Local Monotonic Attention Mechanism
for End-to-End Speech and Language Processing

Andros Tjandra, Sakriani Sakti, and **Satoshi Nakamura**
Graduate School of Information Science
Nara Institute of Science and Technology, Japan
{andros.tjandra.ai6, ssakti, s-nakamura}@is.naist.jp

Abstract

Recently, encoder-decoder neural networks have shown impressive performance on many sequence-related tasks. The architecture commonly uses an attentional mechanism which allows the model to learn alignments between the source and the target sequence. Most attentional mechanisms used today is based on a global attention property which requires a computation of a weighted summarization of the whole input sequence generated by encoder states. However, it is computationally expensive and often produces misalignment on the longer input sequence. Furthermore, it does not fit with monotonous or left-to-right nature in several tasks, such as automatic speech recognition (ASR), grapheme-to-phoneme (G2P), etc. In this paper, we propose a novel attention mechanism that has local and monotonic properties. Various ways to control those properties are also explored. Experimental results on ASR, G2P and machine translation between two languages with similar sentence structures, demonstrate that the proposed encoder-decoder model with local monotonic attention could achieve significant performance improvements and reduce the computational complexity in comparison with the one that used the standard global attention architecture.

1 Introduction

End-to-end training is a newly emerging approach to sequence-to-sequence mapping tasks, that allows the model to directly learn the mapping between variable-length representation of different modalities (i.e., text-to-text sequence (Bahdanau et al., 2014; Sutskever et al., 2014), speech-to-text sequence (Chorowski et al., 2014; Chan et al., 2016), image-to-text sequence (Xu et al., 2015), etc).

One popular approaches in the end-to-end mapping tasks of different modalities is based on encoder-decoder architecture. The earlier version of an encoder-decoder model is built with only two different components (Sutskever et al., 2014; Cho et al., 2014b): (1) an encoder that processes the source sequence and encodes them into a fixed-length vector; and (2) a decoder that produces the target sequence based on information from fixed-length vector given by encoder. Both the encoder and decoder are jointly trained to maximize the probability of a correct target sequence given a source sequence. This architecture has been applied in many applications such as machine translation (Sutskever et al., 2014; Cho et al., 2014b), image captioning (Karpathy and Fei-Fei, 2015), and so on.

However, such architecture encounters difficulties, especially for coping with long sequences. Because in order to generate the correct target sequence, the decoder solely depends only on the last hidden state of the encoder. In other words, the network needs to compress all of the information contained in the source sequence into a single fixed-length vector. (Cho et al., 2014a) demonstrated a decrease in the performance of the encoder-decoder model associated with an increase in the length of the input sentence sequence. Therefore, (Bahdanau et al., 2014) introduced attention mechanism to address these issues. Instead of relying on a fixed-length vector, the decoder is assisted by the attention module to get the related context from the encoder sides, depends on the current decoder states.

Most attention-based encoder-decoder model

used today has a "global" property ([Bahdanau et al., 2014](); [Luong et al., 2015]()). Every time the decoder needs to predict the output given the previous output, it must compute a weighted summarization of the whole input sequence generated by the encoder states. This global property allows the decoder to address any parts of the source sequence at each step of the output generation and provides advantages in some cases like machine translation tasks. Specifically, when the source and the target languages have different sentence structures and the last part of the target sequence may depend on the first part of the source sequence. However, although the global attention mechanism has often improved performance in some tasks, it is very computationally expensive. For a case that requires mapping between long sequences, misalignments might happen in standard attention mechanism ([Kim et al., 2017]()). Furthermore, it does not fit with monotonous or left-to-right natures in several tasks, such as ASR, G2P, etc.

In this paper, we propose a novel attention module that has two important characteristics to address those problems: local and monotonicity properties. The local property helps our attention module focus on certain parts from the source sequence that the decoder wants to transcribe, and the monotonicity property strictly generates alignment left-to-right from beginning to the end of the source sequence. In case of speech recognition task that need to produces a transcription given the speech signal, the attention module is now able to focus on the audio's specific timing and always move in one direction from the start to the end of the audio. Similar way can be applied also for G2P or machine translation (MT) between two languages with similar sentences structure, i.e., Subject-Verb-Object (SVO) word order in English and French languages. Experimental results demonstrate that the proposed encoder-decoder model with local monotonic attention could achieve significant performance improvements and reduce the computational complexity in comparison with the one that used the standard global attention architecture.

