

DIALOG-CONTEXT AWARE END-TO-END SPEECH RECOGNITION

Suyoun Kim¹ and Florian Metze²

¹Electrical & Computer Engineering

²Language Technologies Institute, School of Computer Science

Carnegie Mellon University

{suyoung1|fmetze}@andrew.cmu.edu

ABSTRACT

Existing speech recognition systems are typically built at the sentence level, although it is known that dialog context, e.g. higher-level knowledge that spans across sentences or speakers, can help the processing of long conversations. The recent progress in end-to-end speech recognition systems promises to integrate all available information (e.g. acoustic, language resources) into a single model, which is then jointly optimized. It seems natural that such dialog context information should thus also be integrated into the end-to-end models to improve recognition accuracy further. In this work, we present a dialog-context aware speech recognition model, which explicitly uses context information beyond sentence-level information, in an end-to-end fashion. Our dialog-context model captures a history of sentence-level contexts, so that the whole system can be trained with dialog-context information in an end-to-end manner. We evaluate our proposed approach on the Switchboard conversational speech corpus, and show that our system outperforms a comparable sentence-level end-to-end speech recognition system.

Index Terms— end-to-end speech recognition

1. INTRODUCTION

As voice-driven interfaces to devices become mainstream, spoken dialog systems that can recognize and understand long dialogs are becoming increasingly important. Dialog context, dynamic contextual flow across multiple sentences, provides important information that can improve speech recognition. To build speech recognition models, the long dialogs typically split into short utterance-level audio clips to make training the speech recognition models computationally feasible. Thus, such speech recognition models built on sentence-level speech data may lose important dialog-context information.

In language model trained only on text data, several recent studies have attempted to use document-level or dialog-level context information to improve language model performance. Recurrent neural network (RNN) based language models [1] have shown success in outperforming conventional n-gram

based models due to their ability to capture long-term information. Based on the success of RNN based language models, recent research has developed a variety of ways to incorporate document-level or dialog-level context information [2–5]. Mikolov et al. proposed a context-dependent RNN language model [2] using a context vector which is produced by applying latent Dirichlet allocation [6] on the preceding text. Wang et al. proposed using a bag-of-words to represent the context vector [3], and Ji et al. proposed using the last RNN hidden states from the previous sentence to represent the context vector [4]. Liu et al. proposed using an external RNN to model dialog context between speakers [5]. All of these models have been developed and optimized on text data, and therefore must still be combined with conventional acoustic models, which are optimized separately without any context information beyond sentence-level. The recent study [7] attempted to integrated such dialog session-aware language model with acoustic models, however, it requires disjoint training procedure.

There have been no studies of speech recognition models that incorporate dialog-context information in end-to-end training approach on both speech and text data. The recently proposed *end-to-end* speech recognition models, a neural network is trained to convert a sequence of acoustic feature vectors into a sequence of graphemes (characters) rather than senones. Unlike sequences of senone predictions, which need to be decoded using a pronunciation lexicon and a language model, the grapheme sequences can be directly converted to word sequences without any additional models. The end-to-end models proposed in the literature to use a Connectionist Temporal Classification framework [8–12], an attention-based encoder-decoder framework [13–16], or both [17, 18].

Our goal is to build speech recognition model that explicitly use a dialog-level context information beyond sentence-level information especially in an end-to-end manner so that the whole system can be trained with the long context information. In this paper, we present a dialog-context aware end-to-end speech recognition model that can capture a history of sentence-level contexts within an end-to-end speech recognition models. We also present a method to *serialize* datasets for

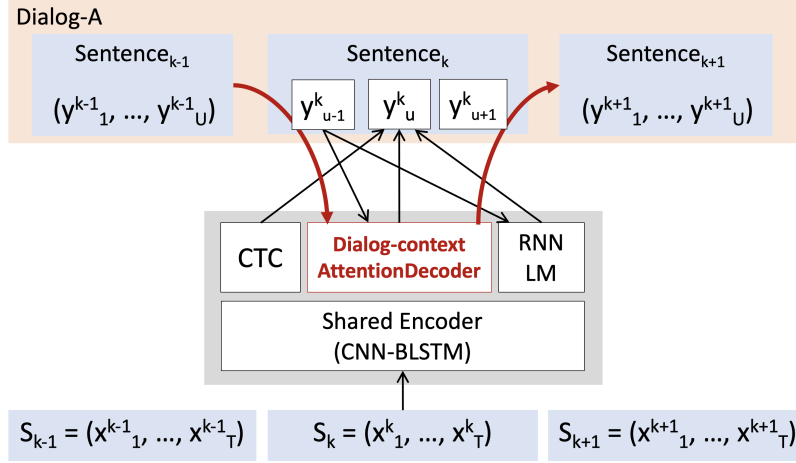


Fig. 1: The architecture of our dialog-context end-to-end speech recognition model. The red curved line represents the context information flow within the same dialog.

