SparseBERT: Rethinking the Importance Analysis in Self-attention

Han Shi [1] **Jiahui Gao** [2] **Xiaozhe Ren** [3] **Hang Xu** [3] **Xiaodan Liang** [4] **Zhenguo Li** [3] **James T. Kwok** [1]

Abstract

Transformer-based models are popularly used in natural language processing (NLP). Its core component, self-attention, has aroused widespread interest. To understand the self-attention mechanism, a direct method is to visualize the attention map of a pre-trained model. Based on the patterns observed, a series of efficient Transformers with different sparse attention masks have been proposed. From a theoretical perspective, universal approximability of Transformer-based models is also recently proved. However, the above understanding and analysis of self-attention is based on a pre-trained model. To rethink the importance analysis in self-attention, we study the significance of different positions in attention matrix during pre-training. A surprising result is that diagonal elements in the attention map are the least important compared with other attention positions. We provide a proof showing that these diagonal elements can indeed be removed without deteriorating model performance. Furthermore, we propose a Differentiable Attention Mask (DAM) algorithm, which further guides the design of the SparseBERT. Extensive experiments verify our interesting findings and illustrate the effect of the proposed algorithm.

1. Introduction

The Transformer (Vaswani et al., 2017) has been commonly used in various natural language processing (NLP) tasks such as text classification (Wang et al., 2018a), text translation (Ott et al., 2018), and question answering (Rajpurkar et al., 2016). The recent use of Transformer for image classification (Dosovitskiy et al., 2021), object detection (Carion et al., 2020) also demonstrates its potential in computer vision. Two notable descendants from the Transformer include the BERT (Devlin et al., 2019), which achieves state-of-the-art performance on a wide range of NLP tasks, and GPT-3 (Brown et al., 2020) which applies the Transformer's decoder on generative downstream tasks.

Self-attention is a core component in Transformer-based architectures. Recently, its interpretation has aroused a lot of interest. Visualization has been commonly used to understand the attention map during inference (Park et al., 2019; Gong et al., 2019; Kovaleva et al., 2019). For example, Park et al. (2019) and Gong et al. (2019) randomly select a sentence from the corpus and visualize the attention maps of different heads in a pre-trained Transformer model. Kovaleva et al. (2019) summarizes five attention patterns and estimates their ratios in different tasks. A common observation from these studies is that local attention and global attention are both important for token understanding.

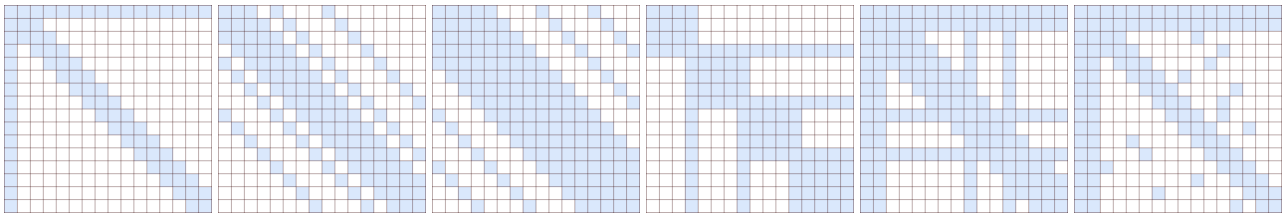
While self-attention is powerful, a main concern is its efficiency bottleneck. As each token has to attend to all n tokens in the sequence, the complexity scales as $\mathcal{O}(n^2)$. This can be expensive on long sequences. To alleviate this problem, sparse attention allows each token to attend to only a token subset. A series of Transformer variants have been proposed along this direction (Guo et al., 2019; Child et al., 2019; Li et al., 2019; Beltagy et al., 2020). However, these sparse attention schemes are designed manually. It is still an open issue on how to find a suitable attention scheme.

Recently, there is a growing interest in understanding self-attention mechanism from a theoretical perspective. Results show that the Transformer and its variants are universal approximators of arbitrary continuous sequence-to-sequence functions (Yun et al., 2019; 2020; Zaheer et al., 2020). A key part of their proofs is that self-attention layers implement contextual mappings of the input sequences. Yun et al. (2019) constructs the self-attention model as a selective shift operation such that contextual mapping can be implemented. Zaheer et al. (2020) shows that universal approximation holds for their sparse Transformer BigBird if its attention structure contains the star graph. Yun et al. (2020) provides a unifying framework for the universal approximation of sparse Transformers. Note that they all emphasize the importance of diagonal elements in the attention map.

To guide the design of an efficient Transformer, it is useful to investigate the importance of different positions in

[1]Hong Kong University of Science and Technology, Hong Kong [2]The University of Hong Kong, Hong Kong [3]Huawei Noah's Ark Lab [4]Sun Yat-sen University, China. Correspondence to: Han Shi <hshiac@cse.ust.hk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.



(a) Star. (b) LogSparse. (c) Strided. (d) Fixed. (e) Longformer. (f) BigBird.

Figure 1. Examples of existing attention masks (with $n = 16$).

self-attention. In this paper, we study this using differentiable search (Liu et al., 2019; Xie et al., 2018). A learnable attention distribution is constructed and the score of each position is learned in an end-to-end manner during pre-training. While existing theoretical and empirical findings suggest the importance of diagonal elements in the self-attention matrix, we observe that they are indeed the least important compared to other entries. Furthermore, neighborhood tokens and special tokens (such as the first token `[CLS]` and last token `[SEP]`) are also prominent, which is consistent with previous observations in (Park et al., 2019; Gong et al., 2019; Kovaleva et al., 2019; Clark et al., 2019). Besides, using the Gumbel-sigmoid function (Maddison et al., 2017), we propose the Differentiable Attention Mask (DAM) algorithm to learn the attention mask in an end-to-end manner. Extensive experiments using masks with different sparsity ratios on various NLP tasks demonstrate the effect of the proposed algorithm. Specifically, highly sparse structured attention masks (with 91.3% sparsity ratio) can already achieve 80.9% average score on the GLUE development set (Wang et al., 2018a). The code is available at `https://github.com/han-shi/SparseBERT`.

2. Related Work

2.1. Transformer Block and Self-Attention

The Transformer block is a basic component in the Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019) architectures. Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ be the input to a Transformer block, where n is the number of input tokens and d is the embedding size. Each block consists of a self-attention layer and a feed-forward layer.

