Extremely Sparse Transformer with Top-k Selective Attention

Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Qi Su, Xu Sun

{*zhaoguangxiang, linjunyang, zzy1210, sukia, xusun*}*@pku.edu.cn*

MOE Key Lab of Computational Linguistics, School of EECS, Peking University

Peking University, No.5 Yiheyuan Road, Haidian District, Beijing, P.R.China 100871

Abstract

The attention mechanism lets a token attends to each token in a given sequence. In Transformer, the original full attention without sparsity leads to irrelevant information extraction on long sequences. Previous solutions include locally sparse attention and globally sparse attention. Locally sparse attention utilizes windows for locality but loses global information, while globally sparse attention requires predefined attention patterns that ignore the input samples. To enable globally sparse attention without predefined patterns, inspired by the selective attention mechanism in psychology, we propose Top-k Selective Attention that lets a position attends to the positions in a given sequence with the top-k highest attention scores to obtain the most critical information from a given sequence. Compared to the previous locally sparse attention methods, the sparse attention patterns in our method are not limited to a local window. Compared to previous globally sparse attention methods, the attention patterns in our method are dynamic since the attention selection is based on the features of the input samples. Our method also achieves non-trivial empirical results. It removes redundancy and improves over sparse and full attention methods on sentence-level and document-level machine translation. Moreover, our method keeps the most helpful information and hardly drops performance for high sparsity on machine translation tasks and enables extreme sparsity for long sequences on language modeling tasks. In addition, our method has better interpretability for understanding self-attention in Transformer.

Keywords: Transformer; Attention Mechanism; Sparse Attention; Natural Language Processing; Machine Translation; Language Modeling.

1. Introduction

Understanding natural language requires the ability to pay attention to the most relevant information. For example, people tend to focus on the most relevant segments to search for the answers to their questions during reading. Psychologists formally describe this focus as the selective attention mechanism, a cognitive psychological behavior that allows an individual to select and focus on a particular input. However, retrieving problems may occur if irrelevant segments impose negative impacts on reading comprehension. Such distraction hinders the understanding process, which calls for an effective attention method. This principle is also applicable to the computation systems for natural language.

Self-attention-based Transformer has demonstrated success in a number of tasks across domains, including natural language understanding [1, 2], natural language generation [3, 4], computer vision [5, 6], protein generation [7], decision making [8, 9], visual question answering [10], etc. However, the vanilla self-attention assigns credits to all context components, resulting in a lack of focus. As illustrated in Figure 1, “tim” assigns non-zero attention scores to the irrelevant words such as ”him” in the Transformer.

Recent years have witnessed the progress of sparse attention for Transformer. Generally, we can categorize previous sparse

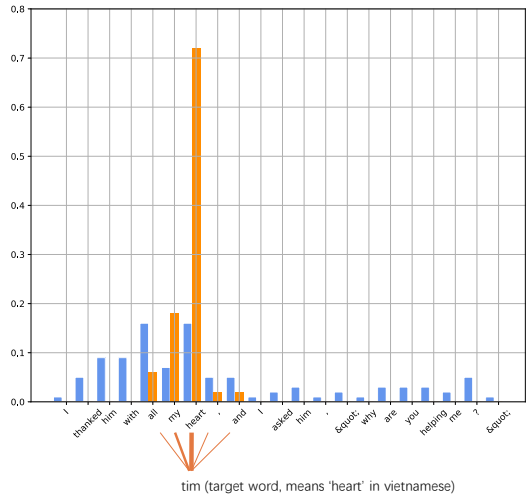


Figure 1: An illustration of attention scores. The orange bar denotes the attention scores with our proposed method, while the blue bar denotes the Transformer's full attention scores. The orange line represents the attention between the target word "tim" and the selected top-k positions in the sequence. In the Transformer's full attention, "tim" assigns too many non-zero attention scores to the irrelevant words. However, in the proposal, the top-k most significant attention scores remove the distraction from irrelevant words and concentrate the attention.

attention mechanisms into locally sparse attention and globally sparse attention. Locally sparse attention uses a local attention window or splits the long sequences into blocks to sparsify the attention weights [11, 12, 13, 14]. However, windows or blocks harm global context modeling, and they cannot filter unrelated information. Globally sparse attention requires manually predefined attention patterns, e.g., Global Attention [15] or Random Attention [15, 16, 17]. In this scenario, human prior knowledge plays a crucial role in such pattern design, and the predefined patterns are input-agnostic.

In this work, we propose a new sparse attention method called **Top-k Selective Attention** that enables globally and dynamically selecting a few states in attention modules of the Transformer. In Top-k Selective Attention, each position in our method only pays attention to the k most contributive states. Compared with locally sparse attention, our method can attend to the whole context without adding local dependency constraints, and it can efficiently filter the impact of unrelated words. Compared with globally sparse attention, it learns attention patterns

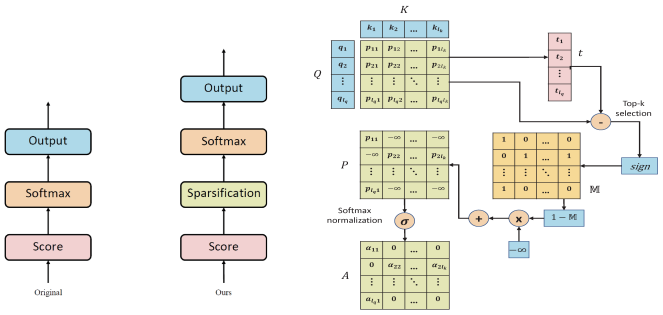


Figure 2: This figure compares the full attention and the proposed Top-k Selective Attention. The subfigure on the left briefly describes the process, and the subfigure on the right illustrates the detailed procedure. We assign only the most contributive positions with attention probabilities according to the mask based on top-k attention selection.

without predefined heuristics. We compare the Top-k Selective Attention with vanilla attention in Figure 2. The orange bars show that the proposal selects the top-k most significant attention scores and removes the distraction from irrelevant words.

