Attention-Based Models for Speech Recognition

Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow

Abstract

Recurrent sequence generators conditioned on input data through an attention mechanism have recently shown very good performance on a range of tasks including machine translation, handwriting synthesis [1, 2] and image caption generation [3]. We extend the attention-mechanism with features needed for speech recognition. We show that while an adaptation of the model used for machine translation in [2] reaches a competitive 18.7% phoneme error rate (PER) on the TIMIT phoneme recognition task, it can only be applied to utterances which are roughly as long as the ones it was trained on. We offer a qualitative explanation of this failure and propose a novel and generic method of adding location-awareness to the attention mechanism to alleviate this issue. The new method yields a model that is robust to long inputs and achieves 18% PER in single utterances and 20% in 10-times longer (repeated) utterances. Finally, we propose a change to the attention mechanism that prevents it from concentrating too much on single frames, which further reduces PER to 17.6% level.

1 Introduction

Recently, attention-based recurrent networks have been successfully applied to a wide variety of tasks, such as handwriting synthesis [1], machine translation [2], image caption generation [3] and visual object classification [4].[1] Such models iteratively process their input by selecting relevant content at every step. This basic idea significantly extends the applicability range of end-to-end training methods, for instance, making it possible to construct networks with external memory [6, 7].

We introduce extensions to attention-based recurrent networks that make them applicable to speech recognition. Learning to recognize speech can be viewed as learning to generate a sequence (transcription) given another sequence (speech). From this perspective it is similar to machine translation and handwriting synthesis tasks, for which attention-based methods have been found suitable [2, 1]. However, compared to machine translation, speech recognition principally differs by requesting much longer input sequences (thousands of frames instead of dozens of words), which introduces a challenge of distinguishing similar speech fragments[2] in a single utterance. It is also different from handwriting synthesis, since the input sequence is much noisier and does not have as clear structure. For these reasons speech recognition is an interesting testbed for developing new attention-based architectures capable of processing long and noisy inputs.

Application of attention-based models to speech recognition is also an important step toward building fully end-to-end trainable speech recognition systems, which is an active area of research. The

---

[1] An early version of this work was presented at the NIPS 2014 Deep Learning Workshop [5].
[2] Explained in more detail in Sec. 2.1.

dominant approach is still based on hybrid systems consisting of a deep neural acoustic model, a tri-phone HMM model and an n-gram language model [8, 9]. This requires dictionaries of hand-crafted pronunciation and phoneme lexicons, and a multi-stage training procedure to make the components work together. Excellent results by an HMM-less recognizer have recently been reported, with the system consisting of a CTC-trained neural network and a language model [10]. Still, the language model was added only at the last stage in that work, thus leaving open a question of how much an acoustic model can benefit from being aware of a language model during training.

In this paper, we evaluate attention-based models on the widely-used TIMIT phoneme recognition task. For each generated phoneme, an attention mechanism selects or weighs the signals produced by a trained feature extraction mechanism at potentially all of the time steps in the input sequence (speech frames). The weighted feature vector then helps to condition the generation of the next element of the output sequence. Since the utterances in this dataset are rather short (mostly under 5 seconds), we measure the ability of the considered models in recognizing much longer utterances which were created by artificially concatenating the existing utterances.

We start with a model proposed in [2] for the machine translation task as the baseline. This model seems entirely vulnerable to the issue of similar speech fragments but despite our expectations it was competitive on the original test set, reaching 18.7% phoneme error rate (PER). However, its performance degraded quickly with longer, concatenated utterances. We provide evidence that this model adapted to track the absolute location in the input sequence of the content it is recognizing, a strategy feasible for short utterances from the original test set but inherently unscalable.

