Only 5% Attention Is All You Need: Efficient Long-range Document-level Neural Machine Translation

Zihan Liu[1,2*], **Zewei Sun**[2], **Shanbo Cheng**[2], **Shujian Huang**[1], **Mingxuan Wang**[2]
[1] National Key Laboratory for Novel Software Technology, Nanjing University
[2] ByteDance
liuzh@smail.nju.edu.cn, huangsj@nju.edu.cn
{sunzewei.v,chengshanbo,wangmingxuan.89}@bytedance.com

Abstract

Document-level Neural Machine Translation (DocNMT) has been proven crucial for handling discourse phenomena by introducing document-level context information. One of the most important directions is to input the whole document directly to the standard Transformer model. In this case, efficiency becomes a critical concern due to the quadratic complexity of the attention module. Existing studies either focus on the encoder part, which cannot be deployed on sequence-to-sequence generation tasks, e.g., Machine Translation (MT), or suffer from a significant performance drop. In this work, we keep the translation performance while gaining 20% speed up by introducing extra selection layer based on lightweight attention that selects a small portion of tokens to be attended. It takes advantage of the original attention to ensure performance and dimension reduction to accelerate inference. Experimental results show that our method could achieve up to 95% sparsity (only 5% tokens attended) approximately, and save 93% computation cost on the attention module compared with the original Transformer, while maintaining the performance.

1 Introduction

Recent developments in neural machine translation have focused on the translation of individual sentences, but research has shown that document-level information is crucial for handling discourse phenomena such as lexical consistency and pronominal anaphora, which rely on long-range context. As a result, various attention mechanisms (Zhang et al., 2018; Maruf et al., 2019; Zheng et al., 2020; Bao et al., 2021) that encode document-level context information have been proposed.

However, the computation cost of these attention mechanisms increases quadratically with the length of the input sequence. To address this issue, researchers have proposed efficient transformer models (Tay et al., 2020b) that aim to reduce the computation cost of attention through techniques such as sparsity patterns (Tay et al., 2020a; Child et al., 2019; Zaheer et al., 2020; Beltagy et al., 2020) that limit the number of tokens to attend to, memory or global tokens that compress contextual tokens into a single representation (Lee et al., 2019; Ma et al., 2021), approximation to softmax with kernel methods (Choromanski et al., 2020; Qin et al., 2022; Peng et al., 2021), or a combination of above (Tay et al., 2021a; Zhu et al., 2021).

Despite the emergence of various efficient transformer models, long-range sequence-to-sequence tasks such as document-level machine translation still need more exploration.

On the one hand, some of the existing efficient models (Wang et al., 2020; Zaheer et al., 2020; Lee-Thorp et al., 2022) focus on the encoder part and can not be used for generation because of the auto-regressive property. Some (Tay et al., 2021b; Child et al., 2019; Beltagy et al., 2020) have a strong relationship to the position of tokens thus can not be applied to cross attention where no alignment is obvious between query and key.

On the other hand, the studies that target on efficient sequence-to-sequence generation only verify their methods on normal sentence-level translation benchmarks like WMT EN-DE test sets (Peng et al., 2021; Petrick et al., 2022; Ma et al., 2021). In our preliminary experiments, we find that almost all the work severely drops in BLEU when dealing with real document translation tasks.

To address this issue, we try to reduce the computation cost while ensuring the translation performance. In this paper, we mainly focus on the attention mechanism following other efficient transformer models.

Specifically, we want to select important tokens (Sun et al., 2020, 2022a) and only conduct

* Work was done while Z. Liu was an intern at ByteDance.

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 733–743

attention to them. Previous studies sharing a similar motivation either design sparsity patterns with human prior like a fixed sliding window(Beltagy et al., 2020; Tay et al., 2020a; Zaheer et al., 2020) which lack flexibility, or try to learn the sparsity pattern by clustering methods. However, the poor performance of learnable pattern methods on DocNMT reflects that the query does not attend to the keys expected in original attention.

In order to ensure the performance, we take advantage of the original attention and propose Lightweight Attention Selection Transformer (Lasformer). Lasformer incorporates selection layers that utilize lightweight attention, whose distribution is guided by supervision from the original attention. The achievement of lightweight processing is attained by reducing the hidden dimension, while the selection process involves retaining tokens with the highest attention scores, a strategy validated for its efficacy by (Zhao et al., 2019). By employing these mechanisms, we are able to efficiently filter out insignificant tokens at a comparatively low expense, resulting in a reduction of the overall computational burden, particularly when a significant proportion of tokens can be filtered out.

