Towards Better Modeling Hierarchical Structure for Self-Attention with Ordered Neurons

Jie Hao[*]
Florida State University
haoj8711@gmail.com

Xing Wang
Tencent AI Lab
brightxwang@tencent.com

Shuming Shi
Tencent AI Lab
shumingshi@tencent.com

Jinfeng Zhang
Florida State University
jinfeng@stat.fsu.edu

Zhaopeng Tu
Tencent AI Lab
zptu@tencent.com

Abstract

Recent studies have shown that a hybrid of self-attention networks (SANs) and recurrent neural networks (RNNs) outperforms both individual architectures, while not much is known about why the hybrid models work. With the belief that modeling hierarchical structure is an essential complementary between SANs and RNNs, we propose to further enhance the strength of hybrid models with an advanced variant of RNNs – *Ordered Neurons* LSTM (ON-LSTM, Shen et al., 2019), which introduces a syntax-oriented inductive bias to perform tree-like composition. Experimental results on the benchmark machine translation task show that the proposed approach outperforms both individual architectures and a standard hybrid model. Further analyses on targeted linguistic evaluation and logical inference tasks demonstrate that the proposed approach indeed benefits from a better modeling of hierarchical structure.

1 Introduction

Self-attention networks (SANs, Lin et al., 2017) have advanced the state of the art on a variety of natural language processing (NLP) tasks, such as machine translation (Vaswani et al., 2017), semantic role labelling (Tan et al., 2018), and language representations (Devlin et al., 2018). However, a previous study empirically reveals that the hierarchical structure of the input sentence, which is essential for language understanding, is not well modeled by SANs (Tran et al., 2018). Recently, hybrid models which combine the strengths of SANs and recurrent neural networks (RNNs) have outperformed both individual architectures on a machine translation task (Chen et al., 2018). We attribute the improvement to that RNNs complement SANs on the representation limitation of hierarchical structure, which is exactly the strength of RNNs (Tran et al., 2018).

Starting with this intuition, we propose to further enhance the representational power of hybrid models with an advanced RNNs variant – *Ordered Neurons* LSTM (ON-LSTM, Shen et al., 2019). ON-LSTM is better at modeling hierarchical structure by introducing a syntax-oriented inductive bias, which enables RNNs to perform tree-like composition by controlling the update frequency of neurons. Specifically, we stack SANs encoder on top of ON-LSTM encoder (*cascaded encoder*). SANs encoder is able to extract richer representations from the input augmented with structure context. To reinforce the strength of modeling hierarchical structure, we propose to simultaneously expose both types of signals by explicitly combining outputs of the SANs and ON-LSTM encoders.

We validate our hypothesis across a range of tasks, including machine translation, targeted linguistic evaluation, and logical inference. While machine translation is a benchmark task for deep learning models, the last two tasks focus on evaluating how much structure information is encoded in the learned representations. Experimental results show that the proposed approach consistently improves performances in all tasks, and modeling hierarchical structure is indeed an essential complementary between SANs and RNNs.

The contributions of this paper are:

- We empirically demonstrate that a better modeling of hierarchical structure is an essential strength of hybrid models over the vanilla SANs.
- Our study proves that the idea of augmenting RNNs with ordered neurons (Shen et al., 2019) produces promising improvement on machine translation, which is one potential criticism of ON-LSTM.

[*]Work done when interning at Tencent AI Lab.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1336–1341,
Hong Kong, China, November 3–7, 2019.

2 Approach

Partially motivated by Wang et al. (2016) and Chen et al. (2018), we stack a SANs encoder on top of a RNNs encoder to form a cascaded encoder. In the cascaded encoder, hierarchical structure modeling is enhanced in the bottom RNNs encoder, based on which SANs encoder is able to extract representations with richer hierarchical information. Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be the input sequence, the representation of the cascaded encoder is calculated by

$$\mathbf{H}^K_{\text{RNNs}} = \text{ENC}_{\text{RNNs}}(\mathbf{X}), \quad (1)$$
$$\mathbf{H}^L_{\text{SANs}} = \text{ENC}_{\text{SANs}}(\mathbf{H}^K_{\text{RNNs}}), \quad (2)$$

where $\text{ENC}_{\text{RNNs}}(\cdot)$ is a K-layer RNNs encoder that reads the input sequence, and $\text{ENC}_{\text{SANs}}(\cdot)$ is a L-layer SANs encoder that takes the output of RNNs encoder as input.