training and decoding based on their onset times to learn dialog flow. We evaluate our proposed approach on the Switchboard conversational speech corpus [19, 20], and show that our model outperforms the sentence-level end-to-end speech recognition model.

2. DIALOG-CONTEXT END-TO-END ASR

2.1. End-to-end models

We perform end-to-end speech recognition using a joint CTC/Attention-based approach with graphemes as the output symbols [17, 18]. The key advantage of the joint CTC/Attention framework is that it can address the weaknesses of the two main end-to-end models, Connectionist Temporal Classification (CTC) [8] and attention-based encoder-decoder (Attention) [21], by combining the strengths of the two. With CTC, the neural network is trained according to a maximum-likelihood training criterion computed over all possible segmentations of the utterance's sequence of feature vectors to its sequence of labels while preserving left-right order between input and output. With attention-based encoder-decoder models, the decoder network can learn the language model jointly without relying on the conditional independent assumption.

Given a sequence of acoustic feature vectors, \mathbf{x} , and the corresponding graphemic label sequence, \mathbf{y} , the joint CTC/Attention objective is represented as follows by combining two objectives with a tunable parameter $\lambda : 0 \leq \lambda \leq 1$:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda) \mathcal{L}_{\text{Attention}}. \quad (1)$$

Each loss to be minimized is defined as the negative log likelihood of the ground truth character sequence \mathbf{y}^* , is computed from:

$$\mathcal{L}_{\text{CTC}} \triangleq -\ln \sum_{\pi \in \Phi(\mathbf{y})} p(\pi|\mathbf{x}) \quad (2)$$

$$\mathcal{L}_{\text{Attention}} \triangleq -\sum_u \ln p(y_u^*|\mathbf{x}, y_{1:u-1}^*) \quad (3)$$

where π is the label sequence allowing the presence of the *blank* symbol, Φ is the set of all possible π given u -length \mathbf{y} , and $y_{1:u-1}^*$ is all the previous labels.

2.2. Dialog-context models

In this section, we present our *DialogAttentionDecoder* sub-network. We extend the *AttentionDecoder* which is a subnetwork of standard end-to-end models in order to employ context information. The core modeling idea of this work is to integrate the context information from the previous sentence into that of the current sentence.

Let we have a dataset consists of N-number of dialogs, $D = \{d_1, \dots, d_N\}$ and each dialog $d_i = (s_1, \dots, s_K)$ has K utterances. k -th utterance s_k is represented as a sequence of U -length output characters (\mathbf{y}) and T -length input acoustic features (\mathbf{x}). Given the high-level representation (\mathbf{h}) of input acoustic features (\mathbf{x}) generated from *Encoder*, both the standard *AttentionDecoder* and our proposed *DialogAttentionDecoder* generates the probability distribution over characters (y_u), conditioned on (\mathbf{h}), and all the characters seen previously ($y_{1:u-1}$). Our proposed *DialogAttentionDecoder* additionally conditioning on dialog context vector (c_k), which represents the information of the preceding utterance in the same dialog as:

$$\mathbf{h} = \text{Encoder}(\mathbf{x}) \quad (4)$$

$$y_u \sim \begin{cases} \text{standard decoder network:} \\ \quad \text{AttentionDecoder}(\mathbf{h}, y_{1:u-1}) \\ \text{proposed decoder network:} \\ \quad \text{DialogAttentionDecoder}(\mathbf{h}, y_{1:u-1}, c_k) \end{cases} \quad (5)$$