Determining the appropriate number of tokens to retain is of utmost importance, as they must contribute sufficient information to ensure optimal performance, while also minimizing their quantity to enhance efficiency. In our approach, the sparsity is learned adaptively, which gradually increases during the training process until it reaches an optimal level that strikes a balance between performance and efficiency for each selection layer.

Experiments show that Lasformer can effectively reduce the computation of attention. Only 5% of tokens are used in attention and translation performance remains almost unchanged. For the long sequence of thousands of words, our method can lower the attention cost to 7%. And end-to-end inference speed can be enhanced to 1.2x.

2 Related Work

2.1 Document-level Machine Translation

Document-level machine translation involves an additional source and target context to improve the translation in terms of coherence and consistency (Voita et al., 2019; Müller et al., 2018; Lopes et al., 2020; Bawden et al., 2018). There exist two lines of methods to use context. One introduces an extra encoder to encode context and integrate it into the current sentence (Zhang et al., 2018; Maruf et al., 2019). The limitation is that the same sentence might be encoded multiple times thus increasing the complexity. It is solved by recent works by sharing the parameters of context encoder and current sentence encoder (Zheng et al., 2020; Ma et al., 2020).

Another line of work concatenates the context and the current sentence and translate it as if it is a single sentencec (Tiedemann and Scherrer, 2017; Sun et al., 2022b). However, the concatenation results in a long input sequence and makes it difficult to train the model, because of the high entropy of attention distribution. To alleviate the problem, locality bias is introduced, where sentence-level information is augmented (Bao et al., 2021).

In short, the former is based on sentence translation while integrating the context. The latter tries to translate the whole document while introducing sentence-level locality. And they seem to reach the same scheme that uses both local attention and global attention.

The local attention implies human-designed sparsity pattern and it is natural to introduce learnable sparsity pattern to global attention in document-level machine translation.

2.2 Efficient Transformer

There have been several previous methods for efficient Transformers that have focused on the properties of attention, specifically sparsity and low rank to reduce the computation cost.

Sparsity refers to the idea that only a few tokens receive a significant amount of attention, while the rest contribute little to the output. Some methods (Tay et al., 2021a; Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020) have proposed handcrafted patterns such as the sliding window or dilated window, which is inspired by human prior knowledge that close tokens contribute the most attention. Other methods (Kitaev et al., 2020; Wang et al., 2022; Tay et al., 2020a; Roy et al., 2021) have attempted to make the sparsity pattern learnable with a lower cost by using techniques like clustering, based on the idea that similar tokens are expected to attend to each other and belong to the same cluster. These clustering methods can include techniques like locality sensitive hashing (Kitaev et al., 2020), K-means (Roy et al., 2021), or learnable sorting networks (Tay et al., 2020a).