In this work, we replace the standard RNNs with recently proposed ON-LSTM for better modeling of hierarchical structure, and directly combine the two encoder outputs to build even richer representations, as described below.

Modeling Hierarchical Structure with Ordered Neurons ON-LSTM introduces a new syntax-oriented inductive bias – *Ordered Neurons*, which enables LSTM models to perform tree-like composition without breaking its sequential form (Shen et al., 2019). Ordered neurons enables dynamic allocation of neurons to represent different time-scale dependencies by controlling the update frequency of neurons. The assumption behind ordered neurons is that some neurons always update more (or less) frequently than the others, and that order is pre-determined as part of the model architecture. Formally, ON-LSTM introduces novel ordered neuron rules to update cell state:

$$w_t = \tilde{f}_t \circ \tilde{i}_t, \quad (3)$$
$$\hat{f}_t = f_t \circ w_t + (\tilde{f}_t - w_t), \quad (4)$$
$$\hat{i}_t = i_t \circ w_t + (\tilde{i}_t - w_t), \quad (5)$$
$$c_t = \hat{f}_t \circ c_{t-1} + \hat{i}_t \circ \hat{c}_t, \quad (6)$$

where forget gate f_t, input gate i_t and state $\hat{c}_t$ are same as that in the standard LSTM (Hochreiter and Schmidhuber, 1997). The *master forget gate* $\tilde{f}_t$ and the *master input gate* $\tilde{i}_t$ are newly introduced to control the erasing and the writing behaviors respectively. w_t indicates the overlap, and when the overlap exists ($\exists k, w_{tk} > 0$), the corresponding neurons are further controlled by the standard gates f_t and i_t.

An ideal master gate is in binary format such as $(0, 0, 1, 1, 1)$, which splits the cell state into two continuous parts: 0-part and 1-part. The neurons corresponding to 0-part and 1-part are updated with more and less frequencies separately, so that the information in 0-part neurons will only keep a few time steps, while the information in 1-part neurons will last for more time steps. Since such binary gates are not differentiable, the goal turns to find the splitting point d (the index of the first 1 in the ideal master gate). To this end, Shen et al. (2019) introduced a new activation function:

$$\text{CU}(\cdot) = \text{CUMSUM}(softmax(\cdot)), \quad (7)$$

where $softmax(\cdot)$ produces a probability distribution (e.g.$(0.1, 0.2, 0.4, 0.2, 0.1)$) to indicate the probability of each position being the splitting point d. CUMSUM is the cumulative probability distribution, in which the k-th probability refers to the probability that d falls within the first k positions. The output for the above example is $(0.1, 0.3, 0.7, 0.9, 1.0)$, in which different values denotes different update frequencies. It also equals to the probability of each position's value being 1 in the ideal master gate. Since this ideal master gate is binary, $\text{CU}(\cdot)$ is the expectation of the ideal master gate.

Based on this activation function, the master gates are defined as

$$\tilde{f}_t = \text{CU}_f(\mathbf{x}_t, \mathbf{h}_{t-1}), \quad (8)$$
$$\tilde{i}_t = 1 - \text{CU}_i(\mathbf{x}_t, \mathbf{h}_{t-1}), \quad (9)$$

where $\mathbf{x}_t$ is the current input and $\mathbf{h}_{t-1}$ is the hidden state of previous step. CU_f and CU_i are two individual activation functions with their own trainable parameters.

Short-Cut Connection Inspired by previous work on exploiting deep representations (Peters et al., 2018; Dou et al., 2018), we propose to simultaneously expose both types of signals by explicitly combining them with a simple *short-cut connection* (He et al., 2016).

Similar to positional encoding injection in Transformer (Vaswani et al., 2017), we add the output of the ON-LSTM encoder to the output of SANs encoder:

$$\widehat{\mathbf{H}} = \mathbf{H}^K_{\text{ON-LSTM}} + \mathbf{H}^L_{\text{SANs}}, \quad (10)$$

#	Encoder Architecture	Para.	BLEU
	Base Model		
1	6L SANs	88M	27.31
2	6L LSTM	97M	27.23
3	6L ON-LSTM	110M	27.44
4	6L LSTM + 4L SANs	104M	27.78$^{\uparrow}$
5	6L ON-LSTM + 4L SANs	123M	28.27$^{\Uparrow}$
6	3L ON-LSTM + 3L SANs	99M	28.21$^{\Uparrow}$
7	+ Short-Cut	99M	**28.37**$^{\Uparrow}$
	Big Model		
8	6L SANs	264M	28.58
9	Hybrid Model + Short-Cut	308M	**29.30**$^{\Uparrow}$