$$(6)$$

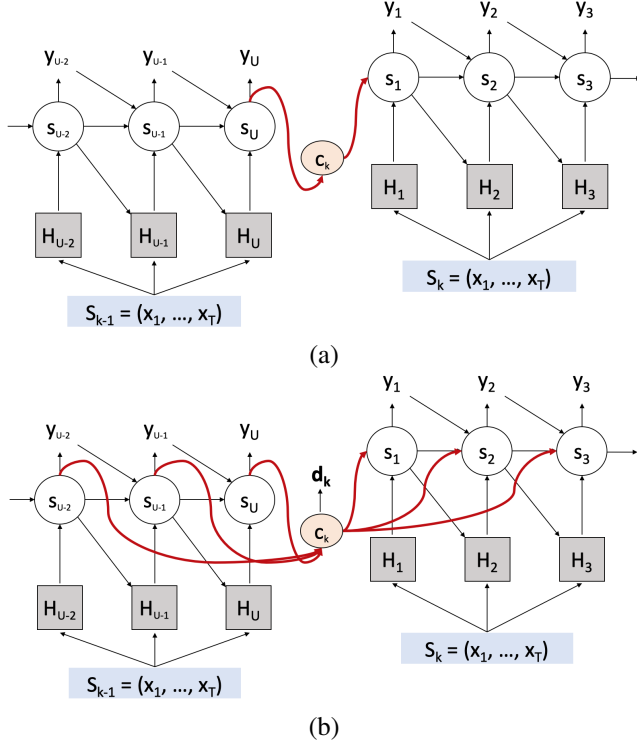


Fig. 2: Two different types of our dialog-context Attention-Decoder network. At each output step (u), the character distribution (y_u), is generated conditioning on 1) the dialog context vector (c_k) from the previous sentence, in addition to 2) the attended inputs (H_u) from Encoder network, 3) the decoder state (s_{u-1}), and 4) the history of character y_{u-1} . The red curved line represents the context information flow.

We represent the context vector, c_k , in two different ways as illustrated in Figure 2. In method (a), the last decoder state of the previous sentence represents the context vector, c_k . The context vector c_k is propagated to the initial decoder state of current sentence. In method (b), every output information is integrated with additional attention mechanism and represents the context vector, c_k , then is propagated to every decoder state for each output time step of current sentence. Motivated by prior work [4], we produce the context vector for the (a)-type of our decoder from the final hidden representation of the previous sentence s_{k-1} :

$$c_k = s_U^{s_{k-1}}. \quad (7)$$

For the (b)-type of our decoder, we embed a subnetwork (*DialogContextGenerator*) with an additional attention mechanism to incorporate every previous context into a single context vector c_k . We use the character distributions of previous sentence ($y_{*1} \cdots y_{*U}$) as input to generate (c_k). This subnetwork is optimized towards minimizing the dialog classifica-

tion error:

$$d_k \sim \text{DialogContextGenerator}(\mathbf{y}^*) \quad (8)$$

$$\mathcal{L}_{\text{Dialog}} \triangleq -\ln p(d_k^* | \mathbf{y}^*) \quad (9)$$

The additional loss $\mathcal{L}_{\text{Dialog}}$ is added to the \mathcal{L} in Eq. (1) so that the whole model is optimized jointly. Once the context vector of previous sentence is generated by either proposed method (a) or (b), it is stored for the next sentence prediction.

The following equation represents how to update the hidden states of *DialogAttentionDecoder* with high-level input features (\mathbf{h}_u) generated from *Encoder*, 2) the previous character (y_{u-1}), and 3) the context vector (c_k) from previous sentence:

$$\hat{s}_{u-1} = f(s_{u-1}, c_k) \quad (10)$$

$$s_u = \text{RNN}(\hat{s}_{u-1}, \mathbf{h}_u, y_{u-1}) \quad (11)$$

$$y_u \sim \text{softmax}(s_u) \quad (12)$$

where $f(\cdot)$ is a function that combines the two inputs, s_{u-1} , and c_k :

$$f(s_{u-1}, c_k) = \tanh(Ws_{u-1} + Vc_k + b) \quad (13)$$

where W, V, b are trainable parameters. In this work, we use \tanh for the non-linear activation function.

2.3. Dataset serialization

In order to learn and use the dialog context during training and decoding, we serialized the sentences based on their onset times and their dialogs rather than random shuffling of data. We first grouped the sentences based on their dialogs, then ordered the sentences according to their onset times. Instead of generating a minibatch set in a typical way that randomly chooses sentences, we created a minibatch to contain the sentences from each one of different dialogs. For example, a size-30 minibatch had 30 sentences from d_1 d_{30} . We did not shuffle the minibatch sets for training or decoding, so that the context information generated from the previous minibatch that contains preceding sentences can propagate to the next minibatch.

