Improving BERT with Syntax-aware Local Attention

Zhongli Li[1]*, **Qingyu Zhou**[2]†, **Chao Li**[2], **Ke Xu**[1], **Yunbo Cao**[2]
[1]Beihang University
[2]Tencent Cloud Xiaowei
{lizhongli@,kexu@nlsde.}buaa.edu.cn
{qingyuzhou,diegoli,yunbocao}@tencent.com

Abstract

Pre-trained Transformer-based neural language models, such as BERT, have achieved remarkable results on varieties of NLP tasks. Recent works have shown that attention-based models can benefit from more focused attention over local regions. Most of them restrict the attention scope within a linear span, or confine to certain tasks such as machine translation and question answering. In this paper, we propose a syntax-aware local attention, where the attention scopes are restrained based on the distances in the syntactic structure. The proposed syntax-aware local attention can be integrated with pretrained language models, such as BERT, to render the model to focus on syntactically relevant words. We conduct experiments[1] on various single-sentence benchmarks, including sentence classification and sequence labeling tasks. Experimental results show consistent gains over BERT on all benchmark datasets. The extensive studies verify that our model achieves better performance owing to more focused attention over syntactically relevant words.

1 Introduction

Recently, Transformer (Vaswani et al., 2017) has performed remarkably well, standing on the multi-headed dot-product attention which fully takes into account the global contextualized information. Several studies find that self-attention can be enhanced by local attention, where the attention scopes are restricted to important local regions. Luong et al. (2015); Yang et al. (2018); Xu et al. (2019); Nguyen et al. (2020) utilize dynamic or fixed windows to perform local attention. Strubell et al. (2018); Zhang et al. (2020); Bugliarello and Okazaki (2020) explore to utilize syntax to restrain attention for better performance, but each of them confines to a certain task.

In this work, we propose a **s**yntax-aware **l**ocal **a**ttention (**SLA**) which is adaptable to several tasks, and integrate it with BERT (Devlin et al., 2019). We first apply dependency parsing to the input text, and calculate the distances of input words to construct the self-attention masks. The local attention scores are calculated by applying these masks to the dot-product attention. Then we incorporate the syntax-aware local attention with the Transformer global attention. A gate unit is employed for each token in each layer, which determines how much attention is paid to syntactically relevant words. We lift weights from existing pre-trained BERT, and evaluate our models on several single-sentence benchmarks, including sentence classification and sequence labeling tasks. Experimental results show that our method achieves consistent performance gains over BERT and outperforms previous syntax-based approaches on the average performance. Furthermore, we compare our syntax-aware local attention with the window-based local attention. We find that the syntax-aware local attention is more involved in the aggregation of local and global attention. The attention visualization also validates the syntactic information supports to capture important local regions.

To summarize, this paper makes the following contributions: i) SLA can capture the information of important local regions on the syntactic structure. ii) SLA can be easily integrated to Transformer, which allows initialization from pre-trained BERT by increasing very few parameters. iii) Experiments show the effectiveness of SLA on various single-sentence benchmarks.

*Contribution done during internship at Tencent Cloud Xiaowei.

†Corresponding author.

[1]The code is available at https://github.com/Neutralzz/syntax_aware_local_attention