2 Attention-based Encoder Decoder Neural Network

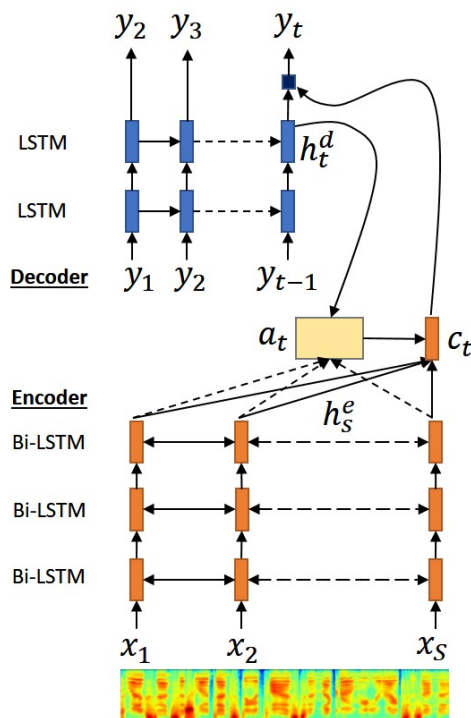The encoder-decoder model is a neural network that directly models conditional probability

Figure 1: Attention-based encoder-decoder architecture.

$p(\mathbf{y}|\mathbf{x})$, where $\mathbf{x} = [x_1, ..., x_S]$ is the source sequence with length S and $\mathbf{y} = [y_1, ..., y_T]$ is the target sequence with length T. Figure 1 shows the overall structure of the attention-based encoder-decoder model that consists of encoder, decoder and attention modules.

The encoder task processes input sequence $\mathbf{x}$ and outputs representative information $\mathbf{h}^e = [h_1^e, ..., h_S^e]$ for the decoder. The attention module is an extension scheme for assisting the decoder to find relevant information on the encoder side based on the current decoder hidden states ([Bahdanau et al., 2014](); [Luong et al., 2015]()). Usually, attention modules produces context information c_t at the time t based on the encoder and decoder hidden states:

$$c_t = \sum_{s=1}^{S} a_t(s) * h_s^e \qquad (1)$$

$$a_t(s) = \text{Align}(h_s^e, h_t^d)$$

$$= \frac{\exp(\text{Score}(h_s^e, h_t^d))}{\sum_{s=1}^{S} \exp(\text{Score}(h_s^e, h_t^d))} \qquad (2)$$

432

There are several variations for score functions:

$$\text{Score}(h_s^e, h_t^d) = \begin{cases} \langle h_s^e, h_t^d \rangle, & \text{dot product} \\ h_s^{e\mathsf{T}} W_s h_t^d, & \text{bilinear} \\ V_s^{\mathsf{T}} \tanh(W_s[h_s^e, h_t^d]), & \text{MLP} \end{cases}$$
$$(3)$$

where $\text{Score} : (\mathbb{R}^M \times \mathbb{R}^N) \to \mathbb{R}$, M is the number of hidden units for encoder and N is the number of hidden units for decoder. Finally, the decoder task, which predicts the target sequence probability at time t based on previous output and context information c_t can be formulated:

$$\log p(\mathbf{y}|\mathbf{x}) = \sum_{t=1}^{T} \log p(y_t|y_{<t}, c_t) \quad (4)$$

For speech recognition task, most common input $\mathbf{x}$ is a sequence of feature vectors like Mel-spectral filterbank and/or MFCC. Therefore, $\mathbf{x} \in \mathbb{R}^{S \times D}$ where D is the number of features and S is the total frame length for an utterance. Output $\mathbf{y}$, which is a speech transcription sequence, can be either phoneme or grapheme (character) sequence. In text-related task such as machine translation, $\mathbf{x}$ and $\mathbf{y}$ are a sequence of word or character indexes.

3 Locality and Monotonicity Properties

In the previous section, we explained the standard global attention-based encoder-decoder model. However, in order to control the area and focus attention given previous information, such mechanism requires to apply the scoring function into all the encoder states and normalizes them with a softmax function. Another problem is we cannot explicitly enforce the probability mass generated by the current attention modules that are always moving incrementally to the end of the source sequence. In this section, we discuss and explain how to model the locality and monotonicity properties on the attention module. This way, we could improve the sensitivity of capturing regularities and ensure to focus only an important subset instead of whole sequence.

