Improved training of end-to-end attention models for speech recognition

Albert Zeyer[1,2,3], *Kazuki Irie*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52062 Aachen, Germany,
[2]AppTek, USA, http://www.apptek.com/,
[3]NNAISENSE, Switzerland, https://nnaisense.com/
{zeyer, irie, schlueter, ney}@cs.rwth-aachen.de

Abstract

Sequence-to-sequence attention-based models on subword units allow simple open-vocabulary end-to-end speech recognition. In this work, we show that such models can achieve competitive results on the Switchboard 300h and LibriSpeech 1000h tasks. In particular, we report the state-of-the-art word error rates (WER) of 3.54% on the dev-clean and 3.82% on the test-clean evaluation subsets of LibriSpeech. We introduce a new pretraining scheme by starting with a high time reduction factor and lowering it during training, which is crucial both for convergence and final performance. In some experiments, we also use an auxiliary CTC loss function to help the convergence. In addition, we train long short-term memory (LSTM) language models on subword units. By shallow fusion, we report up to 27% relative improvements in WER over the attention baseline without a language model.

Index Terms: attention, end-to-end, speech recognition

1. Introduction

Conventional speech recognition systems [1] with neural network (NN) based acoustic models using the hybrid hidden Markov models (HMM) / NN approach [2, 3] usually operate on the phone level, given a phonetic pronunciation lexicon (from phones to words). They require a pretraining scheme with HMM and Gaussian mixture models (GMM) as emission probabilities to bootstrap good alignments of the HMM states. Context-independent phones are used initially because context-dependent phones need a good clustering, which is usually created on good existing alignments (via a Classification And Regression Tree (CART) clustering [4]). This boot-strapping process is iterated a few times. Then a hybrid HMM / NN is trained with frame-wise cross entropy. Recognition with such a model requires a sophisticated beam search decoder. Handling out-of-vocabulary words is also not straightforward and increases the complexity. There was certain work to remove the GMM dependency in the pretraining [5], or to be able to train without an existing alignment [6–8], or to avoid the lexicon [9], which simplifies the pretraining procedure but still is not end-to-end.

An *end-to-end model* in speech recognition generally denotes a simple single model which can be trained from scratch, and usually directly operates on words, sub-words or characters/graphemes. This removes the need for a pronunciation lexicon and the whole explicit modeling of phones, and it greatly simplifies the decoding.

Connectionist temporal classification (CTC) [10] has been often used as an end-to-end model for speech recognition, often on characters/graphemes [11–16] or on sub-words [17] but also directly on words [18, 19].

The *encoder-decoder framework with attention* has become the standard approach for machine translation [20–22] and many other domains such as images [23]. Recent investigations have shown promising results by applying the same approach for speech recognition [24–28]. In this work, we also investigate techniques to improve recurrent encoder-attention-decoder based systems for speech recognition. We use long short-term memory (LSTM) neural networks [29] for the encoder and the decoder. Our model is similar to the architecture used in machine translation [30], except for encoder time reduction. This generality of the model and the simplicity is its strength. Although a valid argument against this model for speech recognition is that it is in fact too powerful because it does not require monotonicity in its implicit alignments. There are attempts to restrict the attention to become monotonic in various ways [31–38]. In this work, our models are without these modifications and extensions.

Recently, alternative models for end-to-end modeling were also suggested, such as inverted HMMs [39], the recurrent transducer [40–42], or the recurrent neural aligner [43]. In many ways, these can all be interpreted in the same encoder-decoder-attention framework, but these approaches often use some variant of hard latent monotonic attention instead of soft attention.

Our models operate on *subword units* which are created via *byte-pair encoding (BPE)* [44]. We introduce a *pretraining scheme* applied on the encoder, which grows the encoder in layer depth, as well as decreases the initial high encoder time reduction factor. To the best of our knowledge, we are the first to apply pretraining for encoder-attention-decoder models. We use RETURNN [30, 45] based on TensorFlow [46] for its computation. We have implemented our own flexible and efficient beam search decoder and efficient LSTM kernels in native CUDA. In addition, we train subword-level LSTM language models [47], which we integrate in the beam search by *shallow fusion* [48]. The source code is fully open[1], as well as all the setups of the experiments in this paper[2]. We report competitive results on the 300h-Switchboard and LibriSpeech [49]. In particular on LibriSpeech, our system achieves WERs of 3.54% on the dev-clean and 3.82% on the test-clean evaluation subsets, which are the best results obtained on this task to the best of our knowledge.