Since the number of sentences varies across the dialogs, some dialogs may not have enough sentences to construct the minibatches. In this case, we included dummy input/output data for the dialogs that have fewer sentences to maintain the minibatch size and not to lose context information of the other dialogs that have more sentences. We then masked out the loss from the dummy data for the objective function. Once every sentence in d_1 d_{30} was processed, then the sentences from the other dialogs d_{31} d_{60} would be processed.

Figure 3 illustrates the example minibatches that are serialized according to their onset times and their dialogs. The size of minibatch is 3 and the sentences are from dialogs, d_A , d_B , and d_C . In second and third minibatches, the dummy input/output data is included in the position for the d_B and d_C , which have fewer sentences.

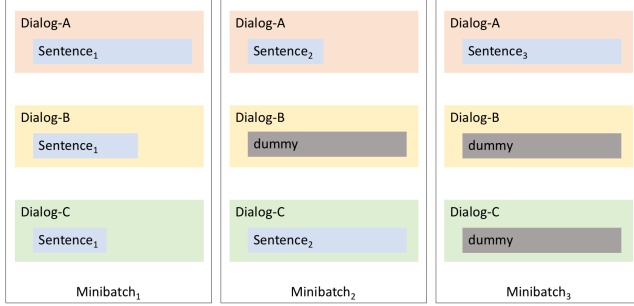


Fig. 3: A method to make the minibatch set for training and evaluating our models. The example minibatches are serialized according to their onset times and their dialogs. The size of example minibatch is 3 and the sentences are from three dialogs, d_A , d_B , and d_C . In second and third minibatches, the dummy input/output data is included in the position for the d_B and d_C , which have fewer sentences.

Table 1: Experimental dataset description. We used the Switchboard dataset which has a 300 hours training set. Note that any pronunciation lexicon or external text data was not used.

	Train_nodup	Train_dev	CH	SWB
# dialog	2,402	34	20	20
# sentences	192,656	4,000	2,627	1,831

3. EXPERIMENTS

3.1. Experimental corpora

We investigated the performance of the proposed dialog-context aware model on the Switchboard LDC corpus (97S62) which has a 300 hours training set. Note that we did not use the Fisher dataset. We split the Switchboard data into two groups, then used 285 hours of data (192 thousand sentences) for model training and 5 hours of data (4 thousand sentences) for hyper-parameter tuning. Evaluation was carried out on the HUB5 Eval 2000LDC corpora (LDC2002S09, LDC2002T43), which have 3.8 hours of data (4.4 thousand sentences), and we show separate results for the Callhome English (CH) and Switchboard (SWB) evaluation sets. We denote train_nodup, train_dev, SWB, and CH as our training, development, and two evaluation datasets for CH and SWB, respectively. Table 1 shows the number of dialogs per each dataset.

We sampled all audio data at 16kHz, and extracted 80-dimensional log-mel filterbank coefficients with 3-dimensional pitch features, from 25 ms frames with a 10ms frame shift. We used 83-dimensional feature vectors to input to the network in total. We used 49 distinct labels: 26 characters, 10 digits, apostrophe, period, dash, underscore, slash, am-

persand, noise, vocalized-noise, laughter, unknown, space, start-of-speech/end-of-speech, and blank tokens.

Note that no pronunciation lexicon was used in any of the experiments.

3.2. Training and decoding

We used joint CTC/Attention end-to-end speech recognition architecture [17, 18] with ESPnet toolkit [24]. We used a CNN-BLSTM encoder as suggested in [25, 26]. We followed the same six-layer CNN architecture as the prior study, except we used one input channel instead of three, since we did not use delta or delta delta features. Input speech features were downsampled to (1/4 x 1/4) along with the time-frequency axis. Then, the 4-layer BLSTM with 320 cells was followed by the CNN. We used a location-based attention mechanism [15], where 10 centered convolution filters of width 100 were used to extract the convolutional features.

The decoder network of both our proposed models and the baseline models was a one-layer LSTM with 300 cells. Our dialog-context aware models additionally requires one-layer with 300 hidden units for incorporating the context vector with decoder states, and attention network with 2402-dimensional output layer to generate the context vector. We also built a character-level RNNLM (Char-RNNLM) on the the same Switchboard text dataset. The Char-RNNLM network was a two-layer LSTM with 650 cells, trained separately only on the training transcription. This network was used only for decoding. Note that we did not use any extra text data other than the training transcription.