Figure 2 illustrates the overall mechanism of our proposed local monotonic attention, and details are described blow.

1. **Monotonicity-based Prediction of Central Position**
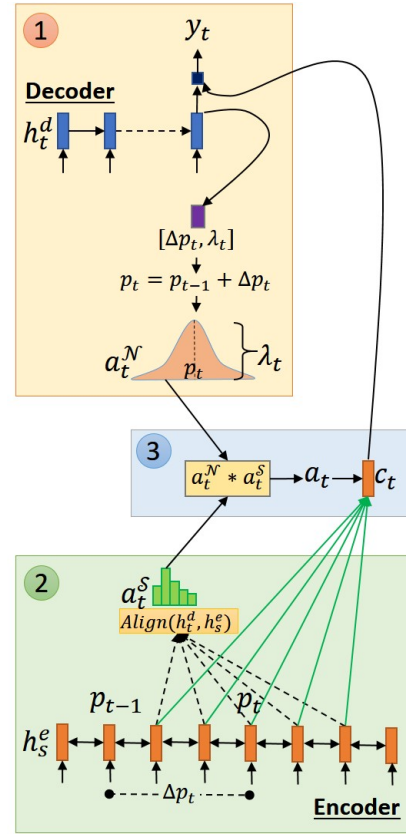   First, we define how to predict the next central position of the alignment illustrated in

Figure 2: Local monotonic attention.

Part (1) of Figure 2. Assume we have source sequence with length S, which is encoded by the stack of Bi-LSTM (see Figure 1) into S encoded states $\mathbf{h^e} = [h_1^e, ..., h_S^e]$. At time t, we want to decode the t-th target output given the source sequence, previous output y_{t-1}, and current decoder hidden states $h_t^d \in \mathbb{R}^N$. In standard approaches, we use hidden states h_t^d to predict the position difference Δp_t with a multilayer perceptron (MLP). We use variable Δp_t to determine how far we should move the center of the alignment compared to previous center p_{t-1}.

In this paper, we propose two different formulations for estimating Δp_t to ensure a forward or monotonicity movement:

- **Constrained position prediction:**
  We limit maximum range from Δp_t with hyperparameter C_{max} with the following equation:

  $$\Delta p_t = C_{max} * \text{sigmoid}(V_p^{\mathsf{T}} \tanh(W_p h_t^d)) \quad (5)$$

  Here we can control how far our next center of alignment position p_t

433

relies on our datasets and guarantee $0 \leq \Delta p_t \leq C_{max}$. However, it requires us to handle hyperparameter C_{max}.

- **Unconstrained position prediction:**
  Compared to a previous formulation, since we do not limit the maximum range of Δp_t, here we can ignore hyperparameter C_{max} and use exponential (exp) function instead of sigmoid. We can also use another function (e.g softplus) as long as the function satisfy $f : \mathbb{R} \rightarrow \mathbb{R}_0^+$ and the result of $\Delta p_t \geq 0$. We formulate unconstrained position prediction with following equation:

$$\Delta p_t = \exp(V_p^\intercal \tanh(W_p h_t^d)) \quad (6)$$

Here $V_p \in \mathbb{R}^{K \times 1}$, $W_p \in \mathbb{R}^{K \times N}$, N is the number of decoder hidden units and K is the number of hidden projection layer units. We omit the bias for simplicity. Both equations guarantee monotonicity properties since $\forall t \in [1..T], p_t \geq (p_{t-1} + \Delta p_t)$.

Additionally, we also used scaling variable λ_t to scale the unnormalized Gaussian distribution that depends on h_t. We calculated λ_t with following equation:

$$\lambda_t = \exp(V_\lambda^\intercal \tanh(W_p h_t^d)) \quad (7)$$

where $V_\lambda \in \mathbb{R}^{K \times 1}$. In our initial experiments, we discovered that we improved our model performance by scaling with λ_t for each time-step. The main objective of this step is to generate a scaled Gaussian distribution $a_t^\mathcal{N}$:

$$a_t^\mathcal{N}(s) = \lambda_t * \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right). \quad (8)$$

where p_t is the mean and σ is the standard deviation, both of which are used to calculate the weighted sum from the encoder states to generate context vector c_t later. In this paper, we treat σ as a hyperparameter.

