Modeling Localness for Self-Attention Networks

Baosong Yang
University of Macau
nlp2ct.baosong@gmail.com

Zhaopeng Tu*
Tencent AI Lab
zptu@tencent.com

Derek F. Wong*
University of Macau
derekfw@umac.mo

Fandong Meng
Tencent AI Lab
fandongmeng@tencent.com

Lidia S. Chao
University of Macau
lidiasc@umac.mo

Tong Zhang
Tencent AI Lab
bradymzhang@tencent.com

Abstract

Self-attention networks have proven to be of profound value for its strength of capturing global dependencies. In this work, we propose to model localness for self-attention networks, which enhances the ability of capturing useful local context. We cast localness modeling as a learnable Gaussian bias, which indicates the central and scope of the local region to be paid more attention. The bias is then incorporated into the original attention distribution to form a revised distribution. To maintain the strength of capturing long distance dependencies and enhance the ability of capturing short-range dependencies, we only apply localness modeling to lower layers of self-attention networks. Quantitative and qualitative analyses on Chinese⇒English and English⇒German translation tasks demonstrate the effectiveness and universality of the proposed approach.

1 Introduction

Recently, a new simple architecture, the TRANSFORMER (Vaswani et al., 2017), that based solely on attention mechanisms has attracted increasing attention in machine translation community. Instead of using complex recurrent or convolutional neural networks, TRANSFORMER implements encoder and decoder as self-attention networks to draw global dependencies between input and output. By further parallel performing (multi-head) and stacking (multi-layer) attentive functions, TRANSFORMER has achieved state-of-the-art performance on various translation tasks (Shaw et al., 2018; Hassan et al., 2018).

One strong point of self-attention networks is the strength of capturing long-range dependencies by explicitly attending to all the signals. In this way, a representation is allowed to build a direct relation with another long-distance representation. Accordingly, it can serve as the role of RNN and CNN to capture both the short- and long-range relations among the representations.

Self-attention networks fully take into account all the signals with a weighted averaging operation. We argue that such operation disperses the distribution of attention, which results in overlooking the relation of neighboring signals. Recent works have shown that self-attention networks benefit from locality modeling. For example, Shaw et al. (2018) introduced relative position encoding to consider the relative distances between sequence elements, which produces substantial improvements on the translation task. Sperber et al. (2018) modeled the local information by restricting self-attention model to neighboring representations, which boosts performance on long-sequence acoustic modeling. Although not for self-attention, Luong et al. (2015) proposed a local attention model for translation task, which looks at only a subset of source words at a time. Inspired by these studies, we propose more flexible strategies for modeling localness for self-attention networks in this work.

Specifically, we cast the localness modeling as a learnable Gaussian bias, in which a central position (i.e. mean of the position) and a dynamic window (i.e. deviation of the distribution) are predicted with the intermediate representations in the self-attention network. Intuitively, the central position and the window respectively denote the center and the scope of the locality to be paid more attention. The learned Gaussian bias is then incorporated into the original attention distribution to form a revised distribution, which considers the expected local context.

Some researchers may doubt that self-attention networks augmented localness modeling focuses

* Zhaopeng Tu and Derek F. Wong are the co-corresponding authors of the paper. This work was conducted when Baosong Yang was interning at Tencent AI Lab.

leanings toward local context, which weakens its strength of capturing long-range dependencies. Our extensive analyses can dispel such doubt by showing that the potential problem is compensated by multi-layer multi-head self-attention networks. First, multi-head attention attends to local regions centered at different positions, which can constitute the complete information of an input sequence. Second, we found that self-attention models tend to capture short-range dependencies among neighboring words in lower layers, while capture long-range dependencies beyond phrase boundaries in higher layers. Accordingly, we only apply localness modeling to lower layers.

We conducted experiments on two widely-used WMT14 English⇒German and WMT17 Chinese⇒English translation tasks. The proposed approach consistently improves translation performance over the strong TRANSFORMER baseline, demonstrating its effectiveness and universality. In addition, our approach is complementary to the relative position encoding (Shaw et al., 2018), and combining them can further improve translation performance.

2 Background

Attention model has recently been a basic module of most deep learning models. The mechanism allows to dynamically select related representations as needed. In particular, it is very useful for generation models such as machine translation (Bahdanau et al., 2015; Luong et al., 2015; Yang et al., 2017) and image captioning (Xu et al., 2015).

