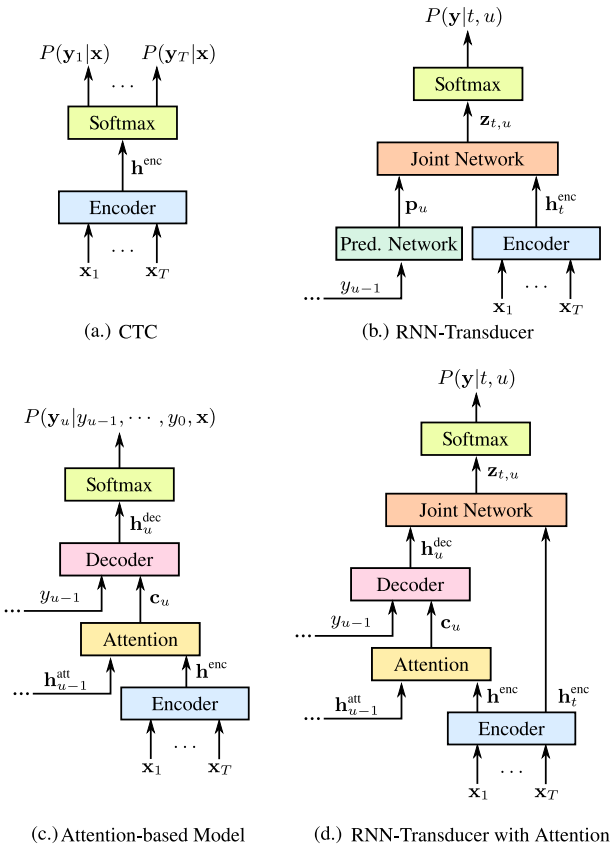

Figure 1: *A schematic representation of various sequence-to-sequence modeling approaches compared in this work.*

sume that the input speech waveform has been suitably parameterized in to a sequence of d-dimensional feature vectors, which we denote by $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbf{R}^d$. We denote the set of grapheme symbols output by the model by $\mathcal{Y}$, and denote the output sequence by $\mathbf{y} = (y_1, y_2, ..., y_L)$. For ASR, the number of output graphemes, L, is typically much smaller than the number of acoustic frames, T.

2.1. Connectionist Temporal Classification (CTC)

The CTC criterion was proposed by Graves et al. [13] as a way of training end-to-end models without requiring a frame-level alignment of the target labels for a training utterance. CTC augments the set of target labels with an additional "blank" symbol, denoted $\langle b \rangle$. Given a target label sequence $\mathbf{y}$ corresponding to the utterance $\mathbf{x}$, let $\mathcal{B}(\mathbf{y}, \mathbf{x})$ be the set of all sequences consisting of the labels in $\mathcal{Y} \cup \{\langle b \rangle\}$, which are of length $|\mathbf{x}| = T$, and which are identical to $\mathbf{y}$ after first collapsing consecutive repeated targets and then removing any blank symbols (e.g., "$x\langle b \rangle xx\langle b \rangle y \rightarrow xxy$"). Thus, any sequence in $\mathcal{B}(\mathbf{y}, \mathbf{x})$ corresponds to a frame-level assignment of the label sequence in $\mathbf{y}$. CTC then defines the probability of the output sequence conditioned on the acoustics as:

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\hat{\mathbf{y}} \in \mathcal{B}(\mathbf{y},\mathbf{x})} \prod_{t=1}^{T} P(\hat{y}_t|\mathbf{x}) \quad (1)$$

where, we assume that labels at each time step are independent, given the acoustics.

The conditional probability of the labels at each time step, $P(\hat{\mathbf{y}}_t|\mathbf{x})$, is estimated using a deep recurrent neural network, which we refer to as the *encoder*. The encoder computes a sequence of vectors $\mathbf{h}^{\text{enc}} = (\mathbf{h}_1^{\text{enc}}, \cdots, \mathbf{h}_T^{\text{enc}})$, which are treated as logits and passed to a single softmax layer which predicts a probability distribution over the set of blank-augmented output symbols, $\mathcal{Y} \cup \{\langle b \rangle\}$, as illustrated in Figure 1 (a.). The model can be trained to maximize Equation 1 by using gradient descent, where the required gradients can be computed using the forward-backward algorithm [13].