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 645–653

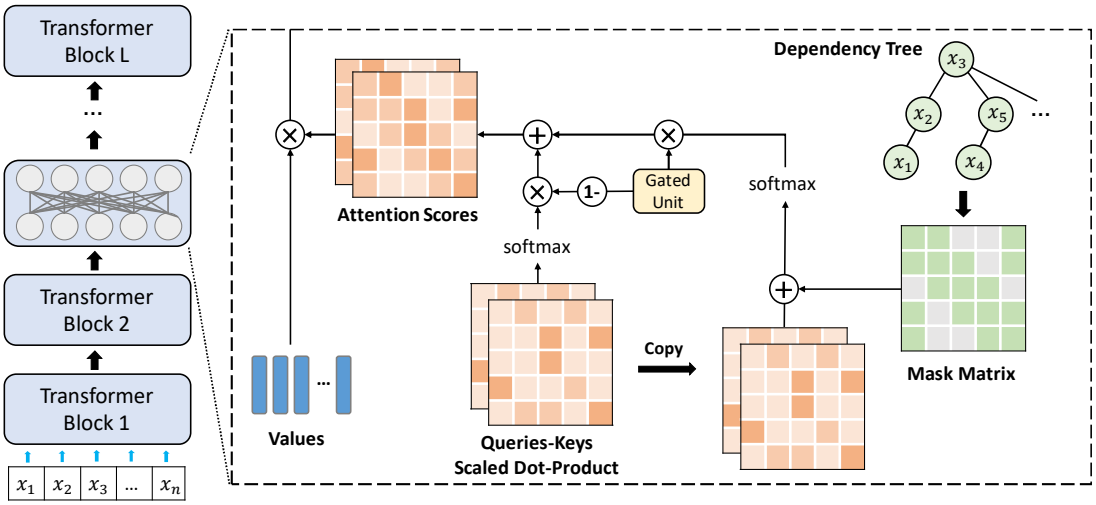

Figure 1: An overview of our model.

2 Related Work

2.1 Transformer Attention

Transformer (Vaswani et al., 2017) use stacked self-attentions to encode contextual information for input tokens. The calculation of self-attention depends on the three components of queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$, which are projected from the hidden vectors of the previous layer. Then the attention output $\mathbf{A}$ of one head is computed as follows:

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases}$$
$$\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{M})\mathbf{V} \quad (1)$$

where d is the dimension of keys and the mask matrix $\mathbf{M}$ controls whether two tokens can attend each other. Within the standard self-attention layer, global attention mechanism is employed that each token provides information to other tokens in the input sentence.

2.2 Local Attentions

Local attention involves limiting each token to attend to a subset of the other tokens in the input. Many works utilize a fixed or dynamic window to derive the important local regions. Luong et al. (2015) first propose a Gaussian-based local attention and increase BLEU scores for neural machine translation. Yang et al. (2018) improve the method of Luong et al. (2015) by predicting a central position and window size to model localness. Compared with Yang et al. (2018), Nguyen et al. (2020) attempt to derive the local window span by a soft-masking method. However, Levy and Goldberg (2014) suggest that more informative representations can be learned from the syntactic structure, instead of a window of surrounding tokens. Strubell et al. (2018) propose to train one attention head to attend to each token's syntactic parent for semantic role labeling. Zhang et al. (2020) also leverage the syntactic information to self-attention, but confine to question answering. Thus, we explore to take advantage of the syntactic structure to improve the model performance on various benchmarks.

3 Approach

In this section, we first introduce the syntax-aware local attention, and then integrate it with standard Transformer attention. As shown in Figure 1, we extend the Transformer layer with the syntax-aware local attention. Syntax-based masking is applied to the dot-product of queries and keys. The final attention scores are computed by incorporating local attention with standard global attention. We stack new layers and initialize weights from pre-trained BERT.

3.1 Syntax-aware Local Attention

We derive syntactic structure from dependency parsing, and treat it as an undirected tree. Each token x_i is mapped to a tree node v_i, and the distance of node v_i and v_j is denoted by $dis(v_i, v_j)$. However, the input may be an ungrammatical sentence in some tasks, and the dependency parser is not very accurate. Thus, we calculate the distance

from neighboring tokens of x_i to token x_j as:

$$D(i,j) = \min dis(v_k, v_j), \quad k \in [i-1, i+1] \quad (2)$$

The motivation is that many attention heads specialize in attending heavily on the next or previous token (Clark et al., 2019). Then, in order to determine whether token x_j can attend to token x_i, a threshold m is applied to restrict the distance $D(i,j)$. For simplification, the mask matrix $\mathbf{M}^{loc}$ calculation can be formulated as:

$$\mathbf{M}^{loc}_{ij} = \begin{cases} 0, & D(i,j) \leq m \\ -\infty, & otherwise \end{cases} \quad (3)$$

Given the query $\mathbf{Q}$ and key $\mathbf{K}$ projected from the hidden vectors $\mathbf{H}$, the syntax-aware local attention scores $\mathbf{S}^{loc}$ are formally defined as:

$$\mathbf{S}^{loc} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}^{loc}) \quad (4)$$

where d is the dimension of keys. In this local attention, two tokens can attend to each other only if they are close enough in the dependency tree.