2.1 Self-Attention Model

Recently, self-attention networks (Vaswani et al., 2017; Shaw et al., 2018; Shen et al., 2018a) have attracted increasing attention due to their flexibility in parallel computation and dependency modeling. Self-attention networks calculate attention weights between each pair of tokens in a single sequence, thus can capture long-range dependency more directly than their RNN counterpart.

Formally, given an input sequence $\mathbf{x} = \{x_1, \ldots, x_I\}$, each hidden state in the l-th layer is constructed by attending to the states in the $(l-1)$-th layer.[1] Specifically, the $(l-1)$-th layer $H^{l-1} \in \mathbb{R}^{I \times d}$ is first transformed into the queries $Q \in \mathbb{R}^{I \times d}$, the keys $K \in \mathbb{R}^{I \times d}$, and the values $V \in \mathbb{R}^{I \times d}$ with three separate weight matrices. The l-th layer is calculated as:

$$H^l = \textsc{Att}(Q, K)\, V, \quad (1)$$

where $\textsc{Att}(\cdot)$ is a dot-product attention model, defined as:

$$\textsc{Att}(Q, K) = softmax(energy) \quad (2)$$

$$energy = \frac{QK^T}{\sqrt{d}}, \quad (3)$$

where $\sqrt{d}$ is the scaling factor with d being the dimensionality of layer states.

2.2 Motivation

The self-attention network models the global dependencies without regard to their distances, by directly attending to all the positions in an input sequence (i.e. Equation 3). We argue that self-attention can be further improved by taking into account the local context. However, since the conventional self-attention models consider all of the words in a sequence, the weighted averaging inhibits the relation among the neighboring words.

From a linguistic intuition, when a word x_i is aligned to another word x_j, we also expect x_i to align mainly to the neighboring words of x_j, so as to capture the phrasal patterns that contain useful local context information. Take Figure 1 as an example, if "*Bush*" is aligned to "*held*" with high probability, we expect the self-attention model to pay more attention to the neighboring words "*a talk*". Consequently, the model is guided to capture the phrase "*held a talk*".

3 Localness Modeling

Figure 1 shows an example. We first learn a Gaussian bias, which is centered around the word "*talk*" (it is not necessary to be consistent with the original attention distribution), with a window size being 2 (in practice, it is a float number in our model). The distribution of attention is then regularized with the learned Gaussian bias to produce the final distribution, which pays more attention to the local context around the word "*talk*".

3.1 Localness Modeling as a Gaussian Bias

Specifically, a Gaussian bias G is placed to mask the logit similarity $energy$ in Equation 2, namely:

$$\textsc{Att}(Q, K) = softmax(energy + G). \quad (4)$$

The first term is the original dot product self-attention model. $G \in \mathbb{R}^{I \times I}$ is a favor alignment

[1] The first layer is the word embedding layer.

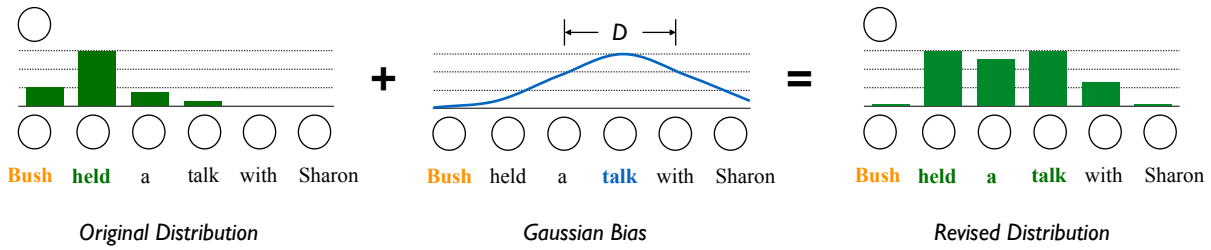

Figure 1: Illustration of the proposed approach. In this example, window size of 2 is used ($D = 2$).