2.2. RNN Transducer

The CTC model is similar to an *acoustic model* in a traditional ASR system. The RNN transducer [3, 4] augments the encoder network from the CTC model architecture with a separate recurrent *prediction network* over the output symbols, as illustrated in Figure 1 (b.). Intuitively, the encoder can be thought of as an acoustic model, while the prediction network is analogous to a language model. The prediction network receives as input the previous grapheme label prediction, $y_{u-1} \in \mathcal{Y} \cup \{\langle \text{sos} \rangle\}$, and computes an output vector $\mathbf{p}_u$, which is dependent on the entire sequence of labels $y_0, \cdots, y_{u-1}$. The special label $\langle \text{sos} \rangle$, which indicates the start of the sentence, is input to the prediction network at the first time step, y_0.

The encoder outputs $\mathbf{h}_t^{\text{enc}}$ and the prediction outputs $\mathbf{p}_u$ are passed to a *joint* network, which computes output logits $\mathbf{z}_{t,u}$ for each input, t, in the encoder sequence and label, u, from the prediction network, as follows:

$$\mathbf{h}_{t,u}^{\text{joint}} = \tanh(A\mathbf{h}_t^{\text{enc}} + B\mathbf{p}_u + b) \quad (2)$$

$$\mathbf{z}_{t,u} = D\mathbf{h}_{t,u}^{\text{joint}} + d \quad (3)$$

where, A, B, b, D, d are parameters of the model. The logits $\mathbf{z}_{t,u}$ are then passed to a softmax layer which defines a probability distribution over the set of output targets and the blank symbol ($\mathcal{Y} \cup \{\langle b \rangle\}$), for each combination of acoustic frame t and output label u.

The model can be optimized using gradient descent by computing the required gradients using a dynamic programming algorithm; we refer the interested reader to [3, 4] for additional details on training and inference. We note here that inference in the RNN transducer is performed in a frame-synchronous manner, and thus the model can be used to perform streaming recognition if a unidirectional encoder is used. In this work, however, we use a bidirectional encoder to ensure that the results are comparable to the attention-based model, which uses a bidirectional encoder.

2.3. Attention-based Models

An attention-based model (e.g., Listen-Attend-and-Spell [6]) contains an encoder network, as in the RNN transducer model. However, unlike the RNN transducer, in which the encoder and the prediction network are modeled independently and combined in the joint network, an attention-based model uses a single *decoder* to produce a distribution over the labels conditioned on the full sequence of previous predictions and the acoustics: $P(\mathbf{y}_u|y_{u-1}, \cdots, y_0, \mathbf{x})$.

The decoder network consists of a number of recurrent layers. Denoting by $\mathbf{h}_{u-1}^{\text{att}}$ the state of the lowest layer of the decoder after predicting the previous labels, $y_1, \cdots, y_{u-1}$, the model computes attention weights $\alpha_u = (\alpha_{1,u}, \cdots, \alpha_{T,u})$ for each frame in the encoder output, $\mathbf{h}^{\text{enc}}$, in order to compute a

single context vector, $\mathbf{c}_u$:

$$\beta_{t,u} = \langle \phi(\mathbf{h}_t^{\text{enc}}), \psi(\mathbf{h}_{u-1}^{\text{att}}) \rangle \quad (4)$$

$$\alpha_{t,u} = \frac{e^{\beta_{t,u}}}{\sum_{i=1}^{T} e^{\beta_{i,u}}} \quad (5)$$

$$\mathbf{c}_u = \sum_t \alpha_{t,u} \mathbf{h}_t^{\text{enc}} \quad (6)$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ represents the inner-product between vectors $\mathbf{x}$ and $\mathbf{y}$, and $\phi(\cdot)$ and $\psi(\cdot)$ are linear embeddings learned jointly with the other model parameters. The context vector is then input to the decoder network, along with the previously predicted label, y_{u-1}, to generate logits $\mathbf{h}_u^{\text{dec}}$ from the final layer in the decoder. Finally, these logits are input into a softmax layer, which outputs a probability distribution over the next label, conditioned on all previous predictions: $P(\mathbf{y}_u | y_{u-1}, \cdots, y_0, \mathbf{x})$. The label inventory is augmented with two special symbols: $\langle \text{sos} \rangle$, which is input to the decoder at the first time-step y_0, and $\langle \text{eos} \rangle$, which indicates the end of a sentence. During inference, the next label prediction process terminates when the $\langle \text{eos} \rangle$ label is predicted.