We verify our idea using experimental results on various tasks, including language modeling and sentence-level and document-level machine translation. First, we show that the Top-k Selective Attention results in highly sparsified attention compared to locally and globally sparse attention on sentence-level and document-level machine translation tasks. Second, we show that the top-k attention can compensate for the drawback of locally sparse Window Attention and propose Top-k Selective Attention (OOW), which outperforms other sparse attention methods by $1-2$ BLEU scores on document-level machine translation tasks. Third, we find that combining Top-k Selective Attention with locally sparse attention can further reduce the redundancy of attention (i.e., we are the first to further remove at most 99.9% attention weights over a locally sparse attention method [14] on language modeling tasks). Lastly, our visual analysis shows that Top-k Selective Attention exhibits a higher potential in performing a high-quality alignment. The contributions of this paper are as follows:

- We present a new sparse attention method called Top-k Selective Attention, which dynamically generates globally sparse attention patterns without the need for pre-defined attention patterns. This method is simple and easy to plug

into the widely used Transformer model.

- We are the first to systematically compare sparse attention methods on machine translation. On these tasks, our method outperforms these sparse attention methods, especially in the settings of low sparsity. Moreover, our idea can make up for the shortcoming of the commonly-used Window Attention and improve its performance by about 2 BLEU scores.

- Our method brings higher sparsity than many other sparse attention methods and enables Extremely Sparse Transformer. We are the first to achieve extreme sparsity by focusing on at least 0.1% of the local context spans in language modeling.

2. Related Work

The attention mechanism is the dominant module of the Transformer model. It models the relation between positions in different sequences [18] or in the same sequence [19, 20, 21]. However, the dense attention in Transformer is often blamed. Researchers try to achieve sparse attention by pre-defining locally or globally sparse attention patterns but ignore the variety of input sequences.

2.1. Locally Sparse Attention

Local attention that restricts context into a local window has been successfully applied to machine translation [22, 23, 24]. Recent studies also achieve state-of-the-art performance in language modeling [12, 13, 14] and language understanding [17, 15]. Local attention can be categorized into two main types. The first type is Block Attention that divides the long sequence into blocks and only performs intra-block attention [25, 12, 24]. Block Attention can be combined with the memory cache mechanism, in which each block can still access the cached memory of previous blocks. The block size can be static in Transformer-XL [13], or adaptive in Adaptive Sparse Attention [14]. The second type is Window Attention, in which each word only pays attention to its surrounding positions [22, 23, 17, 15, 26].

However, Window Attention with small windows can miss global context modeling, and it cannot filter irrelevant information. In comparison, the Top-k Selective Attention does not add local dependency constraints. It enables more compact attention and more efficient filtering of irrelevant words. Furthermore, the proposed attention can improve the sparsification methods with stationary attention span [13] or adaptive attention span [14] in language modeling. Besides language modeling, we also show that our method improves locally sparse attention methods in machine translation [22, 23, 24].

Another related work constructs quasi-globally sparse attention. This approach still restricts the attention patterns and assumes that each position should not attend to two unrelated positions [27]. In contrast, in the Top-k Selective Attention each position automatically searches for its global attention patterns through learning.

2.2. Globally Sparse Attention

Hard attention is a straightforward way to introduce global sparsity to attention mechanisms. It is consistent with our intuition that a position cannot attend to all contexts at a time. Researchers have successfully applied hard attention to cross attention [28, 29, 30].

Recent studies achieve hard sparse attention by substituting the softmax function with another activation function [31, 32]. Sparsemax projects the probabilities from euclidean space to simplex. Our method still uses the softmax function to give probabilities. we normalize attention between each position to top-k related positions.

There are two representative and simple global attention mechanisms. One is Random Attention, which allows each position randomly to pay attention to the part of the target sequence [15, 16, 17]. The other is Global Attention, which allows part of the sequence to pay attention to or be paid attention to by all the positions of the target sequence [15, 17]. These global attention methods ignore the diversity among examples, while in the Top-k Selective Attention each example learn globally sparse attention patterns.

3. Background

3.1. Attention Mechanism

The attention mechanism is to learn the alignment between the target-side context and the source-side context [18], and Luong *et al.* [33] formulated several versions for local and global attention. In general, the attention mechanism maps a query and a key-value pair to an output. The attention score function and softmax normalization can turn the query Q and the key K into a distribution α. Following the distribution α, the attention mechanism computes the expectation of the value V and finally generates the output C.

Take the original attention mechanism in Machine Translation (MT) as an example. Both key $K \in \mathbb{R}^{n\times d}$ and value $V \in \mathbb{R}^{n\times d}$ are the sequence of output states from the encoder. Query $Q \in \mathbb{R}^{m\times d}$ is the sequence of output states from the decoder, where m is the length of Q, n is the length of K and V, and d is the dimension of the states. Thus, the attention mechanism is:

$$C = \mathrm{softmax}(f(Q, K))V \tag{1}$$

where f refers to the parametric function for computing attention scores.

3.2. Attention in Transformer

Transformer [21], which is entirely based on the attention mechanism, demonstrates state-of-the-art performance in a series of natural language generation tasks. There are two types of attention modules in Transformer. Self-attention module models relation between tokens in one sequence, either for the source sequence in the encoder or for the target sequence in the decoder. Cross-attention module bridges the encoder and the decoder by allowing the target sequence to attend tokens in the source sentence. In Transformer, self-attention and cross-attention can both have multiple heads.

The ideology of self-attention is, as the name implies, the attention over the context itself. In the implementation, the query Q, key K and value V are the linear transformation of the input x, so that $Q = W_Q x$, $K = W_K x$ and $V = W_V x$ where W_Q, W_K and W_V are learn-able parameters. Therefore, the computation is as below:

$$C = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d}}\right)V \tag{2}$$

where d refers to the dimension of the states.