The AdaDelta algorithm [27] with gradient clipping [28] was used for optimization. We used $\lambda = 0.5$ for joint CTC/Attention training. We bootstrap the training our proposed dialog-context aware end-to-end models from the baseline end-to-end models. For decoding of the models, we used joint decoder which combines the output label scores from the AttentionDecoder, CTC, and Char-RNNLM by using shallow fusion [26]:

$$\begin{aligned} \mathbf{y}^* = \operatorname{argmax} \{ & \log p_{att}(\mathbf{y}|\mathbf{x}) \\ & + \alpha \log p_{att}(\mathbf{y}|\mathbf{x}) \\ & + \beta \log p_{rnnlm}(\mathbf{y}) \} \end{aligned} \quad (14)$$

The scaling factor of CTC, and RNNLM scores were $\alpha = 0.3$, and $\beta = 0.3$, respectively. We used a beam search algorithm similar to [29] with the beam size 20 to reduce the computation cost. We adjusted the score by adding a length penalty, since the model has a small bias for shorter utterances. The final score $s(\mathbf{y}|\mathbf{x})$ is normalized with a length penalty 0.1.

The models were implemented by using the Chainer deep learning library [30], and ESPnet toolkit [17, 18, 24].

Table 2: Comparison of hypothesis between baseline and our proposed model. Two examples of two consecutive sentences are manually chosen from evaluation dataset. They show that our model correctly predicted the word bolded in current sentence, while the baseline incorrectly predicted it. The preceding sentence includes the context information related to the word, and our model seems to be benefit this information.

model	previous sentence	current sentence
REF	<i>yes it is so hot in the building have you ever been in it</i>	<i>it is like a sauna</i>
Baseline	<i>yeah if when he said that is like just so hot in the belly have ever been but</i>	<i>it is like i saw</i>
Ours	<i>yeah if when he is in this like it is so hot in the belief I have never been but</i>	<i>it is like a sauna</i>
REF	<i>if we go we like check into a to a to a hotel but</i>	<i>i know but it is much more comfortable</i>
Baseline	<i>if we go we like check until the law that you know to do</i>	<i>i know that is much more comes of one</i>
Ours	<i>if we go we like check into a law if it does a job hotel</i>	<i>i know that is much more comfortable</i>

Table 3: Word Error Rate (WER) on the Switchboard dataset. None of our experiments used any lexicon information or external text data other than the training transcription. The models were trained on 300 hours of Switchboard data only.

Train (~ 300hrs) Models	CH WER %	SWB WER %
<i>sentence-level end2end</i>		
Seq2Seq A2C [22]	40.6	28.1
CTC A2C [10]	31.8	20.0
CTC A2C [12]	32.1	19.8
CTC A2W(Phone/external-LM init.) [23]	23.6	14.6
<i>sentence-level end2end</i>		
Our baseline (CTC/Seq2Seq)	34.4	19.0
<i>dialog-context aware end2end</i>		
Our proposed model(a)	34.1	18.2
Our proposed model(b)	33.2	18.6

4. RESULTS

We evaluated both the end-to-end speech recognition model which was built on sentence-level data (*sentence-level end2end*) and our proposed dialog-context aware end-to-end speech recognition model which leveraged dialog-context information within and beyond the sentence (*dialog-context aware end2end*).

Table 3 shows the WER of our baseline, proposed models, and several other published results those were only trained on 300 hours Switchboard training data. Note that CTC A2W(Phone/external-LM init.) [23] was initialized from Phone CTC model and use external word embeddings. As shown in Table 3, we obtained a performance gain over our baseline *sentence-level end2end* by using the dialog-context information. Our proposed model (a) performed best on SWB evaluation set showing 4.2% relative improvement over our baseline. Our proposed model (b) performed best on CH evaluation set showing 3.4% relative improvement over our baseline.

Table 4 shows the insertion, deletion, and substitution

Table 4: Substitution rate (Sub), Deletion rate (Del), and Insertion rate (Ins) for the baseline and our proposed model.