2. Pretraining

Compared to machine translation, the input sequences are much longer in speech recognition, relatively to the output sequence (e.g. with BPE 10K subword units, and audio feature frames every 10ms, more than 30 times longer on Switchboard on average). However, as the original input is continuous, some sort of down-scaling in the time dimension works, such as concatenation in the feature dimension of consecutive time-frames [7, 24, 42, 50]. We use max-pooling in the time-dimension which is simpler. The time reduction can be done directly on the features or al-

[1] https://github.com/rwth-i6/returnn
[2] https://github.com/rwth-i6/returnn-experiments/tree/master/2018-asr-attention

ternatively at multiple steps inside the encoder, e.g. after every encoder layer [24]. This is also what we do. This allows the encoder to better compress any necessary information.

We observed that a high time reduction factor makes the training much simpler. In fact, without careful tuning, usually the model will not converge without a high time reduction factor (16 or 32), as has also been observed in the literature [24]. However, we also observed that a low time reduction factor (e.g. 8) can perform better after all, when pretrained with a high time reduction factor.

Also, it has been shown that deep LSTM models can benefit from layer-wise pretraining, by starting with 1 or 2 layers and adding more and more layers [1]. We apply the same pretraining.

To improve the convergence further, we disable label smoothing during pretraining and only enable it after pretraining. Also, we disable dropout during the first few pretraining epochs in the encoder.

3. Model

We use a deep bidirectional LSTM encoder network, and LSTM decoder network. After every layer in the encoder, we optionally do max-pooling in the time dimension to reduce the encoder length. I.e. for the input sequence x_1^T, we end up with the encoder state

$$h_1^{T'} = \mathrm{LSTM}_{\#\mathrm{enc}} \circ \cdots \circ \text{max-pool}_1 \circ \mathrm{LSTM}_1(x_1^T),$$

where $T' = \mathrm{red} \cdot \mathrm{T}$ for the time reduction factor red, and $\#\mathrm{enc}$ is the number of encoder layers, with $\#\mathrm{enc} \geq 2$. We use the MLP attention [20, 21, 31, 32, 51]. Our model closely follows the machine translation model presented by Bahar et al. [51] and Bahdanau et al. [20] and we use a variant of attention weight / fertility feedback [52], which is inverse in our case, to use a multiplication instead of a division, for better numerical stability. More specifically, the attention energies $e_{i,t} \in \mathbb{R}$ for encoder time-step t and decoder step i are defined as

$$e_{i,t} = v^\top \tanh(W[s_i, h_t, \beta_{i,t}]),$$

where v is a trainable vector, W a trainable matrix, s_i the current decoder state, h_t the encoder state, and $\beta_{i,t}$ is the attention weight feedback, defined as

$$\beta_{i,t} = \sigma(v_\beta^\top h_t) \cdot \sum_{k=1}^{i-1} \alpha_{k,t},$$

where v_β is a trainable vector. Then the attention weights are defined as

$$\alpha_i = \mathrm{softmax}_t(e_i)$$

and the attention context vector is given as

$$c_i = \sum_t \alpha_{i,t} h_t.$$

The decoder state is recurrent function implemented as

$$s_i = \mathrm{LSTMCell}(s_{i-1}, y_{i-1}, c_{i-1})$$

and the final prediction probability for the output symbol y_i is given as

$$p(y_i | y_{i-1}, x_1^T) = \mathrm{softmax}(\mathrm{MLP}_{\mathrm{readout}}(s_i, y_{i-1}, c_i)).$$

In our case we use $\mathrm{MLP}_{\mathrm{readout}} = \mathrm{linear} \circ \mathrm{maxout} \circ \mathrm{linear}$.

4. Sub-word units

Characters/graphemes are probably the most generic and simple output units for generating texts but it has been shown that sub-word units can perform better [26] and they can be just as generic since the characters can be included in the set of subword units. Using words as output units is also possible but does not permit recognition of out-of-vocabulary words and it requires a large softmax output and thus is computationally expensive. An inhomogeneous length distribution as well as an imbalance in the label occurence can also make training harder.

In all the experiments, we use byte-pair encoding (BPE) [44] to create subword units, which are the output targets of the decoder. The beam search decoding will go over these BPE units, and then select the best hypothesis. Therefore, our system is open-vocabulary. At the end of decoding, the BPE units are merged into words in order to obtain the best hypothesis on word level. In addition, we add the special tokens from the transcriptions which denote noise, vocalized-noise and laughter in our BPE vocabulary set. Our recognizer can also potentially recognize these special events.