position matrix (I denotes the sequence length). The element $G_{i,j} \in [0, -\infty)$ measures the tightness between the word x_j and the predicted central position P_i:

$$G_{i,j} = -\frac{(j - P_i)^2}{2{\sigma_i}^2}, \tag{5}$$

where σ_i denotes the standard deviation which is empirically set as $\sigma_i = \frac{D_i}{2}$, and D_i is a window size. Note that, due to the exponential operation in $softmax$ function, adding the logit similarity $energy$ with a bias $\in [0, -\infty)$ approximates to multiplying the attention distribution by a weight $\in [1, 0)$. The position and window size can be calculated as:

$$\begin{bmatrix} P_i \\ D_i \end{bmatrix} = I \cdot \text{sigmoid}(\begin{bmatrix} p_i \\ z_i \end{bmatrix}). \tag{6}$$

The scalar factor I is used to regulate P_i and D_i to real value numbers between 0 and the length of input sequence. The predictions are conditioned on two scalar p_i and z_i respectively.

3.2 Central Position Prediction

Since the prediction of each central position depends on its corresponding query vector,[2] we simply apply a feed-forward network to transform Q_i into a positional hidden state, which is then mapped into the scalar p_i by a linear projection $U_p \in \mathbb{R}^d$, namely:

$$p_i = {U_p}^T \tanh(W_p Q_i), \tag{7}$$

where $W_p \in \mathbb{R}^{d \times d}$ is the model parameter.

3.3 Window Size Prediction

Several alternative strategies are proposed to select the window size. Except a non-parametric approach, the other two define parametric windows. Among the parametric methods, the first strategy assigns a unified window size to all the hidden states in a layer, so as to consider the context of the sequence, while the second one calculates a distinct window size for each hidden state.

Fixed-Window A simple choice is to use a predefined window size D, which is a constant number throughout the whole training and testing process. In this study, following the common practice (Luong et al., 2015), D is set to 10.

Layer-Specific Window Furthermore, an interpretable way to select the window size is to account for the context of the sequence by summarizing the information from all the representations in a layer. In this study, we assign the mean of keys $\overline{\mathbf{K}}$ to represent the semantic context. Thus, the unified scalar z of a layer is defined as:

$$z = U_d^T \tanh(W_d \overline{\mathbf{K}}), \tag{8}$$

where $W_d \in \mathbb{R}^{d \times d}$ and $U_d \in \mathbb{R}^d$ are learnable parameters.

Query-Specific Window The last strategy provides a more flexible manner to differentiate the scope by conditioning on each query. Similar to the prediction of the central position (Equation 7), the query-specific window can be formally expressed as:

$$z_i = {U_d}^T \tanh(W_p Q_i). \tag{9}$$

Here, $U_d \in \mathbb{R}^d$ is a trainable linear projection. Note that, Equations 7 and 9 share same parameter W_p but use different U_p and U_d. The intuition behind this design is that the central position and window size interdependently locate the local scope, hence condition on the same hidden state. The distinct linear projections U_p and U_d are sufficient in distinguishing the two scalars, resulting in a smaller parameter size and faster computational speed than that of the layer-specific model.

[2] For the input of feed-forward network, we also tried an additive term to consider the weighted context O_i (Equation 1), namely: $\tanh(W_p Q_i + W_o O_i)$. Our experimental results showed that there is no progressive improvement.

3.4 Incorporating into TRANSFORMER

We evaluate our model on the advanced TRANSFORMER model (Vaswani et al., 2017), which builds an encoder-decoder framework merely using attention networks. Both the encoder and decoder are composed of a stack of $L = 6$ layers, each of which has two sub-layers. The first is a multi-head self-attention layer, and the second is a position-wise fully connected feed-forward layer. In this section, we describe how to apply our approach to TRANSFORMER by adapting to *multi-head* and *multi-layer* self-attention networks.

Adapting to Multi-Head Self-Attention Instead of performing a single attention function, the multi-head mechanism employs M separate attention models with distinct parameters to jointly attend to the information from different representation subspaces at different positions. Accordingly, we assign a distinct Gaussian bias to each attention head, and rewrite Equation 6 as:

$$\begin{bmatrix} P_i^m \\ D_i^m \end{bmatrix} = I \cdot \text{sigmoid}(\begin{bmatrix} p_i^m \\ z_i^m \end{bmatrix}), \quad (10)$$

where p_i^m and z_i^m are trained with distinct parameters to predict the central position and window size for the m-th attention head.

We argue that multi-head self-attention may benefit more from localness modeling. Multi-head attention captures different features by attending to different positions, which complements the localness modeling that may potentially ignore the global information. Experimental results in Table 5 confirm our hypothesis by showing that localness modeling achieves more significant improvement when working with multi-head attention than its single-head counterpart.