We can regard the mechanism mentioned above as the one head attention and view the attention computation separated into g heads as the multi-head attention. Thus, we can compute multiple parts of the inputs individually. For the i-th head, the output is in the following formula:

$$C^{(i)} = \mathrm{softmax}\left(\frac{Q^{(i)}K^{(i)\mathrm{T}}}{\sqrt{d_k}}\right)V^{(i)} \tag{3}$$

where $C^{(i)}$ refers to the output of the head, $Q^{(i)}$, $K^{(i)}$ and $V^{(i)}$ are the query, key and value of the head, and d_k refers to the size of each head ($d_k = d/g$). Finally, we concentrate outputs of all heads to get an overall output:

$$C = [C^{(1)}, \cdots, C^{(i)}, \cdots, C^{(g)}] \tag{4}$$

In common practice, C is sent through a linear transformation with weight matrix W_c for the final output of multi-head attention. The formalization of cross-attention is similar to self-attention, except that query Q is from the target sequence and K, V is from the source sequence.

However, the original full attention can assign weights to more words that are less relevant to the query. Therefore, to improve concentration in attention for effective information extraction, we propose Top-k Selective Attention that enables Extremely Sparse Transformer. We also compare it with previous sparse attention methods.

4. Extremely Sparse Transformer with Top-k Selective Attention

Lack of concentration in the attention can lead to the failure of relevant information extraction. To this end, we propose a novel method, Top-k Selective Attention, which focuses on only a few elements through explicit selection. We can apply the proposed method to the vanilla Transformer or its variants to build an Extremely Sparse Transformer model.

We code-name it as Extremely Sparse Transformer since it globally sparsifies the attention based on the input and retains the most useful information with the top-k operation. Thus, it loses the least information and enables extreme sparsity. Compared with the conventional attention, this model assigns no credit to the target token that is not highly correlated to the query token .

We provide a comparison between the attention of the vanilla Transformer and that of the Extremely Sparse Transformer in Figure 2. We can see that the difference is that our method sparsifies the attention scores. The attention degenerates to sparse attention through top-k selection. In this way, we reserve the most contributive components for attention and remove the other irrelevant information. This selective attention is effective in preserving essential information and removing noise. The attention can be much more concentrated on the most contributive elements of the value.

In the following, we describe the forward and backward propagation of our method.

4.1. The Forward-Propagation Process of Top-k Selective Attention

In the uni-head self-attention, the key components, the query $Q[l_Q, d]$, key $K[l_K, d]$ and value $V[l_V, d]$, are the linear transformation of the source context, namely the input of each layer, where $Q = W_Q x$, $K = W_K x$ and $V = W_V x$. Extremely Sparse Transformer first generates the attention scores P as demonstrated below:

$$P = \frac{QK^{\mathrm{T}}}{\sqrt{d}} \tag{5}$$

Then the model evaluates the values of the scores P based on the hypothesis that scores with larger values demonstrate higher relevance. The sparse attention masking operation $\mathcal{M}(\cdot)$ is implemented upon P in order to select the top-k contributive elements. Specifically, we select the k largest elements of each row in P and record their positions in the position matrix (i, j), where k is a hyper-parameter. To be specific, say the k-th largest value of row i is t_i, if the value of the j-th component is larger than t_i, the position (i, j) is recorded. We concatenate the threshold value of each row to form a vector $t = [t_1, t_2, \cdots, t_{l_Q}]$. The masking functions $\mathcal{M}(\cdot, \cdot)$ is illustrated as follows:

$$\mathcal{M}(P, k)_{ij} = \begin{cases} P_{ij} & \text{if } P_{ij} \geq t_i \text{ (k-th largest value of row } i) \\ -\infty & \text{if } P_{ij} < t_i \text{ (k-th largest value of row } i) \end{cases} \tag{6}$$

The top-k selection method explicitly selects the high attention scores. This explicit manner is different from dropout, which randomly abandons the scores. Such explicit selection can guarantee the preservation of essential components and simplify the model since k is usually a small number such as 8. Please refer section 6.1 for detailed analysis. The next step after top-k selection is normalization:

$$A = \mathrm{softmax}(\mathcal{M}(P, k)) \tag{7}$$

where A refers to the normalized scores. Since scores that are smaller than the top-k largest scores are set to negative infinity by the masking function $\mathcal{M}(\cdot, \cdot)$, these normalized scores approximate 0. We show the back-propagation process of top-k selection in Section 4.2. The output representation of self-attention C is below:

$$C = AV \tag{8}$$

The output is the expectation of the value following the sparsified attention distribution A. Following the distribution of the selected components, the attention in the Top-k Selective Attention can obtain more focused attention. Additionally, such sparse attention can extend to cross-attention. In the implementation, we replace Q with $W_Q S$, where W_Q is still a learnable matrix. Resembling but different from the self-attention mechanism, the Q is no longer the linear transformation of the source context but the decoding states S.

In brief, the attention in our proposed Extremely Sparse Transformer sparsifies the attention weights. The attention can then focus on the most contributive elements. Besides, Top-k Selective Attention is compatible with both self-attention and cross-attention.