5. Language model combination

We also improve the recognition accuracy of our recognizer using external language models. We train LSTM language models [47] on the same BPE vocabulary set as the end-to-end model, using RETURNN with TensorFlow. For Switchboard, the training set of 27M words concatenating Switchboard and Fisher parts of transcriptions was used. For LibriSpeech, we use the 800M-word dataset officially available[3] for training language models. It can be noted that in the case of Switchboard, there is some overlap between the training data for language models and the transcription used to train the end-to-end model: 3M out of 27M words are used to train the end-to-end system. While for the LispriSpeech, 800M-word data is fully external to the end-to-end models. Our experiments show that this difference in amount of external data directly affects the performance improvements by the use of external language model. For both tasks, we use two LSTM layers with 2048 nodes. The input projection layer size is 256 and 512 respectively for Switchboard and LibriSpeech. We apply dropout at the input of all hidden layers with the rate of 0.2. The standard stochastic gradient descent with global gradient clipping is used for optimization to train all LSTM LMs.

We integrate the external language model in the beam search by shallow fusion [48]. The weight for the language model has been optimized by grid search on the development set WER. We found 0.23 and 0.36 to be optimal respectively for Switchboard and LibriSpeech (the weight on the attention model is 1).

For LibriSpeech, we also train Kneser-Ney smoothed n-gram count based language models [53] on the same BPE vocabulary set using SRILM toolkit [54]. The comparison of perplexities can be found in Table 1. We also report WERs using the 4-gram count model by shallow fusion with a weight of 0.01, for comparison to the performance of the LSTM LM.

Table 1: *Perplexities (PPL) on the concatenation of dev-clean and dev-other sets of LibriSpeech. All models have the same vocabulary of 10K BPE.*

LM	3-gram	4-gram	5-gram	LSTM
PPL	104.6	88.2	85.1	65.9

6. Experiments

All attention models and neural network language models were trained and decoded with RETURNN. For both Switchboard and LibriSpeech, we first used the BPE vocabulary of 10K subword units to tune the hyperparameters of the model, then trained the

[3] http://www.openslr.org/11/

models with 1K and 5K BPE units. We found 1K and 10K to be optimal for Switchboard and LibriSpeech respectively. We use label smoothing [55], dropout [56], Adam [57], learning rate warmup [26], and automatic learning rate scheduling according to a cross-validation set ("Newbob") [1].

6.1. Pretraining

In all cases we use layer-wise pretraining for the encoder, where we start with two encoder layers and a single max-pool in between with factor 32. Then we add a LSTM layer and a max-pool in between, and we reduce the first max-pool to factor 16 and the new one with factor 2 such that we always keep the same total encoder time reduction factor of 32. Only when we end up at 6 layers, we remove some of the max-pooling ops to get a final total time reduction factor of e.g. 8. Directly starting with a time reduction factor of 8 with and with 2 layers did not work for us. Also directly starting with 6 layers and time reduction factor of 32 did not work for us. Similar experiments for translation converged also without pretraining, however with much worse performance compared when layer-wise pretraining was used [30]. With more careful tuning or more training data, it might have worked without pretraining as it is seen in the literature, however, that is not necessary with pretraining.

We were interested in the optimal final total time reduction factor, after the pretraining with time reduction factor 32. We tried factor 8, 16 and 32, and ended up with 20.4, 21.0 and 21.9 WER% respectively, on the full Hub5'00 set (Switchboard + Callhome). Thus we continue to use a final reduction factor of 8 in all further experiments. Note that a lower factor requires more memory and more computation for the global attention and was not feasible with our hardware and computational resources.

6.2. Switchboard 300h

Switchboard consists of about 300 hours of training data. There is also the additional Fisher training dataset, so combined it makes the total of about 2000h. In this work, we only use the 300h-Switchboard training data. We use 40-dimensional Gammatone features [58], and the feature extraction was done with RASR [59]. Results are shown in Table 2. We observe that our attention model performs better on the easier Switchboard subset of the dev set Hub5'00, where it is the best end-to-end model we know. On the harder Callhome part, it also performs well compared to other end-to-end models but the relative difference is not as high.