Adapting to Multi-Layer Self-Attention Recent work shows that different layers capture different types of features. Anastasopoulos and Chiang (2018) indicated that higher-level layers are more representative than lower-level layers, while Peters et al. (2018) showed that higher-level layers capture context-dependent aspects of word meaning while lower-level layers model aspects of syntax. One question naturally arises: *is it necessary to model localness for all layers?*

In this work, we investigate which levels of layers benefit most from the localness modeling. In addition, we visualize the Gaussian biases across layers, to better understand the behaviors of different attentive layers.

4 Experiments

4.1 Setup

To compare with the results reported by previous work (Gehring et al., 2017; Vaswani et al., 2017; Hassan et al., 2018), we conducted experiments on both Chinese⇒English (Zh⇒En) and English⇒German (En⇒De) translation tasks. For the Zh⇒En task, the models were trained using all of the available parallel corpus from WMT17 dataset with maximum length limited to 50, consisting of about 20.62 million sentence pairs. We used newsdev2017 as the development set and newstest2017 as the test set. For the En⇒De task, we trained on the widely-used WMT14 dataset consisting of about 4.56 million sentence pairs. The models were validated on newstest2013 and examined on newstest2014. The Chinese sentences were segmented by the word segmentation toolkit *Jieba,*[3] and the English and German sentences were tokenized using the scripts provided in Moses. Then, all tokenized sentences were processed by byte-pair encoding (BPE) to alleviate the Out-of-Vocabulary problem (Sennrich et al., 2016) with 32K merge operations for both language pairs. The 4-gram NIST BLEU score (Papineni et al., 2002) is used as the evaluation metric.

We evaluated the proposed approaches on advanced TRANSFORMER model (Vaswani et al., 2017), and implemented on top of an open-source toolkit – THUMT[4] (Zhang et al., 2017). We followed Vaswani et al. (2017) to set the configurations and reproduced their reported results on the En⇒De task. We tested both the *Base* and *Big* models, which differ at the layer size (512 vs. 1024) and the number of attention heads (8 vs. 16). All the models were trained on eight NVIDIA P40 GPUs, each of which is allocated a batch of 4096 tokens. In consideration of the computation cost, we studied the variations of the *Base* model on Zh⇒En task, and evaluated the overall performance with the *Big* model on both Zh⇒En and En⇒De translation tasks.

4.2 Ablation Study

In the first series of experiments, we evaluated the impact of different components on the Zh⇒En validation set using the TRANSFORMER-BASE. First, we investigated the effect of different strategies to predict the localness window. Then, we

[3] https://github.com/fxshy/jieba
[4] https://github.com/thumt/THUMT

Model	Speed	Dev	△
Baseline	1.20	22.59	-
Fixed	1.14	23.07	+ 0.48
Layer-Spec.	1.07	23.13	+ 0.54
Query-Spec.	1.11	23.13	+ 0.54

Table 1: Evaluation of various window prediction strategies for localness modeling, which is only applied to encoder-side self-attention network. "Speed" denotes training speed measured in steps per second.

examined whether it is necessary to apply localness modeling to all the layers. Finally, given that TRANSFORMER consists of encoder and decoder side self-attention as well as encoder-decoder attention networks, we checked which types of attention networks benefit most from the localness modeling. To eliminate the influence of control variables, we conducted the first two ablation studies on encoder-side self-attention networks only.

Window Prediction Strategies As shown in Table 1, all the proposed window prediction strategies consistently improve the model performance over the baseline, validating the importance of localness modeling in self-attention networks. Among them, layer-specific and query-specific window outperform[5] their fixed counterpart, showing the benefit that flexible mechanism is able to capture varying local context according to layer and query information. Moreover, the flexible strategy does not reply on the handcrafted parameters (e.g. the pre-defined window size), which makes model robustly applicable to other language pairs and NLP tasks. Considering the training speed, we use the query-specific prediction mechanism as the default setting in subsequent experiments.

Layers to be Applied In this experiment, we investigated the question of which layers should be applied with the localness modeling. Recent works show that different layers tend to capture different features, thus there may have different needs for the local context. We applied localness modeling to different combinations of layers, as shown in Table 2. Clearly, modeling the localness for part of the layers consistently outperforms all layers in terms of the training speed and translation quality, which again validates our claim.

