A comprehensive analysis on attention models

Albert Zeyer[1,2], **André Merboldt**[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/
{zeyer, schlueter, ney}@cs.rwth-aachen.de, andre.merboldt@rwth-aachen.de

Abstract

Sequence-to-sequence attention-based models are a promising approach for end-to-end speech recognition. The increased model power makes the training procedure more difficult, and analyzing failure modes of these models becomes harder because of the end-to-end nature. In this work, we present various analyses to better understand training and model properties. We investigate on pretraining variants such as growing in depth and width, and their impact on the final performance, which leads to over 8% relative improvement in word error rate. For a better understanding of how the attention process works, we study the encoder output and the attention energies and weights. Our experiments were performed on Switchboard, LibriSpeech and Wall Street Journal.

1 Introduction

The encoder-decoder framework with attention [Bahdanau et al., 2015, Luong et al., 2015, Wu et al., 2016] has been successfully applied to automatic speech recognition (ASR) [Chan et al., 2015, Chiu et al., 2017, Toshniwal et al., 2018, Krishna et al., 2018, Zeyer et al., 2018b, Zeghidour et al., 2018a, Weng et al., 2018, Sabour et al., 2018] and is a promising end-to-end approach. The model outputs are words, sub-words or characters, and training the model can be done from scratch without any prerequisites except the training data in terms of audio features with corresponding transcriptions.

In contrast to the conventional hybrid hidden Markov models (HMM) / neural network (NN) approach [Bourlard and Morgan, 1994, Robinson, 1994], the encoder-decoder model does not model the alignment explicitly. In the hybrid HMM/NN approach, a latent variable of hidden states is introduced, which model the phone state for any given time position. Thus by searching for the most probable sequence of hidden states, we get an explicit alignment. There is no such hidden latent variable in the encoder decoder model. Instead there is the attention process which can be interpreted as an implicit soft alignment. As this is only implicit and soft, it is harder to enforce constraints such as monotonicity, i.e. that the attention of future label outputs will focus also only to future time frames. Also, the interpretation of the attention weights as a soft alignment might not be completely valid, as the encoder itself can shift around and reorder evidence, i.e. the neural network could learn to pass over information in any possible way. E.g. the encoder could compress all the information of the input into a single frame and the decoder can learn to just attend on this single frame. We observed this behavior in early stages of the training. Thus, studying the temporal "alignment" behavior of the attention model becomes more difficult.

Other end-to-end models such as connectionist temporal classification [Graves et al., 2006] has often been applied to ASR in the past [Graves and Jaitly, 2014, Hannun et al., 2014, Miao et al., 2015, Amodei et al., 2016, Soltau et al., 2017, Audhkhasi et al., 2017, Krishna et al., 2018, Zenkel et al., 2018, Zhang and Lei, 2018]. Other approaches are e.g. the inverted hidden Markov / segmental encoder-decoder model [Doetsch et al., 2017, Beck et al., 2018a], the recurrent transducer [Rao et al., 2017, Battenberg et al., 2017, Prabhavalkar et al., 2017a], or the recurrent neural aligner [Sak et al.,

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

2017, Dong et al., 2018]. Depending on the interpretation, these can all be seen as variants of the encoder decoder approach. In some of these models, the attention process is not soft, but a hard decision. This hard decision can also become a latent variable such that we include several choices in the beam search. This is also referred to as hard attention. Examples of directly applying this idea on the usual attention approach are given by Raffel et al. [2017], Aharoni and Goldberg [2016], Chiu* and Raffel* [2018], Luo et al. [2017], Lawson et al. [2018].

We study recurrent NN (RNN) encoder decoder models in this work, which use long short-term memory (LSTM) units [Hochreiter and Schmidhuber, 1997]. Recently the transformer model [Vaswani et al., 2017] gained attention, which only uses feed-forward and self-attention layers, and the only recurrence is the label feedback in the decoder. As this does not include any temporal information, some positional encoding is added. This is not necessary for a RNN model, as it can learn such encoding by itself, which we demonstrate later for our attention encoder.

We study attention models in more detail here. We are interested in when, why and how they fail and do an analysis on the search errors and relative error positions. We study the implicit alignment behavior via the attention weights and energies. We also analyze the encoder output representation and find that it contains information about the relative position and that it specially marks frames which should not be attended to, which correspond to silence.

2 Related work

Karpathy [2015] analyzes individual neuron activations of a RNN language model and finds a neuron which becomes sensitive to the position in line. Belinkov and Glass [2017] analyzed the hidden activations of the DeepSpeech 2 [Amodei et al., 2016] CTC end-to-end system and shows their correlation to a phoneme frame alignment. Palaskar and Metze [2018] analyzed the encoder state and the attention weights of an attention model and makes similar observations as we do. Attention plots were used before to understand the behaviour of the model [Chorowski et al., 2015]. Beck et al. [2018b] performed a comparison of the alignment behavior between hybrid HMM/NN models, the inverted HMM and attention models. [Prabhavalkar et al., 2017b] investigate the effects of varying block sizes, attention types, and sub-word units. Understanding the inner working of a speech recognition system is also subject in [Tang et al., 2017], where the authors examine activation distribution and temporal patterns, focussing on the comparison between LSTM and GRU systems.