In order to circumvent this undesired behavior, in this paper, we propose to modify the attention mechanism such that it explicitly takes into account both (a) the location of the focus from the previous step, as in [6] and (b) the features of the input sequence, as in [2]. This is achieved by adding as inputs to the attention mechanism auxiliary *convolutional features* which are extracted by convolving the attention weights from the previous step with trainable filters. We show that a model with such convolutional features performs significantly better on the considered task (18.0% PER). More importantly, the model with convolutional features robustly recognized utterances many times longer than the ones from the training set, always staying below 20% PER.

Therefore, the contribution of this work is three-fold. For one, we present a novel purely neural speech recognition architecture based on an attention mechanism, whose performance is comparable to that of the conventional approaches on the TIMIT dataset. Moreover, we propose a generic method of adding location awareness to the attention mechanism. Finally, we introduce a modification of the attention mechanism to avoid concentrating the attention on a single frame, and thus avoid obtaining less "effective training examples", bringing the PER down to 17.6%.

2 Attention-Based Model for Speech Recognition

2.1 General Framework

An attention-based recurrent sequence generator (ARSG) is a recurrent neural network that stochastically generates an output sequence $(y_1, \ldots, y_T)$ from an input x. In practice, x is often processed by an *encoder* which outputs a sequential input representation $h = (h_1, \ldots, h_L)$ more suitable for the attention mechanism to work with.

In the context of this work, the output y is a sequence of phonemes, and the input $x = (x_1, \ldots, x_{L'})$ is a sequence of feature vectors. Each feature vector is extracted from a small overlapping window of audio frames. The encoder is implemented as a deep bidirectional recurrent network (BiRNN), to form a sequential representation h of length $L = L'$.

At the i-th step an ARSG generates an output y_i by focusing on the relevant elements of h:

$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1}, h) \tag{1}$$

$$g_i = \sum_{j=1}^{L} \alpha_{i,j} h_j \tag{2}$$
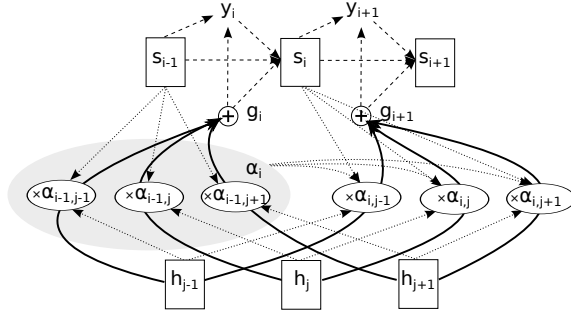
$$y_i \sim Generate(s_{i-1}, g_i), \tag{3}$$

Figure 1: Two steps of the proposed attention-based recurrent sequence generator (ARSG) with a hybrid attention mechanism (computing α), based on both content (h) and location (previous α) information. The dotted lines correspond to Eq. (1), thick solid lines to Eq. (2) and dashed lines to Eqs. (3)–(4).

where s_{i-1} is the $(i-1)$-th state of the recurrent neural network to which we refer as the *generator*, $\alpha_i \in \mathbb{R}^L$ is a vector of the *attention weights*, also often called the alignment [2]. Using the terminology from [4], we call g_i a *glimpse*. The step is completed by computing a new generator state:

$$s_i = Recurrency(s_{i-1}, g_i, y_i) \tag{4}$$

Long short-term memory units (LSTM, [11]) and gated recurrent units (GRU, [12]) are typically used as a recurrent activation, to which we refer as a *recurrency*. The process is graphically illustrated in Fig. 1.

Inspired by [6] we distinguish between location-based, content-based and hybrid attention mechanisms. *Attend* in Eq. (1) describes the most generic, hybrid attention. If the term α_{i-1} is dropped from *Attend* arguments, i.e., $\alpha_i = Attend(s_{i-1}, h)$, we call it content-based (see, e.g., [2] or [3]). In this case, *Attend* is often implemented by scoring each element in h separately and normalizing the scores:

$$e_{i,j} = Score(s_{i-1}, h_j), \tag{5}$$

$$\alpha_{i,j} = \exp(e_{i,j}) \left/ \sum_{j=1}^{L} \exp(e_{i,j}) \right. . \tag{6}$$

The main limitation of such scheme is that identical or very similar elements of h are scored equally regardless of their position in the sequence. This is the issue of "similar speech fragments" raised above. Often this issue is partially alleviated by an encoder such as e.g. a BiRNN [2] or a deep convolutional network [3] that encode contextual information into every element of h. However, capacity of h elements is always limited, and thus disambiguation by context is only possible to a limited extent.