2. **Locality-based Alignment Generation**
   After calculating new position p_t, we generate locality-based alignment, as shown in Part (2) of Figure 2. Based on predicted position p_t, we follow (Luong et al., 2015) to generate

alignment $a_t^\mathcal{S}$ only within $[p_t - 2\sigma, p_t + 2\sigma]$:

$$a_t^\mathcal{S}(s) = \text{Align}(h_s^e, h_t^d), \quad (9)$$
$$\forall s \in [p_t - 2\sigma, p_t + 2\sigma].$$

Since p_t is a real number and the indexes for the encoder states are integers, we convert p_t into an integer with floor operation. After we know the center of the position p_t, we only need to calculate the scores (Eq. 3) for each encoder states in $[p_t - 2\sigma, .., p_t + 2\sigma]$ then calculate the context alignment scores (Eq. 2).

Compared to the standard global attention, we can reduce the decoding computational complexity $O(T * S)$ into $O(T * \sigma)$ where $\sigma \ll S$ and σ is constant, T is total decoding step, S is the length of the encoder states.

3. **Context Calculation**
   In the last step, we calculate context c_t with alignments $a_t^\mathcal{N}$ and $a_t^\mathcal{S}$, as shown in Part (3) of Figure 2:

$$c_t = \sum_{s=(p_t - 2\sigma)}^{(p_t + 2\sigma)} \left(a_t^\mathcal{N}(s) * a_t^\mathcal{S}(s)\right) * h_s^e \quad (10)$$

Context c_t and current hidden state h_t^d will later be utilized for calculating current output y_t.

Overall, we can rephrase the first step as generating "prior" probabilities $a_t^\mathcal{N}$ based on the previous p_{t-1} position and the current decoder states. Then the second step task generates "likelihood" probabilities $a_t^\mathcal{S}$ by measuring the relevance of our encoder states with the current decoder states. In the third step, we combine our "prior" and "likelihood" probability into an unnormalized "posterior" probability a_t and calculate expected context c_t.

4 Experiment on Speech Recognition

We applied our proposed architecture on ASR task. The local property helps our attention module focus on certain parts from the speech that the decoder wants to transcribe, and the monotonicity property strictly generates alignment left-to-right from beginning to the end of the speech.

4.1 Speech Data

We conducted our experiments on the TIMIT [1] (Garofolo et al., 1993) dataset with the same set-

---

[1] https://catalog.ldc.upenn.edu/ldc93s1

up for training, development, and test sets as defined in the Kaldi s5 recipe (Povey et al., 2011). The training set contains 3696 sentences from 462 speakers. We also used another sets of 50 speakers for the development set and the test set contains 192 utterances, 8 each from 24 speakers. For every experiment, we used 40-dimensional fbank with delta and acceleration (total 120-dimension feature vector) extracted from the Kaldi toolkit. The input features were normalized by subtracting the mean and divided by the standard deviation from the training set. For our decoder target, we re-mapped the original target phoneme set from 61 into 39 phoneme class plus the end of sequence mark (eos).

4.2 Model Architectures

On the encoder sides, we projected our input features with a linear layer with 512 hidden units followed by tanh activation function. We used three bidirectional LSTMs (Bi-LSTM) for our encoder with 256 hidden units for each LSTM (total 512 hidden units for Bi-LSTM). To reduce the computational time, we used hierarchical subsampling (Graves, 2012; Bahdanau et al., 2016), applied it to the top two Bi-LSTM layers, and reduced their length by a factor of 4.

On the decoder sides, we used a 64-dimensional embedding matrix to transform the input phonemes into a continuous vector, followed by two unidirectional LSTMs with 512 hidden units. For every local monotonic model, we used an MLP with 256 hidden units to generate Δp_t and λ_t. Hyperparameter 2σ was set to 3, and C_{max} for constrained position prediction (see Eq. 5) was set to 5. Both hyperparameters were empirically selected and generally gave consistent results across various settings in our proposed model. For our scorer module, we used bilinear and MLP scorers (see Eq 3) with 256 hidden units. We used an Adam (Kingma and Ba, 2014) optimizer with a learning rate of $5e - 4$.

In the recognition phase, we generated transcriptions with best-1 (greedy) search from the decoder. We did not use any language model in this work. All of our models were implemented on the Chainer framework (Tokui et al., 2015).

For comparison, we evaluated our proposed model with the standard global attention-based encoder-decoder model and *local-m* attention (Luong et al., 2015) as the baseline. Most of the con-

figurations follow the above descriptions, except the baseline model that does not have an MLP for generating Δp_t and λ_t.