A number of saliency methods [Simonyan et al., 2014, Luisa M Zintgraf and Welling, 2017, Sundararajan et al., 2017] are used for interpreting model decisions.

3 ASR tasks and baselines

In all cases, we use the RETURNN framework [Zeyer et al., 2018a] for neural network training and inference, which is based on TensorFlow [TensorFlow Development Team, 2015] and contains some custom CUDA kernels. In case of the attention models, we also use RETURNN for decoding. All experiments are performed on single GPUs, we did not take advantage of multi-GPU training. In some cases, the feature extraction, and in the hybrid case the decoding, is performed with RASR [Wiesler et al., 2014]. All used configs as well as used source code are published.[1]

3.1 Switchboard 300h

The Switchboard corpus [Godfrey et al., 2003] consists of English telephone speech. We use the 300h train dataset (LDC97S62), and a 90% subset for training, and a small part for cross validation, which is used for learning rate scheduling and to select a few models for decoding. We decode and report WER on Hub5'00 and Hub5'01. We use Hub5'00 to select the best model which we report the numbers on.

Our hybrid HMM/NN model uses a deep bidirectional LSTM as described by Zeyer et al. [2017]. Our baseline has 6 layers with 500 nodes in each direction. It uses dropout of 10% on the non-recurrent input of each LSTM layer, gradient noise with standard deviation of 0.3, Adam with Nesterov

[1] https://github.com/rwth-i6/returnn-experiments/tree/master/2018-nips-irasl-paper

Table 1: Switchboard results. [1]is our baseline, and we selected the best model from multiple runs. [2]is our best model with improved pretraining, see Section 5, Table 7.

model	paper	LM	label unit	WER[%]			
				Hub5'00			Hub5'01
				Σ	SWB	CH	Σ
hybrid	[Povey et al., 2016]	4-gram	CDp		9.6	19.3	
	[Weng et al., 2018]	4-gram	CDp		9.6	19.3	
	[Zeyer et al., 2018b]	LSTM	CDp		**8.3**	**17.3**	**12.9**
	this work	4-gram	CDp	14.3	9.6	19.0	14.5
inverted HMM	[Beck et al., 2018a]	4-gram	CDp	19.3	13.0	25.6	
CTC	[Zweig et al., 2017]	none	chars		24.7	37.1	
	[Zweig et al., 2017]	n-gram	chars		19.8	32.1	
	[Zweig et al., 2017]	word RNN	chars		14.0	25.3	
attention	[Lu et al., 2016]	none	words		26.8	48.2	
	[Lu et al., 2016]	3-gram	words		25.8	46.0	
	[Toshniwal et al., 2017]	none	chars		23.1	40.8	
	[Weng et al., 2018]	none	chars		12.2	23.3	
	[Zeyer et al., 2018b]	none	BPE 1k	19.6	13.1	26.1	19.7
	[Zeyer et al., 2018b]	LSTM	BPE 1k	18.8	11.8	25.7	18.1
attention	this work[1]	none	BPE 1k	19.1	12.8	25.3	19.0
	this work[2]	none	BPE 1k	17.8	11.9	23.7	17.7
	this work[2]	LSTM	BPE 1k	**17.1**	**11.0**	**23.1**	**16.6**

momentum (Nadam) [Dozat, 2015], Newbob learning rate scheduling [Zeyer et al., 2017], and focal loss [Lin et al., 2017].

Our attention model uses byte pair encoding [Sennrich et al., 2015] as subword units. We follow the baseline with about 1000 BPE units as described by Zeyer et al. [2018b]. All our baselines and a comparison to results from the literature are summarized in Table 1.

3.2 LibriSpeech 1000h

The LibriSpeech dataset [Panayotov et al., 2015a] are read audio books and consists of about 1000h of speech. A subset of the training data is used for cross-validation, to perform learning rate scheduling and to select a number of models for full decoding. We use the dev-other set for selecting the final best model.

The end-to-end attention model uses byte pair encoding (BPE) [Sennrich et al., 2015] as subword units with a vocabulary of 10k BPE units. We follow the baseline as described by Zeyer et al. [2018b]. A comparison of our baselines and other models are in Table 2.

3.3 Wall Street Journal 80h

The Wall Street Journal (WSJ) dataset [Paul and Baker, 1992] is read text from the WSJ. We use 90% of si284 for training, the remaining for cross validation and learning rate scheduling, dev93 for validation and selection of the final model, and eval92 for the final evaluation.