Alternatively, a location-based attention mechanism computes the alignment from the generator state and the previous alignment only such that $\alpha_i = Attend(s_{i-1}, \alpha_{i-1})$. For instance, Graves [1] used the location-based attention mechanism using a Gaussian mixture model in his handwriting synthesis model. In the case of speech recognition, this type of location-based attention mechanism would have to predict the distance between consequent phonemes using s_{i-1} only, which we expect to be hard due to large variance of this quantity.

For these limitations associated with both content-based and location-based mechanisms, we argue that a hybrid attention mechanism is a natural candidate for speech recognition. Informally, we would like an attention model that uses the previous alignment α_{i-1} to select a short list of elements from h, from which the content-based attention, in Eqs. (5)–(6), will select the relevant ones without confusion.

2.2 Proposed Model: ARSG with Convolutional Features

We start from the ARSG-based model with the content-based attention mechanism proposed in [2]. This model can be described by Eqs. (5)–(6), where

$$e_{i,j} = w^\top \tanh(W s_{i-1} + V h_j + b). \tag{7}$$

w and b are vectors, W and V are matrices.

3

We extend this content-based attention mechanism of the original model to be location-aware by making it take into account the alignment produced at the previous step. First, we extract k vectors $f_{i,j} \in \mathbb{R}^k$ for every position j of the previous alignment α_{i-1} by convolving it with a matrix $F \in \mathbb{R}^{k \times r}$:

$$f_i = F * \alpha_{i-1}. \tag{8}$$

These additional vectors $f_{i,j}$ are then used by the scoring mechanism $e_{i,j}$:

$$e_{i,j} = w^\top \tanh(W s_{i-1} + V h_j + U f_{i,j} + b) \tag{9}$$

2.3 Score Normalization: Sharpening and Smoothing

There are three potential issues with the normalization in Eq. (6).

First, when the input sequence h is long, the glimpse g_i is likely to contain noisy information from many irrelevant feature vectors h_j, as the normalized scores $\alpha_{i,j}$ are all positive and sum to 1. This makes it difficult for the proposed ARSG to focus clearly on a few relevant frames at each time i. Second, the attention mechanism is required to consider all the L frames each time it decodes a single output y_i while decoding the output of length T, leading to a computational complexity of $O(LT)$. This may easily become prohibitively expensive, when input utterances are long (and issue that is less serious for machine translation, because in that case the input sequence is made of words, not of 20ms acoustic frames).

The other side of the coin is that the use of *softmax* normalization in Eq. (6) prefers to mostly focus on only a single feature vector h_j. This prevents the model from aggregating multiple top-scored frames to form a glimpse g_i.

Sharpening   There is a straightforward way to address the first issue of a noisy glimpse by "sharpening" the scores $\alpha_{i,j}$. One way to sharpen the weights is to introduce an *inverse temperature* $\beta > 1$ to the softmax function such that $a_{i,j} = \exp(\beta e_{i,j}) \big/ \sum_{j=1}^{L} \exp(\beta e_{i,j})$, or to keep only the top-k frames according to the scores and re-normalize them. These sharpening methods, however, still requires us to compute the score of every frame each time ($O(LT)$), and they worsen the second issue, of overly narrow focus.

