SPARSE TRANSFORMER: CONCENTRATED ATTENTION THROUGH EXPLICIT SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-attention-based Transformer has demonstrated the state-of-the-art performances in a number of natural language processing tasks. Self attention is able to model long-term dependencies, but it may suffer from the extraction of irrelevant information in the context. To tackle the problem, we propose a novel model called **Sparse Transformer**. Sparse Transformer is able to improve the concentration of attention on the global context through an explicit selection of the most relevant segments. Extensive experimental results on a series of natural language processing tasks, including neural machine translation, image captioning, and language modeling, all demonstrate the advantages of Sparse Transformer in model performance. Sparse Transformer reaches the state-of-the-art performances in the IWSLT 2015 English-to-Vietnamese translation and IWSLT 2014 German-to-English translation. In addition, we conduct qualitative analysis to account for Sparse Transformer's superior performance.

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 in mind during reading. 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.

This principle is also applicable to the computation systems for natural language. Attention has been a vital component of the models for natural language understanding and natural language generation. Recently, Vaswani et al. (2017) proposed Transformer, a model based on the attention mechanism for Neural Machine Translation(NMT). Transformer has shown outstanding performance in natural language generation tasks. More recently, the success of BERT (Devlin et al., 2018) in natural language processing shows the great usefulness of both the attention mechanism and the framework of Transformer.

However, the attention in vanilla Transformer has a obvious drawback, as the Transformer assigns credits to all components of the context. This causes a lack of focus. Ke et al. (2018) pointed out that no attempt of sparse attentive weight has been made on Transformer. With this motivation, we propose a novel model called **Sparse Transformer** which is equipped with our sparse attention. We implement an explicit selection method based on top-k selection. Unlike vanilla Transformer, Sparse Transformer only pays attention to the k most contributive states. Thus Sparse Transformer can perform more concentrated attention than vanilla Transformer.

As illustrated in Figure 1, the attention in vanilla Transformer assigns high credits to many irrelevant words, while in Sparse Transformer, it concentrates on the most relevant k words. For the word "tim", the most related words should be ”heart” and the immediate words. Yet the attention in vanilla Transformer does not focus on them but gives credits to some irrelevant words such as "him". Sparse Transformer is designed to alleviate this problem and make the attention more concentrated.

For further investigation, we also conduct a series of qualitative analyses, including ablation studies and attention visualization. Through the analyses, we demonstrate that our proposed mechanism can enhance the model performance in different aspects. We also find out that the context attention at the top layer of the vanilla Transformer focuses on the end position of the text. This discourages effective

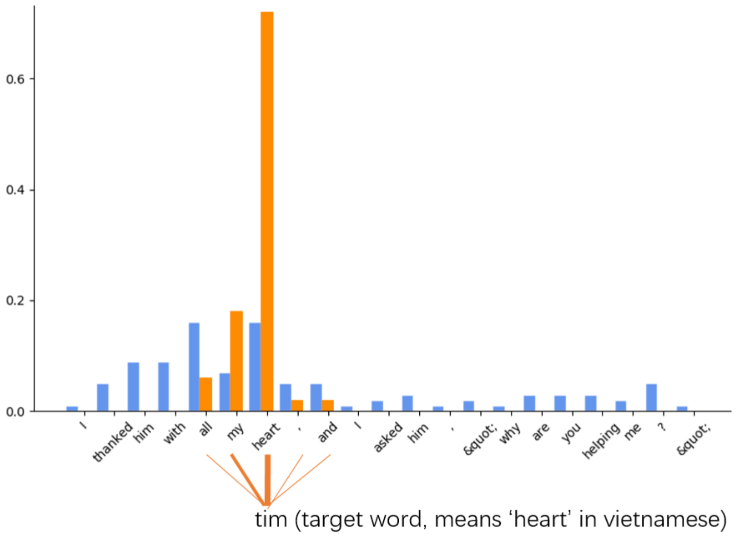


Figure 1: Illustration of self attention in the models. The orange bar denotes the attention score of our proposed Sparse Transformer while the blue bar denotes the attention scores of the vanilla Transformer. The orange line denotes the attention between the target word “tim” and the selected top-k positions in the sequence. In the attention of vanilla Transformer, ”tim” assigns too many non-zero attention scores to the irrelevant words. But for the proposal, the top-k largest attention scores removes the distraction from irrelevant words and the attention becomes concentrated.

information extraction from the source context. Devoid of such a defect, Sparse Transformer exhibits a higher potential in performing a high-quality alignment.