The model is trained to optimize cross-entropy on the training data [6], and is illustrated in Figure 1 (c.).[2]

2.4. RNN Transducer with Attention

Finally, we consider a modification to the RNN transducer that replaces the prediction network in a standard RNN transducer with an attention-based decoder network used in the LAS model, as depicted in Figure 1 (d.). Unlike the RNN transducer, where the prediction network only has access to the sequence of previous predictions, the RNN transducer with attention allows the decoder network to include acoustic information. Since the outputs $\mathbf{h}_u^{\text{dec}}$ computed from the decoder network depend on the entire encoder representation $\mathbf{h}^{\text{enc}}$, and are not dependent on a particular choice of segmentation (i.e., $\mathcal{B}(\mathbf{y}, \mathbf{x})$), the forward-backward algorithm [3] used to compute the RNN transducer loss can be used exactly as before; inference in this model is performed using the same frame-synchronous decoding algorithm as in the standard RNN transducer model [3]. Thus, we only include a $\langle \text{sos} \rangle$ label, and exclude the $\langle \text{eos} \rangle$ label, since the decoding process terminates when all encoder frames have been processed.

3. Experimental Details

The various sequence-to-sequence modeling approaches described in Section 2 are evaluated on an LVCSR task. Our models are trained on a set of $\sim$15M hand-transcribed anonymized utterances extracted from Google voice-search traffic, which corresponds to $\sim$12,500 hours of training data. In order to improve system robustness to noise and reverberation, multi-condition training (MTR) data are generated: training utterances are artificially distorted using a room simulator, by adding in noise samples extracted from YouTube videos and environmental recordings of daily events; the overall SNR is between 0dB and 30dB, with an average of 12dB.

[2] Unlike the RNN transducer, which can be used for streaming recognition if a unidirectional encoder is used, attention-based models cannot be deployed in applications where streaming recognition is required, since they must examine the entire input sequence before they can output any labels. We note, however, that recent work on "online" attention [16] is a step towards enabling streaming attention-based models.

Results are reported on three types of test sets, which consist of hand-transcribed anonymized utterances extracted from Google traffic: a set of $\sim$13K utterances ($\sim$124K words) from the domain of open-ended dictation (denoted, dict); a set of $\sim$12.9K utterances ($\sim$63K words) from the domain of voice-search queries (denoted, vs); and a set of $\sim$12.6K utterances ($\sim$75K words) corresponding to instances of times, dates, etc., where the expected transcript corresponds to the *written domain* (denoted, numeric) (e.g., the *spoken domain* utterance "twelve fifty one p m" must be recognized as "12:51 p.m."). Noisy versions of the *dictation* and *voice-search* sets are created by artificially adding noise drawn from the same distribution as in training (denoted, noisy-dict, and noisy-vs, respectively).

The input acoustic signal is represented with 80-dimensional log-mel features, computed with a 25ms window, and a 10ms frame-shift. Following previous work [17], we stack three consecutive frames and present only every third stacked frame as input to the encoder. The same acoustic frontend is used for all experiments described in this work.

All sequence-to-sequence models in this work are trained to directly output grapheme targets; the grapheme inventory includes the 26 lower-case letters a–z, the numerals 0–9, a label representing 'space', and punctuation symbols (e.g., the apostrophe symbol ('), hyphen (-), etc.).

A CTC model is trained to predict graphemes as output targets, as described in Section 2.1. The encoder in this model consists of 5 layers of 700 bidirectional [18] long short-term memory (BLSTM) cells [19] (i.e., 350 cells in the forward and reverse directions in each layer). When trained to convergence, the weights from this encoder are used to initialize the encoders in all other sequence-to-sequence models, since this was found to significantly speed up convergence; thus, all encoders used in the sequence-to-sequence models are identical in size and configuration, although the encoder parameters differ after the models have been trained.