Table 1: Case-sensitive BLEU scores on the WMT14 English⇒German translation task. "↑ / ⇑": significant over the conventional self-attention counterpart ($p < 0.05/0.01$), tested by bootstrap resampling. "6L SANs" is the state-of-the-art Transformer model. "nL LSTM + mL SANs" denotes stacking n LSTM layers and m SANs layers subsequently. "Hybrid Model" denotes "3L ON-LSTM + 3L SANs".

where $\mathbf{H}^K_{\text{ON-LSTM}} \in \mathbb{R}^{N\times d}$ is the output of ON-LSTM encoder, and $\mathbf{H}^L_{\text{SANs}} \in \mathbb{R}^{N\times d}$ is output of SANs encoder.

3 Experiments

We chose machine translation, targeted linguistic evaluation and logical inference tasks to conduct experiments in this work. The first and the second tasks evaluate and analyze models as the hierarchical structure is an inherent attribute for natural language. The third task aims to directly evaluate the effects of hierarchical structure modeling on artificial language.

3.1 Machine Translation

For machine translation, we used the benchmark WMT14 English⇒German dataset. Sentences were encoded using byte-pair encoding (BPE) with 32K word-piece vocabulary (Sennrich et al., 2016). We implemented the proposed approaches on top of TRANSFORMER (Vaswani et al., 2017) – a state-of-the-art SANs-based model on machine translation, and followed the setting in previous work (Vaswani et al., 2017) to train the models, and reproduced their reported results. We tested on both the *Base* and *Big* models which differ at hidden size (512 vs. 1024), filter size (2048 vs. 4096) and number of attention heads (8 vs. 16). All the model variants were implemented on the encoder. The implementation details are introduced in Appendix A.1. Table 1 lists the results.

#	Encoder Architecture	Para.	BLEU
1	3L ON-LSTM → 3L SANs	99M	28.21
2	3L SANs → 3L ON-LSTM	99M	27.39
3	8L LSTM	102.2M	27.25
4	10L SANs	100.6M	27.76

Table 2: Results for encoder strategies. Case-sensitive BLEU scores on the WMT14 English⇒German translation task. "A → B" denotes stacking B on the top of A. The model in Row 1 is the hybrid model in Table 1.

Baselines (Rows 1-3) Following Chen et al. (2018), the three baselines are implemented with the same framework and optimization techniques as used in Vaswani et al. (2017). The difference between them is that they adopt SANs, LSTM and ON-LSTM as basic building blocks respectively. As seen, the three architectures achieve similar performances for their unique representational powers.

Hybrid Models (Rows 4-7) We first followed Chen et al. (2018) to stack 6 RNNs layers and 4 SANs layers subsequently (Row 4), which consistently outperforms the individual models. This is consistent with results reported by Chen et al. (2018). In this setting, the ON-LSTM model significantly outperforms its LSTM counterpart (Row 5), and reducing the encoder depth can still maintain the performance (Row 6). We attribute these to the strength of ON-LSTM on modeling hierarchical structure, which we believe is an essential complementarity between SANs and RNNs. In addition, the Short-Cut connection combination strategy improves translation performances by providing richer representations (Row 7).

Stronger Baseline (Rows 8-9) We finally conducted experiments on a stronger baseline – the TRANSFORMER-BIG model (Row 8), which outperforms its TRANSFORMER-BASE counterpart (Row 1) by 1.27 BLEU points. As seen, our model consistently improves performance over the stronger baseline by 0.72 BLEU points, demonstrating the effectiveness and universality of the proposed approach.

Assessing Encoder Strategies We first investigate the encoder stack strategies on different stack orders. From Table 2, to compare with the proposed hybrid model, we stack 3-layers ON-LSTM on the top of 3-layers SANs (Row 2). It performs worse than the strategy in the proposed hybrid model. The result support the viewpoint that the