6.3. LibriSpeech 1000h

LibriSpeech training dataset consist of about 1000 hours of read audio books. The dev and test sets were split into simple ("clean") and harder ("other") subsets [49]. We do 40-dim. MFCC feature extraction on-the-fly in RETURNN, based on librosa [62]. We use CTC as an additional loss function applied on top of the decoder to help the convergence, although this is not used in decoding [63]. We initially trained only using the train-clean set and restricting it to sequences not longer than 75 characters in the orthography. Results are shown in Table 3. Our end-to-end system achieves competitive performance even without using language models. We observed that the shallow fusion with LSTM LM brings from 17% to 27% relative improvements in terms of WER on different subsets. This improvement is much larger than in the case of Switchboard. The amount of data is most likely the reason for this observation. For LibriSpeech, the external data of 800M words is used to train the language models, which is 80 times larger than the 10M words corresponding to the transcription of 1000 hours of audio. In addition, this 10M transcription is not part of the language model training data. In case of Switchboard, the LM is trained only on about 27M words, including 3M of transcription used to train the end-to-end system. Text data for conversational speech is not as readily available as for read speech. The WER of 3.54% on the dev-clean and 3.82% on the test-clean subsets are the best performance on this task to the best of our knowledge for systems trained only using LibriSpeech data.[4]

Table 2: *Comparisons on Switchboard 300h. The hybrid HMM/NN model is a 6 layer deep bidirectional LSTM. The attention model has a 6 layer deep bidirectional LSTM encoder and a 1 layer LSTM decoder. CDp are (clustered) context-dependend phones. Byte-pair encoding (BPE) are sub-word units. SWB and CH are from Hub5'00.* [1]*added noise from external data.* [2]*added the lexicon, i.e. also additional data.*

model	LM	label unit	WER[%] Hub5'00 SWB	WER[%] Hub5'00 CH	WER[%] Hub5'01
LF MMI, 2016 [7]	4-gram	CDp	9.6	19.3	
hybrid	4-gram	CDp	9.8	19.0	14.7
hybrid	LSTM	CDp	**8.3**	**17.3**	12.9
CTC[1], 2014 [12]	RNN	chars	20.0	31.8	
CTC, 2015 [60]	none	chars	38.0	56.1	
CTC, 2015 [60]	RNN	chars	21.4	40.2	
attention, 2016 [61]	none	chars	32.8	52.7	
attention, 2016 [61]	5-gram	chars	30.5	50.4	
attention, 2016 [61]	none	words	26.8	48.2	
attention, 2016 [61]	3-gram	words	25.8	46.0	
CTC, 2017 [16]	none	chars	24.7	37.1	
CTC, 2017 [16]	n-gram	chars	19.8	32.1	
CTC[2], 2017 [16]	word RNN	chars	14.0	**25.3**	
attention, 2017 [28]	none	chars	23.1	40.8	
attention	none	BPE 10K	13.5	27.1	19.9
		BPE 1K	13.1	26.1	19.7
	LSTM	BPE 1K	**11.8**	25.7	18.1

6.4. Beam search prune error analysis

Beam search is an approximation for the decision rule

$$x_1^T \to \hat{w}_1^N := \arg\max_{w_1^N} p(w_1^N | x_1^T).$$

The approximation is the pruning we apply due to the beam size. Beam search decoding for hybrid models is very sophisticated and uses a dynamic beam size based on the partial hypothesis scores which can become very large (on the order of thousands) [67]. The beam search for attention models works directly on the labels, i.e. on the BPE units in our case, and usually a static fixed very low beam size (e.g. 10) is used. It has been shown that increasing the beam size much more does not increase net performance. This indicates that we do not have a search problem but we wanted to analyze this in more detail. Specifically, we are interested in how many errors we are making due to the pruning for our attention models, and we can count that by calculating the search score of the reference target sequence, and compare it to the search score of the decoded sequence. If the decoded sequence has a higher score than the reference target sequence, we have not made a search error but it is a model error. We count the number of sequences where the decoded sequence has a lower score than the reference target sequence. We report our

[4] After the Interspeech submission deadline, better WERs for LibriSpeech have been reported in [64] with 3.51% / 3.19% WER on test-clean using a hybrid HMM-LSTM / system combination.