The prediction network used in the RNN transducer models, with and without attention, consists of a single layer of 700 gated recurrent unit (GRU) cells [20], and the joint network consists of a single feed-forward layer of 700 units with a $\tanh$ activation function as described in Section 2.2. The decoder networks in our attention-based models also consist of GRU cells; we compare using either one or two layers of 700 cells in the decoder.

As our baselines, we train state-of-the-art unidirectional and bidirectional CTC models which predict 8,192 CD phonemes. The bidirectional model uses an encoder with the same size and configuration as our other models (i.e., 5 layers of 700 BLSTM cells), and the unidirectional model uses 5 layers of 700 LSTM cells. The baselines are first trained to optimize the CTC-criterion, followed by discriminative sequence training to optimize the state-level minimum Bayes risk (sMBR) criterion [21]. These models are decoded using a pruned, first-pass, 5-gram language model which is subsequently rescored with a large 5-gram language model. These systems use a vocabulary which consists of millions of words, as well as a separate expert-curated pronunciation model.

In contrast, all sequence-to-sequence models examined in this work are directly decoded to extract an output grapheme sequence, *without using a separate pronunciation model or an external language model*. The RNN transducer and attention-based models are decoded using a beam-search algorithm [6, 3], where at most 15 highest scoring candidates are retained at every step during decoding; however, performance is not found to be sensitive to the choice of beam size. For RNN transducer

Table 1: *WERs (%) on various test sets for the models compared in this work. The attention-based model with two decoder layers is the single best sequence-to-sequence model.*

Model	Clean		Noisy		numeric
	dict	vs	dict	vs	
Baseline Uni. CDP	6.4	9.9	8.7	14.6	11.4
Baseline BiDi. CDP	5.4	8.6	6.9	-	11.4
End-to-end systems					
CTC-grapheme[3]	39.4	53.4	-	-	-
RNN Transducer	6.6	12.8	8.5	22.0	9.9
RNN Trans. with att.	6.5	12.5	8.4	21.5	9.7
Att. 1-layer dec.	6.6	11.7	8.7	20.6	9.0
Att. 2-layer dec.	**6.3**	**11.2**	**8.1**	**19.7**	**8.7**

models, candidates are pruned after processing the t-th acoustic frame (*frame-synchronous decoding*), whereas for the attention-based models, candidates are pruned after producing the u-th output symbol from the decoder network (*label-synchronous decoding*). For the attention-based models, in order to prevent the model from outputting very short utterances, the next label prediction process is only allowed to terminate if the model outputs an $\langle \text{eos} \rangle$ label whose probability is above a threshold.

All models are trained using asynchronous stochastic gradient descent [22] with learning rate decay, and are implemented in TensorFlow [23].

4. Results

Our results are presented in Table 1. As expected, the CTC grapheme-based system, which is decoded without a language model, performs significantly worse than the baseline systems. The RNN transducer, which combines the acoustic modeling component of the grapheme CTC system with a grapheme-based LM, significantly improves performance across all test sets. Similarly, if we compare the performance of the RNN transducer model against the RNN transducer augmented with attention, we observe that the model with attention obtains a small but consistent improvement across all test sets.

Comparing the performance of the two attention-based systems, we note that increasing the number of layers in the decoder network improves performance between 3.3%–7.0% across the various test sets, suggesting that depth in both the encoder and the decoder networks is beneficial for attention-based models.

A comparison of the various sequence-to-sequence models against each other also reveals some interesting conclusions. First, we note that the performance of all sequence-to-sequence systems (other than the grapheme CTC system) on both clean and noisy dictation test sets is comparable with the *strong* unidirectional baseline system. It is also interesting to note that the RNN transducer systems as well as the attention-based systems significantly outperform the baseline on the numeric test set by a large margin. The "numeric" test set is particularly challenging since the mapping from the *spoken* to the *written* form often depends on context: e.g., "twelve fifty two p m" must be normalized to "12:52 p.m.", but "twelve fifty two main street" should be normalized to "1252 main street"). In particular, we note that switching from a unidirectional to a bidirectional acoustic model in the baseline did not improve performance on the numeric test set, although it resulted in a 13–20% improvement across the other test sets. We hypothesize that the ability to examine the input acoustics in addition to the previous sequence of predicted tokens is particularly helpful on this test set. This hypothesis is further supported by the fact that both attention-based systems significantly outperform the RNN transducer on this test set. However, examining the performance of these models on the voice-search test sets reveals that these systems are 13–35% worse than the baseline in this case (although we note that the gap is significantly less than has been reported in previous work, when decoding without an LM [6, 11]). We briefly analyze the nature of these errors in Section 4.1.