The self-attention layer output can be written as:

$$\begin{aligned} Attn(\boldsymbol{X}) &= \boldsymbol{X} + \sum_{k=1}^{H} \sigma(\boldsymbol{X}\boldsymbol{W}_Q^k(\boldsymbol{X}\boldsymbol{W}_K^k)^\top)\boldsymbol{X}\boldsymbol{W}_V^k\boldsymbol{W}_O^{k\top} \\ &= \boldsymbol{X} + \sum_{k=1}^{H} \boldsymbol{A}^k(\boldsymbol{X})\boldsymbol{V}^k(\boldsymbol{X})\boldsymbol{W}_O^{k\top}, \end{aligned} \tag{1}$$

where H is the number of heads, σ is the softmax function, and $\boldsymbol{W}_Q^k, \boldsymbol{W}_K^k, \boldsymbol{W}_V^k, \boldsymbol{W}_O^k \in \mathbb{R}^{d\times d_h}$ (where $d_h = d/H$ is the dimension of a single-head output) are weight matrices for the query, key, value, and output, respectively of the kth head. In particular, the self-attention matrix

$$\boldsymbol{A}(\boldsymbol{X}) = \sigma(\boldsymbol{X}\boldsymbol{W}_Q(\boldsymbol{X}\boldsymbol{W}_K)^\top) \tag{2}$$

in (1) plays a key role in the self-attention layer (Park et al., 2019; Gong et al., 2019; Kovaleva et al., 2019).

The fully-connected layer usually has two layers with residual connection:

$$FF(\boldsymbol{X}) = Attn(\boldsymbol{X}) + ReLU(Attn(\boldsymbol{X})\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2,$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d\times d_{\text{ff}}}, \boldsymbol{W}_2 \in \mathbb{R}^{d_{\text{ff}}\times d}$ (d_{ff} is the size of the intermediate layer) are the weight matrices, and $\boldsymbol{b}_1, \boldsymbol{b}_2$ are the biases. As in (Yun et al., 2019; Zaheer et al., 2020; Yun et al., 2020), we drop the scale product and layer-normalization layer to simplify analysis.

2.2. Sparse Transformers

To reduce the quadratic complexity in self-attention, a number of sparse Transformers have been recently proposed. In these models, each token can attend to only a subset of fixed positions (Guo et al., 2019; Child et al., 2019; Li et al., 2019; Beltagy et al., 2020). This can be seen as being controlled by an attention mask $\boldsymbol{M} = [0, 1]^{n\times n}$, where $M_{i,j} = 1$ indicates that token i can attend to token j, and 0 otherwise. For example, Figure 1(a) shows the attention mask of the Star-Transformer (Guo et al., 2019). It uses ring connections for local attention, and radical connections to an auxiliary relay node (the first token in the figure) to represent global attention. Li et al. (2019) proposes the LogSparse self-attention, in which each token only attends to itself and its previous tokens with an exponential stepsize (Figure 1(b)), resulting in $\mathcal{O}(n \log n)$ complexity. Child et al. (2019) performs sparse factorization on the attention matrix, and reduces its complexity to $\mathcal{O}(n\sqrt{n})$ with the use of two attention masks. The strided mask (Figure 1(c)) attends to every lth location (where l is the stride step), while the fixed mask (Figure 1(d)) allows specific positions to be attended to. The very recent Longformer (Beltagy et al., 2020) (Figure 1(e)) and BigBird (Zaheer et al., 2020) (Figure 1(f)) models use a number of attention patterns, and reduce their complexities

to $\mathcal{O}(n)$. With these sparse Transformers, BERT is shown to be more efficient for long document understanding (Child et al., 2019; Qiu et al., 2020).

3. Which Attention Positions are Important?

Previous works on sparse Transformers only provide a crude understanding of the self-attention module that local attention and global attention are both important. In this section, we study the self-attention matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ in Eq. (2) in more detail. To emphasize its role, we write the output of the self-attention layer as $Attn(\boldsymbol{X}, \boldsymbol{A}(\boldsymbol{X}, \boldsymbol{M}))$, where $\boldsymbol{M}$ is a fixed attention mask. Since the nonzero elements of the attention matrix are fixed, one only needs to perform computations related to these positions. We define the sparsity of an attention mask as $\rho = 1 - |\boldsymbol{M}|/n^2$. The complexity of the self-attention layer is thus reduced to $\mathcal{O}((1-\rho)n^2)$.

With a low self-attention sparsity, a token can attend to more tokens given the same amount of computational cost, and is thus expected to have better performance. On the other hand, at high self-attention sparsity, model performance may drop. It is natural to ask which positions in self-attention are more important. In other words, which attention mask is better for a given sparsity. We formulate this problem as the search for a mask in $[0,1]^{n\times n}$ such that the balance between performance and efficiency is optimized.

3.1. Continuous Relaxation

In this section, we investigate the importance of different positions in self-attention. A similar study on the importance of different heads in the Transformer is recently performed in (Michel et al., 2019). However, while Michel et al. (2019) performs the ablation study with only 16 heads, there are $2^{n\times n}$ possible attention distributions here. This huge search space makes the study very challenging.

In neural architecture search (NAS) (Elsken et al., 2019), one has to find a good architecture from a huge search space. Inspired by the similarity with our problem, we propose to use continuous relaxation as in differentiable architecture search (DARTS) (Liu et al., 2019). Specifically, we associate an $\alpha_{i,j}$ with each position (i,j) in the self-attention matrix $\boldsymbol{A}(\boldsymbol{X})$, and define the attention probability as

$$P_{i,j} = \text{sigmoid}(\alpha_{i,j}) \in [0,1]. \tag{3}$$

For symmetric structure, we enforce $\alpha_{i,j} = \alpha_{j,i}$. Analogous to Eq. (1), the soft-masked self-attention is then

$$Attn(\boldsymbol{X}) = \boldsymbol{X} + \sum_{k=1}^{H} (\boldsymbol{P}^k \odot \boldsymbol{A}^k(\boldsymbol{X}))\boldsymbol{V}^k(\boldsymbol{X})\boldsymbol{W}_O^{k\top}, \tag{4}$$

where $\odot$ is the element-wise product. Obviously, when $P_{i,j} = 1$ for all (i,j)'s, this reduces to Eq. (1). However, the above multiplicative attention mask will result in unnormalized attention distributions. To solve this problem, we introduce the renormalization trick, which replaces the multiplicative attention mask with an additive mask before the softmax function as follows.

$$\hat{\boldsymbol{A}}(\boldsymbol{X}) = \sigma(\boldsymbol{X}\boldsymbol{W}_Q(\boldsymbol{X}\boldsymbol{W}_K)^\top + \boldsymbol{Q}), \tag{5}$$

$$Q_{i,j} = -c(1-P_{i,j}), \tag{6}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ is the addictive attention mask, c is a large constant such that $\hat{A}_{i,j} = 0$ if $P_{i,j} = 0$, and $\hat{A}_{i,j}$ reduces to the original attention score if $P_{i,j} = 1$.