Table 3: *Comparisons on LibriSpeech 1000h. The attention model has a 6 layer deep bidirectional LSTM encoder and a 1 layer LSTM decoder. CDp are (clustered) context-dependend phones. Byte-pair encoding (BPE) are sub-word units. Lattice-free (LF) maximum mutual information (MMI) [7] is a sequence criterion to train a hybrid HMM/NN model. Auto SeGmentation (ASG) [65] can be seen as a variant of the CTC criterion and model. Policy learning is a sequence training method, applied here on a CTC model [15]. If not specified, the official 4-gram word LM is used. The remaining attention models are all our models.*

model	LM	label unit	WER[%]			
			dev		test	
			clean	other	clean	other
hybrid, FFNN, 2015 [49]	4-gram	CDp	4.90	12.98	5.51	13.97
LF MMI, LSTM, 2016 [7]	4-gram	CDp			4.28	
CTC, 2015 [66]	4-gram	chars			5.33	13.25
ASG (CTC), 2017 [65]	4-gram	chars			4.80	14.50
ASG (CTC), 2017 [65]	none	chars			6.70	20.80
CTC, PL, 2017 [15]	4-gram	chars	5.10	14.26	5.42	14.70
attention	none	BPE	4.87	14.37	4.87	15.39
	4-gram	BPE	4.79	14.31	4.82	15.30
	LSTM	BPE	**3.54**	**11.52**	**3.82**	**12.76**

results in Table 4. We observe that for our standard beam size 12, the number of search errors are well below 1%, and also the WER will not noticeably improve with a larger beam size. Note that we only analyzed the search errors regarding reaching the reference target sequence. We did not count search errors regarding reaching any sequence with lower WER. However, our results still suggest that we do not seem to have a search problem but a model problem.

Table 4: *Beam search error analysis, performed on LibriSpeech, without language model. We provide both the number of reference-related search errors, relative to the number of sequences, and also the corresponding WER.*

beam size	search errors [%] (WER [%])			
	dev		test	
	clean	other	clean	other
4	1.52 (4.87)	1.68 (14.53)	1.07 (4.87)	1.70 (15.49)
8	0.96 (4.88)	0.98 (14.40)	0.76 (4.87)	1.02 (15.39)
12	0.81 (4.87)	0.59 (14.37)	0.61 (4.86)	0.71 (15.39)
16	0.70 (4.87)	0.52 (14.36)	0.50 (4.86)	0.58 (15.37)
32	0.26 (4.87)	0.14 (14.34)	0.19 (4.86)	0.20 (15.34)

7. Conclusions

We presented an encoder-decoder-attention model for speech recognition operating on BPE subword units. We introduced a new method for pretraining the encoder, which was crucial for both convergence and the performance in terms of WER. We further improved our recognition accuracy by a joint beam search with a LSTM LM trained on the same subword vocabulary. We carried out experiments on two standard datasets. On the 300h-Switchboard, we achieved competitve results compared to the previously reported end-to-end models, while the WERs are still higher than the conventional hybrid systems. On the 1000h-LibriSpeech task, we obtained competitive results across different evaluation subsets. To the best of our knowledge, the WERs of 3.54% on the dev-clean and 3.82% on the test-clean subsets are the best results reported on this task, when only the official LibriSpeech training data is used.

8. Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537, project "SEQCLAS"). The work reflects only the authors' views and the ERC Executive Agency is not responsible for any use that may be made of the information it contains. The GPU cluster used for the experiments was partially funded by Deutsche Forschungsgemeinschaft (DFG) Grant INST 222/1168-1.

9. References

[1] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," in *ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 2462–2466.

[2] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994, vol. 247.

[3] A. J. Robinson, "An application of recurrent nets to phone probability estimation," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 298–305, 1994.

[4] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.

[5] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "GMM-free DNN acoustic model training," in *ICASSP*, 2014.

[6] A. Zeyer, E. Beck, R. Schlüter, and H. Ney, "CTC in the context of generalized full-sum HMM training," in *Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 944–948.

[7] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.

[8] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Interspeech*, 2015.

[9] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *ICASSP*, Orlando, FL, USA, May 2002, pp. 845–848.

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*. ACM, 2006, pp. 369–376.

[11] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *ICML*, T. Jebara and E. P. Xing, Eds. JMLR Workshop and Conference Proceedings, 2014, pp. 1764–1772.

[12] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep-Speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[13] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *ASRU*. IEEE, 2015, pp. 167–174.

[14] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[15] Y. Zhou, C. Xiong, and R. Socher, "Improving end-to-end speech recognition with policy learning," *arXiv preprint arXiv:1712.07101*, 2017.