5 Result and Discussion for Speech Recognition

Table 1 summarizes our experiments on our proposed local attention models and compares them to the baseline model using several possible scenarios.

5.1 Constrained vs Unconstrained Position Prediction

Considering the use of constrained and unconstrained position prediction Δp_t, our results show that the model with the unconstrained position prediction (exp) model gives better results than one based on the constrained position prediction (sigmoid) model on both MLP and bilinear scorers. We conclude that it is more beneficial to use the unconstrained position prediction formulation since it gives better performance and we do not need to handle the additional hyperparameter C_{max}.

5.2 Alignment Scorer vs Non-Scorer

Next we investigate the importance of the scorer module by comparing the results between a model with and without it. Our results reveal that, by only relying on Gaussian alignment $a_t^{\mathcal{N}}$ and set $a_t^S = 1$, our model performance's was worse than one that used both the scorer and Gaussian alignment. This might be because the scorer modules are able to correct the details from the Gaussian alignment based on the relevance of the encoder states in the current decoder states. Thus, we conclude that alignment with the scorer is essential for our proposed models.

5.3 Overall comparison to the baseline

Overall, our proposed encoder-decoder model with local monotonic attention significantly improved the performance and reduced the computational complexity in comparison with one that used standard global attention mechanism (we cannot compare directly with (Chorowski et al., 2014) since its pretrained with HMM state alignment). We also tried *local-m* attention from (Luong et al., 2015), however our model cannot converge and we hypothesize the reason is because ratio length between the speech and their corresponding text is larger than 1, therefore the

Table 1: Results from baseline and proposed models on ASR task with TIMIT test set.

Model	Test PER (%)
Global Attention Model (Baseline)	
Att Enc-Dec (pretrained with HMM align) (Chorowski et al., 2014)	18.6
Att Enc-Dec (Pereyra et al., 2017)	23.2
Att Enc-Dec (Luo et al., 2016)	24.5
Att Enc-Dec with MLP Scorer (ours)	23.8
Att Enc-Dec with *local-m* (ours) (Luong et al., 2015)	-

Local Attention Model (Proposed)			
Monotonicity	**Locality**		
Pos Prediction Δp_t	**Alignment Score**(h_s^e, h_t^d)	**Func. Type**	**Test PER (%)**
Const ($sigmoid$)	No	-	23.2
Const ($sigmoid$)	Yes	Bilinear	21.9
Const ($sigmoid$)	Yes	MLP	21.7
Unconst (exp)	No	-	23.1
Unconst (exp)	Yes	Bilinear	**20.9**
Unconst (exp)	Yes	MLP	21.4

Δp_t cannot be represented by fixed value. The best performance achieved by our proposed model with unconstrained position prediction and bilinear scorer, and provided 12.2% relative error rate reduction to our baseline.

6 Experiment on Grapheme-to-Phoneme

We also investigated our proposed architecture on G2P conversion task. Here, the model need to generate corresponding phoneme given small segment of characters and its always moving from left to right. The local property helps our attention module focus on certain parts from the grapheme source sequence that the decoder wants to convert into phoneme, and the monotonicity property strictly generates alignment left-to-right from beginning to the end of the grapheme source sequence.

6.1 Dataset

Here, we used the CMUDict dataset[2]. It contains 113438 words for training and 12753 for testing (12000 unique words). For validation, we randomly select 3000 sentences from the training set. The evaluation metrics for this task are phoneme error rate (PER) and word error rate (WER). In the evaluation process, there are some words has multiple references (pronunciations). Therefore, we

---
[2]CMUdict: https://sourceforge.net/projects/cmusphinx/files/G2P%20Models/phonetisaurus-cmudict-split.tar.gz

select one of the references that has lowest PER between compared to our hypothesis, and if the hypothesis completely match with one of those references, then the WER is not increasing. For our encoder input, we used 26 letter (A-Z) + single quotes ('). For our decoder target, we used 39 phonemes plus the end of sequence mark (eos).

6.2 Model Architectures

On the encoder sides, the characters input were projected into 256 dims using embedding matrix. We used two bidirectional LSTMs (Bi-LSTM) for our encoder with 512 hidden units for each LSTM (total 1024 hidden units for Bi-LSTM). On the decoder sides, the phonemes input were projected into 256 dims using embedding matrix, followed by two unidirectional LSTMs with 512 hidden units. For local monotonic model, we used an MLP with 256 hidden units to generate Δp_t and λ_t. For this task, we only used the unconstrained formulation because based on previous sections, we able to achieved better performance and we didn't need to find optimal hyperparameter for C_{max}. For our scorer module, we used MLP scorer with 256 hidden units.