3.2 Attention Aggregation

As shown in Figure 1, the final attention is the aggregation of syntax-aware local attention and Transformer attention. We denote the Transformer attention scores by $\mathbf{S}^{glb}$. A gated unit is used to combine the global and local attention scores. The gate value g_i for each token x_i is calculated as follows:

$$g_i = \sigma(\mathbf{W}_g h_i + b_g), \quad (5)$$

where h_i is the hidden vector of token x_i from the previous layer, $\mathbf{W}_g$ is a learnable linear transformation and b_g is the bias. Then the attention output $\hat{\mathbf{A}}_i$ is calculated as a weighted average over values $\mathbf{V}$, and the weights are derived from global and local attention scores:

$$\hat{\mathbf{A}}_i = (g_i \mathbf{S}^{loc}_i + (1 - g_i)\mathbf{S}^{glb}_i)\mathbf{V}. \quad (6)$$

A larger gate value means more focused attention over syntactically relevant words. It can be seen that, if all the outputs of gated units are equal to 0, we could obtain the standard Transformer attention. Compared with the original architecture, our self-attention layer has one more input ($\mathbf{M}_{loc}$) and two more trainable parameters ($\mathbf{W}_g$ and b_g). Thus, we can easily lift weights from existing pre-trained BERT models.

4 Experiments

4.1 Experimental Setup

Benchmarks We use two English single-sentence classification datasets from the GLUE benchmark (Wang et al., 2018). We test on the CoLA and SST-2 datasets for acceptability and sentiment classification. Besides, we evaluate our method on two sequence labeling tasks: named entity recognition (NER) and grammatical error detection (GED). We use the CoNLL-2003 and FCE datasets for NER and GED, respectively. The training procedures are introduced in Appendix A.1.

Configuration All the training experiments are based on BERT. We use the uncased version of BERT for CoLA and SST-2, and the cased version for CoNLL-2003 and FCE. We derive dependency tree using Spacy[2]. More implementation details are reported in Appendix A.2.

Baselines We apply the syntax-aware local attention (**SLA**) to BERT. In addition to comparing with BERT, we also investigate the following approaches:

SGNet Zhang et al. (2020) present a syntax-guided self-attention layer, where each word is limited to interact with all of its syntactic ancestor words. Then they stack this layer on the top of the pre-trained BERT model[3], instead of modifying the Transformer architecture.

LISA Strubell et al. (2018) restrict each token to attend to its syntactic parent in one attention head[4]. We apply it to BERT and add the corresponding supervision at the last attention head in each Transformer layer.

Besides, we implement the window-based local attention (**WLA**), which allows each token to attend to the neighboring tokens within a window size $2k + 1$ (varying k in {3,4,5}). Then it is also integrated with BERT as shown in Section 3.2.

4.2 Main Results

Experimental results are shown in Table 1. We report results on the dev set of CoLA and SST-2 and the test set of CoNLL-2003 and FCE. We employ t-tests to see if the mean difference differed from 0 between the standard attention and our proposed attention. It can be seen that our

[2] https://spacy.io/
[3] https://github.com/cooelf/SG-Net
[4] https://github.com/strubell/LISA

Models	Params	Avg.	CoLA MCC	SST-2 Acc	CoNLL-2003 P	CoNLL-2003 R	CoNLL-2003 F_1	FCE (M2) P	FCE (M2) R	FCE (M2) $F_{0.5}$
State-of-the-art Models										
ERNIE 2.0 (Sun et al., 2020)	-	-	65.4	96.0	-	-	-	-	-	-
T5 (Raffel et al., 2019)	-	-	71.6	97.5	-	-	-	-	-	-
BERT-MRC (Li et al., 2020)	-	-	-	-	92.3	94.6	93.0	-	-	-
BERT-GED (Bell et al., 2019)	-	-	-	-	-	-	-	65.0	38.9	57.3
Base-size Models										
BERT (Devlin et al., 2019)	110M	-	58.9	92.7	-	-	**92.4**	-	-	-
LISA (Strubell et al., 2018)	110M	74.8	59.8	92.0	90.7	92.2	91.4	**63.4**	38.6	**56.1**
SGNet (Zhang et al., 2020)	133M	74.8	59.2	93.1	90.9	92.6	91.7	60.9	40.7	55.4
BERT (Our reimplementation)	110M	74.6	58.7	93.1	91.0	92.3	91.6	60.5	40.0	54.9
+ WLA	+0.01M	75.0	59.6	92.8	91.3	92.9	92.1	60.4	**41.3**	55.3
+ SLA	+0.01M	**75.3**	**60.0**$^{\uparrow}$	**93.3**	**91.5**$^{\uparrow}$	**92.9**$^{\uparrow}$	92.2$^{\uparrow}$	61.0$^{\uparrow}$	41.3$^{\uparrow}$	55.7$^{\uparrow}$
Large-size Models										
BERT (Devlin et al., 2019)	340M	-	60.6*	93.2*	-	-	92.8	-	-	-
LISA (Strubell et al., 2018)	340M	76.2	62.2	92.7	91.3	92.6	92.0	63.4	43.2	57.9
SGNet (Zhang et al., 2020)	381M	76.6	63.3	93.6	91.5	92.8	92.1	63.1	42.5	57.5
BERT (Our reimplementation)	340M	76.9	63.9	94.0	91.7	93.1	92.4	62.7	42.6	57.3
+ WLA	+0.02M	76.6	62.7	93.9	91.5	93.1	92.3	61.9	**44.5**	57.4
+ SLA	+0.02M	**77.4**	**64.5**$^{\uparrow}$	**94.3**$^{\uparrow}$	**92.3**$^{\uparrow}$	**93.4**	**92.9**	**63.9**$^{\uparrow}$	42.3	**58.0**$^{\uparrow}$