The contributions of this paper are presented below:

- We propose a novel model called Sparse Transformer, which enhances the concentration of the Transformer’s attention through explicit selection.
- We conducted extensive experiments on three natural language processing tasks, including Neural Machine Translation, Image Captioning and Language Modeling. Compared with vanilla Transformer, Sparse Transformer demonstrates better performances in the above three tasks. Specifically, our model reaches the state-of-the-art performances in the IWSLT 2015 English-to-Vietnamese translation and IWSLT 2014 German-to-English translation.
- We conducted a series of qualitative analyses for Sparse Transformer. The analyses show that our proposed sparsification method implemented can be applied to any part of the model and bring improvement to the model performance. The attention in Sparse Transformer demonstrates higher concentration and leads to more accurate alignment compared to vanilla Transformer.

2 Background

Prior to the methodology part, a review to the attention mechanism and the attention-based framework of Transformer is provided as follows.

2.1 Attention Mechanism

Bahdanau et al. (2014) first introduced the attention mechanism to learn the alignment between the target-side context and the source-side context, and Luong et al. (2015) 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.

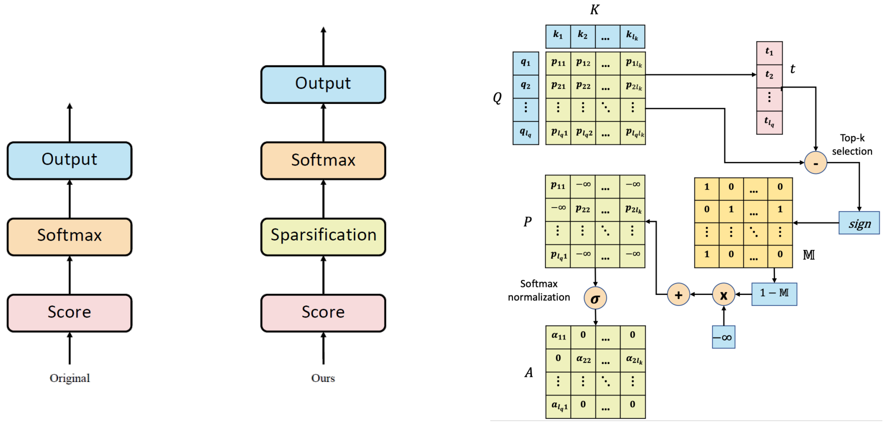


Figure 2: The comparison between the attentions of vanilla Transformer and Sparse Transformer and the illustration of the attention module of Sparse Transformer. With the mask based on top-k selection and softmax function, only the most contributive elements are assigned with probabilities.

Take the original attention mechanism in NMT 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 formulated as:

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

where f refers to the attention score computation.

2.2 Transformer

Transformer (Vaswani et al., 2017), which is fully based on the attention mechanism, demonstrates the state-of-the-art performances in a series of natural language generation tasks. Specifically, we focus on self attention and multi-head attention.

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 learnable parameters. Therefore, the computation can be formulated as below:

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

where d refers to the dimension of the states.

The aforementioned mechanism can be regarded as the unihead attention. As to the multi-head attention, the attention computation is separated into g heads (namely 8 for basic model and 16 for large model in the common practice). Thus multiple parts of the inputs can be computed individually. For the i-th head, the output can be computed as in the following formula:

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

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, the output of each head are concatenated for the 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.

However, soft attention can assign weights to a lot more words that are less relevent to the query. Therefore, in order to improve concentration in attention for effective information extraction, we study the problem of sparse attention in Transformer and propose our model Sparse Transformer.

3 SPARSE TRANSFORMER

Lack of concentration in the attention can lead to the failure of relevant information extraction. To this end, we propose a novel model, **Sparse Transformer**, which enables the focus on only a few elements through explicit selection. Compared with the conventional attention, no credit will be assigned to the value that is not highly correlated to the query. We provide a comparison between the attention of vanilla Transformer and that of Sparse Transformer in Figure 2.