We trained an end-to-end attention model using BPE subword units, with a vocabulary size of about 1000 BPE units. Our preliminary results are shown in Table 3. Our attention model is based on the improved pretraining scheme as described in Section 5.

4 Error analysis

We analyze the errors in the decoding process during beam search. In Fig. 1 we collected the correspondence between the beam size and the WER or the amount of search errors. We just count

Table 2: LibriSpeech results. [1]is our baseline, and we selected the best model from multiple runs. [2]is our best model with improved pretraining, see Section 5.

model	paper	LM	label unit	WER[%]			
				dev		test	
				clean	other	clean	other
hybrid	[Panayotov et al., 2015b]	4-gram	CDp	4.90	12.98	5.51	13.97
	[Han et al., 2018]	4-gram	CDp	3.35	8.78	3.63	8.94
	[Han et al., 2018]	RNN	CDp	**3.12**	**8.28**	**3.51**	**8.58**
CTC	[Amodei et al., 2016]	4-gram	chars			5.33	13.25
	[Zhou et al., 2017]	4-gram	chars	5.10	14.26	5.42	14.70
ASG	[Liptchinsky et al., 2017]	none	chars			6.70	20.80
	[Liptchinsky et al., 2017]	4-gram	chars			4.80	14.50
	[Zeghidour et al., 2018b]	CNN	chars	**3.16**	**10.05**	**3.44**	**11.24**
attention	[Zeyer et al., 2018b]	none	BPE 10k	4.87	14.37	4.87	15.39
	[Zeyer et al., 2018b]	LSTM	BPE 10k	3.54	11.52	3.82	12.76
	[Sabour et al., 2018]	none	BPE 10k			**4.5**	**13.3**
attention	this work[1]	none	BPE 10k	4.68	14.27	4.81	15.43
	this work[2]	none	BPE 10k	4.71	13.95	4.70	15.20

Table 3: WSJ results. Marked are the best results with and without language model.

model	paper	comment	LM	label unit	WER[%]	
					dev93	eval92
GMM	[Panayotov et al., 2015a]		3-gram	CDp	9.39	6.26
hybrid	[Panayotov et al., 2015a]	feed-forward	3-gram	CDp	6.97	3.92
	[Chan and Lane, 2015]		3-gram	CDp	6.58	**3.47**
CTC	[Liu et al., 2017]	Gram-CTC	none	word piece		16.7
	[Liu et al., 2017]	Gram-CTC	LM	word piece		**6.7**
attention	[Chan et al., 2016]	LSD	none	word piece		9.6
	[Chorowski and Jaitly, 2016]	LS	none	chars	13.7	10.6
	[Chorowski and Jaitly, 2016]	LS	3-gram	chars	9.7	**6.7**
	[Zhang et al., 2017]	quite deep	none	chars		10.5
	[Renduchintala et al., 2018]	augmentation	none	chars	22.7	17.5
	[Sabour et al., 2018]	OCD	none	chars		**9.3**
attention	this work	SWB best config	none	BPE 1k	16.1	14.0
	this work	improved pretrain	none	BPE 1k	15.3	13.6

the search errors where the models recognized sentence (via beam search) has a worse model score than the ground truth sentence. We observe that we do only very few search errors, and the amount of search errors seems independent from the final WER performance. Thus we conclude that we mostly have a problem in the model.

We also were interested in the score difference between the best recognized sentence and the ground truth sentence. The results are in Fig. 2. We can see that they concentrate on the lower side, around 10%, which is an indicator why a low beam size seems to be sufficient.

5 Analysis on pretraining

It has been observed that pretraining can be substantial for good performance, and sometimes to get a converging model at all [Zeyer et al., 2018a,b]. We provide a study on cases with attention-based models where pretraining benefits convergence, and compare the performance with and without pretraining.

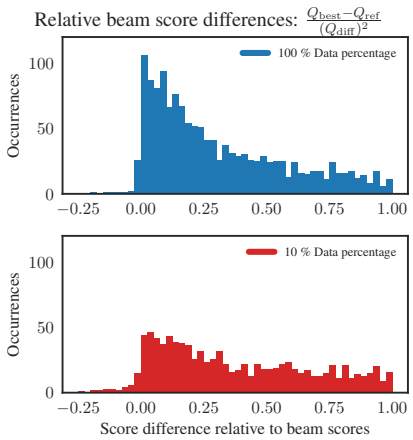


Figure 1: Beam score difference within a beam, relative to the score variations within the beam. This is for an attention model using beam size 12 on LibriSpeech (test-other).

