Structure Representation Learning by Jointly Learning to Pool and Represent

Anonymous ACL submission

Abstract

Structure representation learning is a task to provide an overall representation for a given structure (e.g., sequential text, non-sequential graph). This representation characterizes the property of that structure. Previous methods decompose the task into an element representation learning phase and a pooling phase to aggregate element representations. Their pooling phase only considers the final representation of each element without considering the relationship between these elements that are used only to construct representations of elements. In this paper, we conjecture that classification performance suffers from the lack of relation exploitation while pooling and propose the Self-Attention Pooling to dynamically provide centrality scores for pooling based on the self-attention scores from the element representation learning. Simply applying Self-Attention Pooling improves model performance on 3 sentence classification tasks (↑ **2.9**) and 5 graph classification tasks (↑ **2.1**) on average[1].

1 Introduction

We use structure representation learning to denote learning a summary representation for a natural structure like a sequence or a non-sequential graph. For example, we can predict the property of a sentence that consists of a sequence of words, with its representations (Wang et al., 2019). In addition to the sequence, the structure can also be a non-sequential graph that is composed of nodes (Reimer and Hahn, 1988; Yao et al., 2018). This task usually follows a pipeline that first learns the representation of the elements and then pools the representations of these elements based on their final representations (Kim, 2014). The pooling layer first predicts the centrality of each element and then either weighted-sum element representations according to their centrality or selects element representations with significant centrality.

[1]We compare with CLS Pooling from BERT for sequence pooling and SAGPooling for non-sequence pooling.

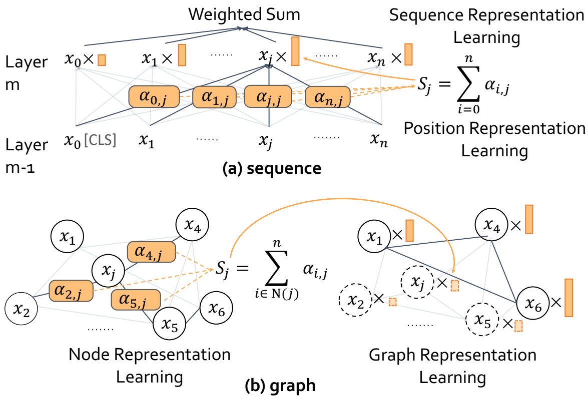

Figure 1: In Self-Attention Pooling, we jointly learn element representations and their centrality for pooling.

Most recently proposed models follow an element representation-based pooling method. For example, in sentence classification, scoring is obtained through the attention of artificial [CLS] token to natural words (Radford et al., 2018; Devlin et al., 2019). In graph classification, to get the centrality of each node, we can exploit the static graph topology (Lee et al., 2019) in addition to the representation of the nodes (Gao and Ji, 2019). A potential issue with this element representation-based pooling method is that obtaining the structure representation by separately considering the representation of the elements does not exploit the relation between the elements. This issue makes the model overly dependent on the element representation to encode the relationship between them and sequentially learns the representations of elements and the pool operation. The relation between elements that help learn element representations can also help learn structures (Voita et al., 2019; Jawahar et al., 2019).

To address this issue, we propose jointly learning to represent the elements and pool the elements by sharing the self-attention modules from element representation learning. Specifically, we utilize the accumulation of all the attention the element receives to indicate its centrality. We also design specific applications on sentence classification with the BERT model and graph classification with the Graph Attention model (refer to Fig. 1). In sentence classification, we extend BERT finetuning so that the relationship between natural words can be applied to pooling instead of just using the relationship between the artificial [CLS] token and natural tokens. We extend graph representation learning on graph classification by exploiting automatically learned node relations instead of just using static graph topology.

2 Related Work

Pooling plays an important role in both sequential (Socher et al., 2011; Chen et al., 2015; Safari et al., 2020) and non-sequential structure representation learning (Lee et al., 2019). Most methods separately learn element representations and pooling and do not exploit the relation between elements (Kim, 2014; Ying et al., 2018).