On the other hand, low-rank methods are based

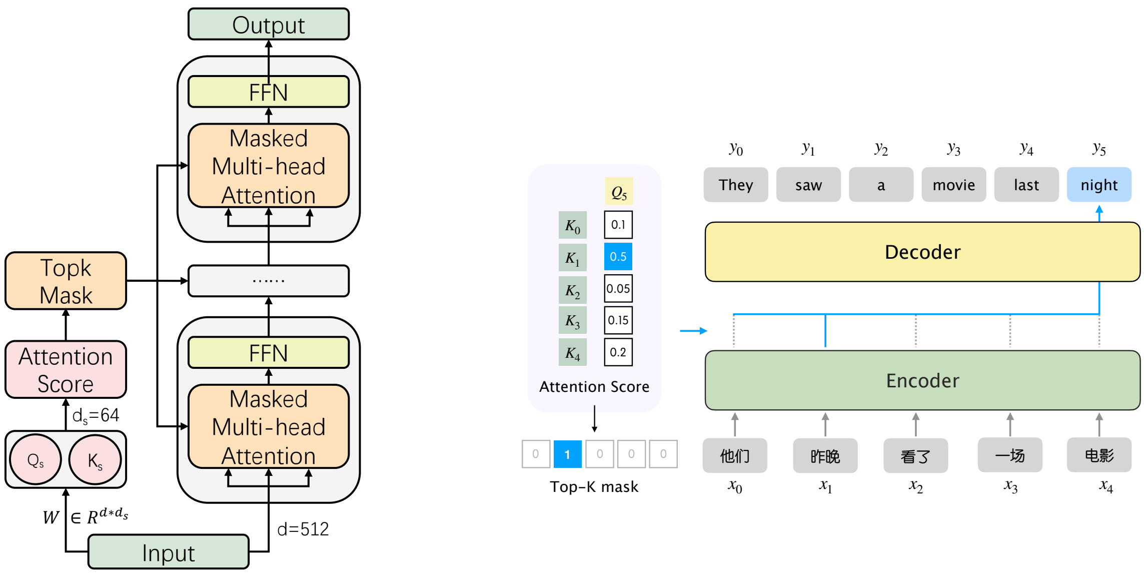


Figure 1: The left figure is the whole architecture of Lasformer. Its left part represents the selection module. It accepts a low-dimensional input to calculate lightweight attention. With the rough attention, we mask the unimportant tokens in the main module (right part of the left figure). In addition, the selection mask is shared across some layers. The right figure illustrates the masking procedure. Taking cross-attention as an example, the current query only attends to those tokens with high rough attention.

on the idea that N dimensional features can be compressed into fewer dimensions. Some work (Lee et al., 2019; Ma et al., 2021; Jaegle et al., 2021) has used global tokens or memory to compress long-range information into a limited number of embeddings or has used kernel methods (Peng et al., 2021; Qin et al., 2022) to approximate softmax scores, allowing the computation of keys and values first and reducing the complexity from $O(N^2d)$ to $O(Nd^2)$ (where d is the dimension of self-attention).

While sparsity methods maintain a token-to-token attention structure, low-rank methods use a compressed embedding for attention. The token-to-token approach is more interpretable but may lose some information, while the other may contain more information but may also be noisier. Since the information in DocNMT is sparse(Lupo et al., 2022), the noise of low-rank methods might be much more severe and thus we exploit the sparsity methods.

3 Method

Sequence-to-sequence document-level translation aims at fully capturing the distant context. It is achieved by attention mechanism, which allows each query attending to all of the keys, resulting in quadratically growing computation cost with the sequence length. However, only a quite small part of tokens is truly relevant. Therefore, it is important to select those important ones and filter the others to reduce the number of aligned objects.

Specifically, we take advantage of the origin attention mechanism and distill it into a lightweight attention with lower hidden dimension to select important tokens, as shown in Figure 1. It still takes O(N^2) calculation but has much less computation. After filtration, only remaining tokens will be attended. Although the selection introduces extra cost, the total efficiency can be improved as far as the aligned range is limited enough.

Basically, we divide our methods into four parts: lightweight attention, attention supervision, adaptive sparsity, and layer sharing, which will be introduced in the following sections.

3.1 Lightweight Attention

Suppose the sequence has N tokens in total and we need to select kN tokens that are important for the current token. k is the selection ratio. Since the selection is only a preliminary process and should only take very little calculation, we project the hidden state of all the tokens from d to d_s (e.g. from 512 to 64). Then a lightweight version of attention is conducted with those low-dimensional hidden states:

$$A_s = \text{softmax}(Q_s K_s^T / \sqrt{d_s}) \quad (1)$$

where Q_s, K_s, and A_s represent the projected query, key, and attention. $Q_s = XW_Q$, $K_s = XW_K$, and $W_Q \in \mathbf{R}^{d \times d_s}$, $W_K \in \mathbf{R}^{d \times d_s}$.

After sorting all the logits, we only preserve the top k keys for each query token and mask the others:

$$mask = \text{top-k}(A_s) \tag{2}$$

Obviously, the top-k function is not differentiable. To train the selection network, we use the reparameter trick from Gumbel Softmax(Jang et al., 2017) to make the parameters learnable:

$$mask = mask + A_s - SG(A_s) \tag{3}$$

where SG refers to stop-gradient. Then the gradient can be passed to A_s while remaining the value of the mask.

3.2 Attention Supervision

Intuitively, the distribution of lightweight attention should be consistent with the original attention layer to ensuring performance. Therefore, we pulls the former to the latter during the training by an addition KL loss. Such distilling process requires no pretrained Transformer model, but the low and high dimension layers are trained with consistency constraint. However, the utilization of original attention prevent speeding up at training time, so we only focus the inference efficiency.

$$A = \text{softmax}(QK^T/\sqrt{d}) \tag{4}$$

$$L_s = \text{kl\_div}(A_s, A) \tag{5}$$

where Q and K are high-dimensional projected hidden states from the original attention layer. kl_div is the Kullback-Leibler Divergence. The loss is added to the NMT loss with a hyperparameter α:

$$Loss = L_{nmt} + \alpha * L_s \tag{6}$$

3.3 Adaptive Sparsity

k represents the level of sparsity and is important in the whole selection procedure. However, the optimal choice of k is not apparent. We propose an adaptive algorithm to search for it.