Sparse Transformer is still based on the Transformer framework. The difference is in the implementation of self attention. The attention is degenerated to the sparse attention through top-k selection. In this way, the most contributive components for attention are reserved and the other irrelevant information are removed. This selective method is effective in preserving important information and removing noise. The attention can be much more concentrated on the most contributive elements of value. In the following, we first introduce the sparsification in self attention and then extend it to context attention.

In the unihead 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 $V = W_V x$. 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 element of each row in P and record their positions in the position matrix (i, j), where k is a hyperparameter. 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:

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

With the top-k selection, the high are selected through an explicit way. This is different from dropout which randomly abandons the scores. Such explicit selection can not only guarantee the preservation of important components, but also simplify the model since k is usually a small number such as 5 or 10. The next step after top-k selection is normalization:

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

where A refers to the normalized scores. As the scores that are smaller than the top k largest scores are assigned with negative infinity by the masking function $\mathcal{M}(\cdot, \cdot)$, their normalized scores, namely the probabilities, approximate 0. We show the back-propagation process of Top-k selection in A.2. The output representation of self attention C can be computed as below:

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

The output is the expectation of the value following the sparsified distribution A. Following the distribution of the selected components, the attention in the Sparse Transformer model can obtain more focused attention.

Also, such sparse attention can extend to context attention. 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 the implementation, we replace Q with $W_Q s$, where W_Q is still learnable matrix.

In brief, the attention in our proposed Sparse Transformer sparsifies the attention weights. The attention can then become focused on the most contributive elements, and it is compatible to both self attention and context attention.

Model	En-De	En-Vi	De-En
Seq2Seq (Lin et al., 2018)	-	26.9	-
GNMT+RL (Wu et al., 2016)	24.6	-	-
ConvS2S (Gehring et al., 2017)	25.2	-	-
Actor-Critic (Bahdanau et al., 2017)	-	-	28.5
NPMT+LM (Huang et al., 2017)	-	28.1	30.1
SACT (Lin et al., 2018)	-	29.1	-
Var-Attn (Deng et al., 2018)	-	-	33.7
Transformer (Vaswani et al., 2017)	28.4	30.2	34.4
RNMT (Chen et al., 2018)	28.5	-	-
Fixup (Zhang et al., 2019)	-	-	34.5
Weighted Transformer (Ahmed et al., 2017)	28.9	-	-
Universal Transformer (Dehghani et al., 2018)	28.9	-	-
Sparse Transformer	**29.4**	**31.1**	**35.6**

Table 1: Results on the En-De, En-Vi and De-En test sets. Compared with the baseline models, Sparse Transformer reaches improved performances, and it achieves the state-of-the-art performances in En-Vi and De-En.

4 Experiments

We conducted a series of experiments on three natural language processing tasks, including neural machine translation, image captioning and language modeling. Detailed experimental settings are in A.1.

4.1 Neural Machine Translation

Dataset To evaluate the performance of Sparse Transformer in NMT, we conducted experiments on three NMT tasks, English-to-German translation (En-De) with a large dataset, English-to-Vietnamese (En-Vi) translation and German-to-English translation (De-En) with two datasets of medium size. For En-De, we trained Sparse Transformer on the standard dataset for WMT 2014 En-De translation. The dataset consists of around 4.5 million sentence pairs. The source and target languages share a vocabulary of 32K sub-word units. We used the *newstest 2013* for validation and the *newstest 2014* as our test set. We report the results on the test set.

For En-Vi, we trained our model on the dataset in IWSLT 2015 (Cettolo et al., 2014). The dataset consists of around 133K sentence pairs from translated TED talks. The vocabulary size for source language is around 17,200 and that for target language is around 7,800. We used *tst2012* for validation, and *tst2013* for testing and report the testing results. For De-En, we used the dataset in IWSLT 2014. The training set contains 160K sentence pairs and the validation set contains 7K sentences. Following Edunov et al. (2018), we used the same test set with around 7K sentences. The data were preprocessed with byte-pair encoding (Sennrich et al., 2016). The vocabulary size is 14,000.