Sequential Pooling Sequential pooling objects to obtain a representation of a piece of text. Previous methods usually perform an average or maximum operation on every position (Kim, 2014; Ma et al., 2019; Song et al., 2020), or sum the representations of positions with their feature weights (Yang et al., 2016; Wu et al., 2020). The powerful pretrained language model BERT (Devlin et al., 2019) directly applies CLS pooling with an artificial [CLS] token (Devlin et al., 2019), which aggregates information by attending representations of other positions. However, these methods neglect the relation between all positions, and the CLS pooling is only learned in the finetuning phase of BERT. Recent studies find that attention weights can indicate keywords, but they do not study its effectiveness in pooling and downstream tasks like sequence classification (Clark et al., 2019; Ding and Luo, 2021).

Non-sequential Pooling Non-sequential pooling aims to extract the overall representation of a non-sequence. The graph is a well-studied non-sequence. Previous research mainly disassembled it into two parts: node representation learning and graph pooling. Traditional graph pooling takes the node representation into account (Gao and Ji, 2019), and recent methods propose to utilize graph topology to model the node relation (Lee et al., 2019; Murphy et al., 2019; Yuan and Ji, 2020), but the relationship automatically learned in node representation learning is still not considered.

3 Proposal

3.1 Self-Attention Pooling

To model dynamic relation in the structure representation, we propose Self-Attention Pooling. It links the construction of element representation and structure representation (i.e., pooling). For learning element representation, self-attention module updates the representation of each element. For pooling, weights are centrality scores that reflect the importance of each element in a structure. Inspired by PageRank (see Section 5), we define the centrality score of an element by its overall attention scores[2] received from other elements. While for learning structure representation, the centrality scores are ranked for top-k selection or weighted sum of the structure representation. We define X as the input structure, N as the number of elements and $X_j^{(m)}$ as the element j at layer m. Then, the element representation $X_j^{(m)}$ and the centrality $S_j^{(m)}$ can be formulated as follows:

$$X_j^{(m)} = \sum\nolimits_{i=1}^{N} \alpha_{i,j} X_j^{(m-1)} \quad (1)$$

$$S_j^{(m)} = \sum\nolimits_{i=1}^{N} \alpha_{i,j} \quad (2)$$

where $\alpha_{i,j}$ is the self-attention score from element i to j, and $\sum_{j=1}^{N} \alpha_{i,j} = \mathbf{1}$. For conciseness, we omit the description of the non-aggregation neural network and focus on the element aggregation.

3.2 Self-Attention based Sequence Pooling

As illustrated in Fig. 1 (a), for sequential structure, our objective is to learn sequence representation for downstream tasks like sentence classification. Here the element representation can be seen as position representation, e.g., word-level or subword-level representation. For sequence pooling, we study the powerful BERT model and compare its pooling methods. Therefore, we pool the representations from the last hidden layer of the BERT encoder.

We compare with the CLS pooling, mean-pooling, and max-pooling. Although been default in BERT pooling, CLS pooling merely takes $X_0^{(m)} = \sum_{j=0}^{N} \alpha_{0,j} X_j^{(m-1)}$ as the sequence representation. In contrast, BERT is pretrained with all the positions rather than only the CLS position.

[2]For pooling, we use the averaged self-attention scores overheads.

Therefore, the discrepancy between pretraining and CLS finetuning causes the learning of finetuning insufficient. Moreover, CLS pooling ignores the relation between natural tokens. mean-pooling and max-pooling are both typical pooling methods, they are operated along the position dimension here.

For Self-Attention Pooling, we implement pooling on the last hidden layer of the BERT encoder, while calculating $\alpha_{i,j}$ from various layers. According to Eq. 1, we exploit the relation between all positions to obtain centrality scores for each position. The overall sequence representation is $\sum_{j=0}^{N} Xj \cdot S_j$.