Specifically, we set a threshold t, for the sum of attention. The intuition is that, since a small amount of tokens contribute to most of the attention weights, "the most of weights" can be quantified as threshold t. If the current sum of attention is below t, some important tokens might be filtered, so we slightly increase k for a small step, and vice versa:

$$k = \begin{cases} k - step & \text{if sum(topk) > t} \\ k + step & \text{else} \end{cases} \tag{7}$$

We regard k as a percentage, so k is in the range $[0, 1]$, and the step is a small constant such as 0.001. We initialize k as 1 and limit k great than or equal to 1%. For documents with few sentences, at least 10 tokens are attended to avoid poor performance. While k gradually decreases and converges in the training process, the model is encouraged to learn a concentrated attention distribution and get rid of unrelated information. In some layers, especially encoder layer, k might stuck at some point and sometimes never decrease, so we manually disable $k + step$ when k is large.

3.4 Layer Sharing

Furthermore, we share the learned sparsity patterns across layers as (Xiao et al., 2019) has proved that attention weights can be directly reused because intuitively, each query in different layers often attends to the same keys. So the extra selection cost can be further reduced while keeping the translation performance.

Basically, we divide all the selection layers into m groups and each group has $r = n/m$ layers, where n is the original layer number. We only calculate the attention of the lowest selection layer in each group. Then the other selection layers share the same attention as the lowest one:

$$A_{s_i} = A_{s_{\lfloor i/r \rfloor * r}} \tag{8}$$

In this way, we can save $m * (r - 1)$ calculation of attention selection.

3.5 Cost Saving

In the end, we try to formalize the attention cost with these algorithms and parameters. The attention cost of the original Transformer attention (Tay et al., 2020b):

$$C_{Transformer} = 2nN^2d \tag{9}$$

where n is the layer number, N is the sequence length, and d is the dimension of the hidden states. "2" means dot product ($A = QK$) and weighted sum (AV). And Lasformer can achieve a complexity as:

$$C_{Lasformer} = \frac{1}{r} \cdot nN^2 d_s + 2knN^2 d \qquad (10)$$

where r is the layer number in each group, d_s is the dimension of the selection layer, and k is the selection ratio. The first item means the d_s-dimension rough selection. The second item means masked attention.

With a small dimension for selection (d_s), a high sparsity for attention (k), and a large layer group size (r), we can greatly reduce the total computation cost. If we set $n = 6$, $d = 512$ as Transformer base, and $d_s = 64$, $t = 0.95$ ($k = 0.05$), $r = 3$, the attention cost can be only **7%** compared with original Transformer. The detailed results are listed in the following sections.

4 Experiments

4.1 Datasets

We conduct experiments on three English-German datasets and one Chinese-English datasets. The English-German datasets include TED, News, and Europarl, following Maruf et al. (2019). The TED corpus is from the IWSLT 2017, and we use tst2016-2017 as test est and the rest are used for development. News are aligned document-delimited News Commentary-v11 corpus, and WMT'16 newstest2015 and news-test2016 are used for development and testing, respectively. Europarl is extracted as proposed in Maruf et al. (2019). For Chinese-English datasets, we follow Sun et al. (2022b), using PDC which is crawled bilingual news corpus with diverse domains.

The above training data are organized into a mix of sentence-level data and document-level data as used in Sun et al. (2022b). All of the data are cut into sub-words using BPE with 32k merge operations.

4.2 Model settings

We build our translation model based on Transformer base (Vaswani et al., 2017) using fairseq(Ott et al., 2019), including 6 layers, 512 dimensions, 8 heads, 2048 feed-forward hidden size, for both encoders and decoders. We use a small dropout of 0.1, as well as word dropout, on large datasets like Europarl and PDC, and a large dropout of 0.3 on small datasets like TED and News.

As for our proposed selection layer, we use $d_s = 64$ dimensions, $m = 2$ groups, and $r = 3$ layers. The coefficient α is set to 0.01 and the threshold t for dynamic top-k is set to 0.95.