When all of the sequence-to-sequence modeling approaches are compared head-to-head, and trained with identical data and initialization, we find that the attention-based model with two decoder layers is the single best system, achieving significant improvements over the other sequence-to-sequence models on the voice-search and numeric test sets.

4.1. Error Analysis

Sequence-to-sequence models show the promise of being capable of end-to-end speech recognition. However, as indicated by the results presented in Table 1, these models are unable to match the performance of traditional state-of-the-art systems on harder tasks such as voice-search. A brief analysis of the errors made by the sequence-to-sequence systems on the voice-search test sets suggest that many of the errors are due to the absence of large language models. The language models used in typical ASR systems are trained on vast amounts of text data, which are more easily obtained than transcribed acoustic data. A particular example demonstrates this: the query `who wrote tortoise and the hare` is recognized by the sequence-to-sequence models as `who wrote tortoise and the hair`. Although `hair` is acoustically confusable with `hare`, the language model in a traditional ASR system enables the model to successfully pick `hare` in this context.

We also find that the sequence-to-sequence models struggle with recognition of proper nouns, such as the names of places and entities, which are common in voice-search utterances. For example, the models recognize the query `keuka lake` (a lake in the U.S. state of New York; pronounced /ˈkju:kə/) as `cuca lake`, which is incorrect. These examples suggests that the incorporation of a stronger language model is vital in order for such sequence-to-sequence models to achieve state-of-the-art performance. We hope to explore this in future work.

5. Conclusions

In this work, we compared a number of sequence-to-sequence modeling approaches on an LVCSR task. In experimental evaluations, we find that the RNN transducer, attention-based models and a novel RNN transducer augmented with attention are comparable in performance to a strong state-of-the-art baseline on a dictation test set, even when evaluated without the use of an external pronunciation or language model. On voice-search test sets, however, we find that these models are still outperformed by the baseline by a large margin. Finally, we find that the sequence-to-sequence models, particularly the attention-based models, significantly outperform the baseline on the test set of numeric entities, which require the model to map utterances from the *spoken* to the *written domain*.

6. Acknowledgements

The authors would like to thank Michiel Bacchiani, Françoise Beaufays, Yanzhang He, Haşim Sak, Trevor Strohman, Chris Thornton, and Anshuman Tripathi for helpful comments and suggestions on this work.

[3]The grapheme CTC model is directly decoded to obtain the one-best grapheme sequence without using a separate language model, which explains why performance is much worse than the other systems.

7. References

[1] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *Proc. Interspeech*, 2014.

[2] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in *Proc. ICASSP*, 2015.

[3] A. Graves, "Sequence Transduction with Recurrent Neural Networks," *CoRR*, vol. abs/1211.3711, 2012.

[4] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Neural Networks," in *Proc. ICASSP*, 2013.

[5] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *Proc. ICML*, 2014.

[6] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, Attend and Spell," *CoRR*, vol. abs/1508.01211, 2015.

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

[8] 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.

[9] W. Chan, Y. Zhang, Q. Le, and N. Jaitly, "Latent Sequence Decompositions," *CoRR*, vol. abs/1610.03035, 2016.

[10] Y. Zhang, W. Chan, and N. Jaitly, "Very Deep Convolutional Networks for End-To-End Speech Recognition," *CoRR*, vol. abs/1610.03022, 2016.

[11] L. Lu, X. Zhang, and S. Renals, "On Training the Recurrent Neural Network Encoder-Decoder for Large Vocabulary End-to-End Speech Recognition," in *Proc. ICASSP*, 2016.

[12] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in All-Neural Speech Recognition," in *Proc. ICASSP*, 2017.

[13] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labeling Unsegmented Seuqnece Data with Recurrent Neural Networks," in *Proc. ICML*, 2006.

[14] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

[15] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," in *Proc. ICASSP*, 1992.

[16] 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.

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

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

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

[20] K. Cho, B. van Merriënboer, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proc. EMNLP*, 2014.

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

[22] 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, pp. 1223–1231.

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