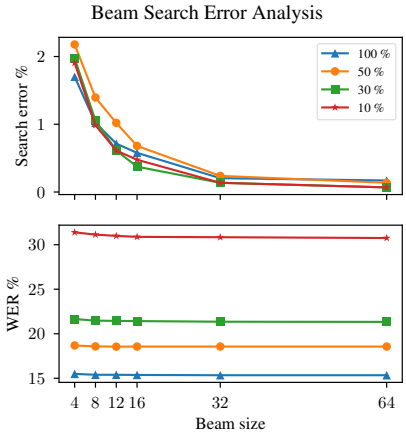


Figure 2: Beam search errors and word error rates as a function of the beam size using different training data percentages. Search error occurance per sentence. This is with an attention model on LibriSpeech (test-other).

Table 4: Comparison of different encoder depth and width with the original pretraining scheme enabled or disabled. We had to lower the initial learning rate to allow the model to converge: [1]lr $5 \cdot 10^{-4}$, [2]lr 10^{-4}

Encoder		Hub5'01 WER[%]	
		Pretraining	
Layers	Hidden units	No	Yes
4	500	19.4	19.9
	700	19.0	19.4
	1000	**18.3**	19.1
5	500	19.9	20.5
	700	19.0	20.5
	1000[1]	19.1	19.7
6	500	20.1	19.8
	700	19.4	19.4
	1000[2]	20.9	19.7

Table 5: Comparison of different start number of layers and start time reduction factor in pretraining. In all cases, the pretraining scheme ends up with a 6 layer encoder, and time reduction factor 8.

Encoder pretrain start		WER[%]			
	Time	Hub5'00			Hub5'01
Layers	reduction	Σ	CH	SWB	Σ
2	32	19.6	26.2	13.1	19.7
3	32	19.2	25.7	12.7	18.6
4	32	18.9	25.2	12.6	18.5
5	32	**18.7**	**24.9**	**12.5**	**18.2**
6	32	>100	>100	>100	>100
3	8	19.0	25.5	12.6	18.9
4	8	**18.4**	**24.3**	**12.5**	**18.3**
5	8	>100	>100	>100	>100
6	8	>100	>100	>100	>100

The pretraining variant of the Switchboard baseline (6 layers, time reduction 8 after pretraining) consists of these steps: 1. starts with 2 layers (layer 1 and 6), time reduction 32, and dropout as well as label smoothing disabled; 2. enable dropout; 3. 3 layers (layer 1, 2 and 6); 4. 4 layers (layer 1, 2, 3 and 6); 5. 5 layers (layer 1, 2, 3, 4 and 6); 6. all 6 layers; 7. decrease time reduction to 8; 8. final model, enable label smoothing. Each pretrain step is repeated for 5 epochs, where one epoch corresponds to 1/6 of the whole train corpus. In addition, a linear learning rate warm-up is performed from 1e-4 to 1e-3 in 10 epochs. We have to start with 2 layers as we want to have the time pooling in between the LSTM layers. In Table 4, performed on Switchboard, we varied the number of encoder layers and encoder LSTM units, both with and without pretraining. We observe that the overall best model is with 4 layers without the pretraining variant. I.e. we showed that we can directly start with 4 layers and time reduction 8 and yield very good results. We even can start directly with 6 layer with a reduced learning rate. This was surprising to us, as this was not possible in earlier experiments. This might be due to a reduced and improved BPE vocabulary. We note that overall all the pretraining experiments seems to run more stable. We also can see that with 6 layers (and also more), pretraining yields better results than no pretraining.

Table 6: Comparison of number of pretrain step repetitions. In all cases, the pretraining scheme ends up with a 6 layer encoder, and time reduction factor 8.

Pretrain repetitions	WER[%] Hub5'00 Σ	CH	SWB	Hub5'01 Σ
1	19.0	25.2	12.8	18.5
2	19.1	25.3	12.8	18.8
3	19.1	25.4	12.9	18.7
4	18.5	24.7	**12.3**	**18.3**
5	**18.4**	**24.3**	12.5	**18.3**

Table 7: Comparison of different time reduction factors (always the same during pretraining and in the final model), adding the LSTM layer always on top instead of in between, and growing in width.

Time red.	Pretraining add top l.	grow width	WER[%] Hub5'00 Σ	CH	SWB	Hub5'01 Σ
8	no	no	18.4	24.3	12.5	18.3
	yes	no	18.4	24.3	12.5	17.8
6	no	no	18.3	24.4	12.0	18.2
	no	yes	17.9	23.8	**11.9**	18.0
	yes	no	17.9	23.8	**11.9**	**17.7**
	yes	yes	**17.8**	**23.7**	**11.9**	**17.7**

These results motivated us to perform further investigations into different variants of pretraining. It seems that pretraining allows to train deeper model, however using too much pretraining can also hurt. We showed that we can directly start with a deeper encoder and lower time reduction. In Table 5, we analyzed the *optimal initial number of layers*, and the *initial time reduction*. We observed that starting with a deeper network improves the overall performance, but also it still helps to then go deeper during pretraining, and starting too deep does not work well. We also observed that directly starting with time reduction 8 also works and further improves the final performance, but it seems that this makes the training slightly more unstable. In further experiments, we directly start with 4 layers and time reduction 8. We were also interested in the *optimal number of repetitions* of each pretrain step, i.e. how much epochs to train with each pretrain step; the baseline had 5 repetitions. We collected the results in Table 6. In further experiments, we keep 5 repetitions as the default.