[16] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *ICASSP*. IEEE, 2017, pp. 4805–4809.

[17] H. Liu, Z. Zhu, X. Li, and S. Satheesh, "Gram-ctc: Automatic unit selection and target decomposition for sequence labelling," *arXiv preprint arXiv:1703.00096*, 2017.

[18] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. Interspeech*, 2017, pp. 3707–3711.

[19] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," in *Proc. Interspeech*, 2017, pp. 959–963.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[21] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches

to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[22] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[23] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[24] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[25] P. Doetsch, A. Zeyer, and H. Ney, "Bidirectional decoder networks for attention-based end-to-end offline handwriting recognition," in *International Conference on Frontiers in Handwriting Recognition*, Shenzhen, China, Oct. 2016, pp. 361–366.

[26] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[27] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *ASRU*, Okinawa, Japan, Dec. 2017, pp. 206–213.

[28] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," in *Proc. Interspeech*, 2017, pp. 3532–3536.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Annual Meeting of the Assoc. for Computational Linguistics*, Melbourne, Australia, Jul. 2018.

[31] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: first results," *arXiv preprint arXiv:1412.1602*, 2014.

[32] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *NIPS*, 2016, pp. 5067–5075.

[33] R. Aharoni and Y. Goldberg, "Morphological inflection generation with hard monotonic attention," *arXiv preprint arXiv:1611.01487*, 2016.

[34] C. Raffel, T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," *arXiv preprint arXiv:1704.00784*, 2017.

[35] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382*, 2017.

[36] A. Tjandra, S. Sakti, and S. Nakamura, "Local monotonic attention mechanism for end-to-end speech and language processing," in *IJCNLP*, vol. 1, 2017, pp. 431–440.

[37] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of "attention" in sequence-to-sequence models,"," in *Proc. of Interspeech*, 2017.

[38] J. Hou, S. Zhang, and L. Dai, "Gaussian prediction based attention for online end-to-end speech recognition," in *Proc. Interspeech*, 2017, pp. 3692–3696.

[39] P. Doetsch, M. Hannemann, R. Schlueter, and H. Ney, "Inverted alignments for end-to-end automatic speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1265–1273, Dec. 2017.

[40] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *ASRU*. IEEE, 2017, pp. 193–199.

[41] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, D. Seetapun, A. Sriram *et al.*, "Exploring neural transducers for end-to-end speech recognition," *arXiv preprint arXiv:1707.07413*, 2017.

[42] 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, pp. 939–943.

[43] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Interspeech*, 2017.

[44] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL*, Berlin, Germany, August 2016, pp. 1715–1725.

[45] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: the RWTH extensible training framework for universal recurrent neural networks," in *ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 5345–5349.

[46] TensorFlow Development Team, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[47] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling." in *Interspeech*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[48] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[49] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[50] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models." in *Interspeech*, 2016, pp. 22–26.

[51] P. Bahar, J. Rosendahl, N. Rossenbach, and H. Ney, "The RWTH Aachen machine translation systems for IWSLT 2017," in *Int. Workshop on Spoken Language Translation*, Tokyo, Japan, Dec. 2017, pp. 29–34.

[52] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *ACL*, 2016.

[53] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *ICASSP*, Detroit, MI, USA, May 1995, pp. 181–184.

[54] A. Stolcke, "SRILM-an extensible language modeling toolkit." in *Interspeech*, Denver, CO, USA, Sep. 2002.

[55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[58] R. Schlüter, I. Bezrukov, H. Wagner, and H. Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *ICASSP*, Honolulu, HI, USA, Apr. 2007, pp. 649–652.

[59] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *ICASSP*, Florence, Italy, May 2014, pp. 3313–3317.

[60] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, "Lexicon-free conversational speech recognition with neural networks," in *Proc. NAACL*, 2015.

[61] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *ICASSP*. IEEE, 2016, pp. 5060–5064.

[62] librosa Development Team, "librosa 0.5.0," Feb. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.293021

[63] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Interspeech*, 2017.

[64] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane, "The CAPIO 2017 conversational speech recognition system," *arXiv preprint 1801.00059 v2*, 2018.

[65] V. Liptchinsky, G. Synnaeve, and R. Collobert, "Letter-based speech recognition with gated convnets," *arXiv preprint arXiv:1712.09444*, 2017.

[66] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[67] D. Nolden, "Progress in decoding for large vocabulary continuous speech recognition," Ph.D. dissertation, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, Apr. 2017.