Task	S	O	Hybrid + Short-Cut		
	Final	Final	$\mathbf{H}_O$	$\mathbf{H}_S$	Final
Surface Tasks					
SeLen	*92.71*	90.70	91.94	89.50	89.86
WC	81.79	76.42	*90.38*	79.10	80.37
Avg	87.25	83.56	**91.16**	84.30	85.12
Syntactic Tasks					
TrDep	44.78	52.58	51.19	52.55	*53.28*
ToCo	84.53	86.32	86.29	*87.92*	87.89
BShif	52.66	*82.68*	81.79	82.05	81.90
Avg	60.66	73.86	73.09	74.17	**74.36**
Semantic Tasks					
Tense	84.76	86.00	83.88	*86.05*	85.91
SubN	85.18	85.44	85.56	84.59	*85.81*
ObjN	81.45	*86.78*	85.72	85.80	85.38
SoMo	49.87	49.54	49.23	49.12	*49.92*
CoIn	68.97	72.03	72.06	72.05	*72.23*
Avg	74.05	**75.96**	75.29	*75.52*	75.85

Table 3: Performance on the linguistic probing tasks of evaluating linguistics embedded in the learned representations. "**S**" and "**O**" denote the SAN and ON-LSTM baseline models. "$\mathbf{H}_O$" and "$\mathbf{H}_S$" are respectively the outputs of the ON-LSTM encoder and the SAN encoder in the hybrid model, and "Final" denotes the final output exposed to decoder.

SANs encoder is able to extract richer representations if the input is augmented with sequential context (Chen et al., 2018).

Moreover, to dispel the doubt that whether the improvement of hybrid model comes from the increasement of parameters. We investigate the 8-layers LSTM and 10-layers SANs encoders (Rows 3-4) which have more parameters compared with the proposed hybrid model. The results show that the hybrid model consistently outperforms these model variants with less parameters and the improvement should not be due to more parameters.

3.2 Targeted Linguistic Evaluation

To gain linguistic insights into the learned representations, we conducted probing tasks (Conneau et al., 2018) to evaluate linguistics knowledge embedded in the final encoding representation learned by model, as shown in Table 3. We evaluated SANs and proposed hybrid model with Short-Cut connection on these 10 targeted linguistic evaluation tasks. The tasks and model details are described in Appendix A.2.

Experimental results are presented in Table 3. Several observations can be made here. The proposed hybrid model with short-cut produces more

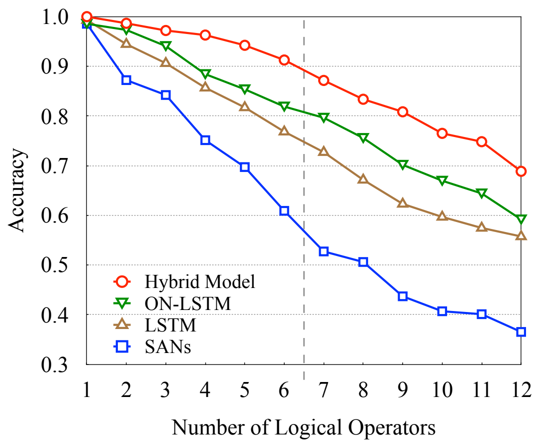

Figure 1: Accuracy of logical inference when training on logic data with at most 6 logical operators in the sequence.

informative representation in most tasks ("Final" in "S" vs. in "Hybrid+Short-Cut"), indicating that the effectiveness of the model. The only exception are surface tasks, which is consistent with the conclusion in Conneau et al. (2018): as a model captures deeper linguistic properties, it will tend to forget about these superficial features. Short-cut further improves the performance by providing richer representations ("$\mathbf{H}_S$" vs. "Final" in "Hybrid+Short-Cut"). Especially on syntactic tasks, our proposed model surpasses the baseline more than 13 points (74.36 vs. 60.66) on average, which again verifies that ON-LSTM enhance the strength of modeling hierarchical structure for self-attention.

3.3 Logical Inference

We also verified the model's performance in the logical inference task proposed by Bowman et al. (2015). This task is well suited to evaluate the ability of modeling hierarchical structure. Models need to learn the hierarchical and nested structures of language in order to predict accurate logical relations between sentences (Bowman et al., 2015; Tran et al., 2018; Shen et al., 2019). The artificial language of the task has six types of words {a, b, c, d, e, f} in the vocabulary and three logical operators {or, and, not}. The goal of the task is to predict one of seven logical relations between two given sentences. These seven relations are: two entailment types ($\sqsubset, \sqsupset$), equivalence ($\equiv$), exhaustive and non-exhaustive contradiction ($\wedge, |$), and semantic independence ($\#, \smile$).

We evaluated the SANs, LSTM, ON-LSTM and proposed model. We followed Tran et al. (2018) to use two hidden layers with Short-Cut connection in all models. The model details and hyperparameters are described in Appendix A.3.