Interestingly, the performance generally goes up with the increase of layers from bottom to top (Rows 2-4), while the trend does not hold when reaching the 4th-layer (Row 5). In addition, the lower three layers benefit more from the localness modeling than that of the higher three layers (Rows 4 and 6). These results reveal that lower-level layers benefit more from the local context. Accordingly, we only model the localness in the lower three layers in the following experiments.

[5]Although the differences are not always significant, the flexible strategy consistently outperforms its fixed counterpart across language pairs. For example, the query-specific strategy improves performance over the fixed-window model by +0.07 and +0.23 BLEU points on Zh-En and En-De validation sets, respectively.

#	Layers	Speed	Dev	△
1	[1-6]	1.11	23.13	-
2	[1-1]	1.18	23.20	+ 0.07
3	[1-2]	1.17	23.23	+ 0.10
4	[1-3]	1.15	23.29	+ 0.16
5	[1-4]	1.14	23.26	+ 0.13
6	[4-6]	1.15	23.22	+ 0.09

Table 2: Evaluation of different layers in the encoder, which are implemented as self-attention with query-specific localness modeling.

Enc	Dec	Enc-Dec	Speed	Dev
✓	×	×	1.15	23.29
✓	✓	×	1.10	23.27
✓	×	✓	1.08	23.33
✓	✓	✓	1.02	23.19

Table 3: Effect of localness modeling on different types of attention networks. "Enc" and "Dec" denote the encoder and decoder side self-attention networks respectively, while "Enc-Dec" represents the encoder-decoder attention network.

Attention Networks to be Applied Table 3 lists the results of localness modeling on different types of attention networks. As observed, modeling localness for decoder-side self-attention and encoder-decoder attention networks only marginally improves or even harms the translation quality. We attribute the marginal improvement of the encoder-decoder attention network to the fact that it exploits the top-layer of encoder representations, which already embeds useful local context. Concerning decoder-side self-attention network, Zhang et al. (2018) pointed out

System	Architecture	Zh⇒En		En⇒De	
		# Para.	BLEU	# Para.	BLEU
Existing NMT systems					
(Wu et al., 2016)	GNMT	n/a	n/a	n/a	26.30
(Gehring et al., 2017)	ConvS2S	n/a	n/a	n/a	26.36
(Vaswani et al., 2017)	Transformer-Base	n/a	n/a	65M	27.3
	Transformer-Big	n/a	n/a	213M	28.4
(Hassan et al., 2018)	Transformer-Big	n/a	24.2	n/a	n/a
Our NMT systems					
this work	Transformer-Base	107.9M	24.13	88.0M	27.64
	+ Rel_Pos (Shaw et al., 2018)	108.0M	24.53	88.1M	27.94
	+ Localness	108.7M	24.77$^{\Uparrow}$	88.8M	28.11$^{\uparrow}$
	+ Localness + Rel_Pos	108.8M	24.96$^{\Uparrow}$	88.9M	28.54$^{\Uparrow}$
	Transformer-Big	303.9M	24.56	264.1M	28.58
	+ Localness	307.2M	25.03$^{\uparrow}$	267.4M	28.89
	+ Localness + Rel_Pos	307.3M	25.28$^{\Uparrow}$	267.5M	29.18$^{\Uparrow}$

Table 4: Comparing with the existing NMT systems on WMT17 Zh⇒En and WMT14 En⇒De test sets. "# Para." denotes the trainable parameter size of each model (M = million). "$\uparrow$ / $\Uparrow$": significant over the conventional self-attention counterpart ($p < 0.05/0.01$), tested by bootstrap resampling (Koehn, 2004).

that it tends to only focus on its nearby representation, which poses difficulties to modeling localness on the decoder side. In the main experiments, we only applied localness modeling to the lower three layers of the encoder, which employs a query-specific window prediction strategy.

4.3 Main Results

In this section, we evaluated the proposed approach on both WMT17 Zh⇒En and WMT14 En⇒De translation tasks, as listed in Table 4. Our baseline models, both Transformer-Base and Transformer-Big, outperform the reported results on the same data, which we believe make the evaluation convincing. As seen, modeling localness ("Localness") consistently achieves improvement across language pairs and model variations, demonstrating the efficiency and universality of the proposed approach.