4.2. The Back-propagation Process of Top-k Selective Attention

In this subsection, we prove that our method does not hinder back-propagation by calculating the gradients. In Formula 6, we

defined $M = \mathcal{M}(P, k)$, its gradient with respect to P is:

$$\frac{\partial M_{ij}}{\partial P_{kl}} = 0 \quad (i \neq k \text{ or } j \neq l) \tag{9}$$

$$\frac{\partial M_{ij}}{\partial P_{ij}} = \begin{cases} 1 & \text{if } P_{ij} \geq t_i \ (k\text{-th largest value of row } i) \\ 0 & \text{if } P_{ij} < t_i \ (k\text{-th largest value of row } i) \end{cases} \tag{10}$$

The next step after top-k selection is normalization:

$$A = \text{softmax}(\mathcal{M}(P, k)) \tag{11}$$

where A refers to the normalized scores. When back-propagating,

$$\frac{\partial A_{ij}}{\partial P_{kl}} = \sum_{m=1}^{l_Q} \sum_{n=1}^{l_K} \frac{\partial A_{ij}}{\partial M_{mn}} \frac{\partial M_{mn}}{\partial P_{kl}} \tag{12}$$

$$= \frac{\partial A_{ij}}{\partial M_{kl}} \frac{\partial M_{kl}}{\partial P_{kl}} \tag{13}$$

$$= \begin{cases} \frac{\partial A_{ij}}{\partial M_{kl}} & \text{if } P_{ij} \geq t_i \ (k\text{-th largest value of row } i) \\ 0 & \text{if } P_{ij} < t_i \ (k\text{-th largest value of row } i) \end{cases} \tag{14}$$

The softmax function is differential; therefore, we have calculated the gradient involved in top-k selection.

4.3. *Comparison between the Proposal and other Sparse Attention Methods*

In addition to the original **Full Attention** method, we also compare our method with six canonical sparse attention methods. We show the comparison in Figure 3. The descriptions of these methods are as follows:

- **Block Attention**: This method first breaks a long sequence into blocks, and then each token in a block only attends other tokens in that block [25, 12, 24]. Block Attention can be combined with the cache mechanism, in which each block can still access the memory of more than the current block through the caching mechanism [13, 14].
- **Window Attention**: Each token attends its nearby tokens [12, 23, 17, 15]. This method is the most commonly used sparse attention method, but it can only model local dependency.

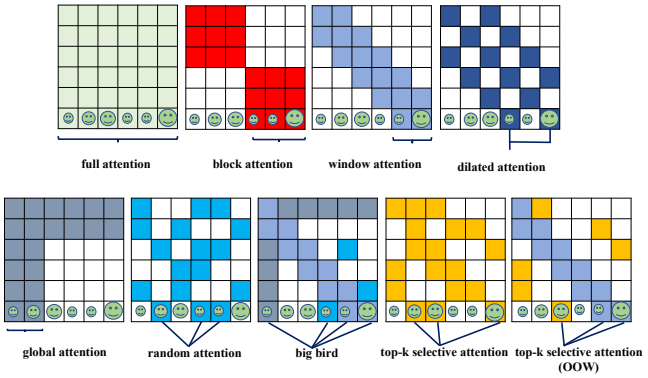


Figure 3: These nine matrices denote nine forms of attention. The size of the face indicates the raw attention quantity. Block Attention, Window Attention, Dilated Attention are three locally sparse attention methods. Global Attention, Random Attention, and Big Bird are there globally sparse attention methods. We present Top-k Selective Attention. This method introduces selective attention and generates sample-dependent sparse attention patterns. The Top-k Selective Attention achieves attention sparsity while approximating the full global attention. We also design its variant, the Top-k Selective Attention (OOW), in which each token sparsely attends tokens Out Of a Window (OOW).

- **Dilated Attention**: Each token can sparsely access a larger window of context, compared to the case in Window Attention [34]. It is a compromise between full attention and local attention. But some nearby tokens are never been attended.
- **Global Attention**: Let a part of the whole sequence pay attention to or be paid attention to by all the positions of the target sequence [15, 17]. Most tokens can only focus on the first small part of the whole sequence.
- **Random Attention**: Each token randomly attends a subset of tokens in the whole sequence [15, 16, 17]. This method can model non-local dependency, but Random Attention is not interpretable and anti the attention in psychology. When the sparsity is high, each token ignores most related tokens. We can apply Random Attention to the cross-attention, but we can only apply the other four sparse attention to self-attention.
- **Big Bird**: Big Bird combines three sparse attention methods: Random Attention, Window Attention, and Global Attention [15]. Let k be the tokens attended by each token.

In our implementation, we divide this attention sparsity into three parts and assign $k/2$ to the Window Attention, $k/4$ to Global Attention, and $k/4$ to Random Attention. Inspired by their papers and preliminary experimental results, we set this division.

We present a novel globally sparse attention method, the Top-k Selective Attention. We also design its variant, the Top-k Selective Attention (OOW), which applies Top-k Selective Attention to non-nearby tokens that are beyond a sliding window.

- **Top-k Selective Attention**: Each token selects top-k related tokens to attend. Window, Block, Dilated Attention discard large context, so they fail to model long-term dependency. Global Attention and Random Attention restrict the globally sparse attention with pre-defined patterns, so they fail to extract most related tokens. However, our sparse attention is determined by both samples and models, and we get these globally sparse attention patterns in an end-to-end way. Moreover, the top-k operation lets us retain the most useful information, so our methods are beneficial for extreme sparsity. We also provide the empirical comparisons in Section 6.1.

- **Top-k Selective Attention (OOW)**: It applies Top-k Selective Attention to tokens Out Of a sliding Window (OOW). Specifically, each token selects top-$k/2$ tokens from remaining tokens beyond a local window of size $k/2$. We display it in the last sub-figure in Figure 3. Top-k Selective Attention (OOW) can make up for window attention for sparsely capturing long dependency with top-k attention selection. The difference between Top-k Selective Attention (OOW) and Big Bird is that we let the input and the model learn the non-local attention patterns, rather than using predefined non-local attention patterns. We show our Top-k Selective Attention can significantly improve Window Attention and outperforms Big Bird in Section 6.2.

5. Experimental Settings

We conduct a series of experiments on three natural language processing tasks, sentence-level and document-level machine translation and language modeling. The hyper-parameters, including the beam size and the number of training steps, are tuned on the validation set.