It has already been shown by Zeyer et al. [2018b] that a lower final time reduction performed better. So far, the *lowest time reduction* was 8 in our experiments. By having a pool size of 3 in the first time max pooling layer, we achieve a better-performing model with time reduction factor of 6 as shown in Table 7. So far we always kept the top layer (layer 6) during pretraining as our intuition was that it might help to get always the same time reduction factor as an input to this layer. When directly starting with the low time reduction, we do not need this scheme anymore, and we can always add a new layer on top. Comparisons are collected in Table 7. We can conclude that this simpler scheme to add layers on top performs better.

We also did experiments with *growing the encoder width* / number of LSTM units during pretraining. We do this orthogonal to the growing in depth / number of layers. As before, our final number of LSTM units in each direction of the bidirectional deep LSTM encoder is 1024. Initially, we start with 50% of the final width, i.e. with 512 units. In each step, we linearly increase the number of units such that we have the final number of units in the last step. We keep the weights of existing units, and weights from/to newly added units are randomly initialized. We also decrease the dropout rate by the same factor. We can see that this width growing scheme performs better. This leads us to our current best model.

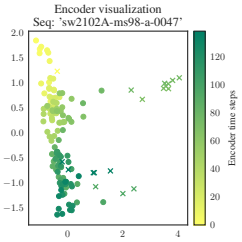
Our findings are that pretraining is in general more stable, esp. for deep models. However, the pretraining scheme is important, and less pretraining can improve the performance, although it becomes more unstable. We also used the same improved pretraining scheme and time reduction 6 for WSJ as well as LibriSpeech and observed similar improvements, compare Table 2 and Table 3.

6 Analysis on training variance

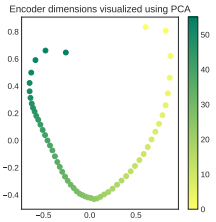
We have observed that training attention models can be unstable, and careful tuning of initial learning rate, warm-up and pretraining is important. Related to that, we observe a high training variance. I.e. with the same configuration but different random seeds, we get some variance in the final WER performance. We observed this even for the same random seed, which we suspect stems from non-deterministic behaviour in TensorFlow operations such as `tf.reduce_sum` based on kernels

Table 8: Training variance. Results on Switchboard 300h, with an attention model with 157M parameters, and a hybrid HMM/LSTM model with 41M parameters. We select the best epoch w.r.t. the best overall Hub5'00 result. Experiments with different random seeds, and same random seed but multiple different runs, i.e. showing non-determinism in the implementation.

model	variant	WER[%] (min–max, μ, σ)			
		Hub5'00			Hub5'01
		Σ	CH	SWB	Σ
attention	5 seeds	19.3–19.9, 19.6, 0.24	25.6–26.6, 26.1, 0.38	12.8–13.3, 13.1, 0.17	19.0–19.7, 19.4, 0.20
attention	5 runs	19.1–19.6, 19.3, 0.22	25.3–26.3, 25.8, 0.40	12.7–13.0, 12.9, 0.12	18.9–19.6, 19.2, 0.27
hybrid	5 seeds	14.3–14.5, 14.4, 0.08	19.0–19.3, 19.1, 0.12	9.6– 9.8, 9.7, 0.06	14.3–14.7, 14.5, 0.16
hybrid	5 runs	14.3–14.5, 14.4, 0.07	19.0–19.2, 19.1, 0.09	9.6– 9.8, 9.7, 0.08	14.4–14.6, 14.5, 0.07

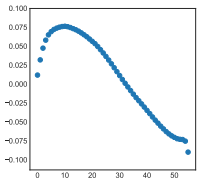


(a) Dimension reduction using PCA on a specific sequence. The different points are colored according to their relative position, and • indicates non-silence frames and × silence frames.

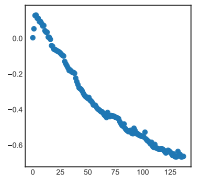


(b) Encoder dimensions visualized across the whole validation subset used for Switchboard. For this, the whole dataset is length-normalized to the mean encoder length, then all encoder activations are added together.

Figure 3: PCA from encoder output.



(a) Averaged encoder activation is plotted along the mean encoder length for a neuron in the encoder. X axis represents encoder position and Y axis activation.



(b) Encoder neuron unit activation for a specific sequence, axis are the same as for (c), as is the neuron.