3.3 Self-Attention based Graph Pooling

As shown in Fig. 1 (b), for graph structure, nodes and graphs represent elements and structures respectively. Here X_j (j=1,2,...,N) denote the feature of each node. We compare our method with two baselines for graph representation: gPool that considers only node features, formulated[3] as $Z = X^{(l)}\Theta^{(l)} / \left\|\Theta^{(l)}\right\|$. SAGPool that considers both features of nodes and the overall graph topology, roughly[4] described as $Z = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta\right)$. Different from previous work, Self-Attention Pooling exploits node relations from the graph attention mechanism (Veličković et al., 2018) (GAT) directly, which is also crucial for node representation. It is slightly different from Eq. 1 because $\alpha_{i,j}$ is only calculated among each node and its neighbors. In GAT, e_{ij} is a logit calculated from concentrated element representation, $N(i)$ denotes node i and its neighbours. The centrality scores are calculated as:

$$\alpha_{ij} = \mathrm{softmax}_j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(e_{ik}\right)} \tag{3}$$

$$S_j = Z_j + \sum_{i \in \mathrm{N}(j)}^{N} \alpha_{i,j} \tag{4}$$

Since the attention in graph is local, we propose iterative Self-Attention Pooling as:

$$w_j = \sum_{i \in \mathrm{N}(j)}^{N} \alpha_{i,j},\ \alpha_i = w_i \cdot \alpha_i,\ S_j = Z_j + \sum_{i \in \mathrm{N}(j)}^{N} \alpha_{i,j} \tag{5}$$

[3]The superscript represents the layer. Θ, N and $\tilde{A} \in \mathbb{R}^{N \times N}$ stands for learnable parameters, the input features of the graph and the adjacency matrix respectively.

[4]$\tilde{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix with self-connections, (i.e. $\tilde{A} = A + I_N$), $\tilde{D} \in \mathbb{R}^{N \times N}$ is the degree matrix of $\tilde{A}$. For details of the formulas of gPool and SAGPool, refer to the SAGPool paper (Lee et al., 2019).

After getting the centrality score S_j of each node in the current graph, we can mask out the nodes with low importance and retrain the others for further calculation.

Dataset	CoLA	RTE	MRPC	
Metric	Matt.	Acc.	Acc.	F1
CLS Pooling	56.5	65.7	84.1	88.9
Mean Pooling	59.2	64.3	84.6	89.0
Max Pooling	59.1	63.5	81.4	87.7
S.A. Pooling (Ours)	**59.8**	**69.7**	**86.6**	**90.6**

Table 1: Results on three sequence classification tasks. S.A. Pooling refers to Self-Attention Pooling. Matt. denotes Matthews correlation coefficient. Acc. abbreviates Accuracy. F1 refers to F1 score.

4 Experiments

4.1 Datasets

Sequence Classification In our experiments, we consider a single sentence or a sentence pair as a sequence. We use CoLA for single sentence classification, MRPC, and RTE for sentence-Pair classification. CoLA (Warstadt et al., 2018) is expertly annotated for grammatical acceptability, consisting of 10,657 sentences from 23 linguistics publications. MRPC (Dolan and Brockett, 2005) is used to classify whether two sentences are paraphrases or not. It consists of 5,801 sentence pairs collected from newswire articles. RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) is a dataset for natural language inference. Given a premise and a hypothesis, models are expected to select the best answer between entailment, neutral, and contraction.

Graph Classification We experiment with five large graph datasets from the benchmark datasets (Kersting et al., 2016). D&D (Dobson and Doig, 2003; Shervashidze et al., 2011) and PROTEINS (Dobson and Doig, 2003; Borgwardt et al., 2005) are both protein datasets that are classified as enzymes or non-enzymes. Nodes represent the amino acids and two nodes are connected by an edge if they are less than 6 Angstroms apart. NCI (Wale et al., 2008) is a biological dataset used for anticancer activity classification. NCI1 and NCI109 are commonly used. FRANKENSTEIN (Orsini et al., 2015) is a set of molecular graphs (Costa and De Grave, 2010). Its label denotes whether a molecule is a mutagen or non-mutagen. D&D, PROTEINS, NCI, NCI109,