5.1. Sentence-level and Document-level Machine Translation

We first verify on the machine translation datasets that the Top-k Selective Attention is more conducive to extreme sparsity than other methods.

5.1.1. Machine Translation (MT) Datasets

For sentence-level machine translation, we mainly conduct experiments on two machine translation datasets, IWSLT2017 French-to-English Translation (Fr-En) [35], and WMT2016 English-to-Romanian Translation (En-Ro) [36]. For data preprocessing, we follow the preprocessing scripts from the codebase of fairseq. To be specific, we use the Moses tokenizer[1] first to tokenize each sentence and then encode it into a sequence of sub-word units (”word pieces”). We also experiment on IWSLT 2015 English-to-Vietnamese (En-Vi) [37] and preprocess without subword encoding to visualize attention distribution of the Full Attention and the Top-k Selective Attention at the word level.

For document-level machine translation, following Maruf *et al.* [38], we perform experiments on TED talks from IWSLT 2017 MT track [35], and News-Commentary (v11) from WMT 2016 [36].

5.1.2. Training

We use the default scripts of fairseq[2] v0.10.2 to preprocess the Fr-En and En-Ro dataset. We adopt a mini model for Fr-En translation with only two layers in the encoder and decoder respectively, following Wang *et al.* [39]. For En-vi and En-Ro translation, we adopt the Transformer base model. For En-Vi

[1] Moses decoder: https://github.com/moses-smt/mosesdecoder
[2] https://github.com/pytorch/fairseq

translation, we use the default scripts and hyper-parameter setting of tensor2tensor[3] v1.11.0 to preprocess, train and evaluate our model. For the document-level machine translation datasets, we adopt the setting from Bao *et al.* [40].

5.1.3. Evaluation

We use the BLEU (bilingual evaluation understudy) score [41] for the evaluation of the performance on sentence-level machine translation. BLEU evaluates the similarity of predicted target sentences to the target sentences translated by a professional human, and high BLEU scores indicate good performance. We use d-BLEU [42] that calculates BLEU scores at the document level to evaluate the performance on document-level machine translation. The difference between d-BLEU and BlEU is that d-BLEU matches n-grams in the whole document because we do not have the alignments between translation and source sentences.

5.1.4. Context Length

For sentence-level machine translation, the average context length is about 25. For document-level machine translation, we follow Bao *et al.* [40] to split each document into blocks of 512 tokens. Thus, the context for each token is 512 tokens.

5.2. Language Modeling

We then show that we can further sparsify the common-used sparse-attention methods in language modeling tasks.

5.2.1. Datasets

We experiment on two large-scale character-level language modeling datasets Enwiki8 and Text8[4]. They both contain **100M** bytes of unprocessed Wikipedia texts. The inputs include Latin alphabets, non-Latin alphabets, XML markups, and special characters. The vocabulary contains 205 characters, including one to denote unknown characters. We used the same preprocessing method following Chung *et al.* [43]. The training set contains 90M bytes of data, and the validation set and the test set contain 5M, respectively.

5.2.2. Training and Evaluation

Following the previous work [43, 13], we use BPC that stands for the average number of Bits-Per-Character, for evaluation. Lower BPC indicates better performance. Transformer-XL [13] and Adaptive Attention Span [14] are two state-of-the-art language modeling baselines that are capable of generating long sequences with their sparse attention. Since we want to apply our method upon Transformer-XL and Adaptive Attention Span, we directly use their implementation for training and evaluation, respectively. As to the model implementation, we perform experiments with models of base version ($\sim$40M parameters). Due to our limited resources (TPU), we did not implement the big version of these language modeling baselines.

5.2.3. Context Length

Since the sequence length is too large, the common practice is splitting the long sequence into blocks (analogy to Block Attention). The dominant methods in language modeling are Block Attention with cache mechanism [13, 14], they can access larger context beyond the current block. For example, Transformer-XL [13] has 512 tokens in each block, but it can access the previous block too (This is like the sliding window). Adaptive Attention Span [14] has an adaptive number of tokens in each block. Still, tokens in the current block can access the memory, which has up to 8192 previous tokens. The average length of context span is 390 on Enwiki-8 and 314 on Text8. We will show in Section 6.3 that the Top-k Selective Attention can reduce the number of attended positions from 390 to 8 by applying it to Adaptive Attention Span. Since the test set has 5M characters, the context length is 5M if we adopt the original Transformer with full attention.

6. Results

This section presents the experimental results on sentence-level machine translation, document-level machine translation,

[3] https://github.com/tensorflow/tensor2tensor

[4] The link to Enwiki8 and Text8 dataset is: http://mattmahoney.net/dc/text.html

Table 1: We show the comparison between results of Top-k Selective Attention and results of other sparse attention methods. The proposed method enables much more sparse attention. The average sequence length is about 25 for these two datasets.

Sentence-level MT Datasets		IWSLT French-English						WMT English-Romanian					
# Attended Positions (Sparsity)		**Full**	**16**	**8**	**4**	**2**	**AVG**	**Full**	**16**	**8**	**4**	**2**	**AVG**
Full Attention [21]		37.10			-			33.73			-		
Locally	Block Attention [25, 12, 24]	-	35.04	34.27	33.63	31.80	33.69	-	32.25	30.52	29.22	29.09	30.27
	Window Attention [12, 23, 17, 15]	-	37.44	37.03	36.69	31.98	35.79	-	0.85	2.49	26.27	29.83	14.86
	Dilated Attention [34]	-	0.03	11.13	24.31	30.23	16.43	-	0.36	0.16	4.83	24.47	9.82
Globally	Global Attention [15, 17]	-	12.87	6.81	2.16	1.65	5.87	-	9.88	2.01	0.48	0.01	3.10
	Random Attention [15, 16, 17]	-	23.89	14.55	6.83	2.82	12.02	-	25.84	18.06	7.90	1.34	13.29
	Top-k Selective Attention	-	36.95	37.64	37.25	36.62	**37.12**	-	33.99	33.61	33.86	32.43	**33.48**