Figure 4: Single neuron output.

using CUDA atomics. [2] This training variance seems to be about the same as due to random seeds, which is higher than we expected. Note that it also depends a lot on other hyper parameters. For example, in a previous iteration of the model using a larger BPE vocabulary, we have observed more unstable training with higher variance, and even sometimes some models diverge while others converge with the same settings. We also compare that to hybrid HMM/LSTM models. It can be observed that it is lower compared to the attention model. We argue that is due to the more difficult optimization problem, and also due to the much bigger model. All the results can be seen in Table 8.

7 Analysis of the encoder output

The encoder creates a high-level representation of the input. It also arguably represents further information needed for the decoder to know where to attend to. We try to analyze the output of the encoder and identify and examine the learned function. In Fig. 5, we plotted the encoder output and the attention weights, as well as the word positions in the audio.

One hypothesis for an important function of the encoder is the detection of frames which should not be attended on by the decoder, e.g. which are silent or non-speech. Such a pattern can be observed in Fig. 5. By performing a dimensionality reduction (PCA) on the encoder output, we can identify the most important distinct information, which we identify as *silence detection* and *encoder time position*, compare Fig. 3. Similar behavior was shown by Palaskar and Metze [2018]. We further try to identify *individual cells* in the LSTM which encodes the positional information. By qualitatively inspecting the different neurons activations, we have identified multiple neurons which perform the hypothesized function as shown in Fig. 4.

[2] This problem has been acknowledged in this GitHub issue.

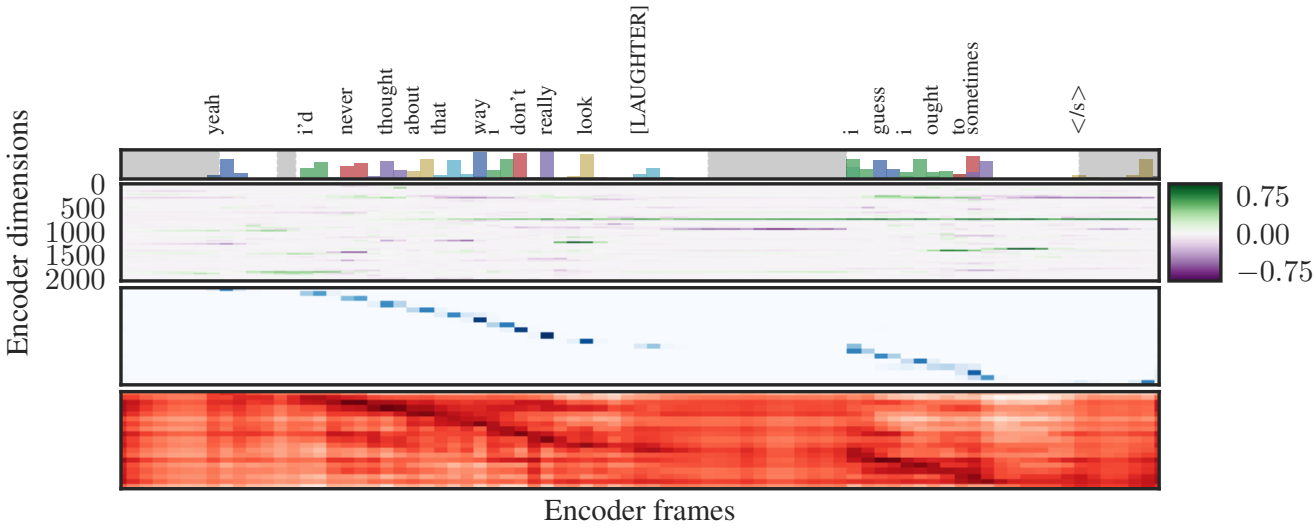


Figure 5: Encoder output combined with attention weights/energies of a sequence in the Switchboard dataset. The histogram at the top of the plot shows the different attention weight activations, each subsequent output has a different color assigned, model outputs corresponding to the same subword-unit are colored the same (green for "i" in the plot). Silence frames as aligned by our hybrid baseline are marked as gray areas. The matrix in the upper half of the figure corresponds to the $\tanh$ output of the last bidirectional LSTM layer in the encoder. Each row represents a encoder component across the encoder frames. The bottom two plots show the attention weights and energies, respectively. Both plots show in each row the activations for one decoder step across all encoder frames.

We also observed that the attention weights are always very local in the encoder frames, and often focus mostly on a single encoder frame, compare Fig. 5. The sharp behavior in the converged attention weight distribution has been observed before [Chan et al., 2015, Beck et al., 2018b, Palaskar and Metze, 2018]. We conclude that the information about the label also needs to be well-localized in the encoder output. To support this observation, we performed experiments where we explicitly allowed only a local fixed-size window of non-zero attention weights around the $\arg\max$ of the attention energies, to understand how much we can restrict the local context. The results can be seen in Table 9. This confirms the hypothesis that the information is localized in the encoder. We explain the gap in performance with decoder frames where the model is unsure to attend, and where a global attention helps the decoder to gather information from multiple frames at once. We observed that in such case, there is sometimes some relatively large attention weight on the very first and/or very last frame.