Figure 1 shows the results. The proposed hybrid model outperforms both the LSTM-based and the SANs-based baselines on all cases. Consistent with Shen et al. (2019), on the longer sequences (≥ 7) that were not included during training, the proposed model also obtains the best performance and has a larger gap compared with other models than on the shorter sequences (≤ 6), which verifies the proposed model is better at modeling more complex hierarchical structure in sequence. It also indicates that the hybrid model has a stronger generalization ability.

4 Related Work

Improved Self-Attention Networks Recently, there is a large body of work on improving SANs in various NLP tasks (Yang et al., 2018; Wu et al., 2018; Yang et al., 2019a,b; Guo et al., 2019; Wang et al., 2019a; Sukhbaatar et al., 2019), as well as image classification (Bello et al., 2019) and automatic speech recognition (Mohamed et al., 2019) tasks. In these works, several strategies are proposed to improve the utilize SANs with the enhancement of local and global information. In this work, we enhance the SANs with the On-Lstm to form a hybrid model (Chen et al., 2018), and thoroughly evaluate the performance on machine translation, targeted linguistic evaluation, and logical inference tasks.

Structure Modeling for Neural Networks in NLP Structure modeling in NLP has been studied for a long time as the natural language sentences inherently have hierarchical structures (Chomsky, 1965; Bever, 1970). With the emergence of deep learning, tree-based models have been proposed to integrate syntactic tree structure into Recursive Neural Networks (Socher et al., 2013), LSTMs (Tai et al., 2015), CNNs (Mou et al., 2016). As for SANs, Hao et al. (2019a), Ma et al. (2019) and Wang et al. (2019b) enhance the SANs with neural syntactic distance, multi-granularity attention scope and structural position representations, which are generated from the syntactic tree structures.

Closely related to our work, Hao et al. (2019b) find that the integration of the recurrence in SANs encoder can provide more syntactic structure features to the encoder representations. Our work follows this direction and empirically evaluates the structure modelling on the related tasks.

5 Conclusion

In this paper, we adopt the ON-LSTM, which models tree structure with a novel activation function and structured gating mechanism, as the RNNs counterpart to boost the hybrid model. We also propose a modification of the cascaded encoder by explicitly combining the outputs of individual components, to enhance the ability of hierarchical structure modeling in a hybrid model. Experimental results on machine translation, targeted linguistic evaluation and logical inference tasks show that the proposed models achieve better performances by modeling hierarchical structure of sequence.

Acknowledgments

J.Z. was supported by the National Institute of General Medical Sciences of the National Institute of Health under award number R01GM126558. We thank the anonymous reviewers for their insightful comments.

References

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. 2019. Attention augmented convolutional networks. *arXiv*.

Thomas G Bever. 1970. The cognitive basis for linguistic structures. *Cognition and the development of language*, 279(362).

Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. 2015. Tree-structured composition in neural networks without tree-structured architectures. *NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, volume 11. MIT press.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $\$\&!\#*$ vector: Probing sentence embeddings for linguistic properties. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Maosheng Guo, Yu Zhang, and Ting Liu. 2019. Gaussian transformer: a lightweight approach for natural language inference. In *AAAI*.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019a. Multi-granularity self-attention for neural machine translation. In *EMNLP*.

Jie Hao, Xing Wang, Baosong Yang, Longyue Wang, Jinfeng Zhang, and Zhaopeng Tu. 2019b. Modeling recurrence for transformer. In *NAACL*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

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

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.

Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2019. Improving neural machine translation with neural syntactic distance. In *NAACL)*.

Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer. 2019. Transformers with convolutional context for asr. *arXiv*.

Lili Mou, Ge Li, Lu Zhang, Tao Wang, and Zhi Jin. 2016. Convolutional neural networks over tree structures for programming language processing. In *AAAI*.

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*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *ACL*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL-IJCNLP*.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *AAAI*.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *EMNLP*.

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, Longyue Wang, and Shuming Shi. 2019a. Exploiting sentential context for neural machine translation. In *ACL*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019b. Self-attention networks with structural position encoding. In *EMNLP*.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *COLING*.

Wei Wu, Houfeng Wang, Tianyu Liu, and Shuming Ma. 2018. Phrase-level self-attention networks for universal sentence encoding. In *ACL*.

Baosong Yang, Jian Li, Derek F. Wong, Lidia S. Chao, Xing Wang, and Zhaopeng Tu. 2019a. Context-aware self-attention networks. In *AAAI*.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *EMNLP*.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019b. Convolutional self-attention networks. In *NAACL*.