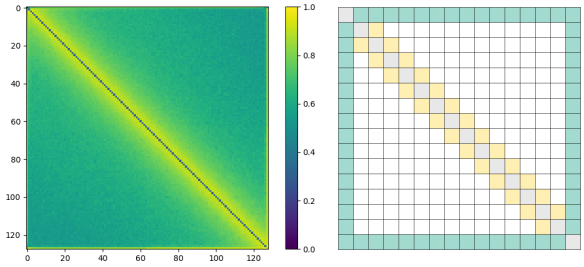
As in DARTS, there are two sets of learnable parameters: parameter $\boldsymbol{w}$ of the Transformer model, and the attention parameter $\boldsymbol{\alpha} = \{\alpha_{i,j}\}$. They can be learned by using either one-level optimization (Xie et al., 2018) or bi-level optimization (Liu et al., 2019) formulations. Recently, Bi et al. (2020) shows that the limitations of one-level optimization can be alleviated when a large data set is used. In our context, as a large data set is often available during pre-training, we apply the simpler one-level optimization here.

3.1.1. Experimental Setup

In this experiment, we empirically study the effect of different positions in the self-attention module using the BERT-base. This model is stacked with 12 Transformer blocks (Section 2.1) with the following hyper-parameters: number of tokens $n = 128$, number of self-attention heads $h = 12$, and hidden layer size $d = 768$. For better comparison with prior works in Figure 1, the parameters of $\boldsymbol{P}$ are also shared among blocks, leading to a total of 12 $\boldsymbol{P}$'s (one for each self-attention head). As for the feed-forward layer, we set the filter size d_{ff} to 3072 as in (Devlin et al., 2019). We follow the standard pre-training experiment setting in (Devlin et al., 2019), and take Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) as pre-training tasks. Data sets BooksCorpus (with 800M words) (Zhu et al., 2015) and English Wikipedia (with 2,500M words) (Devlin et al., 2019) are used. We use the WordPiece embedding (Wu et al., 2016), and 30,000 tokens are contained in the dictionary. The special token `[CLS]` is used as the first token of each sequence. Another special token `[SEP]` is used to separate sentences in a sequence. The pre-training is performed for 40 epochs. All experiments are performed on NVIDIA Tesla V100 GPUs.

3.1.2. Results

Figure 2(a) shows the attention distribution $\boldsymbol{P}$ averaged over the 12 heads after normalization (with $n = 128$). For comparison with existing attention masks in Figure 1, we also illustrate its sketch in Figure 2(b) (with $n = 16$). The following can be observed: (i) Diagonal elements (denoted "diag-attention" in the sequel) are the least important com-



(a) Attention distribution. (b) Sketch.

Figure 2. Visualization of the attention distribution. In Figure 2(b), the dark entries are for diag-attention, yellow for neighborhood attention and green for attention to special tokens.

pared to other positions. Surprisingly, this conflicts with existing observations in (Park et al., 2019; Gong et al., 2019; Kovaleva et al., 2019), which emphasize the importance of diagonal attention. We believe this is because the self-attention layer already has a skip connection (the term $\boldsymbol{X}$ in (1)). Influence of diag-attention can thus be conveyed in the skip connection instead of via the self-attention matrix; (ii) Neighborhood positions are the most significant in the attention distribution matrix; (iii) The special tokens (`[CLS]` and `[SEP]`) are important, which is also observed in (Clark et al., 2019); (iv) The importance of other positions is similar.

3.2. Universal Approximability

Recall that the Transformer and its variants are universal approximators of arbitrary continuous sequence-to-sequence functions (Yun et al., 2019; 2020; Zaheer et al., 2020). In their proofs, diagonal positions in the self-attention matrix play a key role. As Section 3.1 has shown that the diagonal elements are empirically the least important, an interesting question is whether universal approximability will still hold when these diagonal elements (diag-attention) are dropped.

Without diag-attention, the ith token output of the self-attention layer becomes:

$$Attn(\boldsymbol{X})_i = \boldsymbol{X}_i + \sum_{k=1}^{H}\sum_{j\neq i} A^k_{i,j}(\boldsymbol{X})\boldsymbol{V}^k_j(\boldsymbol{X})\boldsymbol{W}^{k\top}_O.$$

Let $\mathcal{T}^{H,d_h,d_{\text{ff}}}$ be a class of Transformers without diag-attention stacks, and F_{CD} be the set of continuous functions $f : [0,1]^{n\times d} \mapsto \mathbb{R}^{n\times d}$. For any $p \geq 1$, the ℓ_p-distance between $f_1, f_2 \in F_{CD}$ is defined as $d_p(f_1, f_2) = (\int \|f_1(\boldsymbol{X}) - f_2(\boldsymbol{X})\|^p_p d\boldsymbol{X})^{1/p}$. The following Theorem shows that the self-attention mechanism without diag-attention is also a universal approximator:

Theorem 1. *Given $1 < p < \infty$, $\epsilon > 0$ and $n > 2$, for any $f \in F_{CD}$, there exists a Transformer network without diag-attention $g \in T^{2,1,4}$, such that $d_p(f, g) < \epsilon$.*

The following shows the proof outline, which is similar to that in (Yun et al., 2019). The main difference is in the contextual mapping step since each token cannot attend to itself in our scenario.

Step 1: Approximate F_{CD} with the set of piecewise-constant functions $\bar{F}_{CD}$. We split input $[0,1]^{n\times d}$ into a set of grids $\mathcal{G}_\delta \in \{0, \delta, \ldots, 1\}^{n\times d}$. We then approximate any input belonging to the same cube $\mathcal{G}_\delta + [0,\delta]^{n\times d}$ by the same value, resulting in a piecewise-constant function $\bar{f}$. With δ small enough, we have $d_p(f, \bar{f}) \leq \epsilon/3$.

Step 2: Approximate $\bar{F}_{CD}$ with the modified Transformer blocks $\bar{\mathcal{T}}^{H,d_h,d_{\text{ff}}}$, which replace the softmax operator and ReLU with the hardmax operator and a piece-wise linear functions (at most three pieces). For each above $\bar{f}$, there exists a closely approximate function $\bar{g} \in \bar{\mathcal{T}}^{2,1,1}$ such that $d_p(\bar{f}, \bar{g}) = O(\delta^{d/p})$.