We also propose and investigate a *windowing* technique. At each time i, the attention mechanism considers only a subsequence $\tilde{h} = (h_{p_i-w}, \ldots, h_{p_i+w-1})$ of the whole sequence h, where $w \ll L$ is the predefined window width and p_i is the median of the alignment α_{i-1}. The scores for $h_j \notin \tilde{h}$ are not computed, resulting in a lower complexity of $O(L + T)$. This windowing technique is similar to taking the top-k frames, and similarly, has the effect of sharpening.

The proposed sharpening based on windowing can be used both during training and evaluation. Later, in the experiments, we only consider the case where it is used during evaluation.

Smoothing   We observed that the proposed sharpening methods indeed helped with long utterances. However, all of them, and especially selecting the frame with the highest score, negatively affected the model's performance on the standard development set which mostly consists of short utterances. This observations let us hypothesize that it is helpful for the model to aggregate selections from multiple top-scored frames. In a sense this brings more diversity, i.e., more effective training examples, to the output part of the model, as more input locations are considered. To facilitate this effect, we replace the unbounded exponential function of the softmax function in Eq. (6) with the bounded logistic sigmoid σ such that $a_{i,j} = \sigma(e_{i,j}) \big/ \sum_{j=1}^{L} \sigma(e_{i,j})$. This has the effect of *smoothing* the focus found by the attention mechanism.

3   Related Work

Speech recognizers based on the connectionist temporal classification (CTC, [13]) and its extension, RNN Transducer [14], are the closest to the ARSG model considered in this paper. They follow earlier work on end-to-end trainable deep learning over sequences with gradient signals flowing
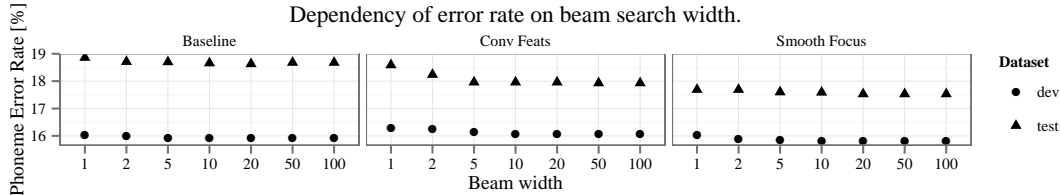
Figure 2: Decoding performance w.r.t. the beam size. For rigorous comparison, if decoding failed to generate ⟨eos⟩, we considered it wrongly recognized without retrying with a larger beams size. The models, especially with smooth focus, perform well even with a beam width as small as 1.

through the alignment process [15]. They have been shown to perform well on the phoneme recognition task [16]. Furthermore, the CTC was recently found to be able to directly transcribe text from speech without any intermediate phonetic representation [17].

The considered ARSG is different from both the CTC and RNN Transducer in two ways. First, whereas the attention mechanism deterministically aligns the input and the output sequences, the CTC and RNN Transducer treat the alignment as a latent random variable over which MAP (maximum a posteriori) inference is performed. This deterministic nature of the ARSG's alignment mechanism allows beam search procedure to be simpler. Furthermore, we empirically observe that a much smaller beam width can be used with the deterministic mechanism, which allows faster decoding (see Sec. 4 and Fig. 2). Second, the alignment mechanism of both the CTC and RNN Transducer is constrained to be "monotonic" to keep marginalization of the alignment tractable. On the other hand, the proposed attention mechanism can result in non-monotonic alignment, which makes it suitable for a larger variety of tasks other than speech recognition.

A hybrid attention model using a convolution operation was also proposed in [6] for neural Turing machines (NTM). At each time step, the NTM computes content-based attention weights which are then convolved with a predicted shifting distribution. Unlike the NTM's approach, the hybrid mechanism proposed here lets learning figure out how the content-based and location-based addressing be combined by a deep, parametric function (see Eq. (9).)

Sukhbaatar et al. [18] describes a similar hybrid attention mechanism, where location embeddings are used as input to the attention model. This approach has an important disadvantage that the model cannot work with an input sequence longer than those seen during training. Our approach, on the other hand, works well on sequences many times longer than those seen during training (see Sec. 5.)