and language modeling. We first show that our method enables much more sparse attention compared to other sparse attention methods on two sentence-level machine translation datasets. For example, we can improve 1.33 BLEU scores on IWSLT French-English translation and 3.21 BLEU scores on WMT English-Romanian translation compared to other sparse attention methods. Then we show the proposal can make up for window attention on two document-level machine translation datasets. For example, we improve 2.16 BLEU scores on TED and 1.13 on News-Commentary. Lastly, we show that our method enables an extremely sparse transformer on two language-modeling datasets since Top-k Selective Attention significantly advances the sparsity of state-of-the-art sparse attention baselines.

6.1. Sentence-level Machine Translation

Since our globally sparse attention is automatically obtained based on each sample and retains the most useful information through Top-k Selective Attention, we can avoid poor BLEU scores towards extreme sparsity. This subsection verifies the claim and shows that our method enables much more sparse attention than other sparse attention methods on two sentence-level machine translation datasets. We have described these sparse attention methods in Section 4.3. The experimental results of them on two sentence-level machine translation datasets are in Table 2. Top-k Selective Attention can build much more sparse attention. For example, when we can only attend to 4 tokens, our method can achieve 37.25 BLEU scores on Fr-En translation and 33.86 BLEU scores on En-Ro Translation. In contrast, the best of other methods only gets 36.69 on Fr-En (Window Attention) and 29.22 on En-Ro (Block Attention). The low BLEU scores of Global Attention and Random Attention verify our claim that it is not easy to pre-define global attention patterns. For example, when we can only attend to 4 tokens, these two baselines only get 0.48 and 7.90 on the En-Ro dataset.

6.2. Document-level Machine Translation

In this subsection, we show that we can use it to make up for the shortcoming of Window Attention to achieve significant improvements on document-to-document Machine Translation. We show an attention pattern example of Top-k Selective Attention (OOW) in the last sub-figure of Figure 3. We select top-$k/2$ tokens beyond a local window of size $k/2$. We only apply Top-k Selective Attention (OOW) to self-attention modules, which is comparable to Window Attention since the latter only applied to self-attention modules too.

Table 1 presents the results of the baselines and our method. Since the sequenced length of the document-level translation task is much longer than the sentence-level translation task (512 vs. 25), Full Attention achieves worse performance than sparse attention methods. The sentence where the current word is located is more critical to the feature learning and translation of

Table 2: Top-k Selective Attention (OOW) outperforms other sparse attention methods in document-to-document machine translation. The full context has 512 tokens.

Document-level MT Datasets		TED					News-Commentary				
# Attended Positions (Sparsity)		**Full**	**32**	**16**	**8**	**AVG**	**Full**	**32**	**16**	**8**	**AVG**
Full Attention [21]		0.76		-			0.60		-		
Locally	Block Attention [25, 12, 24]	-	5.71	3.58	2.66	3.56	-	2.50	1.79	1.40	1.90
	Window Attention [12, 23, 17, 15]	-	23.28	23.68	24.73	23.90	-	23.65	23.68	24.73	24.02
	Dilated Attention [34]	-	20.9	21.19	21.76	21.28	-	18.06	21.19	21.76	20.34
Globally	Global Attention [15, 17]	-	0.39	0.6	0.68	0.56	-	0.49	0.48	0.33	0.43
	Random Attention [15, 16, 17]	-	4.22	0.65	0.23	1.70	-	3.13	0.80	0.40	1.44
	Big Bird [15]	-	0.49	0.52	8.87	3.29	-	0.41	0.39	0.19	0.33
	Top-k Selective Attention (OOW)	-	25.42	25.97	26.80	**26.06**	-	24.84	25.00	25.55	**25.13**

the current word than other sentences in the entire document. Hence, the locally sparse attention methods (Window, Block, and Dilated Attention) achieve better results in these baselines than the predefined global attention methods(Random Attention and Global Attention). Among them, Window Attention allows introductory connections of distant words compared to Block Attention, and the results are better. Dilated Attention tries to increase the window size but loses attention to nearby tokens, and it is worse than Window Attention. Since Top-k Selective Attention (OOW) can dynamically attend critical tokens beyond a window, it outperforms Window Attention, e.g., we bring +2.16 BLEU scores on TED and +1.13 BLEU scores on News Commentary. The application of Top-k Selective Attention to the cross-attention attention brings improvements for the cases of lower sparsity. Big Bird adds global context modeling, but it reduces translation performance since their global attention patterns are predefined and input-agnostic compared with Window Attention. But our Top-k Selective Attention (OOW) dynamically models globally sparse attention patterns that can significantly improve the performance. This result verifies our claim that the mechanism of Top-k Selective Attention serves as a better globally sparse attention method than Random Attention and Global Attention in extracting global features.

6.3. Language Modeling

This subsection shows that our Top-k Selective Attention can further sparse the state-of-the-art sparse attention methods, including Transformer-XL and Adaptive Attention Span. This subsection shows that our method enables an extremely sparse transformer on two language-modeling datasets. The common practice in language modeling is splitting the long sequences (5M tokens in the test set of both datasets) into blocks (e.g., 512 tokens). Transformer-XL [13] is a strong approach that introduces local attention constraints by breaking long sequences into blocks. In a block, each token attends all the tokens in the current block and all the tokens in the former block. The difference between Adaptive Attention Span [14] and Transformer-XL is that Adaptive Attention Span changes the length of the segment adaptively.