In the decoding phase, we used beam search strategy with beam size 3 to generate the phonemes given the character sequences. For comparison, we evaluated our model with standard global attention and *local-m* attention model (Luong et al., 2015) as the baseline.

6.3 Result Discussion

Table 2 summarizes our experiment on proposed local attention models. We compared our proposed models with several baselines from other algorithm as well. Our model significantly improving the PER and WER compared to encoder-decoder, attention-based global softmax and *local-m* attention (fixed-step size). Compared to Bi-LSTM model which was trained with explicit alignment, we achieve slightly better PER and WER with larger window size ($2\sigma = 3$).

7 Experiment on Machine Translation

We also conducted experiment on machine translation task, specifically between two languages with similar sentences structure. By using our proposed method, we able to focus only to a small related segment on the source side and the target

Table 2: Results from baseline and proposed method on G2P task with CMUDict test set

Model	PER (%)	WER (%)
Baseline		
Enc-Dec LSTM (2 lyr) (Yao and Zweig, 2015)	7.63	28.61
Bi-LSTM (3 lyr) (Yao and Zweig, 2015)	5.45	23.55
Att Enc-Dec with Global MLP Scorer (ours)	5.96	25.55
Att Enc-Dec with *local-m* (ours) (Luong et al., 2015)	5.64	24.32
Proposed		
Att Enc-Dec + Unconst (exp) ($2\sigma = 2$)	5.45	**23.15**
Att Enc-Dec + Unconst (exp) ($2\sigma = 3$)	**5.43**	23.19

generation process usually follows the source sentence structure without many reordering process.

7.1 Dataset

We used BTEC dataset (Kikui et al., 2003) and chose English-to-France and Indonesian-to-English parallel corpus. From BTEC dataset, we extracted 162318 sentences for training and 510 sentences for test data. Because there are no default development set, we randomly sampled 1000 sentences from training data for validation set. For all language pairs, we preprocessed our dataset using Moses (Koehn et al., 2007) tokenizer. For training, we replaced any word that appear less then twice with unknown (unk) symbol. In details, we keep 10105 words for French corpus, 8265 words for English corpus and 9577 words for Indonesian corpus. We only used sentence pairs where the source is no longer than 60 words in training phase.

7.2 Model Architecture

On both encoder and decoder sides, the input words were projected into 256 dims using embedding matrix. We used three Bi-LSTM for our encoder with 512 hidden units for each LSTM (total 1024 hidden unit for Bi-LSTM). For our decoder, we used three LSTM with 512 hidden units. For local monotonic model, we used an MLP with 256 hidden units to generate Δp_t and λ_t. Same as previous section, we only used the unconstrained for-

mulation for local monotonic experiment. For our scorer module, we used MLP scorer with 256 hidden units. In the decoding phase, we used beam search strategy with beam size 5 and normalized length penalty with $\alpha = 1$ (Wu et al., 2016). For comparison, we evaluate our model with standard global attention and *local-m* attention model (Luo et al., 2016) as the baseline.

Table 3: Results from baseline and proposed method on English-to-France and Indonesian-to-English translation tasks.

Model	BLEU
BTEC English to France	
Baseline	
Att Enc-Dec with Global MLP Scorer	49.0
Att Enc-Dec with *local-m* (ours) (Luong et al., 2015)	50.4
Proposed	
Att Enc-Dec + Unconst (exp) ($2\sigma = 4$)	**51.2**
Att Enc-Dec + Unconst (exp) ($2\sigma = 6$)	51.1
BTEC Indonesian to English	
Baseline	
Att Enc-Dec with Global MLP Scorer	38.2
Att Enc-Dec with *local-m* (ours) (Luong et al., 2015)	39.8
Proposed	
Att Enc-Dec + Unconst (exp) ($2\sigma = 4$)	40.9
Att Enc-Dec + Unconst (exp) ($2\sigma = 6$)	**41.8**

7.3 Result Discussion

Table 3 summarizes our experiment on proposed local attention models compared to baseline global attention model and *local-m* attention model (Luong et al., 2015). Generally, local monotonic attention had better result compared to global attention on both English-to-France and Indonesian-to-English translation task. Our proposed model were able to improve the BLEU up to 2.2 points on English-to-France and 3.6 points on Indonesian-to-English translation task compared to standard global attention. Compared to *local-m* attention with fixed step size, our proposed model able to improve the performance up to 0.8 BLEU on

English-to-France and 2.0 BLEU on Indonesian-to-English translation task.