4 Experimental Setup

We closely followed the procedure in [16]. All experiments were performed on the TIMIT corpus [19]. We used the train-dev-test split from the Kaldi [20] TIMIT s5 recipe. We trained on the standard 462 speaker set with all SA utterances removed and used the 50 speaker dev set for early stopping. We tested on the 24 speaker core test set. All networks were trained on 40 mel-scale filterbank features together with the energy in each frame, and first and second temporal differences, yielding in total 123 features per frame. Each feature was rescaled to have zero mean and unit variance over the training set. Networks were trained on the full 61-phone set extended with an extra "end-of-sequence" token that was appended to each target sequence. Similarly, we appended an all-zero frame at the end of each input sequence to indicate the end of the utterance. Decoding was performed using the 61+1 phoneme set, while scoring was done on the 39 phoneme set.

Training Procedure One property of ARSG models is that different subsets of parameters are reused different number of times; L times for those of the encoder, LT for the attention weights and T times for all the other parameters of the ARSG. This makes the scales of derivatives w.r.t. parameters vary significantly. We used an adaptive learning rate algorithm, AdaDelta [21] which has two hyperparameters ϵ and ρ. All the weight matrices were initialized from a normal Gaussian distribution with its standard deviation set to 0.01. Recurrent weights were orthogonalized.

As TIMIT is a relatively small dataset, proper regularization is crucial. We used the adaptive weight noise as a main regularizer [22]. We first trained our models with a column norm constraint [23] with
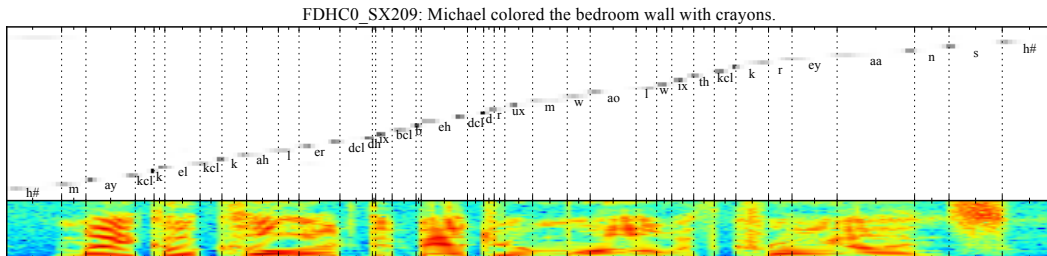
Figure 3: Alignments produced by the baseline model. The vertical bars indicate ground truth phone location from TIMIT. Each row of the upper image indicates frames selected by the attention mechanism to emit a phone symbol. The network has clearly learned to produce a left-to-right alignment with a tendency to look slightly ahead, and does not confuse between the repeated "kcl-k" phrase. Best viewed in color.

the maximum norm 1 until the lowest development negative log-likelihood is achieved.[3] During this time, ϵ and ρ are set to 10^{-8} and 0.95, respectively. At this point, we began using the adaptive weight noise, with the model complexity cost L_C divided by 10, while disabling the column norm constraints. Once the new lowest development log-likelihood was reached, we fine-tuned the model with a smaller $\epsilon = 10^{-10}$, until we did not observe the improvement in the development phoneme error rate (PER) for 100K weight updates. Batch size 1 was used throughout the training.

Evaluated Models   We evaluated the ARSGs with different attention mechanisms. The encoder was a 3-layer BiRNN with 256 GRU units in each direction, and the activations of the 512 top-layer units were used as the representation h. The generator had a single recurrent layer of 256 GRU units. *Generate* in Eq. (3) had a hidden layer of 64 maxout units. The initial states of both the encoder and generator were treated as additional parameters.

