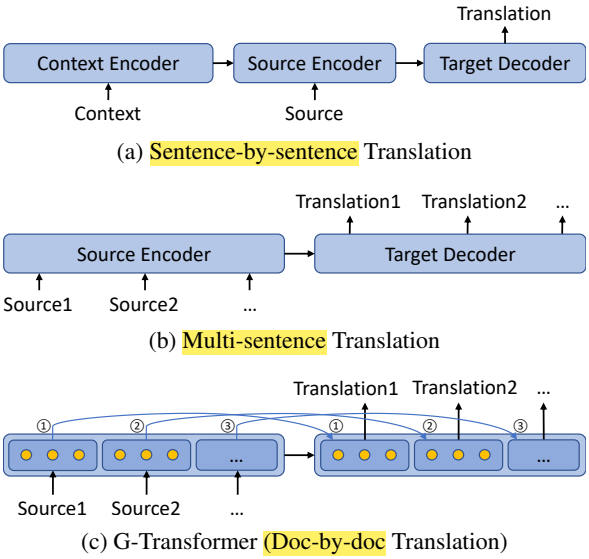
G-Transformer for Document-level Machine Translation

Guangsheng Bao[1,2], **Yue Zhang**[*1,2], **Zhiyang Teng**[1,2], **Boxing Chen**[3] **and Weihua Luo**[3]
[1] School of Engineering, Westlake University
[2] Institute of Advanced Technology, Westlake Institute for Advanced Study
[3] DAMO Academy, Alibaba Group Inc.
{baoguangsheng, zhangyue, tengzhiyang}@westlake.edu.cn
{boxing.cbx, weihua.luowh}@alibaba-inc.com

Abstract

Document-level MT models are still far from satisfactory. Existing work extend translation unit from single sentence to multiple sentences. However, study shows that when we further enlarge the translation unit to a whole document, supervised training of Transformer can fail. In this paper, we find such failure is not caused by overfitting, but by sticking around local minima during training. Our analysis shows that the increased complexity of target-to-source attention is a reason for the failure. As a solution, we propose G-Transformer, introducing locality assumption as an inductive bias into Transformer, reducing the hypothesis space of the attention from target to source. Experiments show that G-Transformer converges faster and more stably than Transformer, achieving new state-of-the-art BLEU scores for both non-pretraining and pre-training settings on three benchmark datasets.



(a) Sentence-by-sentence Translation

(b) Multi-sentence Translation

(c) G-Transformer (Doc-by-doc Translation)

Figure 1: Overview of model structures for document-level machine translation.

1 Introduction

Document-level machine translation (MT) has received increasing research attention (Gong et al., 2011; Hardmeier et al., 2013; Garcia et al., 2015; Miculicich et al., 2018a; Maruf et al., 2019; Liu et al., 2020). It is a more practically useful task compared to sentence-level MT because typical inputs in MT applications are text documents rather than individual sentences. A salient difference between document-level MT and sentence-level MT is that for the former, much larger inter-sentential context should be considered when translating each sentence, which include discourse structures such as anaphora, lexical cohesion, etc. Studies show that human translators consider such contexts when conducting document translation (Hardmeier, 2014; Läubli et al., 2018). Despite that neural models achieve competitive performances on sentence-level MT, the performance of document-level MT is still far from satisfactory.

Existing methods can be mainly classified into two categories. The first category translates a document sentence by sentence using a sequence-to-sequence neural model (Zhang et al., 2018; Miculicich et al., 2018b; Maruf et al., 2019; Zheng et al., 2020). Document-level context is integrated into sentence-translation by introducing additional context encoder. The structure of such a model is shown in Figure 1(a). These methods suffer from two limitations. First, the context needs to be encoded separately for translating each sentence, which adds to the runtime complexity. Second, more importantly, information exchange cannot be made between the current sentence and its document context in the same encoding module.

The second category extends the translation unit from a single sentence to multiple sentences (Tiedemann and Scherrer, 2017; Agrawal et al.,

* Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3442–3455
August 1–6, 2021.