Result Table 1 presents the results of the baselines and our Sparse Transformer on the three datasets. For En-De, Transformer-based models outperform the previous methods. Compared with the result of Transformer (Vaswani et al., 2017), Sparse Transformer reaches 29.4 in BLEU score evaluation, outperforming vanilla Transformer by 1.0 BLEU score. For En-Vi, vanilla Transformer[1] reaches 30.2, outperforming the state-of-the-art method (Huang et al., 2017). Our model, Sparse Transformer, achieves a new state-of-the-art performance, 31.1, by a margin of 0.9 over vanilla Transformer. For De-En, we demonstrate that Transformer-based models outperform the other baselines. Compared with Transformer, our Sparse Transformer reaches a better performance, 35.6. Its advantage is +1.2. To the best of our knowledge, Sparse Transformer reaches a top line performance on the dataset.

[1] While we did not find the results of Transformer on En-Vi, we reimplemented our vanilla Transformer with the same setting.

Model	**BLEU-4**	**METEOR**	**CIDEr**
SAT Bazzani et al. (2018b)	28.2	24.8	92.3
SCST Rennie et al. (2017)	32.8	26.7	106.5
NBT Lu et al. (2018)	34.7	27.1	107.2
AdaAtt Lu et al. (2017)	33.2	26.6	108.5
ARNN Bazzani et al. (2018a)	33.9	27.6	109.8
Transformer	35.3	27.7	113.1
UpDown Anderson et al. (2018)	**36.2**	27.0	113.5
Sparse Transformer	35.7	**28.0**	**113.8**

Table 2: Results on the MSCOCO Karpathy test split.

Model	**Params**	**BPC**
LN HyperNetworks (Ha et al., 2016)	27M	1.34
LN HM-LSTM (Chung et al., 2016)	35M	1.32
RHN (Zilly et al., 2017)	46M	1.27
Large FS-LSTM-4 (Mujika et al., 2017)	47M	1.25
Large mLSTM (Krause et al., 2016)	46M	1.24
Transformer (Al-Rfou et al., 2018)	44M	1.11
Transformer-XL (Dai et al., 2019)	41M	1.06
Sparse Transformer-XL	41M	**1.05**

Table 3: Comparison with state-of-the-art results on enwiki8. Sparse Transformer-XL refers to the Transformer with our sparsification method.

4.2 Image Captioning

Dataset We evaluated our approach on the image captioning task. Image captioning is a task that combines image understanding and language generation. We conducted experiments on the Microsoft COCO 2014 dataset (Chen et al., 2015a). It contains 123,287 images, each of which is paired 5 with descriptive sentences. We report the results and evaluate the image captioning model on the MSCOCO 2014 test set for image captioning. We used the publicly-available splits provided by Karpathy & Li (2015). The validation set and test set both contain 5,000 images.

Result Table 2 shows the results of the baseline models and Sparse Transformer on the COCO Karpathy test split. Transformer outperforms the mentioned baseline models. Sparse Transformer outperforms the implemented Transformer by +0.4 in terms of BLEU-4, +0.3 in terms of METEOR, +0.7 in terms of CIDEr. , which consistently proves its effectiveness in Image Captioning.

4.3 Language Modeling

Dataset Enwiki8[2] is large-scale dataset for character-level language modeling. It contains 100M bytes of unprocessed Wikipedia texts. The inputs include Latin alphabets, non-Latin alphabets, XML markups and special characters. The vocabulary size 205 tokens, including one for unknown characters. We used the same preprocessing method following Chung et al. (2015). The training set contains 90M bytes of data, and the validation set and the test set contains 5M respectively.

Result Table 3 shows the results of the baseline models and Sparse Transformer-XL on the test set of enwiki8. Compared with the other strong baselines, Transformer-XL can reach a better performance. Sparse Transformer outperforms Transformer-XL with an advantage. Regardless of the big version of Transformer-based models with more parameters, its performance, 1.05 BPC, is the best on enwiki8 for language modeling.

[2] http://mattmahoney.net/dc/text.html

Mask Position	**Test(BLEU)**
Transformer	30.2
+ Enc-Sparse	30.6
+ Dec-Sparse	30.6
+ Context-Sparse	30.8
Sparse Transformer	31.1

Table 4: Results of the ablation study on the En-Vi test set. ”Enc-Sparse” represents adding the sparsification to the encoder self attention, ”Dec-Sparse” refers to adding the sparsification to the decoder self attention, and ”Context-Sparse” denotes adding the sparsification to the context attention.