Table 1: Results on single-sentence benchmarks. Results with "$*$" are taken from Liu et al. (2019). "$\uparrow$" means statistically significant improvement over the BERT baseline with p-value < 0.05. Reported results are averaged over 5 runs. "Params" is short for the number of model parameters. "MCC" is short for the Matthews correlation coefficient.

method achieves consistent gains over BERT on single-sentence classification and sequence labeling tasks. Specifically, our model exceeds the published BERT results by 3.9% correlation coefficient on CoLA and 1.1% accuracy on SST-2. For the NER task, even though our reimplementation didn't achieve the performance (92.8 F1) reported by Devlin et al. (2019), our model still outperforms it in large-size. More importantly, the syntax-aware local attention yields state-of-the-art results with 0.7 absolute improvements on FCE.

Besides, the proposed local attention outperforms other approaches leveraging syntactic information on the average performance. Compared with BERT, the syntax-aware local attention improves performances consistently but the window-based local attention can't. This suggest that BERT can benefit from more attention over syntactically relevant words on several datasets.

However, there are still some gaps between our model and the state-of-the-art models on these datasets. We argue that our method just modifies the standard Transformer attention without changing its main architecture, but those models are trained by using more advanced pre-training methods (Sun et al., 2020), larger-scale datasets (Raffel et al., 2019), or learning framework (Li et al., 2020).

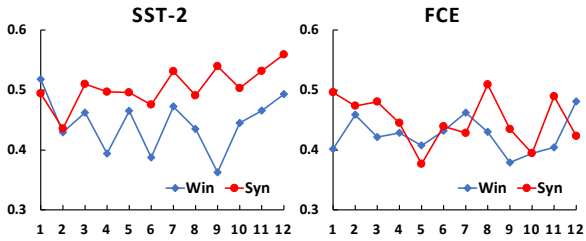

Figure 2: Gate values in different layers on SST-2 and FCE datasets. The blue polyline means that we incorporate the window-based local attention with global attention, and the red polyline corresponds to the syntax-aware local attention.

4.3 Analysis

Gated Unit in Each Layer It can be seen from Equation (6) that a larger gate value means a more important role of local attention in the attention aggregation. We analyze the gate values in different layers on SST-2 and FCE datasets. The gated unit outputs are collected from the best-trained base-size models, and are averaged over all input tokens in each layer.

As shown in Figure 2, on the SST-2 dataset, the syntax-aware local attention has higher values than the window-based local attention in most layers. Even if the sentences of the FCE dataset are ungrammatical, our attention plays a more important role in 8 of 12 layers. It indicates that our