This is the key step related to the contextual mapping. A selective shift operation is proposed to construct an approximation in (Yun et al., 2019). Here, we consider a simple scenario where $n = 3$ and $d = 1$ and let $\boldsymbol{L} = [l_1 \quad l_2 \quad l_3]^\top \in \mathcal{G}_\delta$. Without loss of generality, we assume that $l_1 < l_2 < l_3$. For a Transformer without diag-attention, the selective shift operation, consisting of 2 attention heads of size 1, is constrained as follows:

$$\Psi(\boldsymbol{Z}; b_1, b_2)_{i,1} = \begin{cases} \max_{j\neq i} Z_{j,1} - \min_{j\neq i} Z_{j,1} & \text{if } b_1 < Z_{i,1} < b_2 \\ 0 & \text{otherwise} \end{cases}.$$

We stack $1/\delta$ self-attention layers, with attention parts $\delta^{-1}\Psi(\cdot; l-\delta/2, l+\delta/2)$ for each $l \in \{0, \delta, \ldots, 1-\delta\}$ in increasing order of l. The shift operation is first applied to l_1, resulting in $\widetilde{l_1} = l_1 + \delta^{-1}(l_3 - l_2) > l_3$. The second shift operation is then applied to the second element, resulting in $\widetilde{l_2} = l_2 + \delta^{-1}(\widetilde{l_1} - l_3) = l_2 + \delta^{-1}(l_1 - l_3) + \delta^{-2}(l_3 - l_2) > \widetilde{l_1}$. Finally, a similar operation is applied to l_3, and the shifted result is $\widetilde{l_3} = l_3 + \delta^{-1}(\widetilde{l_2} - \widetilde{l_1}) = l_3 + \delta^{-1}(l_2 - l_1) + \delta^{-2}(l_1 + l_2 - 2l_3) + \delta^{-3}(l_3 - l_2)$. It is easy to check that the map from the original $\boldsymbol{L}$ to $\widetilde{l_3}$ is one-to-one and that $0 < \widetilde{l_1} < \widetilde{l_2} < \widetilde{l_3} < \delta^{-3}$. We then add two additional layers shifting all positive elements, resulting in $[\widetilde{l_1} + \Delta(\widetilde{l_2} + \widetilde{l_3}) + \Delta^2\widetilde{l_3} \quad \widetilde{l_2} + \Delta(\widetilde{l_1} + \widetilde{l_3}) + \Delta^2\widetilde{l_3} \quad \widetilde{l_3} + 2\Delta\widetilde{l_2} + \Delta^2\widetilde{l_3}]^\top$, where $\Delta = (\delta^{-1} - 1)(\delta^{-3} + \delta^{-1} + 1)$. Note that all elements are in the disjoint interval for different $\boldsymbol{L}$'s because $\boldsymbol{L} \to \widetilde{l_3}$ is bijective. Thus, the self-attention layer without diag-attention is a contextual mapping as defined in (Yun et al., 2019).

Step 3: Approximate the modified Transformer blocks $\bar{g} \in \bar{\mathcal{T}}^{2,1,1}$ with standard Transformer blocks $g \in \mathcal{T}^{2,1,4}$ such that we have $d_p(\bar{g}, g) \leq \epsilon/3$.

Table 1. Performance (in %) of the various BERT-base variants on the GLUE data set.

	MNLI (m/mm)	QQP	QNLI	SST-2	COLA	STS-B	MRPC	RTE	Average
Development Set									
BERT-base (ours)	85.4/85.8	88.2	91.5	92.9	62.1	88.8	90.4	69.0	83.8
BERT-base (randomly dropped)	84.6/85.2	87.7	91.1	92.7	62.0	88.9	89.3	68.9	83.4
BERT-base (no diag-attention)	85.6/85.9	88.2	92.0	93.8	63.1	89.2	91.2	67.9	**83.9**
Test Set									
BERT-base (Devlin et al., 2019)	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT-base (ours)	84.8/84.1	71.3	90.9	93.4	52.3	85.3	88.3	66.9	79.7
BERT-base (randomly dropped)	84.5/83.5	70.3	91.1	93.4	52.0	85.8	87.4	66.7	79.4
BERT-base (no diag-attention)	85.5/84.9	71.3	91.1	93.4	53.3	86.3	88.9	67.9	**80.3**

Table 2. Performance (in %) of the various BERT-base variants on the SWAG and SQuAD development sets.

	SWAG	SQuAD v1.1		SQuAD v2.0	
	acc	EM	F1	EM	F1
BERT-base (Devlin et al., 2019)	81.6	80.8	88.5	-	-
BERT-base (ours)	82.5	79.7	87.1	72.9	75.5
BERT-base (randomly dropped)	81.6	79.7	87.0	71.5	74.2
BERT-base (no diag-attention)	83.5	80.3	87.9	73.2	75.9

Summarizing the above three steps, we have:

$$d_p(f, g) \leq d_p(f, \bar{f}) + d_p(\bar{f}, \bar{g}) + d_p(\bar{g}, g) \leq 2\epsilon/3 + O(\delta^{d/p}).$$

With a small enough δ, we have $d_p(f, g) \leq \epsilon$. Thus, Transformers without diag-attention are also universal approximators. The detailed proof is in Appendix A.

3.3. Empirical Verification

In this section, we empirically study the effect of dropping diag-attention from the self-attention mechanism. The fine-tuning experiments are performed on the GLUE benchmark (Wang et al., 2018a), SWAG (Zellers et al., 2018) and SQuAD (Rajpurkar et al., 2016; 2018) data sets.

3.3.1. Data

The GLUE benchmark includes three categories of natural language understanding tasks: (i) single-sentence tasks (CoLA and SST-2); (ii) similarity and paraphrase tasks (MRPC, QQP and STS-B); (iii) inference tasks (MNLI, QNLI and RTE). For MNLI sub-task, we experiment on both the matched (MNLI-m) and mismatched (MNLI-mm) sections. The SWAG data set is for grounded commonsense inference, while the SQuAD data set is for question answering. In SQuAD v1.1, the answers are included in the context; while in SQuAD v2.0, some answers are not included. Descriptions of the data sets are in Appendix B.