We also re-implemented the relative position encoding ("Rel_Pos") that recently proposed by Shaw et al. (2018), which considers the relative distances between sequence elements. Both Shaw et al. (2018) and our work have shown that explicitly modeling locality for self-attention networks can improve the model performance. This indicates that it is necessary to enhance the locality modeling for Transformer. Besides, our approach is complementary to theirs, and combining them is able to further improve the translation performance. We attribute this to the fact that the two models modeling localness from two different aspects: First, the position embeddings are the same across different positions (if the absolute positions or relative positions are the same) and training examples, our model assigns a distinct localness bias to each position from layer to layer. Second, contrast to position encoding which learns the locality through the positional information in embeddings, our model directly revises the attention distribution to focus on a local space.

5 Analysis

We conducted extensive analyses to better understand our model in terms of its compatibility with multi-head and multi-layer attention networks, as well as building the ability of capturing phrasal patterns. All the results are reported on Zh⇒En development set with Transformer-Base, unless otherwise stated.

5.1 Compatibility with Multi-Head Attention

In this section, we investigated whether multi-head attention and localness modeling are compatible from two perspectives: (1) whether multi-head attention benefits more from the localness modeling than its single-head counterpart; and (2) how does multi-head attention work together with localness modeling?

Model	1-Head		8-Head	
	Dev	△	Dev	△
BASE	22.05	-	22.59	-
OURS	22.18	+ 0.13	23.29	+ 0.70

Table 5: Evaluation of localness modeling on top of single and multiple attention heads.

Multi-Head vs. Single-Head The single-head attention and multi-head attention differ at: the former uses a single 512-dimension attention head while the latter uses eight 64-dimension heads. The results in Table 5 confirm our claim by showing that multi-head attention indeed benefits more from our model than the single-head model (+0.70 vs. +0.13). It should be noted that our model marginally improves the performance under single-head setting. One possible reason is that our model focuses more on local context thus may ignore global information, which cannot be complemented by the single-head attention.

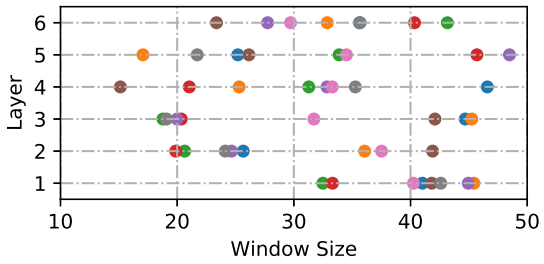

Figure 2: Instructions of the learned window size by head-specific parametric model, where colors distinguish the heads.

Can Multi-Head Separate Locality? To simplistically visualize how heads cooperate in modeling localness, we propose an additional parametric model which is assigned a learnable but unified window size for each head, namely **head-specific**. As a result, the window size D^m of the m-th head is calculated as:

$$D^m = N \cdot \text{sigmoid}(z^m), \tag{11}$$

where the scalar z^m is a trainable parameter, $N = 50$ denotes a pre-defined constant number.

Figure 2 visualizes the distribution of the learned window size of each head, verifying that multi-head attention is able to capture diverse information by selecting suitable window sizes for different heads. For example, in the middle-level layers, heads are assigned to consider both the global and local information by regulating the different window sizes. One interesting finding is that the distributions of window size are not exactly same in different layers, which is explored in more details in the next section.

5.2 Analysis on Multi-Layer Attention

In this section, we try to answer how does each layer learn the localness. We first investigated how the window size varies across layers. Then we checked the specific behavior of the first word embedding layer, which is inconsistent with the trend of other layers.

The Higher Layer, The Larger Scope Shi et al. (2016) and Vaswani et al. (2017) have shown that different layers have the abilities to distinguish and capture diverse syntactic context (e.g. the dependents between words). Figure 3 shows the distribution of local scopes predicted by each layer. Except the first layer, the higher layers are more likely to pay attention to larger scopes, indicating that self-attention models tend to capture short-term dependencies among neighboring words in lower layers, while capture long-range dependencies beyond phrase boundaries in higher layers.

The Special First Layer Inconsistent with the intuition which the lower layers may focus on local information, in common, the first layer is assigned with large scopes of local context. The same phenomenon has also occurred for head-specific model (Figure 2). Since the first layer represents word embeddings that are deficient in context, we argue that the self-attention model at first layer has to encode the representations with global context. In addition, experimental results in Table 2 (Row 2) show that despite its large local size, modeling localness at the first layer is still valid.