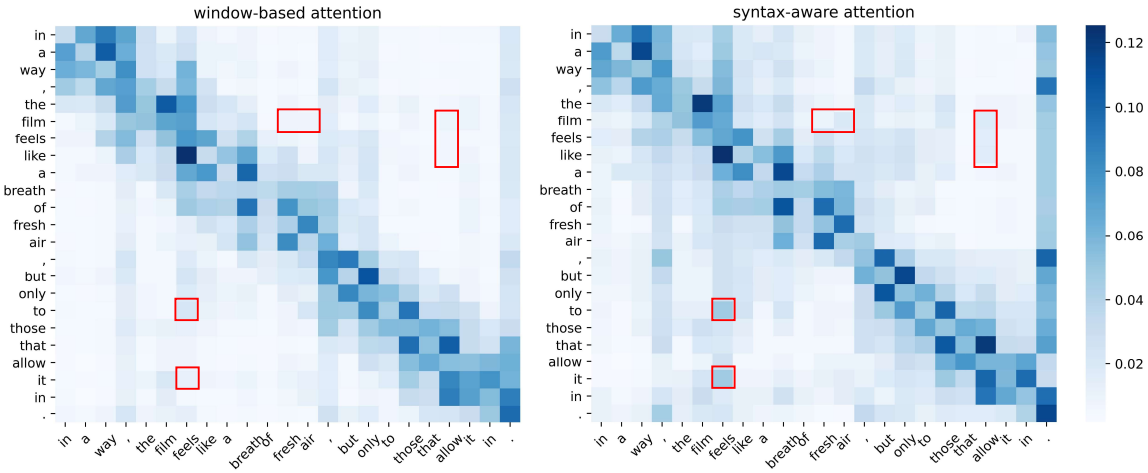

Figure 3: Visualization of attention scores averaged over all heads and all layers. This case is selected from the SST-2 dev set. The red rectangle indicates higher scores on the right side but lower scores on the left side.

Models	QNLI Acc	RTE Acc	MRPC Acc	STS PCC
BERT	**91.7**	**68.6**	87.3	89.5
+SLA	91.4	67.8	**88.5**	**89.9**

Table 2: Experimental results on sentence-pair classification datasets. All models are base-size and results are reported on their dev sets. "PCC" is short for the Pearson correlation coefficient.

local attention is more important in the attention score calculation process. Besides, Table 1 and Figure 2 illustrate that our model achieves better performances owing to more attention on syntactically relevant words.

Attention Visualization In order to compare the syntax-aware attention with the window-based attention, we plot their attention scores in Figure 3. As formulated in Equation (6), the attention scores are calculated from the aggregation of global and local attention. We mainly focus on the interactions of tokens, except for `[CLS]` and `[SEP]`. Then the attention scores are averaged over all heads and layers. This visualization validates the effectiveness of incorporating syntactic information into self-attention. As shown in Figure 3, we can see that there are many informative tokens overlooked by the window-based method (left) but captured by our method (right). For instance, the syntax-aware attention allows the tokens "*fresh air*" and "*allow*" to strongly attend to the token "*film*", but these tokens are paid less attention in the window-based attention.

Testing on Sentence-Pair Classification We attempt to evaluate our model on sentence-pair classification datasets. Given a single sentence, we can easily apply dependency parsing and restrain the attention scopes inside the sentence. But for pairwise classification, one problem is how to limit the scopes between a pair of sentences. So a naive approach is adopted, that each token in a sentence can attend to all tokens in another sentence. We conduct experiments on four pairwise classification datasets from GLUE benchmark (Wang et al., 2018), which cover paraphrase, textual entailment and text similarity.

Experimental results are shown in Table 2. The syntax-aware local attention achieves better performances on MRPC and STS, but doesn't perform well on RTE and QNLI. We suspect that it is because the cross-sentence interactions are more important for textual entailment task.

5 Conclusion

This work verifies that BERT can be further promoted by incorporating syntactic knowledge to the local attention mechanism. With more focused attention over the syntactically relevant words, our model achieves better performance on various benchmarks. Additionally, the extensive experiments demonstrate the universality of our syntax-aware local attention.

References

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is key: Grammatical error detection with contextual word representations. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ruiji Fu, Zhengqi Pei, Jiefu Gong, Wei Song, Dechuan Teng, Wanxiang Che, Shijin Wang, Guoping Hu, and Ting Liu. 2018. Chinese grammatical error diagnosis using statistical and prior knowledge driven features with probabilistic ensemble enhancement. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 52–59, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on EMNLP*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2020. Differentiable window for dynamic local attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6589–6599, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

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 *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Mingzhou Xu, Derek F. Wong, Baosong Yang, Yue Zhang, and Lidia S. Chao. 2019. Leveraging local and global patterns for self-attention networks.