Task	**Base**	**T**	**T&P**
En-Vi (BLEU)	30.2	30.6	31.1

Table 5: Results of the ablation study of the sparsification at different phases on the En-Vi test set. “Base” denotes vanilla Transformer. “T” denotes only adding the sparsification in the training phase, and “T&P” denotes adding it at both phases as the implementation of Sparse Transformer does.

5 Discussion

In this section, we performed several analyses for further discussion of Sparse Transformer. We conducted an ablation study to evaluate the effects of sparsification in different attention of the Transformer, including encoder self attention, decoder self attention and context attention. Moreover, we conducted a series of qualitative analyses to evaluate the effects of sparsification in Transformer.

5.1 Ablation Study

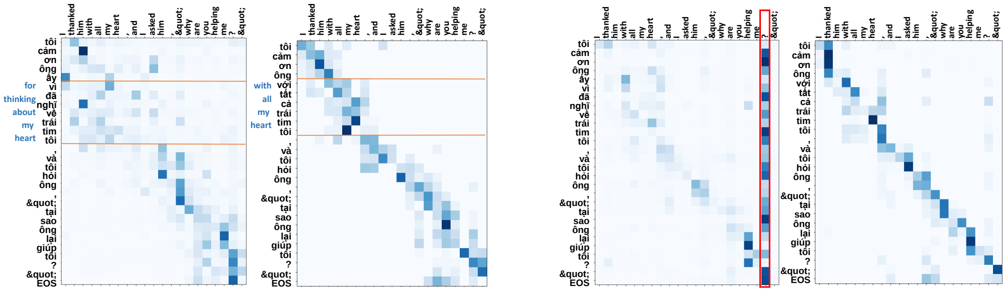
We performed an ablation study on the En-Vi test set and evaluated the effects of our sparsification method in either one of the attention modules of the transformer. The results are shown in Table 4. Adding the sparsification to different attention module can all bring an improvement in the performance. Both adding the sparsification to the encoder self attention and to the decoder self attention bring an improvement of +0.4 BLEU, and adding it to the context attention brings a larger improvement of +0.6 BLEU. Sparse Transformer with the sparsification to all three attention modules achieves the state-of-the-art performance. This demonstrates that the sparsification method can extend to any attention in the Transformer framework.

Another interesting finding is that only adding the sparsification in the training phase can also bring an improvement in the performance. We evaluate it on En-Vi and report the results in Table 5. The results demostrate that only adding it at the training phase can bring an improvement of +0.4 BLEU. This shows that vanilla Transformer may be overparameterized and the sparsification encourages the simplification of the model.

5.2 Attention Visualization

To perform a thorough evaluation of our Sparse Transformer, we conducted a case study and visualize the attention distributions of our model and the baseline for further comparison. Specifically, we conducted the analysis on the test set of En-Vi, and randomly selected a sample pair of attention visualization of both models.

The visualization of the context attention of the decoder's bottom layer in Figure 3(a). The attention distribution of the left figure is fairly disperse. 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 Sparse Transformer, when generating the phrase ”with all my heart”, the attention can focus on the corresponding positions with strong confidence.



(a) Attention of the bottom layer

(b) Attention of the top layer

Figure 3: Figure 3(a) is the attention visualization of Transformer and Figure 3(b) is that of the Sparse Transformer. The red box shows that the attentions in vanilla Transformer at most steps are concentrated on the last token of the context.

The visualization of the decoder's top layer is shown in Figure 3(b). From the figure, the context attention at the top layer of the vanilla Transformer decoder suffers from focusing on the last source token. This is a common behavior of the attention in vanilla Transformer. Such attention with wrong alignment cannot sufficiently extract enough relevant source-side information for the generation. In contrast, Sparse Transformer, with simple modification on the vanilla version, does not suffer from this problem, but instead focuses on the relevant sections of the source context. The figure on the right demonstrating the attention distribution of Sparse Transformer shows that our proposed attention in the model is able to perform accurate alignment.