Dataset	D&D	PROTEINS	NCI1	NCI09	FRANKENSTEIN
gPool (Gao and Ji, 2019)	73.74±0.45	72.80±0.17	70.04±0.44	70.10±1.23	75.97±0.53
SAGPool (Lee et al., 2019)	75.01±0.50	72.99±0.12	72.37±0.22	71.63±0.54	76.09±0.57
S.A. Pooling (Ours)	76.00±0.71	**74.12**±0.40	74.60±0.22	73.81±0.41	79.02±0.70
Iterative S.A. Pooling (Ours)	**76.23**±0.13	74.06±0.40	**74.70**±0.25	**74.33**±0.15	**79.30**±0.68

Table 2: Results on graph classification tasks. gPool gets pooling scores from features. SAGPool uses the graph topology. Self-Attention Pooling introduces learned node relations from node representation learning to pooling.

FRANKENSTEIN have 1178, 1113, 4110, 4127, 4337 graphs respectively.

4.2 Training and Evaluation

Sequence Classification We use the $\text{BERT}_{\text{base}}$ model implemented by Transformers (Wolf et al., 2020), and follow the default setting of their "text-classification" directory without tuning *any* hyper-parameters. We also run all GLUE tasks and report results on them in the Appendix A.

Graph Classification We experiment on the GAT model and run it 3 times; each run contains 20 different train, valid, test splits of the data (split by 0.8, 0.1, 0.1) since a recent study indicates that different dataset splits largely affect the test performance (Shchur et al., 2019). For evaluation, we report test accuracy on the early stopping model with the best valid accuracy.

4.3 Results

As shown in Table 1, mean/max pooling outperforms CLS pooling on single sentence classification, but they are less effective on sentence-pair classification. Compared to CLS pooling, Self-Attention Pooling considers relations between natural tokens. The relations are the self-attention weights that can be easily transferred from the pertaining phase. On average, Self-Attention Pooling outperforms CLS pooling 2.9 points.

Table 2 demonstrates that graph topology is ineffective on the PROTEINS dataset and the FRANKENSTEIN dataset. In our Self-Attention Pooling method, the automatically leaned relation from the node representation learning serves as a good indicator for centrality. On average, Self-Attention Pooling outperforms SAGPool by +1.9 points, and can further achieve +0.2 improvements if we iterate the method twice.

5 Discussion

Relation to PageRank In order to measure the relative importance of web pages, Page et al. (1999) propose PageRank. Its main idea is that the value of a node is determined by the sum of all the nodes pointing to it, while our Self-Attention Pooling extends it to aggregating self-attention weights. Neural Pagerank (Klicpera et al., 2018) equips the PageRank algorithm with Neural Networks but still does into involve attention weights.

Layers Chosen To analyze the effect of layer chosen for Self-Attention Pooling during BERT fine-tuning, we take CLS Pooling as the baseline and experiment with different layer settings. Table 3 demonstrates that the last layers deliver the most substantial improvement.

Layer Metric	CoLA Matt.	RTE Acc.	MRPC Acc.	 F1
CLS_{L12}	56.5 (−)	65.7 (−)	84.1 (−)	88.9 (−)
L12	59.8($\uparrow$ 3.3)	68.2($\uparrow$ 2.5)	86.8($\uparrow$ 2.7)	90.7($\uparrow$ 1.8)
L10-12	**60.1**($\uparrow$ 3.6)	68.6($\uparrow$ 2.9)	**87.3**($\uparrow$ 3.2)	**91.0**($\uparrow$ 2.1)
L9-12	59.8($\uparrow$ 3.3)	**69.7**($\uparrow$ 4.0)	86.6($\uparrow$ 2.5)	90.6($\uparrow$ 1.7)
L1-12	59.5($\uparrow$ 3.0)	**69.7**($\uparrow$ 4.0)	83.8($\downarrow$ 0.3)	88.7($\downarrow$ 0.2)

Table 3: Layer chosen for Self-Attention Pooling.