Table 9: Local attention window on Switchboard, WER on Hub5'00.

Model	Win. size	WER[%]
baseline	∞	19.6
local	10	20.7

8 Conclusion

We provided an overview of our recent attention models results on Switchboard, LibriSpeech and WSJ. We performed an analysis on the beam search errors. By our improved pretraining scheme, we improved our Switchboard baseline by over 8% relative in WER. We pointed out the high training variance of attention models compared to hybrid HMM/NN models. We analyzed the encoder output and identified the representation of the relative input position, both clearly visible in the PCA reduction of the encoder but even represented by individual neurons. Also we found indications that the encoder marks frames which can be skipped by decoder, which correlate to silence.

Acknowledgments

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") and from a Google Focused Award. The work reflects only the authors' views and none of the funding parties is responsible for any use that may be made of the information it contains.

References

Roee Aharoni and Yoav Goldberg. Sequence to sequence transduction with hard monotonic attention. 2016.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016.

Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo. Direct acoustics-to-word models for english conversational speech recognition. In *Proc. Interspeech*, pages 959–963, 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur, Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu. Exploring neural transducers for end-to-end speech recognition. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 206–213, 2017.

Eugen Beck, Mirko Hannemann, Patrick Doetsch, Ralf Schlüter, and Hermann Ney. Segmental encoder-decoder models for large vocabulary automatic speech recognition. In *Interspeech*, pages 766–770, 2018a.

Eugen Beck, Albert Zeyer, Patrick Doetsch, André Merboldt, Ralf Schlüter, and Hermann Ney. Sequence modeling and alignment for LVCSR-systems. In *Proceedings of the 13. ITG Symposium on Speech Communication*, Paderborn, Germany, October 2018b.

Yonatan Belinkov and James Glass. Analyzing hidden representations in end-to-end automatic speech recognition systems. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2441–2451. Curran Associates, Inc., 2017.

Hervé Bourlard and Nelson Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer, 1994.

William Chan and Ian Lane. Deep recurrent neural networks for acoustic modelling. *arXiv preprint arXiv:1504.01482*, 2015.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell. *CoRR*, abs/1508.01211, 2015.

William Chan, Yu Zhang, Quoc Le, and Navdeep Jaitly. Latent sequence decompositions. *arXiv preprint arXiv:1610.03035*, 2016.

Chung-Cheng Chiu* and Colin Raffel*. Monotonic chunkwise attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. *arXiv preprint arXiv:1712.01769*, 2017.

Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*, 2016.

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*, pages 577–585, 2015.

Patrick Doetsch, Mirko Hannemann, Ralf Schlueter, and Hermann Ney. Inverted alignments for end-to-end automatic speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1265–1273, December 2017.

Linhao Dong, Shiyu Zhou, Wei Chen, and Bo Xu. Extending recurrent neural aligner for streaming end-to-end speech recognition in mandarin. In *Interspeech*, 2018.

Timothy Dozat. Incorporating Nesterov momentum into Adam. Technical report, Stanford University, 2015. `http://cs229.stanford.edu/proj2015/054_report.pdf`.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'03, pages 517–520, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7803-0532-9. URL `http://dl.acm.org/citation.cfm?id=1895550.1895693`.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Tony Jebara and Eric P. Xing, editors, *ICML*, pages 1764–1772. JMLR Workshop and Conference Proceedings, 2014.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376. ACM, 2006.

Kyu J. Han, Akshay Chandrashekaran, Jungsuk Kim, and Ian Lane. The CAPIO 2017 conversational speech recognition system. *arXiv preprint 1801.00059 v2*, 2018.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. DeepSpeech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks, May 2015. URL `http://karpathy.github.io/2015/05/21/rnn-effectiveness`. [Online; accessed Dec. 2018].

Kalpesh Krishna, Shubham Toshniwal, and Karen Livescu. Hierarchical multitask learning for CTC-based speech recognition. *arXiv preprint arXiv:1807.06234*, 2018.

Dieterich Lawson, Chung-Cheng Chiu, George Tucker, Colin Raffel, Kevin Swersky, and Navdeep Jaitly. Learning hard alignments with variational inference. In *ICASSP*, pages 5799–5803. IEEE, 2018.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. Letter-based speech recognition with gated convnets. *arXiv preprint arXiv:1712.09444*, 2017.

Hairong Liu, Zhenyao Zhu, Xiangang Li, and Sanjeev Satheesh. Gram-CTC: Automatic unit selection and target decomposition for sequence labelling. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2188–2197, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Liang Lu, Xingxing Zhang, and Steve Renais. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In *ICASSP*, pages 5060–5064. IEEE, 2016.