Table 3 shows the results on the test set of two character-level language model datasets en-wiki8 and text8. When we add the proposed Top-k Selective Attention to the locally sparse attention method of Adaptive Attention Span with an average span size of 390 and a maximum span size of 8192, we can further sparse the attention. For example, on average, we increase the sparsity by 47.75 and 38.25 times on Enwiki8 and Text8, respectively. In the case of a larger attention span (at most 8192), we reduce the number of attended positions by 99.9% (8192 to

Table 3: We report the sparsity–the number of attended positions on character language modeling and corresponding BPC scores. Sparsity and BPC scores are the smaller, the better. We achieve extremely sparse attention by adding our method to local attention with stationary attention span (Transformer-XL) and adaptive attention span (Adaptive Attention Span). These methods also split the long input sequence into blocks but remember the previous clock with a cache mechanism. Results show that our method advances the sparsity of these strong baselines. Transformer-XL does not report results about Text8 with a base model in their paper.

Language Modeling Datasets	Enwiki8		Text8	
Metrics	# Attended Positions	BPC	# Attended Positions	BPC
Transformer (Full) [21]	5M	Intractable	5M	Intractable
Transformer (Block) [44]	512	1.11	512	1.18
Transformer-XL [13]	512	1.07		
+Top-k Selective Attention	64	1.05	-	
Improved Sparsity Ratio	8			
Adaptive Span [14]	390	1.02	314	1.11
+Top-k Selective Attention	8	1.02	8	1.11
Improved Sparsity Ratio	48.75		39.25	

Table 4: The first line are the BPC scores, the last three lines are corresponding span size with three methods, they are both the lower the better. By introducing the top-k attention selection on the sparse attention method of Adaptive Attention Span, the length of context span we need becomes much smaller.

	BPC	1.40	1.30	1.20	1.10
Span Size	Adaptive Attention Span [14]	151	210	295	370
	+Top-k Selective Attention (k=32)	117	170	227	319
	+Top-k Selective Attention (k=8)	78	101	141	193

8). Our method is also additive with the Transformer-XL. We can improve language modeling performance while reducing the redundancy of attention.

We also find that when we apply the Top-k Selective Attention upon the method of Adaptive Attention Span. Adaptive Attention Span reduces span size compared to the Transformer-XL (e.g., from 512 to 390 on Enwiki8) and adaptively changes the span size during training. We can further reduce the span size (e.g., 390 to 220 on Enwiki8). As shown in Table 4, applying the top-k attention selection on the Adaptive Attention Span [14] dramatically reduces the length of attention span. For example, the application of top-8 attention selection can reduce half of the span size while achieving equal performance or achieve 30% improvement with the equal span size.

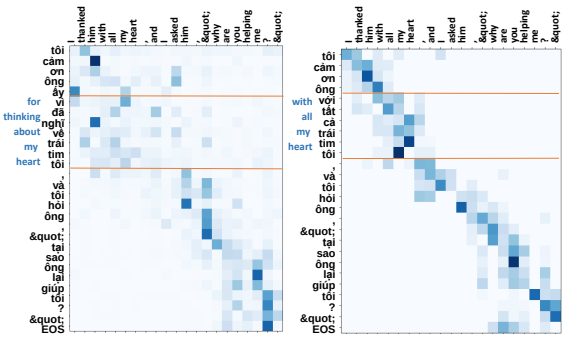
7. Visual Analysis

In this section, we show that our method leads to better attention interpretability and word alignment. To perform a thorough evaluation of our Extremely Sparse Transformer, we conducted a case study and visualized the attention distributions of our model and the baseline for further comparison. Specifically, we analyzed the test set of IWSLT En-Vi, and randomly selected a sample pair of attention visualization of both models.

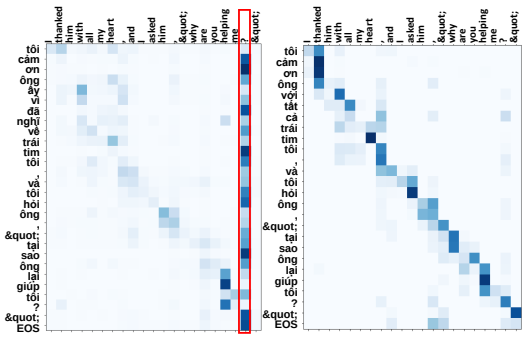
The visualization of the cross-attention of the decoder's bottom layer in Figure 4(a). The attention distribution of the left figure is fairly dispersed. On the contrary, the right figure shows that the sparse attention can choose to focus only on several positions so that the model can be forced to stay focused. For example, when generating the phrase "for thinking about my heart" (Word-to-word translation from Vietnamese), the generated word cannot be aligned to the corresponding words. As to Extremely Sparse Transformer, when generating the phrase "with all my heart", the attention can focus on the corresponding positions with strong confidence.

We show the visualization of the decoder's top layer in Figure 4(b). From the figure, the cross-attention at the top layer of the vanilla Transformer decoder suffers from focusing on the

last source token. This over-attention phenomenon is typical behavior of the attention in vanilla Transformer. Such attention with the wrong alignment cannot sufficiently extract relevant source-side information for the generation. In contrast, with a simple modification on the vanilla version, Extremely Sparse Transformer does not suffer from this problem but instead focuses on the relevant sections of the source context. The figure on the right demonstrates the attention distribution of our method shows that our proposed attention in the model can perform an accurate alignment.



(a) Attention of the bottom layer



(b) Attention of the top layer

Figure 4: Figure 4(a) is the attention visualization of the Transformer, and Figure 4(b) is that of the Extremely Sparse Transformer. The red box shows that the Transformer's attention at most steps concentrates on the context's last token.

8. Conclusion

In this paper, we present a globally sparse attention method called Top-k Selective Attention, which enhances the concentration of the Transformer's attention through explicit selection. Top-k Selective Attention can make the attention in Transformer more concentrated on the most contributive components, and it can automatically obtain the global sparse attention pattern belonging to each sample. Since Top-k Selective Attention can retain information from a wide range of contexts and retain the most helpful information with the top-k operation, extensive experiments show that this idea enables Extremely Sparse Transformer with considerate performance.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019, pp. 4171–4186.