Limitation Our method requires that element representation learning involves self-attention mechanisms. Nevertheless, our scope of application is still wide because the self-attention mechanism has proven to be dramatically useful in various fields, such as natural language processing (Vaswani et al., 2017), graph models (Veličković et al., 2018), and computer vision (Dosovitskiy et al., 2021).

6 Conclusion

We propose Self-Attention Pooling to learn representation and pooling simultaneously, allowing the structure representation learning to take element relation into account. Self-Attention Pooling substantially improves the sequential structure and non-sequential structure.

References

Luisa Bentivogli, Ido Kalman Dagan, Dang Hoa, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC 2009 Workshop*. no publisher.

Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

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*, Florence, Italy. Association for Computational Linguistics.

Fabrizio Costa and Kurt De Grave. 2010. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262. Omnipress; Madison, WI, USA.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*. Springer.

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*, Minneapolis, Minnesota. Association for Computational Linguistics.

Haoran Ding and Xiao Luo. 2021. AttentionRank: Unsupervised keyphrase extraction using self and cross attentions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paul D Dobson and Andrew J Doig. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Hongyang Gao and Shuiwang Ji. 2019. Graph u-nets. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Kristian Kersting, Nils M Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. 2016. Benchmark data sets for graph kernels.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*. PMLR.

Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2019. Universal Text Representation from Bert: An Empirical Study. *arXiv preprint arXiv:1910.07973*.

Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. 2019. Relational pooling for graph representations. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Francesco Orsini, Paolo Frasconi, and Luc De Raedt. 2015. Graph invariant kernels. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

U. Reimer and U. Hahn. 1988. Text condensation as knowledge base abstraction. In *[1988] Proceedings. The Fourth Conference on Artificial Intelligence Applications*.

Pooyan Safari, Miquel India, and Javier Hernando. 2020. Self-attention encoding and pooling for speaker recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*. ISCA.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Pitfalls of graph neural network evaluation.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*.

Richard Socher, Eric Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in neural information processing systems*.

Youwei Song, Jiahai Wang, Zhiwei Liang, Zhiyue Liu, and Tao Jiang. 2020. Utilizing BERT Intermediate Layers for Aspect based Sentiment Analysis and Natural Language Inference. *arXiv preprint arXiv:2002.04815*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Nikil Wale, Ian A Watson, and George Karypis. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, Tao Qi, Xiaohui Cui, and Yongfeng Huang. 2020. Attentive pooling with learnable norms for text representation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph convolutional networks for text classification.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Hao Yuan and Shuiwang Ji. 2020. Structpool: Structured graph pooling via conditional random fields. In *International Conference on Learning Representations*.

Model	CoLA	RTE	MRPC(ACC/F1)	QNLI	SST-2	STS-B	QQP	MNLI	Score
CLS Pooling	56.5	65.7	84.1/88.9	90.7	**92.3**	88.6	90.7	83.9	82.4
Mean Pooling	59.2	64.3	84.6/89.0	90.6	91.2	88.3	90.9	83.8	82.4
Max Pooling	59.1	63.5	81.4/87.7	90.7	91.2	87.9	**91.0**	**84.5**	81.8
Self-Attention Pooling	**59.8**	**69.7**	**86.6/90.6**	**90.8**	91.5	**89.3**	**91.0**	83.9	**83.7**

Table 4: Results on GLUE.

A Results on GLUE

We use the $\text{BERT}_{\text{base}}$ model implemented by Transformers (Wolf et al., 2020), and follow the default setting of their "text-classification" directory for the training and evaluation on GLUE without tuning *any* hyper-parameters. Table 4 shows the full results and average performance. For STS-B, we report Pearson metric. For other new tasks, we report accuracy. On average, Self-Attention Pooling improves CLS Pooling by 1.3 points.

B Experiment Details on Graph Classification

Our experiments on graph classification (Section 4.2) follow the implementation of the "proteins_topk_pool.py" file in pytorch-geometric (Fey and Lenssen, 2019). We set three GNN layers and apply pooling for each layer, retaining 80% nodes at a time. The Self-Attention Pooling implemented on each layer only takes the self-attention of the current layer into account.