Our baseline model is the one with a purely content-based attention mechanism (See Eqs. (5)–(7).) The scoring network in Eq. (7) had 512 hidden units. The other two models use the convolutional features in Eq. (8) with $k = 10$ and $r = 201$. One of them uses the smoothing from Sec. 2.3.

Decoding Procedure   A left-to-right beam search over phoneme sequences was used during decoding [24]. Beam search was stopped when the "end-of-sequence" token ⟨eos⟩ was emitted. We started with a beam width of 10, increasing it up to 40 when the network failed to produce ⟨eos⟩ with the narrower beam. As shown in Fig. 2, decoding with a wider beam gives little-to-none benefit.

5   Results

Table 1: Phoneme error rates (PER). The bold-faced PER corresponds to the best error rate with an attention-based recurrent sequence generator (ARSG) incorporating convolutional attention features and a smooth focus.

Model	Dev	Test
Baseline Model	15.9%	18.7%
Baseline + Conv. Features	16.1%	18.0%
Baseline + Conv. Features + Smooth Focus	15.8%	**17.6%**
RNN Transducer [16]	N/A	17.7%
HMM over Time and Frequency Convolutional Net [25]	13.9%	16.7%

All the models achieved competitive PERs (see Table 1). With the convolutional features, we see 3.7% relative improvement over the baseline and further 5.9% with the smoothing.

To our surprise (see Sec. 2.1.), the baseline model learned to align properly. An alignment produced by the baseline model on a sequence with repeated phonemes (utterance FDHC0_SX209) is presented in Fig. 3 which demonstrates that the baseline model is not confused by short-range repetitions. We can also see from the figure that it prefers to select frames that are near the beginning or

---

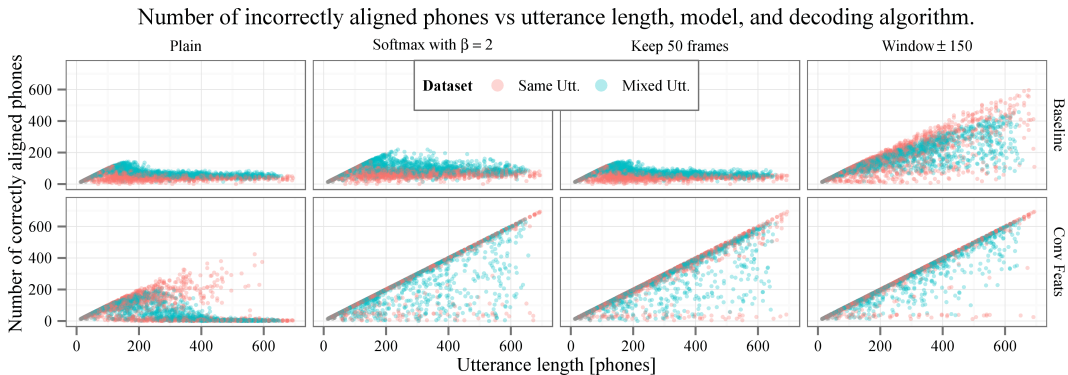[3] Applying the weight noise from the beginning of training caused severe underfitting.

Figure 4: Results of force-aligning the concatenated utterances. Each dot represents a single utterance created by either concatenating multiple copies of the same utterance, or of different, randomly chosen utterances. We clearly see that the highest robustness is achieved when the hybrid attention mechanism is combined with the proposed sharpening technique (see the bottom-right plot.)

even slightly before the phoneme location provided as a part of the dataset. The alignments produced by the other models were very similar visually.

5.1 Forced Alignment of Long Utterances

The good performance of the baseline model led us to the question of how it distinguishes between repetitions of similar phoneme sequences and how reliably it decodes longer sequences with more repetitions. We created two datasets of long utterances; one by repeating each test utterance, and the other by concatenating randomly chosen utterances. In both cases, the waveforms were cross-faded with a 0.05s silence inserted as the "pau" phone. We concatenated up to 15 utterances.