3.3.2. Metric

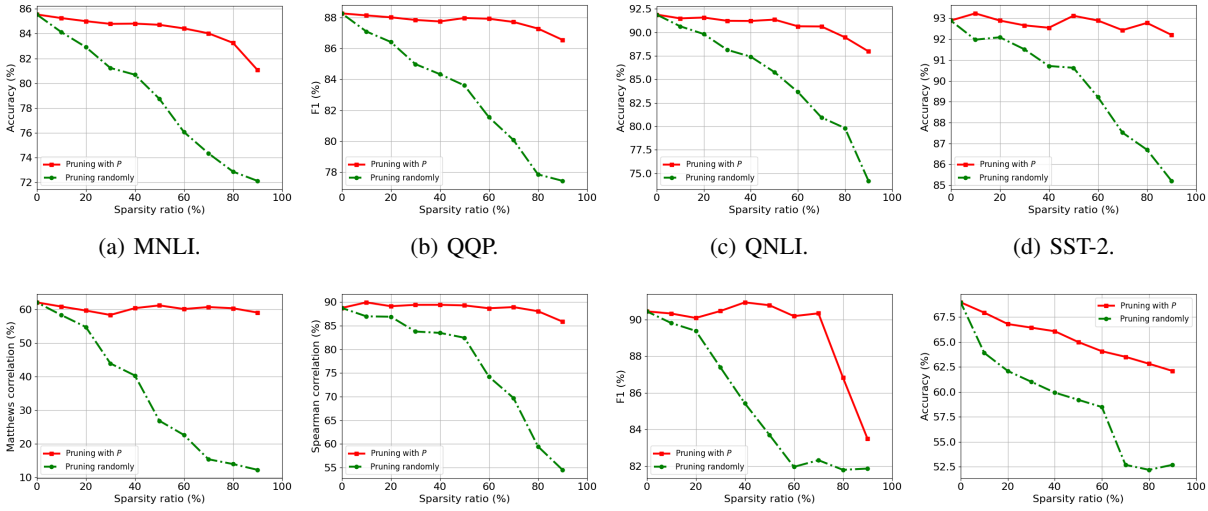
Following BERT (Devlin et al., 2019), we report different metrics for different GLUE sub-tasks. Specifically, we use accuracy for MNLI, QNLI, RTE, SST-2 tasks, F1 score for QQP and MRPC, Spearman correlation for STS-B, and Matthews correlation for CoLA. The results of GLUE test set are evaluated by the evaluation server (`https://gluebenchmark.com`). For SWAG task, we use accuracy for evaluation. For SQuAD v1.1 and v2.0, we use the Exact Match (EM) and F1 scores.

3.3.3. Details

We perform pre-training and fine-tuning. The experiment settings are the same as Devlin et al. (2019). We take the BERT-base model in Section 3.1 as the baseline and verify the effect of diagonal elements in the self-attention matrix. In the variant "BERT-base (no diag-attention)", the diag-attention in each self-attention layer is removed, dropping a total of 128 elements (as $n = 128$). In "BERT-base (randomly dropped)", we remove the same number (128) of entries randomly. To reduce statistical variability, the results are averaged over three random repetitions. Details of other hyper-parameter settings are in Appendix C. We choose the best hyper-parameter combination on the development set and test it on the evaluation server.

3.3.4. Results

Results on the GLUE benchmark are shown in Table 1. For comparison, we also show the BERT-base results reported in Devlin et al. (2019), and BERT-base (ours) is reproduced by ourselves. As can be seen, even by constraining the self-attention mechanism such that each token cannot attend to itself, the performance of this constrained model "BERT-base (no diag-attention)" is still comparable with the



(a) MNLI. (b) QQP. (c) QNLI. (d) SST-2.

(e) CoLA. (f) STS-B. (g) MRPC. (h) RTE.

Figure 3. Performance of the BERT-base at different attention mask sparsity ratios on the GLUE development set. MNLI is the average performance on the MNLI-m and MNLI-mm sections.

original or even better than BERT-base. On the other hand, when the sparse attention masks are randomly chosen, the performance drops. Table 2 shows the results on SWAG and SQuAD, and the observations are similar. This demonstrates that attentions to the diagonal positions are not necessary. Without attending to self-token, the model performance is not deteriorated.

3.4. Progressive Pruning of Self-attention

To further investigate the effectiveness of the searched mask, we perform progressive pruning of self-attention according to the results in Section 3.1.2. Experiments are performed on the GLUE benchmark and the experiment settings are the same. We threshold $\boldsymbol{P}$ to a binary attention mask $\boldsymbol{M}$ so that a part of entries in $\boldsymbol{P}$ are pruned. We investigate the performance with different mask sparsity ratios. Specifically, for each head, we remove the smallest $10\%/20\%/\ldots/90\%$ entries from $\boldsymbol{P}$ and keep other positions active. For comparison, we include a random baseline that randomly removes the same number of entries.

As can be seen from Figure 3, pruning by the attention distribution matrix $\boldsymbol{P}$ consistently outperforms random pruning on all GLUE tasks. This verifies the importance analysis in Section 3.1.2. Moreover, models from different GLUE sub-tasks can have different degrees of redundancy. For instance, performance on CoLA sub-task remains stable when sparsity ratio gets higher. In contrast, the RTE sub-task performance shows a significant drop as the model is sparsified, illustrating the need for a denser self-attention computation. Thus, we can select different sparsity ratios for different tasks to balance performance and efficiency.

4. SparseBERT

As discussed in Section 2.2, vanilla self-attention suffers from quadratic complexity, and a number of sparse Transformers have been proposed (Guo et al., 2019; Child et al., 2019; Li et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020). However, their attention masks are manually designed. In this section, we use the observations in the previous sections to help develop a series of sparse attention masks with different sparsity ratios.

4.1. Differentiable Attention Mask

A straightforward method to generate sparse Transformers is by pruning the attention distribution $\boldsymbol{P}$ as is performed in Section 3.4. However, this involves two separate stages. The resultant performance may be sub-optimal as the whole model is not trained end-to-end. Moreover, discretization of the continuous attention distribution $\boldsymbol{P}$ may mislead the final attention mask.

To enable end-to-end training, we propose to use the Gumbel relaxation (Maddison et al., 2017). Instead of using the sigmoid function to output the attention probability as in (3), we use the Gumbel-sigmoid:

$$M_{i,j} = \text{Gumbel-sigmoid}(\alpha_{i,j}) = \text{sigmoid}((\alpha_{i,j} + G_1 - G_2)/\tau),$$

where G_1, G_2 are independent Gumbel noises generated from the uniform distribution U as:

$$G_k = -\log(-\log(U_k)), U_k \sim U(0,1),$$

and τ is a temperature hyperparameter. As τ approaches zero, the Gumbel-sigmoid output becomes a discrete distribution in $\{0, 1\}$. Thus, we can train the attention mask in

Algorithm 1 Differentiable Attention Mask (DAM).

initialize model parameter $\boldsymbol{w}$ and attention mask parameter $\boldsymbol{\alpha}$.
repeat
  generate mask $M_{i,j} \leftarrow \text{gumbel-sigmoid}(\alpha_{i,j})$;
  obtain the loss with attention mask $\mathcal{L}$;
  update parameter $\boldsymbol{w}$ and $\boldsymbol{\alpha}$ simultaneously;
until convergence.
return attention mask $\boldsymbol{M}$.

an end-to-end manner with the Gumbel-sigmoid variant. To balance mask sparsity with performance, we add the sum absolute values of the attention mask to the loss, as:

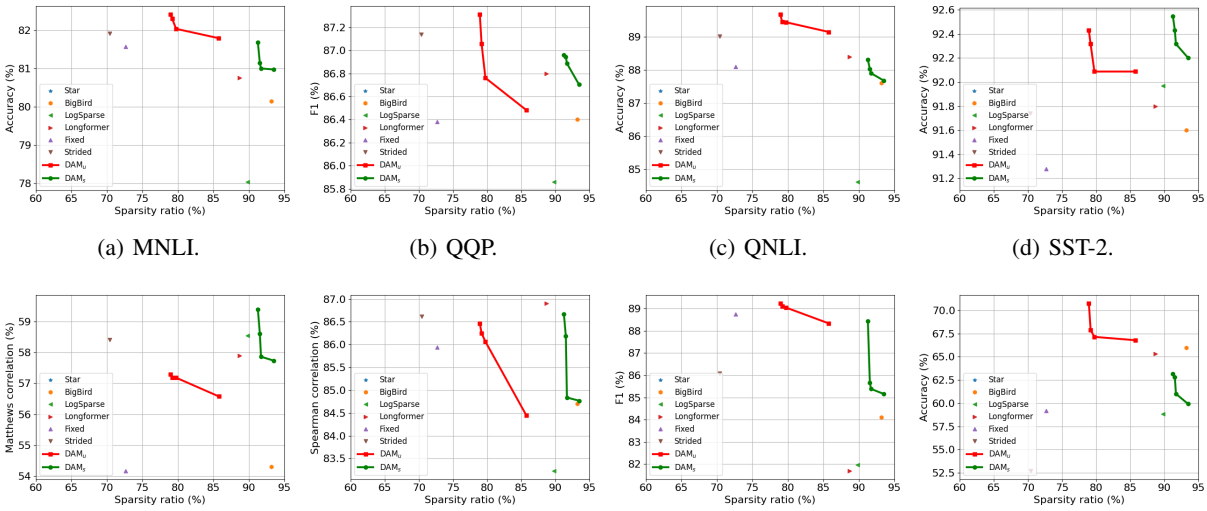
$$\mathcal{L} = l(\text{BERT}(\boldsymbol{X}, \boldsymbol{A}(\boldsymbol{X}) \odot \boldsymbol{M}(\boldsymbol{\alpha}); \boldsymbol{w})) + \lambda \|\boldsymbol{M}(\boldsymbol{\alpha})\|_1, \quad (7)$$

where $l(\text{BERT}(\boldsymbol{X}, \boldsymbol{A}(\boldsymbol{X}); \boldsymbol{w}))$ is the pre-training loss, and λ is a trade-off hyperparameter. When λ is large, we pay more emphasis on efficiency and the learned attention mask $\boldsymbol{M}$ is sparser, and vice versa. We also apply the renormalization trick to normalize the attention distributions in Eq. (7). The obtained one-hot Gumbel sigmoid output can then be directly used as the attention mask. The whole algorithm is shown in Algorithm 1. We optimize both parameter sets simultaneously in one-level optimization until convergence and the generated mask is returned finally.

Note that the attention mask obtained in Algorithm 1 is unstructured, as the $\alpha_{i,j}$'s are all independent of each other. This irregular structure may affect the efficiency of the final CUDA implementation. To alleviate this problem, we use the observed sparsity patterns in Figure 2 to constrain the structure of the attention mask. First, as the special tokens are important, we require the first and last row/column of the attention mask to be active. Second, for all positions on each line parallel to the diagonal, we share their mask parameters including their two Gumbel noises such that the generated mask has $M_{i,j} = M_{i+k,j+k}$ for integer k. Among the previously proposed attention masks, the Sparse Transformer (strided) (Child et al., 2019) and LogSparse Transformer (Li et al., 2019) conform to our definition of structured attention masks, and they can be implemented efficiently by custom CUDA kernels. With the constrained structure, there are now only $n-2$ attention mask parameters. Therefore, the search space is reduced to 2^{n-2} in structured attention mask search. Since the sparsity ratio directly affects the efficiency of the SparseBERT, we will show the performance of the attention mask with its sparsity ratio.

4.2. Experiments

As in previous sections, we evaluate the proposed DAM algorithm by using the BERT-base (Devlin et al., 2019) model on the GLUE data sets (Wang et al., 2018a) with pre-training and fine-tuning paradigm (Devlin et al., 2019). We use the same pre-training settings as in Section 3.1 with the dynamic attention mask. As for fine-tuning, the attention mask is fixed and other experiment settings are the same as Section 3.3. The DAM variant using unstructured attention mask is denoted DAM_u, while the one using structured attention mask is denoted DAM_s. The trade-off hyperparameter λ in (7) is varied in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ to generate masks with different sparsities for both structured and unstructured attention masks.



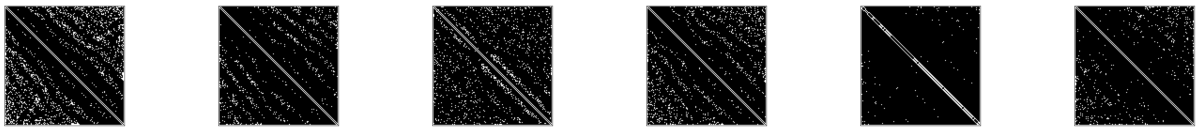
(a) MNLI. (b) QQP. (c) QNLI. (d) SST-2.

(e) CoLA. (f) STS-B. (g) MRPC. (h) RTE.

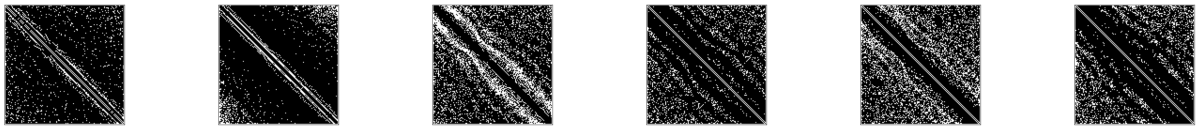
Figure 4. Performance of the BERT-base for different attention masks on the GLUE development set. MNLI shows the average performance on the MNLI-m and MNLI-mm sections.

Table 3. Comparison of BERT-base model with different attention masks on the GLUE development set (%).