2018; Zhang et al., 2020) and the whole document (Junczys-Dowmunt, 2019; Liu et al., 2020). Recently, it has been shown that when the translation unit increases from one sentence to four sentences, the performance improves (Zhang et al., 2020; Scherrer et al., 2019). However, when the whole document is encoded as a single unit for sequence to sequence translation, direct supervised training has been shown to fail (Liu et al., 2020). As a solution, either large-scale pre-training (Liu et al., 2020) or data augmentation (Junczys-Dowmunt, 2019) has been used as a solution, leading to improved performance. These methods are shown in Figure 1(b). One limitation of such methods is that they require much more training time due to the necessity of data augmentation.

Intuitively, encoding the whole input document as a single unit allows the best integration of context information when translating the current sentence. However, little work has been done investigating the underlying reason why it is difficult to train such a document-level NMT model. One remote clue is that as the input sequence grows larger, the input becomes more sparse (Pouget-Abadie et al., 2014; Koehn and Knowles, 2017). To gain more understanding, we make dedicated experiments on the influence of input length, data scale and model size for Transformer (Section 3), finding that a Transformer model can fail to converge when training with long sequences, small datasets, or big model size. We further find that for the failed cases, the model gets stuck at local minima during training. In such situation, the attention weights from the decoder to the encoder are flat, with large entropy values. This can be because that larger input sequences increase the challenge for focusing on a local span to translate when generating each target word. In other words, the hypothesis space for target-to-source attention is increased.

Given the above observations, we investigate a novel extension of Transformer, by restricting self-attention and target-to-source attention to a local context using a guidance mechanism. As shown in Figure 1(c), while we still encode the input document as a single unit, group tags ① ② ③ are assigned to sentences to differentiate their positions. Target-to-source attention is guided by matching the tag of target sentence to the tags of source sentences when translating each sentence, so that the hypothesis space of attention is reduced. Intuitively, the group tags serve as a constraint on attention, which is useful for differentiating the current sentence and its context sentences. Our model, named G-Transformer, can be thus viewed as a combination of the method in Figure 1(a) and Figure 1(b), which fully separate and fully integrates a sentence being translated with its document level context, respectively.

We evaluate our model on three commonly used document-level MT datasets for English-German translation, covering domains of TED talks, News, and Europarl from small to large. Experiments show that G-Transformer converges faster and more stably than Transformer on different settings, obtaining the state-of-the-art results under both non-pretraining and pre-training settings. To our knowledge, we are the first to realize a truly document-by-document translation model. We release our code and model at https://github.com/baoguangsheng/g-transformer.

2 Experimental Settings

We evaluate Transformer and G-Transformer on the widely adopted *benchmark datasets* (Maruf et al., 2019), including three domains for English-German (En-De) translation.

TED. The corpus is transcriptions of TED talks from IWSLT 2017. Each talk is used as a document, aligned at the sentence level. *tst2016-2017* is used for testing, and the rest for development.

News. This corpus uses News Commentary v11 for training, which is document-delimited and sentence-aligned. *newstest2015* is used for development, and *newstest2016* for testing.

Europarl. The corpus is extracted from Europarl v7, where sentences are segmented and aligned using additional information. The train, dev and test sets are randomly split from the corpus.

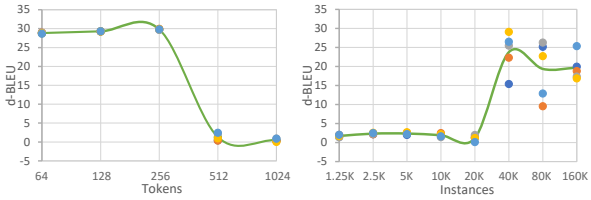
The detailed statistics of these corpora are shown in Table 1. We pre-process the documents by splitting them into instances with up-to 512 tokens, taking a sentence as one instance if its length exceeds 512 tokens. We tokenize and truecase the sentences with MOSES (Koehn et al., 2007) tools, applying BPE (Sennrich et al., 2016) with 30000 merging operations.

We consider three standard *model configurations*.