First, we checked the forced alignment with these longer utterances by forcing the generator to emit the correct phonemes. Each alignment was considered correct if 90% of the alignment weight lies inside the ground-truth phoneme window extended by 20 frames on each side. Under this definition, all phones but the ⟨eos⟩ shown in Fig. 3 are properly aligned.

The first column of Fig. 4 shows the number of correctly aligned frames w.r.t. the utterance length (in frames) for some of the considered models. One can see that the baseline model was able to decode sequences up to about 120 phones when a single utterance was repeated, and up to about 150 phones when different utterances were concatenated. Even when it failed, it correctly aligned about 50 phones. On the other hand, the model with the hybrid attention mechanism with convolutional features was able to align sequences up to 200 phones long. However, once it began to fail, the model was not able to align almost all phones. The model with the smoothing behaved similarly to the one with convolutional features only.

We examined failed alignments to understand these two different modes of failure. Some of the examples are shown in the Supplementary Materials.

We found that the baseline model properly aligns about 40 first phones, then makes a jump to the end of the recording and cycles over the last 10 phones. This behavior suggests that it learned to track its approximate location in the source sequence. However, the tracking capability is limited to the lengths observed during training. Once the tracker saturates, it jumps to the end of the recording.

In contrast, when the location-aware network failed it just stopped aligning – no particular frames were selected for each phone. We attribute this behavior to the issue of noisy glimpse discussed in Sec. 2.3. With a long utterance there are many irrelevant frames negatively affecting the weight assigned to the correct frames. In line with this conjecture, the location-aware network works slightly better on the repetition of the same utterance, where all frames are somehow relevant, than on the concatenation of different utterances, where each misaligned frame is irrelevant.

To gain more insight we applied the alignment sharpening schemes described in Sec. 2.3. In the remaining columns of Fig. 4, we see that the sharpening methods help the location-aware network to find proper alignments, while they show little effect on the baseline network. The windowing
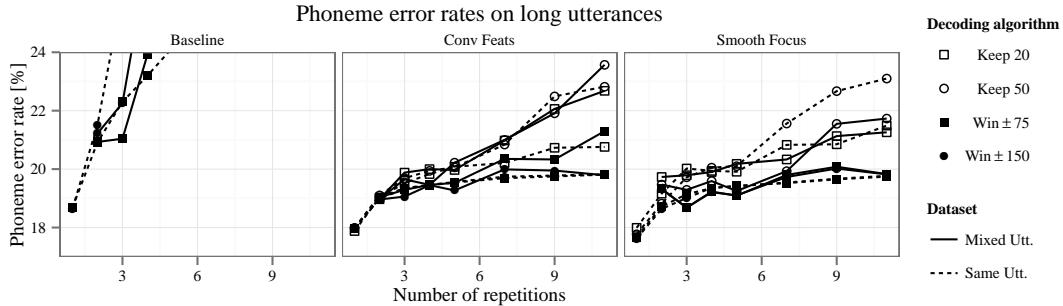
Figure 5: Phoneme error rates obtained on decoding long sequences. Each network was decoded with alignment sharpening techniques that produced proper forced alignments. The proposed ARSG's are clearly more robust to the length of the utterances than the baseline one is.

technique helps both the baseline and location-aware networks, with the location-aware network properly aligning nearly all sequences.

During visual inspection, we noticed that in the middle of very long utterances the baseline model was confused by repetitions of similar content within the window, and that such confusions did not happen in the beginning. This supports our conjecture above.

5.2 Decoding Long Utterances

We evaluated the models on long sequences. Each model was decoded using the alignment sharpening techniques that helped to obtain proper forced alignments. The results are presented in Fig. 5. The baseline model fails to decode long utterances, even when a narrow window is used to constrain the alignments it produces. The two other location-aware networks are able to decode utterances formed by concatenating up to 11 test utterances. Better results were obtained with a wider window, presumably because it resembles more the training conditions when at each step the attention mechanism was seeing the whole input sequence. With the wide window, both of the networks scored about 20% PER on the long utterances, indicating that the proposed location-aware attention mechanism can scale to sequences much longer than those in the training set with only minor modifications required at the decoding stage.