6 Related Work

Attention mechanism has demonstrated outstanding performances in a number of neural-network-based methods, and it has been a focus in the NLP studies (Bahdanau et al., 2014). A number of studies are proposed to enhance the effects of attention mechanism (Luong et al., 2015; Vaswani et al., 2017; Ke et al., 2018). Luong et al. (2015) propose local attention and Yang et al. (2018) propose local attention for self attention. Xu et al. (2015) propose hard attention that pays discrete attention in image captioning. Chandar et al. (2016) propose a combination soft attention with hard attention to construct hierarchical memory network. Lin et al. (2018) propose a temperature mechanism to change the softness of attention distribution. Shen et al. (2018) propose an attention which can select a small proportion for focusing. It is trained by reinforcement learning algorithms (Williams, 1992). Recent advances in sparse attention demonstrate its potential in natural language processing (Martins & Astudillo, 2016; Niculae & Blondel, 2017; Ke et al., 2018; Laha et al., 2018). However, these methods have problems in either restricted range of attention or training difficulty. They did not demonstrate improvements in Transformer. Child et al. (2019) recently propose to use local attention and block attention to sparsify the transformer. Our approach differs from them in that our method does not need to block sentences and still capture long distance dependencies. Besides, we demonstrate the importance of sparse transformer in sequence to sequence learning.

7 Conclusion

In this paper, we propose a novel model called Sparse Transformer. Sparse Transformer is able to make the attention in vanilla Transformer more concentrated on the most contributive components. Extensive experiments show that Sparse Transformer outperforms vanilla Transformer in three different NLP tasks. We conducted a series of qualitative analyses to investigate the reasons why Sparse Transformer outperforms the vanilla Transformer. The results of the ablation study demonstrate that our sparsification method at any attention module of the Transformer can increase the model's performance individually. Furthermore, we find an obvious problem of the attention at the top layer of the vanilla Transformer, and Sparse Transformer can alleviate this problem effectively with improved alignment effects.

REFERENCES

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. Weighted transformer network for machine translation. *CoRR*, abs/1711.02132, 2017.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. *arXiv preprint arXiv:1808.04444*, 2018.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6077–6086, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Loris Bazzani, Tobias Domhan, and Felix Hieber. Image captioning as neural machine translation task in sockeye. *arXiv preprint arXiv:1810.04101*, 2018a.

Loris Bazzani, Tobias Domhan, and Felix Hieber. Image captioning as neural machine translation task in SOCKEYE. *CoRR*, abs/1810.04101, 2018b.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, 2014.

Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. Hierarchical memory networks. *CoRR*, abs/1605.07427, 2016.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pp. 76–86, 2018.

X. Chen, H. Fang, TY Lin, R. Vedantam, S. Gupta, P. Dollr, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv:1504.00325*, 2015a.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015b.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.

Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pp. 2067–2075, 2015.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *CoRR*, abs/1807.03819, 2018.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pp. 9735–9747, 2018.

Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014*, pp. 376–380, 2014.

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

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018 1 (Long Papers)*, pp. 355–364, 2018.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1243–1252. PMLR, 2017.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. Towards neural phrase-based machine translation. *arXiv preprint arXiv:1706.05565*, 2017.

Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR 2015*, pp. 3128–3137. IEEE Computer Society, 2015.

Nan Rosemary Ke, Anirudh Goyal, Olexa Bilaniuk, Jonathan Binas, Michael C. Mozer, Chris Pal, and Yoshua Bengio. Sparse attentive backtracking: Temporal credit assignment through reminding. In *NeurIPS 2018*, pp. 7651–7662, 2018.

Ben Krause, Liang Lu, Iain Murray, and Steve Renals. Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*, 2016.

Anirban Laha, Saneem Ahmed Chemmengath, Priyanka Agrawal, Mitesh M. Khapra, Karthik Sankaranarayanan, and Harish G. Ramaswamy. On controllable sparse alternatives to softmax. In *NeurIPS 2018*, pp. 6423–6433, 2018.

Junyang Lin, Xu Sun, Xuancheng Ren, Muyu Li, and Qi 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*, pp. 2985–2990, 2018.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 375–383, 2017.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7219–7228, 2018.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1412–1421, 2015.

André F. T. Martins and Ramón Fernández Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML 2016*, pp. 1614–1623, 2016.

Asier Mujika, Florian Meier, and Angelika Steger. Fast-slow recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 5915–5924, 2017.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. In *NIPS 2017*, pp. 3340–3350, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, pp. 311–318, 2002.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7008–7024, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, 2016.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. In *IJCAI 2018*, pp. 4345–4352, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS 2017*, pp. 6000–6010, 2017.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR 2015*, pp. 4566–4575, 2015.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pp. 229–256, 1992.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML 2015*, pp. 2048–2057, 2015.