Base Model. Following the standard Transformer base model (Vaswani et al., 2017), we use 6 layers, 8 heads, 512 dimension outputs, and 2048

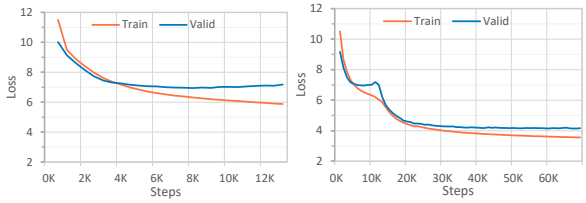
Language	Dataset	#Sentences train/dev/test	#Documents train/dev/test	#Instances train/dev/test	Avg #Sents/Inst train/dev/test	Avg #Tokens/Inst train/dev/test
En-De	TED	0.21M/9K/2.3K	1.7K/92/22	11K/483/123	18.3/18.5/18.3	436/428/429
	News	0.24M/2K/3K	6K/80/154	18.5K/172/263	12.8/12.6/11.3	380/355/321
	Europarl	1.67M/3.6K/5.1K	118K/239/359	162K/346/498	10.3/10.4/10.3	320/326/323

Table 1: En-De datasets for evaluation.



(a) Input Length (Base model with filtered data.) (b) Data Scale (Base model with 512 tokens input.)

Figure 2: Transformer on various input length and data scale.



(a) Failed Model (b) Successful Model

Figure 3: Loss curve of the models and the local minima.

dimension hidden vectors.

Big Model. We follow the standard Transformer big model (Vaswani et al., 2017), using 6 layers, 16 heads, 1024 dimension outputs, and 4096 dimension hidden vectors.

Large Model. We use the same settings of BART large model (Lewis et al., 2020), which involves 12 layers, 16 heads, 1024 dimension outputs, and 4096 dimension hidden vectors.

We use s-BLEU and d-BLEU (Liu et al., 2020) as the *metrics*. The detailed descriptions are in Appendix A.

3 Transformer and Long Inputs

We empirically study Transformer (see Appendix B) on the datasets. We run each experiment five times using different random seeds, reporting the average score for comparison.

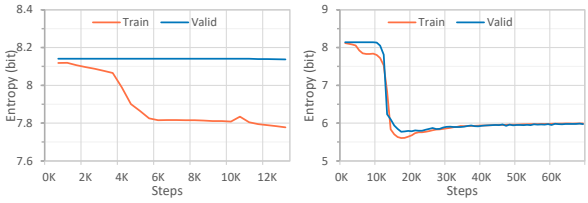
3.1 Failure Reproduction

Input Length. We use the Base model and fixed dataset for this comparison. We split both the training and testing documents from Europarl dataset into instances with input length of 64, 128, 256, 512, and 1024 tokens, respectively. For fair comparison, we remove the training documents with a length of less than 768 tokens, which may favour small input length. The results are shown in Figure 2a. When the input length increases from 256 tokens to 512 tokens, the BLEU score drops dramatically from 30.5 to 2.3, indicating failed training with 512 and 1024 tokens. It demonstrates the difficulty when dealing with long inputs of Transformer.

Data Scale. We use the Base model and a fixed input length of 512 tokens. For each setting, we randomly sample a training dataset of the expected size from the full dataset of Europarl. The results are shown in Figure 2b. The performance increases sharply when the data scale increases from 20K to 40K. When data scale is equal or less than 20K, the BLEU scores are under 3, which is unreasonably low, indicating that with a fixed model size and input length, the smaller dataset can also cause the failure of the training process. For data scale more than 40K, the BLEU scores show a wide dynamic range, suggesting that the training process is unstable.

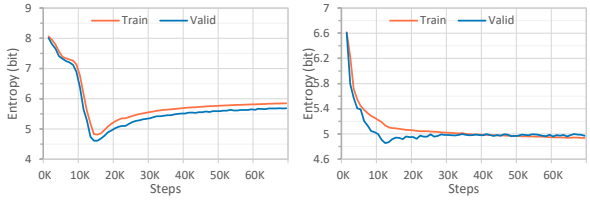
Model Size. We test Transformer with different model sizes, using the full dataset of Europarl and a fixed input length of 512 tokens. Transformer-Base can be trained successfully, giving a reasonable BLEU score. However, the training of the Big and Large models failed, resulting in very low BLEU scores under 3. It demonstrates that the increased model size can also cause the failure with a fixed input length and data scale.