In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3069–3075, Florence, Italy. Association for Computational Linguistics.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020. SG-Net: Syntax-guided machine reading comprehension. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

A Appendices

A.1 Training Procedure

We extend the Transformer encoder layer and lift weights from BERT to our model. Following Devlin et al. (2019), we apply the fine-tuning procedure for various NLP tasks. For classification tasks, the final output of the first token `[CLS]` is taken as the representation of the input. The probability that the input sentence X is labeled as class c is predicted by a linear transformation with softmax:

$$P(c\,|X) = \text{softmax}(\mathbf{W}_c h_{\texttt{[CLS]}} + b_c) \quad (7)$$

where $h_{\texttt{[CLS]}}$ is the representation of the token `[CLS]`, $\mathbf{W}_c$ and b_c are task-specific parameters. For labeling tasks, we apply the BIO annotation (Ratinov and Roth, 2009) to label outputs and compute the probability that token x_i belongs to class c as:

$$P(c\,|x_i) = \text{softmax}(\mathbf{W}_t h_i + b_t) \quad (8)$$

where h_i is the representation of the token x_i, $\mathbf{W}_t$ and b_t are task-specific parameters. Finally, the training objective for all tasks is to minimize the cross-entropy loss.

A.2 Implementation Details

We apply the whitespace tokenization to the input sentence, and obtain the dependency tree using the Spacy parser[5]. However, the BERT inputs are tokenized by WordPiece tokenizer, which means one word may be split into several sub-words. To address this issue, for each word in the dependency tree, the sub-words split by WordPiece tokenizer share the same masking value in the calculation of syntax-aware local attention.

An important detail is that BERT represents the input by adding a `[CLS]` token at the beginning as the special classification embedding and separating sentences with a `[SEP]` token. Clark et al. (2019) find that these special tokens are attached with a substantial amount of BERT's attention. Thus, the `[CLS]` and `[SEP]` tokens are guaranteed to be present and are never masked out in our local attention.

We use the uncased version of BERT for CoLA and SST-2, and the cased version for CoNLL-2003 and FCE. During the training, we empirically select the threshold m from {3,4}. The maximum sequence length is set to 128 for all tasks. We use Adam (Kingma and Ba, 2015) as our optimizer, and perform grid search over the sets of the learning rate as {2e-5, 3e-5} and the number of epochs as {3,5,10} for most tasks. In particular, we use smaller learning rates {5e-6, 1e-5, 2e-5} and train more epochs {30, 60} on CoNLL-2003, but the average F1 of the best 5 runs still hasn't reached the results reported by Devlin et al. (2019). The batch size is fixed to 32 to reduce the search space, and we evaluate models every 500 training steps for all datasets. Furthermore, we experiment with the window-based attention on BERT, which allows each token to pay more attention to the neighboring tokens within a window size $2k+1$. We vary the k within {3,4,5}, and also incorporate the attention scores with global attention scores.

A.3 Testing on Chinese Benchmarks

The ChnSentiCorp dataset is used for sentiment classification task. We treat the ChnSentiCorp as single-sentence datasets although there are some examples including multiple sentences. The MSRA NER and CGED datasets are selected for named entity recognition and grammatical error detection in Chinese. The accuracy (Acc) is used as the metric of ChnSentiCorp, the precision, recall and F_1 are used as metrics of MSRA NER and CGED. In particular, for a fair comparison with the results of iFLYTEK's single model (Fu et al., 2018), we construct the CGED test set from CGED 2016 and 2017 test sets. Then we report detection-level results computed by the official evaluation tool.

Table 3 shows the main results on Chinese datasets. All results are reported on their test set. The proposed syntax-aware local attention outperforms the window-based attention and the basic BERT on all evaluated datasets. We attain 95.7 accuracy on ChnSentiCorp and 94.9 F1 on MSRA NER. Besides, BERT+SLA outperforms the state-of-the-art with a large margin on CGED.

[5] https://spacy.io/

Models	ChnSentiCorp Acc	MSRA NER P	MSRA NER R	MSRA NER F_1	CGED P	CGED R	CGED F_1
State-of-the-art Models							
ERNIE 2.0 (Sun et al., 2020)	95.8	-	-	95.0	-	-	-
BERT-MRC (Li et al., 2020)	-	96.2	95.1	95.7	-	-	-
ePMI Matcher (Fu et al., 2018)	-	-	-	-	83.2	61.0	70.4
Base-size Models							
BERT (Our reimplementation)	94.7	95.0	94.6	94.8	79.9	75.2	77.5
+ WLA	95.1	**95.1**	94.2	94.6	79.9	73.5	76.6
+ SLA	**95.7**	94.9	**95.0**	**94.9**	**81.0**	**76.6**	**78.7**

Table 3: Experimental results on Chinese single-sentence benchmarks. We only show the results of base-size models because Google has not released the large-size model. Reported results are averaged over 5 runs.