We adopt case-insensitive sacreBLEU (Post, 2018) on the whole documents, following all the previous document-level NMT studies.

4.3 Comparison Work

We compare the results with three typical efficient Transformers from different classes of methods and directly use their open-source code to conduct experiments on the datasets:

- **LSH-trans** (Petrick et al., 2022) [1] is based on Reformer and uses locality sensitive hashing to obtain a cluster of tokens to be attended to each other within it.
- **Luna** (Ma et al., 2021)[2] is a low-rank-based model that compresses the long sequence into a fixed number of global tokens using the attention mechanism.
- **RFA-trans** (Wu et al., 2022) extends the RFA (Peng et al., 2021) with sentence level gating mechanism to enhance the locality[3].

There are many other efficient Transformer studies(Beltagy et al., 2020; Zaheer et al., 2020; Tay et al., 2020a, 2021a). However, since they bypass sequence-to-sequence generation tasks and only focus on the encoder-only or decoder-only task, we do not involve them here.

4.4 Results

Table 1 shows the translation results compared to previous document-level translation models. As for efficiency, all related studies achieve cost saving to various extents. They yield better results in terms of cost or speed. However, they face a serious quality drop when dealing with real long-range documents. We find that although they report a comparable result on WMT or IWSLT (with very limited context, around 30 tokens per sentence), there is a large performance decrease on long documents like TED, Europarl, and PDC. These results are obtained by their open-source codes. We suggest that all efficient-related studies should be

[1] https://github.com/rwth-i6/returnn-experiments/tree/master/2022-lsh-attention

[2] https://github.com/XuezheMax/fairseq-apollo

[3] https://github.com/ZhaofengWu/rfa-doc-mt

Models	Efficiency		Quality (BLEU)			
	Attn Cost	Infer Speed	TED	News	Europarl	PDC
♡Transformer (Sun et al., 2022b)	100%	1.0x	27.96	25.05	34.48	27.80
♠LSH-trans (Petrick et al., 2022)	2%	0.8x	9.80	10.04	18.44	17.82
♢Luna (Ma et al., 2021)	12%	1.5x	10.15	9.02	20.32	19.44
♣RFA-trans(Wu et al., 2022)	10%	1.8x	16.93	16.92	26.91	23.48
Lasformer	7%	1.2x	27.24	25.95	34.62	28.04

♡ It adopts the original Transformer and we use MR Doc2Doc setting.
♠ Its complexity is $nCd(N/C)^2 + ndNlogN$, where C is hashing chunk size, and we set C=N/32 as decribed in the paper. The hashing (independent of attention) and sorting take a very long time, yielding a overall low speed.
♢ Its complexity is $2nNdm$, where m is the number of compressed tokens. We set m=64.
♣ Its complexity is $2n(Nd'^2 + Ndd')$, where d' is the projection dim set to 128, n is the number of layers and N is the sequence length. We set n=6 and N = 1000 for the above settings.

Table 1: The results of document-level translation. Except for baseline, we also list the attention cost and inference speed of three typical studies on efficient seq2seq generation. They achieve better efficiency, in terms of cost or speed. However, they face severe drop when dealing with the real document-level translation. Overall, Lasformer achieves the best results of long-range document-level translation.

verified on real long-range sequences. Otherwise, some potential risks may be ignored.

Overall, Lasformer achieves the best results, not only reducing the attention calculation and boosting end-to-end inference speed effectively but also maintaining the translation quality. Notably, we cut down the attention cost to 7%, which is important for the quadratic growth with the sequence length.

Model	TC	CP	PT	TCP
Transformer	56.3	38.1	40.2	44.1
Lasformer	54.4	37.4	41.9	44.0

Table 2: Results on TCP.

Meanwhile, except for BLEU, we conduct experiments on document-level test set to evaluate the capability of utilizing document context. We do not use contrastive test sets(Voita et al., 2019; Bawden et al., 2018) because their instance only contains at most 5 sentences. Instead, we test our model on PDC(Sun et al., 2022b), including Tense Consistency(**TC**), Conjunction Presence(**CP**), Pronoun Translation(**PT**) and an overall score **TCP** that is the geometric mean of above. Table 2 shows that our model achieve comparable results as Transformer baseline and our selection strategy keeps the tokens of importance for handling discourse coherence.

5 Analysis

In this section, we will dive into the method and analyze some important parts and interesting phenomena. Except for extra explanation, the basic setting of all the experiments is as follows: $t = 0.95$, $d_s = 64$, $r = 2$. Datasets are PDC.

