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