[2] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, in: ICLR, 2020.

[3] M. Ott, S. Edunov, D. Grangier, M. Auli, Scaling neural machine translation, in: WMT, 2018, pp. 1–9.

[4] S. Ma, L. Cui, D. Dai, F. Wei, X. Sun, Livebot: Generating live video comments based on visual and textual contexts, in: AAAI, 2019, pp. 6810–6817.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, 2021.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: ECCV, 2020, pp. 213–229.

[7] D. Grechishnikova, Transformer neural network for protein-specific de novo drug generation as a machine translation problem, Scientific reports 11 (1) (2021) 1–13.

[8] E. Parisotto, F. Song, J. Rae, R. Pascanu, C. Gulcehre, S. Jayakumar, M. Jaderberg, R. L. Kaufman, A. Clark, S. Noury, M. Botvinick, N. Heess, R. Hadsell, Stabilizing transformers for reinforcement learning, in: ICML, 2020, pp. 7487–7498.

[9] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, I. Mordatch, Decision transformer: Reinforcement learning via sequence modeling, in: NeurIPS, 2021.

[10] Z. Yu, J. Yu, Y. Cui, D. Tao, Q. Tian, Deep modular co-attention networks for visual question answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6281–6290.

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional sequence to sequence learning, in: ICML, 2017, pp. 1243–1252.

[12] R. Child, S. Gray, A. Radford, I. Sutskever, Generating long sequences with sparse transformers, arXiv preprint arXiv:1904.10509.

[13] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, R. Salakhutdinov,

Transformer-XL: Attentive language models beyond a fixed-length context, in: ACL, 2019, pp. 2978–2988.

[14] S. Sukhbaatar, E. Grave, P. Bojanowski, A. Joulin, Adaptive attention span in transformers, in: ACL, 2019, pp. 331–335.

[15] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, A. Ahmed, Big bird: Transformers for longer sequences, in: NeurIPS, 2020, pp. 17292–17306.

[16] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, C. Zheng, Synthesizer: Rethinking self-attention for transformer models, in: ICML, pp. 10183–10192.

[17] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150.

[18] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: ICLR, 2014.

[19] J. Cheng, L. Dong, M. Lapata, Long short-term memory-networks for machine reading, in: EMNLP, 2016, pp. 551–561.

[20] A. P. Parikh, O. Täckström, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, in: EMNLP, 2016, pp. 2249–2255.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 6000–6010.

[22] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, T. Zhang, Modeling localness for self-attention networks, in: EMNLP, 2018, pp. 4449–4458.

[23] B. Yang, L. Wang, D. F. Wong, L. S. Chao, Z. Tu, Convolutional self-attention networks, in: NAACL-HLT, 2019, pp. 4040–4045.

[24] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, in: ICLR, 2020.

[25] T. Shen, T. Zhou, G. Long, J. Jiang, C. Zhang, Bi-directional block self-attention for fast and memory-efficient sequence modeling, in: ICLR, 2018.

[26] C. Wu, F. Wu, Y. Huang, DA-transformer: Distance-aware transformer, in: NAACL-HLT, 2021, pp. 2059–2068.

[27] Y. Tay, D. Bahri, L. Yang, D. Metzler, D.-C. Juan, Sparse Sinkhorn attention, in: ICML, 2020, pp. 9438–9447.

[28] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: ICML, 2015, pp. 2048–2057.

[29] T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, C. Zhang, Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling, in: IJCAI 2018, 2018, pp. 4345–4352.

[30] J. Lin, X. Sun, X. Ren, M. Li, Q. Su, Learning when to concentrate or divert attention: Self-adaptive attention temperature for neural machine translation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2985–2990.

[31] A. F. T. Martins, R. F. Astudillo, From softmax to sparsemax: A sparse model of attention and multi-label classification, in: ICML, 2016, pp. 1614–1623.

[32] B. Peters, V. Niculae, A. F. T. Martins, Sparse sequence-to-sequence models, in: ACL, Association for Computational Linguistics, 2019, pp. 1504–1519.

[33] T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in: EMNLP, 2015, pp. 1412–1421.

[34] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: ICLR, 2016.

[35] M. Cettolo, M. Federico, L. Bentivogli, N. Jan, S. Sebastian, S. Katsuitho, Y. Koichiro, F. Christian, Overview of the iwslt 2017 evaluation campaign, in: IWSLT, 2017, pp. 2–14.

[36] Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany.

[37] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, M. Federico, The iwslt 2015 evaluation campaign, 2015.

[38] S. Maruf, A. F. T. Martins, G. Haffari, Selective attention for context-aware neural machine translation, in: NAACL-HLT, 2019, pp. 3092–3102.

[39] C. Wang, K. Cho, J. Gu, Neural machine translation with byte-level subwords, in: AAAI, 2020, pp. 9154–9160.

[40] G. Bao, Y. Zhang, Z. Teng, B. Chen, W. Luo, G-transformer for document-level machine translation, in: ACL, 2021, pp. 3442–3455.

[41] K. Papineni, S. Roukos, T. Ward, W. Zhu, Bleu: a method for automatic evaluation of machine translation, in: ACL, 2002, pp. 311–318.

[42] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, L. Zettlemoyer, Multilingual Denoising Pre-training for Neural Machine Translation, TACL 8 (2020) 726–742.

[43] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Gated feedback recurrent neural networks, in: ICML, 2015, pp. 2067–2075.

[44] R. Al-Rfou, D. Choe, N. Constant, M. Guo, L. Jones, Character-level language modeling with deeper self-attention, arXiv preprint arXiv:1808.04444.