	Sparsity ratio	MNLI-(m/mm)	QQP	QNLI	SST-2	COLA	STS-B	MRPC	RTE	Average
Strided (Child et al., 2019)	70.4	81.9/81.9	87.1	89.0	91.7	58.4	86.6	86.1	52.7	79.5
Fixed (Child et al., 2019)	72.7	81.4/81.8	86.4	88.1	91.3	54.2	85.9	88.7	59.2	79.7
Longformer (Beltagy et al., 2020)	88.7	80.5/81.0	86.8	88.4	91.8	57.9	86.9	81.7	65.3	80.1
LogSparse (Li et al., 2019)	89.8	77.9/78.2	85.9	84.6	92.0	58.5	83.2	82.0	58.8	77.9
BigBird (Zaheer et al., 2020)	93.2	80.2/80.1	86.4	87.6	91.6	54.3	84.7	84.1	66.0	79.4
Star (Guo et al., 2019)	96.1	79.1/79.0	86.2	86.4	91.2	59.6	84.7	83.9	60.3	78.9
DAM_u ($\lambda = 10^{-4}$)	78.9	82.2/82.6	87.3	89.7	92.4	57.3	86.5	89.2	70.8	82.0
DAM_u ($\lambda = 10^{-3}$)	79.2	82.2/82.4	87.1	89.5	92.3	57.2	86.2	89.1	67.9	81.6
DAM_u ($\lambda = 10^{-2}$)	79.8	81.7/82.3	86.8	89.4	92.1	57.2	86.1	89.0	67.1	81.3
DAM_u ($\lambda = 10^{-1}$)	85.8	81.4/82.2	86.5	89.1	92.1	56.6	84.4	88.3	66.8	80.8
DAM_s ($\lambda = 10^{-4}$)	91.2	81.7/81.7	87.0	88.3	92.5	59.4	86.7	88.4	63.2	80.9
DAM_s ($\lambda = 10^{-3}$)	91.6	81.0/81.2	86.9	88.0	92.4	58.6	86.2	85.7	62.8	80.3
DAM_s ($\lambda = 10^{-2}$)	91.7	81.1/80.9	86.9	87.9	92.3	57.9	84.8	85.4	61.0	79.8
DAM_s ($\lambda = 10^{-1}$)	93.5	80.9/81.0	86.7	87.7	92.2	57.7	84.8	85.2	59.9	79.6



(a) Head-1. (b) Head-2. (c) Head-3. (d) Head-4. (e) Head-5. (f) Head-6.



(g) Head-7. (h) Head-8. (i) Head-9. (j) Head-10. (k) Head-11. (l) Head-12.

Figure 5. Visualization of attention masks generated by DAM_u ($\lambda = 10^{-1}$). White means with-attention and dark means no-attention.

For comparison, we consider the following baselines with different manually-designed attention masks (Figure 1): (i) Star (Guo et al., 2019); (ii) LogSparse (Li et al., 2019); (iii) Strided (Child et al., 2019); (iv) Fixed (Child et al., 2019); (v) Longformer (Beltagy et al., 2020); and (vi) BigBird (Zaheer et al., 2020). For the Longformer, we set the sliding window size to 2 and two input locations are randomly selected for global attention. For the BigBird, we ignore its block structure for fair comparison and set its random attention size to 2, window attention size to 1, and global attention size to 2. The sparsity ratios of these fixed attention masks can be easily computed.

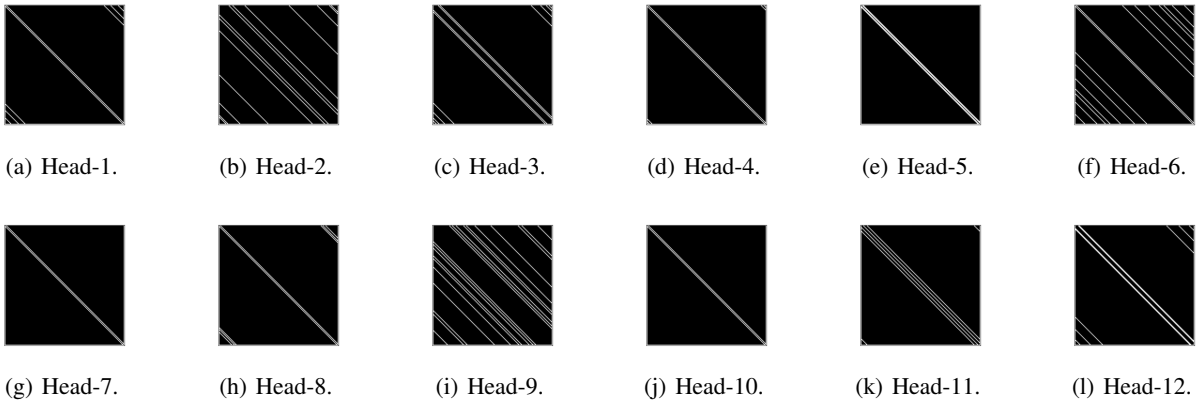
Performance on GLUE sub-tasks are shown in Figure 4 and Table 3. As can be seen, attention masks generated by the proposed method outperform manual-designed masks in almost all cases, and their sparsity ratios can be controlled by λ. For example, compared with BigBird, the proposed DAM_s ($\lambda = 10^{-1}$) achieves higher average performance while using a sparser attention mask.

Figures 5 and 6 show two ensembles of attention masks generated by DAM_u and DAM_s, respectively, after pre-training with $\lambda = 10^{-1}$. As can be seen, the diagonal entries are not attended, while their neighborhood positions are attended in most heads. From the visualizations of attention masks in Figure 1 and Figures 5-6, we can summarize them into three categories: (i) Stride/Fixed/Logsparse, which only contain neighboring attention; (ii) BigBird/Longformer/Star, which contain both neighboring attention and attention from special tokens; (iii) The proposed attention masks, which replace diag-attention with other attention positions from the second category. From the empirical performance on the GLUE benchmark, the first class is the worst, the second is competing while ours are the best, which agrees with the observations in Section 3.1.2. Moreover, DAM learns different attention masks for different heads, which utilizes the multi-head structure of the model.

4.3. Ablation Study

4.3.1. Effect of Diag-attention for Masks

In Section 3, we show the unimportance of diag-attention in the self-attention module. Here, we study the effect of diag-attention in different attention masks. Specifically, we perform ablation experiments for different attention masks



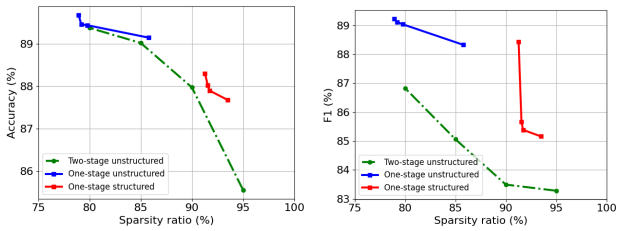
(a) Head-1. (b) Head-2. (c) Head-3. (d) Head-4. (e) Head-5. (f) Head-6.

(g) Head-7. (h) Head-8. (i) Head-9. (j) Head-10. (k) Head-11. (l) Head-12.

Figure 6. Visualization of attention masks generated by DAM_s ($\lambda = 10^{-1}$). White means with-attention and dark means no-attention.

Table 4. Ablation study on the importance of diag-attention in different attention masks. Here, "w/" means using diag-attention and "w/o" means without using diag-attention. As can be seen, dropping diag-attention increases sparsity ratio without harming the performance.