5.1 Sparsity Distribution

Since the efficiency of our model totally relies on the learned topk sparse pattern, it is our major concern that to what extent the sparsity can achieve. As is shown in Table 3, Lasformer yields very sparse attention results.

We also find the degree of sparsity among different modules is different. The decoder self-attention can achieve an extreme sparsity of 2%, showing most past contexts are not crucial to the language model. While encoder self-attention only shows 10% sparsity. We suggest that the distribution of attention on the source side is relatively flat so the model needs more tokens. Considering the encoder is non-autoregressive, the strong reduction of the decoder side, including cross-attention and self-attention, can significantly boost efficiency. And even under such great sparsity, Lasformer can still reach a comparable translation result.

Sparsity	Enc	Crs	Dec
Layer 0	10.0%	3.0%	1.8%
Layer 3	9.8%	2.9%	2.7%

Table 3: Sparsity that different attention modules can achieve. Enc, Crs, Dec refer to encoder self-attention, cross-attention and decoder self-attention respectively. Layer 0-2 and layer 3-5 share the same attention.

5.2 Abalation Study

Table 4 shows the effects of the different modules we proposed.

"- Top-k Selection" means that we abandon the Top-k selection. Instead, we limit the attention range within a fixed window whose center is the query and length is 20. Though getting a lower attention cost, its quality deterioration shows that naive human prior is not robust and leads to quality drop.

"- Attention Supervision" means that we set α in formula 6 to 0, thus not constraining the consistency between the attention of the selection layer and the original layer. Consequently, the BLEU score has a large drop, showing the importance of attention supervision. And the lack of supervision might cause the failure of previous sparsity-based efficient transformers.

"- Re-parameter trick" means that we do not use formula 3 so that the parameters of the selection layer are only trained by attention supervision loss and do not contribute to NMT loss. The BLEU score has a small drop, showing that the re-parameter trick helps.

It achieves a comparable result but significantly raises the computation cost.

	Attn Cost	BLEU
Lasformer	7%	28.04
- Top-k Selection	4%	26.52
- Attention Supervision	7%	12.94
- Re-parameter Trick	7%	27.58

Table 4: Effects of different modules. Dynamic Selection, Attention Supervision, and Re-parameter Trick mainly contribute to the quality maintaince. Layer Sharing mainly contributes to the efficiency boost.

5.3 t and k: Sparsity Effects

The sparsity degree k is also an important index. We conduct some experiments to check the relationship between the attention sum t and the sparsity k. As is shown in Table 5, lower attention sum requirements bring more sparsity but also slightly lower BLEU. We choose $t = 0.95$ as a tradeoff.

Threshold	k	Attn Cost	BLEU
$t = 0.90$	3%	5%	27.90
$t = 0.95$	5%	7%	28.04
$t = 0.99$	14%	16%	28.22

Table 5: Effects of the attention threshold and the sparsity they achieve.

5.4 N: The Longer, The More Efficient

Since our method aims at long-range sequences, it is necessary to look into the effect of the sequence length. We calculate the total cost of attention (including QKV linear projection) with the sequence length.

As is shown in Figure 2, as the sequence gets longer, the attention cost ratio gradually decreases, which means we obtain higher and higher efficiency. For the extremely long sequence like 8K tokens, we can lower the attention cost to 15% (7% if not including QKV linear projection). This shows the extraordinary potential of our methods. As the translation range gets wider and wider (e.g. a whole book or movie), Lasformer can obtain high efficiency.