5.3 Analysis on Phrasal Pattern

As aforementioned, one intuition of our approach is to capture useful phrase patterns. To evaluate the accuracy of phrase translations, we calculate the improvement of the proposed approaches over multiple N-grams, as shown in Figure 4. Although our models underperform the baseline on unigram translations, they consistently outperform the baseline on larger granularities, indicating that modeling locality can raise the ability of

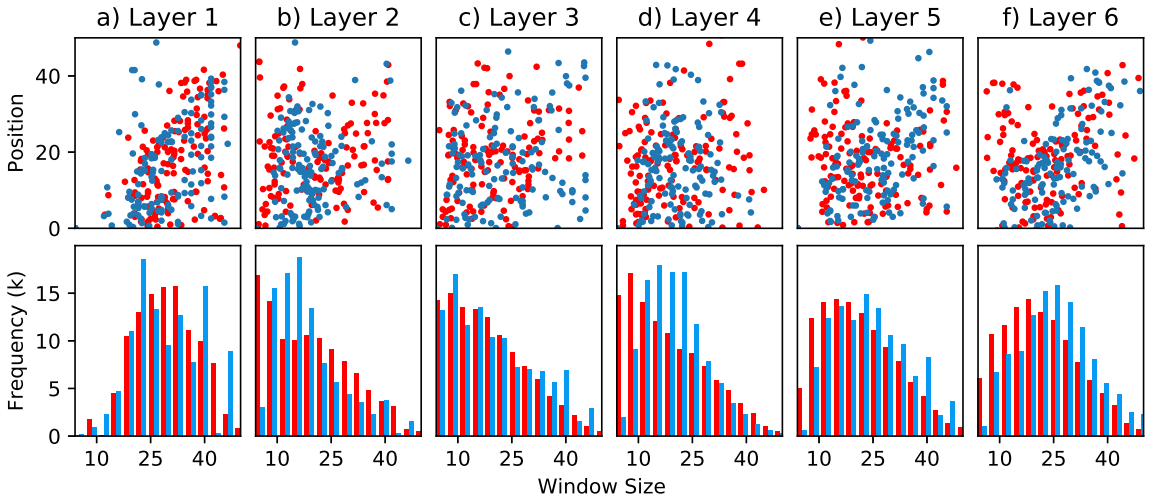

Figure 3: Distribution of the local scopes learned by each attentive layer. The upper figures illustrate the distribution of the predicted pairs of central position (Y-axis) and its correspond window size (X-axis) in each layer, the samples are randomly selected from the development set. The lower figures show the distribution of the window size in each layer. Blue color represents the layer-specific parametric approach, while the query-specific parametric method is indicated in red.

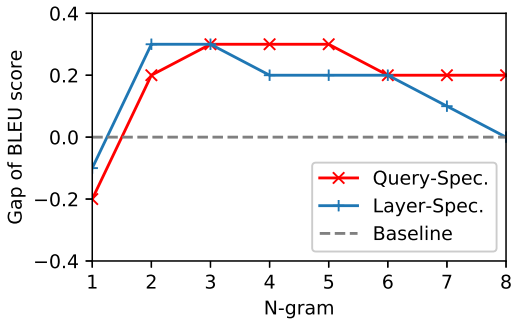

Figure 4: Performance improvement according to N-gram. Y-axis denotes the gap of BLEU score between our model and baseline.

self-attention model on capturing the phrasal information. Concerning the two variations, query-specific localness modeling surpasses its layer-specific counterpart on large phrases (i.g. 4-grams to 8-grams). We attribute this to the more modeling flexibility of query-specific strategy to differentiate the scope by conditioning on each query.

6 Related Work

A successful extension of neural language model is attention mechanism, which can directly capture long-distance dependencies by attending to previously generated words. Daniluk et al. (2017) proposed a *key-value-predict* attention to separate the key addressing, value reading, and word predicting functions explicitly. Im and Cho (2017) and Sperber et al. (2018) adopted self-attention networks for acoustic modeling and natural language inference tasks, respectively.