	Strided		Fixed		Longformer		LogSparse		BigBird		Star		$\text{DAM}_s(\lambda = 10^{-4})$		$\text{DAM}_s(\lambda = 10^{-1})$	
	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o
Sparsity (%)	70.4	71.2	72.7	73.4	88.7	89.5	89.8	90.6	93.2	93.9	96.1	96.9	90.4	91.2	92.7	93.5
GLUE (%)	79.5	80.2	79.7	79.6	80.1	80.1	77.9	77.8	79.4	79.5	78.9	78.6	80.5	80.9	79.3	79.6



(a) QNLI. (b) MRPC.

Figure 7. Comparison of one-stage and two-stage attention mask on QNLI and MRPC development set.

and compare their average scores on the GLUE development set. We illustrate the results on the existing attention masks (Figure 1) and two structured masks of SparseBERT in Table 4.

As can be seen, the average GLUE score of using diag-attention and dropping diag-attention is similar for all attention masks. Thus, dropping diag-attention can increase the sparsity ratio further without harming performance.

4.3.2. One-stage vs Two-stage Pruning

In this section, we compare Differentiable Attention Mask described in Algorithm 1, which generates the attention mask as part of the end-to-end training process (one-stage) with the pruning approach in Section 3.4, which first obtains the attention probabilities in $\boldsymbol{P}$ and then performs thresholding to obtain the binary attention mask (two-stage). For the two-stage attention mask, we prune $80\%/85\%/90\%/95\%$ entries of self-attention for better comparison.

Here we only illustrate the performance comparison on QNLI and MRPC development sets in Figure 7. As can be seen, the attention masks (unstructured and structured) generated by one-stage optimization achieve better performance, which is caused by the gap between continuous $\boldsymbol{P}$ and discrete $\boldsymbol{M}$. However, the two-stage attention mask can easily adjust its sparsity to any desired value, while the one-stage approach cannot set the sparsity directly as it is controlled by the hyper-parameter λ in Eq. (7).

5. Conclusion

In this paper, we investigate the importance of the different attention positions in the self-attention mechanism. By jointly optimizing a soft attention mask with the BERT model, we obtain several interesting findings. In particular, one surprising observation is that the diagonal elements in the attention matrix are the least important, which conflicts with observations in prior works. We then show, both theoretically and experimentally, that these diagonal elements are indeed not useful for universal approximation and empirical performance. Besides, by using the Gumbel-sigmoid function, we propose to optimize the attention mask in an end-to-end manner for efficient Transformer design. Extensive experimental results on a number of NLP tasks demonstrate the usefulness of the proposed algorithm.

References

Beltagy, I., Peters, M., and Cohan, A. Longformer: The Long-Document Transformer. Preprint arXiv:2004.05150, 2020.

Bentivogli, L., Dagan, I., Hoa, D., Giampiccolo, D., and Magnini, B. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC 2009 Workshop*, 2009.

Bi, K., Xie, L., Chen, X., Wei, L., and Tian, Q. GOLD-NAS: Gradual, One-Level, Differentiable. Preprint arXiv:2007.03331, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language Models are Few-Shot Learners. Preprint arXiv:2005.14165, 2020.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*, 2020.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *International Workshop on Semantic Evaluation*, 2017.

Chen, Z., Zhang, H., Zhang, X., and Zhao, L. Quora question pairs. *University of Waterloo*, 2018.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating Long Sequences with Sparse Transformers. Preprint arXiv:1904.10509, 2019.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. What Does BERT Look at? An Analysis of BERT's Attention. In *ACL Workshop BlackboxNLP*, 2019.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Dolan, W. and Brockett, C. Automatically Constructing a Corpus of Sentential Paraphrases. In *International Workshop on Paraphrasing*, 2005.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021.

Elsken, T., Metzen, J., and Hutter, F. Neural Architecture Search: A Survey. *Journal of Machine Learning Research*, 2019.

Gong, L., He, D., Li, Z., Qin, T., Wang, L., and Liu, T. Efficient Training of BERT by Progressively Stacking. In *International Conference on Machine Learning*, 2019.

Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., and Zhang, Z. Star-Transformer. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. Revealing the Dark Secrets of BERT. In *Empirical Methods in Natural Language Processing*, 2019.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y., and Yan, X. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In *Neural Information Processing Systems*, 2019.

Liu, H., Simonyan, K., and Yang, Y. DARTS: Differentiable Architecture Search. In *International Conference on Learning Representations*, 2019.

Maddison, C., Mnih, A., and Teh, Y. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*, 2017.

Michel, P., Levy, O., and Neubig, G. Are Sixteen Heads Really Better than One? In *Neural Information Processing Systems*, 2019.

Ott, M., Edunov, S., Grangier, D., and Auli, M. Scaling Neural Machine Translation. In *Machine Translation*, 2018.

Park, C., Na, I., Jo, Y., Shin, S., Yoo, J., Kwon, B., Zhao, J., Noh, H., Lee, Y., and Choo, J. SANVis: Visual Analytics for Understanding Self-Attention Networks. In *IEEE Visualization Conference*, 2019.

Qiu, J., Ma, H., Levy, O., Yih, W., Wang, S., and Tang, J. Blockwise Self-Attention for Long Document Understanding. In *Empirical Methods in Natural Language Processing*, 2020.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Empirical Methods in Natural Language Processing*, 2016.

Rajpurkar, P., Jia, R., and Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., and Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Empirical Methods in Natural Language Processing*, 2013.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. In *Neural Information Processing Systems*, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *EMNLP Workshop BlackboxNLP*, 2018a.

Wang, W., Yan, M., and Wu, C. Multi-Granularity Hierarchical Attention Fusion Networks for Reading Comprehension and Question Answering. In *Annual Meeting of the Association for Computational Linguistics*, 2018b.

Warstadt, A., Singh, A., and Bowman, S. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 2019.

Williams, A., Nangia, N., and Bowman, S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *North American Chapter of the Association for Computational Linguistics*, 2018.

Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Preprint arXiv:1609.08144, 2016.

Xie, S., Zheng, H., Liu, C., and Lin, L. SNAS: Stochastic Neural Architecture Search. In *International Conference on Learning Representations*, 2018.

Yun, C., Bhojanapalli, S., Rawat, A., Reddi, S., and Kumar, S. Are Transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2019.

Yun, C., Chang, Y., Bhojanapalli, S., Rawat, A., Reddi, S., and Kumar, S. $O(n)$ Connections are Expressive Enough: Universal Approximability of Sparse Transformers. In *Neural Information Processing Systems*, 2020.

Zaheer, M., Guruganesh, G., Dubey, K., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big Bird: Transformers for Longer Sequences. In *Neural Information Processing Systems*, 2020.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *Empirical Methods in Natural Language Processing*, 2018.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *International Conference on Computer Vision*, 2015.