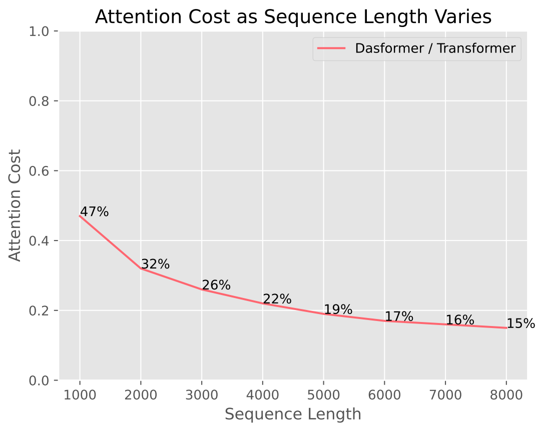
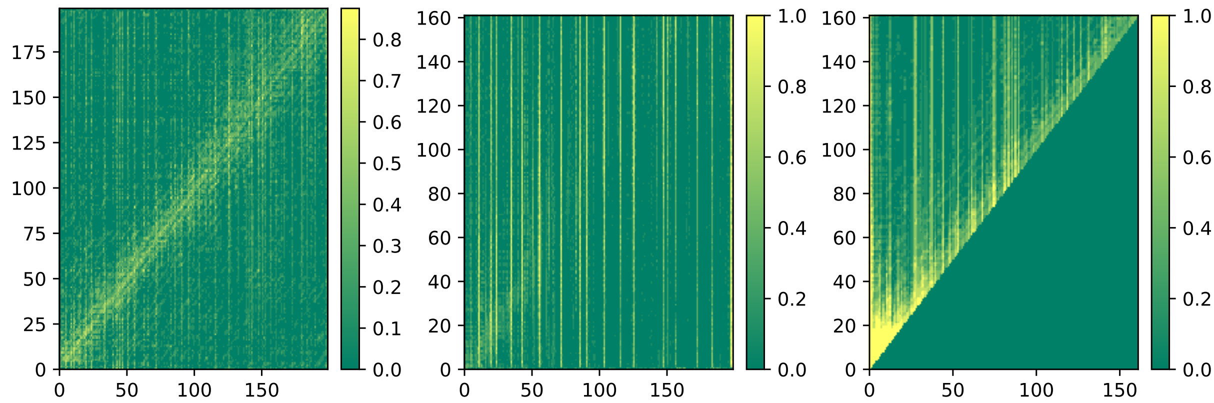


Figure 2: Attention cost (including QKV linear projection) of Lasformer compared to original Transformer with different sequence length.

5.5 d_s: U-shaped Curve with Cost

Obviously, the low dimension sacrifices the model precision to decrease the computation cost. Therefore, it is a tradeoff to balance efficiency and performance. We conduct a series of experiments. Table 6 shows the efficiency and performance under different dimensions.

We find that a low dimension of 32 is enough for a coarse selection, while a dimension of 16 hurts the performance. Also, a lower dimension of the selection layer can bring higher sparsity k which conversely raises the computation cost. Even if we only focus on efficiency, the lowest dimension does not mean the lowest cost. The attention cost goes down and then up as d_s decreases. Therefore, we pick $d_s = 64$ as our final setting.



(a) Encoder Self-attention (b) Cross-attention (c) Decoder Self-attention

Figure 3: Visualization of all three kinds of attention. On the one hand, only a handful of tokens are necessary while most of the others are noise. On the other hand, the distribution shows some regular pattern but many attended tokens is still ruleless.

	k	Attn Cost	BLEU
Transformer	100%	100%	27.80
$d_s = 16$	24%	23%	24.62
$d_s = 32$	15%	16%	27.88
$d_s = 64$	5.0%	7%	28.04
$d_s = 128$	4.8%	9%	28.05
$d_s = 256$	4.3%	13%	28.06
$d_s = 512$	5.2%	21%	28.20

Table 6: Effects of different selection dimensions and the sparsity they achieve.

	k	Attn Cost	BLEU
012345	9%	10%	27.85
012 345	5%	7%	28.04
01 23 45	5%	8%	28.12
0 1 2 3 4 5	5%	11%	28.26

Table 7: Attention sharing in the selection layer. The numbers sharing a underline are in the same group and share the same attention pattern.

5.6 r: Sharing Layers Helps

Another point is that the sparse pattern is obtained in one selection layer and applied to all of the layers within a layer group. We suggest that some adjacent layers share the same function so their attention can be shared together. For example, the lower layers are expected to learn the syntactic information while the higher ones are expected to learn semantic information. So some attention distribution can be shared across the layers. As is shown in Table 7, sharing layers slightly enhance k and drop BLEU. We suggest that sharing too many layers limits the model capacity while not sharing results in some redundancy. Taking $r = 3$ yields the best results.

5.7 Visualization: Attention Patterns

Figure 3 shows the sparsity patterns on encoder self-attention, cross-attention, and decoder self-attention.

On the one hand, there exist some common characteristics, such as: 1) Most tokens prefer to attend to nearby tokens. 2) Some tokens serve as the global token that almost all tokens attend to it, which might be some punctuation. These characteristics shares the same idea with human prior(Child et al., 2019; Beltagy et al., 2020).