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

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *CoRR*, abs/1901.09321, 2019.

Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber. Recurrent highway networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 4189–4198. JMLR. org, 2017.

A Appendix

A.1 Experimental Details

The hyper parameters including beam size and training steps are tuned on the valid set.

Neural Machine Translation We use the default setting in Vaswani et al. (2017) for the implementation of our proposed Sparse Transformer. We use case-sensitive tokenized BLEU score (Papineni et al., 2002) for the evaluation of En-De, and we use case-insensitive BLEU for that of En-Vi and De-En following Lin et al. (2018). For En-Vi translation, we use default scripts and hyper-parameter setting of tensor2tensor[3] v1.11.0 to preprocess, train and evaluate our model. We use the default

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

scripts of fairseq[4] v0.6.1 to preprocess the De-En and En-De dataset. We train the model on the En-Vi dataset for $35K$ steps with batch size of $4K$. For IWSLT 2015 De-En dataset, batch size is also set to $4K$, we update the model every 4 steps and train the model for 20000 updates. For WMT 2014 En-De dataset, we train the model for 72 epochs on 4 GPUs with update frequency of 32 and batch size of 3584.

Image Captioning We still use the default setting of Transformer for training our proposed Sparse Transformer. We report the standard automatic evaluation metrics with the help of the COCO captioning evaluation toolkit[5] (Chen et al., 2015b), which includes the commonly-used evaluation metrics, BLEU-4 Papineni et al. (2002), METEOR Denkowski & Lavie (2014), and CIDEr Vedantam et al. (2015).

Language Models We follow Dai et al. (2019) and use their implementation for our Sparse Transformer. Following the previous work (Chung et al., 2015; Dai et al., 2019), we use BPC ($E[log_2 P(xt+1|ht)]$), standing for the average number of Bits-Per-Character, for evaluation. Lower BPC refers to better performance. As to the model implementation, we implement Sparse Transformer-XL, which is based on the base version of Transformer-XL.[6] Transformer-XL is a model based on Transformer but has better capability of representing long sequences.

A.2 The Back-propagation Process of Top-k Selection

The masking function $\mathcal{M}(\cdot,\cdot)$ is illustrated as follow:

$$\mathcal{M}(P,k)_{ij} = \begin{cases} P_{ij} & \text{if } P_{ij} \geq t_i \text{ (k-th largest value of row i)} \\ -\infty & \text{if } P_{ij} < t_i \text{ (k-th largest value of row i)} \end{cases} \tag{9}$$

Denote $M = \mathcal{M}(P,k)$. We regrad t_i as constants. When back-propagating,

$$\frac{\partial M_{ij}}{\partial P_{kl}} = 0 \quad (i \neq k \text{ or } j \neq l) \tag{10}$$

$$\frac{\partial M_{ij}}{\partial P_{ij}} = \begin{cases} 1 & \text{if } P_{ij} \geq t_i \text{ (k-th largest value of row i)} \\ 0 & \text{if } P_{ij} < t_i \text{ (k-th largest value of row i)} \end{cases} \tag{11}$$

The next step after top-k selection is normalization:

$$A = \mathrm{softmax}(\mathcal{M}(P,k)) \tag{12}$$

where A refers to the normalized scores. When backpropagating,

$$\begin{aligned} \frac{\partial A_{ij}}{\partial P_{kl}} &= \sum_{m=1}^{l_Q}\sum_{n=1}^{l_K} \frac{\partial A_{ij}}{\partial M_{mn}} \frac{\partial M_{mn}}{\partial P_{kl}} && (13) \\ &= \frac{\partial A_{ij}}{\partial M_{kl}} \frac{\partial M_{kl}}{\partial P_{kl}} && (14) \\ &= \begin{cases} \dfrac{\partial A_{ij}}{\partial M_{kl}} & \text{if } P_{ij} \geq t_i \text{ (k-th largest value of row i)} \\ 0 & \text{if } P_{ij} < t_i \text{ (k-th largest value of row i)} \end{cases} && (15) \end{aligned}$$

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

[4] https://github.com/pytorch/fairseq

[5] https://github.com/tylin/coco-caption

[6] Due to our limited resources (TPU), we did not implement the big version of Sparse Transformer-XL.