Model	Test	Sub %	Del %	Ins %	WER %
Baseline	CH	23.9	5.8	4.7	34.4
Proposed model(a)	CH	23.9	5.9	4.3	34.1
Proposed model(b)	CH	22.8	6.3	4.1	33.2
Baseline	SWB	13.1	3.4	2.5	19.0
Proposed model(a)	SWB	12.5	3.4	2.2	18.2
Proposed model(b)	SWB	12.6	3.6	2.4	18.6

rates. We observed that the largest factor of WER improvement was from the substitution rates rather than deletion or insertion. Table 2 shows three example utterances with each previous sentence to show that our context information beyond the sentence-level improves word accuracy.

5. CONCLUSION

We proposed a dialog-context aware end-to-end speech recognition model which explicitly uses context information beyond sentence-level information. A key aspect of this model is that the whole system can be trained with dialog-level context information in an end-to-end manner. Our model was shown to outperform previous end-to-end models trained on sentence-level data. Moving forward, we plan to explore additional methods to represent the context information and evaluate performance improvements that can be obtained by addressing overfitting and data sparsity using larger conversational datasets, e.g., 2,000 hours of Fisher dataset.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. This research was supported by a fellowship from the Center for Machine Learning and Health (CMLH) at Carnegie Mellon University.

7. REFERENCES

- [1] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur, “Recurrent neural network based language model,” in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.
- [2] Tomas Mikolov and Geoffrey Zweig, “Context dependent recurrent neural network language model,” *SLT*, vol. 12, pp. 234–239, 2012.
- [3] Tian Wang and Kyunghyun Cho, “Larger-context language modelling,” *arXiv preprint arXiv:1511.03729*, 2015.
- [4] Yangfeng Ji, Trevor Cohn, Lingpeng Kong, Chris Dyer, and Jacob Eisenstein, “Document context language models,” *arXiv preprint arXiv:1511.03962*, 2015.
- [5] Bing Liu and Ian Lane, “Dialog context language modeling with recurrent neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5715–5719.
- [6] David M Blei, Andrew Y Ng, and Michael I Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [7] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig, “The microsoft 2016 conversational speech recognition system,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5255–5259.
- [8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.
- [9] Alex Graves and Navdeep Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.
- [10] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [11] Yajie Miao, Mohammad Gowayyed, and Florian Metze, “EESN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.
- [12] Geoffrey Zweig, Chengzhu Yu, Jasha Droppo, and Andreas Stolcke, “Advances in all-neural speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4805–4809.
- [13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [14] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “End-to-end continuous speech recognition using attention-based recurrent NN: First results,” *arXiv preprint arXiv:1412.1602*, 2014.
- [15] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, “Attention-based models for speech recognition,” in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.
- [16] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals, “Listen, attend and spell,” *arXiv preprint arXiv:1508.01211*, 2015.
- [17] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, “Joint ctc-attention based end-to-end speech recognition using multi-task learning,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4835–4839.
- [18] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [19] John J Godfrey, Edward C Holliman, and Jane McDaniel, “Switchboard: Telephone speech corpus for research and development,” *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, 1992.
- [20] John J Godfrey, Edward C Holliman, and Jane McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*. IEEE, 1992, vol. 1, pp. 517–520.
- [21] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, “End-to-end attention-based large vocabulary speech recognition,” *arXiv preprint arXiv:1508.04395*, 2015.
- [22] Thomas Zenkel, Ramon Sanabria, Florian Metze, Jan Niehues, Matthias Sperber, Sebastian Stüker, and Alex Waibel, “Comparison of decoding strategies for ctc

- acoustic models,” *arXiv preprint arXiv:1708.04469*, 2017.
- [23] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny, “Building competitive direct acoustics-to-word models for english conversational speech recognition,” *arXiv preprint arXiv:1712.03133*, 2017.
 - [24] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., “Espnet: End-to-end speech processing toolkit,” *arXiv preprint arXiv:1804.00015*, 2018.
 - [25] Yu Zhang, William Chan, and Navdeep Jaitly, “Very deep convolutional networks for end-to-end speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4845–4849.
 - [26] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, “Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm,” *arXiv preprint arXiv:1706.02737*, 2017.
 - [27] Matthew D Zeiler, “Adadelata: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.
 - [28] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, “On the difficulty of training recurrent neural networks,” in *International Conference on Machine Learning*, 2013, pp. 1310–1318.
 - [29] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
 - [30] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton, “Chainer: a next-generation open source framework for deep learning,” in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.