Tameem Adel Luisa M Zintgraf, Taco S Cohen and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Yuping Luo, Chung-Cheng Chiu, Navdeep Jaitly, and Ilya Sutskever. Learning online alignments with continuous rewards policy gradient. In *ICASSP*, pages 2801–2805. IEEE, 2017.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

Yajie Miao, Mohammad Gowayyed, and Florian Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *ASRU*, pages 167–174. IEEE, 2015.

Shruti Palaskar and Florian Metze. Acoustic-to-word recognition with sequence-to-sequence models. *arXiv preprint arXiv:1807.09597*, 2018.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*, pages 5206–5210, 2015a.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: an ASR corpus based on public domain audio books. In *ICASSP*, pages 5206–5210. IEEE, 2015b.

Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992.

Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Interspeech*, pages 2751–2755, 2016.

Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. A comparison of sequence-to-sequence models for speech recognition. In *Proc. Interspeech*, pages 939–943, 2017a.

Rohit Prabhavalkar, Tara N. Sainath, Bo Li, Kanishka Rao, and Navdeep Jaitly. An analysis of "attention" in sequence-to-sequence models. In *Proc. Interspeech 2017*, pages 3702–3706, 2017b.

Colin Raffel, Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. *arXiv preprint arXiv:1704.00784*, 2017.

Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer. In *ASRU*, pages 193–199. IEEE, 2017.

Adithya Renduchintala, Shuoyang Ding, Matthew Wiesner, and Shinji Watanabe. Multi-modal data augmentation for end-to-end asr. *arXiv preprint arXiv:1803.10299*, 2018.

Anthony J Robinson. An application of recurrent nets to phone probability estimation. *Neural Networks, IEEE Transactions on*, 5(2):298–305, 1994.

Sara Sabour, William Chan, and Mohammad Norouzi. Optimal completion distillation for sequence learning. *arXiv preprint arXiv:1810.01398*, 2018.

Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays. Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping. In *Interspeech*, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Hagen Soltau, Hank Liao, and Haşim Sak. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. In *Proc. Interspeech*, pages 3707–3711, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/sundararajan17a.html`.

Zhiyuan Tang, Ying Shi, Dong Wang, Yang Feng, and Shiyue Zhang. Memory visualization for gated recurrent neural networks in speech recognition. pages 2736–2740, 2017.

TensorFlow Development Team. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. In *Proc. Interspeech*, pages 3532–3536, 2017.

Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. A comparison of techniques for language model integration in encoder-decoder speech recognition. *arXiv preprint arXiv:1807.10857*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 6000–6010, 2017.

Chao Weng, Jia Cui, Guangsen Wang, Jun Wang, Chengzhu Yu, Dan Su, and Dong Yu. Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition. In *Interspeech*, pages 761–765, 2018.

Simon Wiesler, Alexander Richard, Pavel Golik, Ralf Schlüter, and Hermann Ney. RASR/NN: The RWTH neural network toolkit for speech recognition. In *ICASSP*, pages 3313–3317, Florence, Italy, May 2014.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Neil Zeghidour, Nicolas Usunier, Gabriel Synnaeve, Ronan Collobert, and Emmanuel Dupoux. End-to-end speech recognition from the raw waveform. In *Interspeech*, 2018a.

Neil Zeghidour, Qiantong Xu, Vitaliy Liptchinsky, Nicolas Usunier, Gabriel Synnaeve, and Ronan Collobert. Fully convolutional speech recognition. *arXiv preprint arXiv:1812.06864*, 2018b.

Thomas Zenkel, Ramon Sanabria, Florian Metze, and Alex Waibel. Subword and crossword units for CTC acoustic models. In *Interspeech*, 2018.

Albert Zeyer, Patrick Doetsch, Paul Voigtlaender, Ralf Schlüter, and Hermann Ney. A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition. In *ICASSP*, pages 2462–2466, New Orleans, LA, USA, March 2017.

Albert Zeyer, Tamer Alkhouli, and Hermann 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, July 2018a.

Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. Improved training of end-to-end attention models for speech recognition. In *Interspeech*, Hyderabad, India, September 2018b.

ShiLiang Zhang and Ming Lei. Acoustic modeling with DFSMN-CTC and joint CTC-CE learning. In *Interspeech*, pages 771–775, 2018.

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*, pages 4845–4849. IEEE, 2017.

Yingbo Zhou, Caiming Xiong, and Richard Socher. Improving end-to-end speech recognition with policy learning. *arXiv preprint arXiv:1712.07101*, 2017.

Geoffrey Zweig, Chengzhu Yu, Jasha Droppo, and Andreas Stolcke. Advances in all-neural speech recognition. In *ICASSP*, pages 4805–4809. IEEE, 2017.