On the other hand, there are also plenty of ruleless distributions, including very far tokens. We suggest that long-range context can contribute to the current token like tense or pronoun (Sun et al., 2022b). These drifting attentions can not be handled by human prior while Lasformer can well cope with it.

6 Conclusion

In this paper, we focus on the long-range document-level translation efficiency due to its quadratic cost growth with the length. However, previous studies suffer severe performance drops when inferring real long sequences. To address this issue, We propose to select important tokens with lightweight attention, which is supervised by the original attention. The proposed Lasformer effectively reduces the attention expense while successfully maintains the translation quality. It turns out that only around

5% of attention is necessary and the attention cost can be reduced to 7%. In the end, we achieve an overall acceleration of 20%.

Limitation

The main limitation of this work is that the reduction of cost does not reflect the actual acceleration, which is influenced by linear modules and GPU optimization.

Linear modules include embedding layers, projection of query, key, value, and feed-forward network. Actually, they are the dominant bottleneck when the sequence length is short. We test the time cost for different modules of various input length and find that the attention modules becomes the bottleneck (over 50%) only when the input length is over 1500 tokens. Therefore, the acceleration is relatively minor when the input is short.

GPU optimization is another important concern. First, due to the parallel computing property, a linear layer of 512 x 32 is not 8x faster than a linear layer of 512 x 512. It depends on the GPU architecture and even batch size. The more GPU cores and a small batch size result in a lower GPU utilization and a small speedup. Second, the sparse model is not as fast as a dense model in terms of GPU memory access and pre-fetching, so more memory reading cost is inevitable, which hurts the final end-to-end speedup.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments. Part of this work is supported by National Science Foundation of China (No. 62376116, 62176120), the Liaoning Provincial Research Foundation for Basic Research (No. 2022-KF-26-02).

References

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3442–3455. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. 2020. Rethinking attention with performers. *CoRR*, abs/2009.14794.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. 2021. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. 2022. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4296–4313. Association for Computational Linguistics.

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal. European Association for Machine Translation.

Lorenzo Lupo, Marco Dinarelli, and Laurent Besacier. 2022. Divide and rule: Effective pre-training for context-aware multi-encoder translation models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4557–4572.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3505–3511, Online. Association for Computational Linguistics.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. 2021. Luna: Linear unified nested attention. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2441–2453.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3092–3102. Association for Computational Linguistics.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. 2021. Random feature attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Frithjof Petrick, Jan Rosendahl, Christian Herold, and Hermann Ney. 2022. Locality-sensitive hashing for long context neural machine translation. In *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL 2022, Dublin, Ireland (in-person and online), May 26-27, 2022*, pages 32–42. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. 2022. cosformer: Rethinking softmax in attention. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.

Zewei Sun, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2022a. Alleviating the inequality of attention heads for neural machine translation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5246–5250. International Committee on Computational Linguistics.

Zewei Sun, Shujian Huang, Hao-Ran Wei, Xinyu Dai, and Jiajun Chen. 2020. Generating diverse translation by manipulating multi-head attention. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8976–8983. AAAI Press.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022b. Rethinking document-level neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3537–3548. Association for Computational Linguistics.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2021a. Synthesizer: Rethinking self-attention for transformer models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10183–10192. PMLR.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020a. Sparse sinkhorn attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9438–9447. PMLR.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021b. Long range arena : A benchmark for efficient transformers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020b. Efficient transformers: A survey. *CoRR*, abs/2009.06732.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Ningning Wang, Guobing Gan, Peng Zhang, Shuai Zhang, Junqiu Wei, Qun Liu, and Xin Jiang. 2022. ClusterFormer: Neural clustering attention for efficient and effective transformer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2390–2402, Dublin, Ireland. Association for Computational Linguistics.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768.

Zhaofeng Wu, Hao Peng, Nikolaos Pappas, and Noah A. Smith. 2022. Modeling context with linear attention for scalable document-level translation. *CoRR*, abs/2210.08431.

Tong Xiao, Yinqiao Li, Jingbo Zhu, Zhengtao Yu, and Tongran Liu. 2019. Sharing attention weights for fast transformer. *arXiv preprint arXiv:1906.11024*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 533–542. Association for Computational Linguistics.

Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. 2019. Explicit sparse transformer: Concentrated attention through explicit selection. *arXiv preprint arXiv:1912.11637*.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Towards making the most of context in neural machine translation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3983–3989. ijcai.org.

Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. 2021. Long-short transformer: Efficient transformers for language and vision. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17723–17736.