8 Related Work

Humans do not generally process all of the information that they encounter at once. Selective attention, which is a critical property in human perception, allows attention to be focused on particular information while filtering out a range of other information. The biological structure of the eye and the eye movement mechanism is one part of visual selective attention that provides the ability to focus attention selectively on parts of the visual space to acquire information when and where it is needed (Rensink, 2000). In the case of the cocktail party effect, humans can selectively focus their attentive hearing on a single speaker among various conversation and background noise sources (Cherry, 1953).

The attention mechanism in deep learning has been studied for many years. But, only recently have attention mechanisms made their way into the sequence-to-sequence deep learning architectures that were proposed to solve machine translation tasks. Such mechanisms provide a model with the ability to jointly align and translate (Bahdanau et al., 2014). With the attention-based model, the encoder-decoder model significantly improved the performance on machine translation (Bahdanau et al., 2014; Luong et al., 2015) and has successfully been applied to ASR tasks (Chorowski et al., 2014; Chan et al., 2016).

However, as we mentioned earlier, most of those attention mechanism are based on "global" property, where the attention module tries to match the current hidden states with all the states from the encoder sides. This approach is inefficient and computationally expensive on longer source sequences. A "local attention" was recently introduced by (Luong et al., 2015) which provided the capability to only focus small subset of the encoder sides. They also proposed monotonic attention but limited to fixed step-size and not suitable for a task where the length ratio between source and target sequence is vastly different. Our proposed method are able to elevated this problem by predicting the step size dynamically instead of using fixed step size. After we constructed our proposed framework, we found work by (Raffel et al., 2017) recently that also proposed a method for producing monotonic alignment by using Bernoulli random variable to control when the alignment should stop and generate output. However, it cannot attend the source sequence outside the range between previous and current position. In contrast with our approach, we are able to control how large the area we want to attend based on the window size.

(Chorowski et al., 2014) also proposed a soft constraint to encourage monotonicity by invoking a penalty based on the current alignment and previous alignments. However, the methods still did not guarantee a monotonicity movement of the attention.

To the best of our knowledge, only few studies have explored about local and monotonicity properties on an attention-based model. This work presents a novel attention module with locality and monotonicity properties. Our proposed mechanism strictly enforces monotonicity and locality properties in their alignment by explicitly modeling them in mathematical equations. The observation on our proposed model can also possibly act as regularizer by only observed a subset of encoder states. Here, we also explore various ways to control both properties and evaluate the impact of each variations on our proposed model. Experimental results also demonstrate that the proposed encoder-decoder model with local monotonic attention could provide a better performances in comparison with the standard global attention architecture and *local-m* attention model (Luong et al., 2015).

9 Conclusion

This paper demonstrated a novel attention mechanism for encoder decoder model that ensures monotonicity and locality properties. We explored various ways to control these properties, including dynamic monotonicity-based position prediction and locality-based alignment generation. The results reveal our proposed encoder-decoder model with local monotonic attention significantly improved the performance on three different tasks and able to reduced the computational complexity more than one that used standard global attention architecture.

10 Acknowledgement

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE.

E Colin Cherry. 1953. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: First results. *arXiv preprint arXiv:1412.1602*.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathon G Fiscus, and David S Pallett. 1993. Darpa TIMIT acoustic-phonetic continous speech corpus cd-rom. NIST speech disc 1-1.1. *NASA STI/Recon technical report n*, 93.

Alex Graves. 2012. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 5–13. Springer.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.

Genichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. 2003. Creating corpora for speech-to-speech translation. In *Eighth European Conference on Speech Communication and Technology*.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4835–4839. IEEE.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Yuping Luo, Chung-Cheng Chiu, Navdeep Jaitly, and Ilya Sutskever. 2016. Learning online alignments with continuous rewards policy gradient. *arXiv preprint arXiv:1608.01281*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

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

Ronald A Rensink. 2000. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence-to-Sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

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

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2048–2057.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *Sixteenth Annual Conference of the International Speech Communication Association*.