Vaswani et al. (2017) applied the idea of self-attention to neural machine translation. Shen et al. (2018a) and Shen et al. (2018b) proposed to improve the self-attention model with directional masks and multi-dimensional features. Although the standard self-attention model can give more bias toward localness,[6] several studies show that explicitly modeling localness for self-attention model can further improve performance. For example, Sperber et al. (2018) showed that restricting the self-attention model on the neighboring representations performs better for longer sequences in acoustic modeling and natural language inference tasks. Closely related to this work, Shaw et al. (2018) introduced relative position encoding to consider the relative distances between sequence elements. While they modeled localness from *static position embedding*, we improve locality modeling from *dynamically revising attention distribution*. Experimental results show

[6] As pointed out by one reviewer, in the original self-attention model, there are some considerations about given more bias toward the localness. For example, base on the definition of the positional embeddings, the adjacent words will have more similar positional embeddings compared with more further words. After summing word embeddings and corresponding positional embeddings together, the model would prefer the local words.

that the two models are complementary to each other, and combining them can further improve performance.

Several researches have shown that explicitly modeling phrases is useful for neural machine translation (Wang et al., 2017; Huang et al., 2018). Our results confirm these findings. Concerning attention models, Luong et al. (2015) proposed a modification to look at only a subset of input words at a time. This can be regarded as a "hard" variation of our fixed-window strategy. In this study, we propose more flexible strategies for placing and zooming the local scope, which yield better results than the fixed scope.

7 Conclusion

In this work, we enhanced the ability of capturing local context for self-attention networks with a learnable Gaussian bias. We proposed several strategies to learn the scope of the local context, and found that a query-specific mechanism yielded the best result due to its more modeling flexibility. Experimental results on widely-used English⇒German and Chinese⇒English translation tasks demonstrate the effectiveness and universality of the proposed approach. By visualizing the scopes of the learned Gaussian biases, we found that the higher the layer, the larger scope the bias, which is consistent with the findings in previous work (Shi et al., 2016; Peters et al., 2018).

As our approach is not limited to specific tasks, it is interesting to validate our model in other tasks, such as reading comprehension, language inference, and stance classification (Xu et al., 2018). Another promising direction is to design more powerful localness modeling techniques, such as incorporating linguistic knowledge (e.g. phrases and syntactic categories). It is also interesting to combine with other techniques (Shaw et al., 2018; Shen et al., 2018a; Dou et al., 2018; Li et al., 2018) to further improve the performance of Transformer.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61672555), the Joint Project of Macao Science and Technology Development Fund and National Natural Science Foundation of China (Grant No. 045/2017/AFJ) and the Multiyear Research Grant from the University of Macau (Grant No. MYRG2017-00087-FST). We thank the anonymous reviewers for their insightful comments.

References

Antonios Anastasopoulos and David Chiang. 2018. Tied Multitask Learning for Neural Speech Translation. In *NAACL*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

Michał Daniluk, Tim Rocktäschel, Johannes Welbl, and Sebastian Riedel. 2017. Frustratingly short attention spans in neural language modeling. In *ICLR*.

Ziyi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting Deep Representations for Neural Machine Translation. In *EMNLP*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *ICML*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *arXiv:1803.05567*.

Po Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. 2018. Towards Neural Phrase-based Machine Translation. In *ICLR*.

Jinbae Im and Sungzoon Cho. 2017. Distance-based Self-Attention Network for Natural Language Inference. *arXiv:1712.02047*.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP*.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-Head Attention with Disagreement Regularization. In *EMNLP*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *ACL*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *NAACL*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018a. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. In *AAAI*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2018b. Bi-Directional Block Self-Attention for Fast and Memory-Efficient Sequence Modeling. In *ICLR*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does String-based Neural MT Learn Source Syntax? In *EMNLP*.

Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. 2018. Self-Attentional Acoustic Models. In *Interspeech*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NIPS*.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017. Translating Phrases in Neural Machine Translation. In *EMNLP*.

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:1609.08144*.

Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. 2018. Cross-Target Stance Classification with Self-Attention Networks. In *ACL*.

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 *ICML*.

Baosong Yang, Derek F Wong, Tong Xiao, Lidia S Chao, and Jingbo Zhu. 2017. Towards Bidirectional Hierarchical Representations for Attention-based Neural Machine Translation. In *EMNLP*.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating Neural Transformer via an Average Attention Network. In *ACL*.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, and Yang Liu. 2017. THUMT: An Open Source Toolkit for Neural Machine Translation. *arXiv:1706.06415*.