6 Conclusions

We proposed and evaluated a novel end-to-end trainable speech recognition architecture based on a hybrid attention mechanism which combines both content and location information in order to select the next position in the input sequence for decoding. One desirable property of the proposed model is that it can recognize utterances much longer than the ones it was trained on. In the future, we expect this model to be used to directly recognize text from speech [10, 17], in which case it may become important to incorporate a monolingual language model to the ARSG architecture [26].

This work has contributed two novel ideas for attention mechanisms: a better normalization approach yielding smoother alignments and a generic principle for extracting and using features from the previous alignments. Both of these can potentially be applied beyond speech recognition. For instance, the proposed attention can be used without modification in neural Turing machines, or by using 2–D convolution instead of 1–D, for improving image caption generation [3].

References

[1] A. Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, August 2013.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. of the 3rd ICLR*, 2015. arXiv:1409.0473.

[3] K. Xu, J. Ba, R. Kiros, et al. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of the 32nd ICML*, 2015. arXiv:1502.03044.

[4] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Proc. of the 27th NIPS*, 2014. arXiv:1406.6247.

[5] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. End-to-end continuous speech recognition using attention-based recurrent NN: First results. *arXiv:1412.1602 [cs, stat]*, December 2014.

[6] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv:1410.5401*, 2014.

[7] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv:1410.3916*, 2014.

[8] M. Gales and S. Young. The application of hidden markov models in speech recognition. *Found. Trends Signal Process.*, 1(3):195–304, January 2007.

[9] G. Hinton, L. Deng, D. Yu, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, November 2012.

[10] A. Hannun, C. Case, J. Casper, et al. Deepspeech: Scaling up end-to-end speech recognition. *arXiv:1412.5567*, 2014.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural. Comput.*, 9(8):1735–1780, 1997.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, October 2014. to appear.

[13] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. of the 23rd ICML-06*, 2006.

[14] A. Graves. Sequence transduction with recurrent neural networks. In *Proc. of the 29th ICML*, 2012.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. *Proc. IEEE*, 1998.

[16] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP 2013*, pages 6645–6649. IEEE, 2013.

[17] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. of the 31st ICML*, 2014.

[18] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. Weakly supervised memory networks. *arXiv:1503.08895*, 2015.

[19] J. S. Garofolo, L. F. Lamel, W. M. Fisher, et al. DARPA TIMIT acoustic phonetic continuous speech corpus, 1993.

[20] D. Povey, A. Ghoshal, G. Boulianne, et al. The kaldi speech recognition toolkit. In *Proc. ASRU*, 2011.

[21] M. D. Zeiler. ADADELTA: An adaptive learning rate method. *arXiv:1212.5701*, 2012.

[22] A. Graves. Practical variational inference for neural networks. In *Proc of the 24th NIPS*, 2011.

[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[24] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proc. of the 27th NIPS*, 2014. arXiv:1409.3215.

[25] L. Tóth. Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. In *Proc. ICASSP*, 2014.

[26] C. Gulcehre, O. Firat, K. Xu, et al. On using monolingual corpora in neural machine translation. *arXiv:1503.03535*, 2015.

[27] J. Bergstra, O. Breuleux, F. Bastien, et al. Theano: a CPU and GPU math expression compiler. In *Proc. SciPy*, 2010.

[28] F. Bastien, P. Lamblin, R. Pascanu, et al. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[29] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, et al. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013.

[30] B. van Merriënboer, D. Bahdanau, V. Dumoulin, et al. Blocks and fuel: Frameworks for deep learning. *arXiv:1506.00619 [cs, stat]*, June 2015.