The results confirm the intuition that the performance will drop with longer inputs, smaller datasets, or bigger models. However, the BLEU scores show a strong discontinuity with the change of input length, data scale, or model size, falling into two discrete clusters. One is successfully trained cases with d-BLEU scores above 10, and the other is failed cases with d-BLEU scores under 3.



(a) Failed Model (b) Successful Model

Figure 4: Cross-attention distribution of Transformer shows that the failed model sticks at the local minima.



(a) Encoder (b) Decoder

Figure 5: For the successful model, the attention distribution shrinks to narrow range (low entropy) and then expands to wider range (high entropy).

3.2 Failure Analysis

Training Convergence. Looking into the failed models, we find that they have a similar pattern on loss curves. As an example of the model trained on 20K instances shown in Figure 3a, although the training loss continually decreases during training process, the validation loss sticks at the level of 7, reaching a minimum value at around 9K training steps. In comparison, the successfully trained models share another pattern. Taking the model trained on 40K instances as an example, the loss curves demonstrate two stages, which is shown in Figure 3b. In the first stage, the validation loss similar to the failed cases has a converging trend to the level of 7. In the second stage, after 13K training steps, the validation loss falls suddenly, indicating that the model may escape successfully from local minima. From the two stages of the learning curve, we conclude that the real problem, contradicting our first intuition, is not about overfitting, but about local minima.

Attention Distribution. We further look into the attention distribution of the failed models, observing that the attentions from target to source are widely spread over all tokens. As Figure 4a shows, the distribution entropy is high for about 8.14 bits on validation. In contrast, as shown in Figure 4b, the successfully trained model has a much lower attention entropy of about 6.0 bits on validation. Furthermore, we can see that before 13K training steps, the entropy sticks at a plateau, confirming with the observation of the local minima in Figure 3b. It indicates that the early stage of the training process for Transformer is difficult.

Source: <s> the Commission shares ... of the European Union institutional framework . </s>① <s> Commission participation is expressly provided for ... of all its preparatory bodies . </s>② <s> only in exceptional circumstances ... be excluded from these meetings . </s>③ ...

Target: <s> die Kommission teilt die Ansicht ... des institutionellen Rahmens der Europischen Union ist . </s>① <s> die Geschftsordnung des Rates ... der Kommission damit ausdrcklich vor . </s>② <s> die Kommission kann nur ... wobei fallweise zu entscheiden ist . </s>③ ...

Figure 6: Example of English-German translation with group alignments.

Figure 5 shows the self-attention distributions of the successfully trained models. The attention entropy of both the encoder and the decoder drops fast at the beginning, leading to a shrinkage of the attention range. But then the attention entropy gradually increases, indicating an expansion of the attention range. Such back-and-forth oscillation of the attention range may also result in unstable training and slow down the training process.

3.3 Conclusion

The above experiments show that training failure on Transformer can be caused by local minima. Additionally, the oscillation of attention range may make it worse. During training process, the attention module needs to identify relevant tokens from whole sequence to attend to. Assuming that the sequence length is N, the complexity of the attention distribution increases when N grows from sentence-level to document-level.

We propose to use locality properties (Rizzi, 2013; Hardmeier, 2014; Jawahar et al., 2019) of both the language itself and the translation task as a constraint in Transformer, regulating the hypothesis space of the self-attention and target-to-source attention, using a simple group tag method.

4 G-Transformer

An example of G-Transformer is shown in Figure 6, where the input document contains more than 3 sentences. As can be seen from the figure, G-Transformer extends Transformer by augmenting the input and output with group tags (Bao and Zhang, 2021). In particular, each token is assigned a group tag, indicating its sentential index. While

source group tags can be assigned deterministically, target tags are assigned dynamically according to whether a generated sentence is complete. Starting from 1, target words copy group tags from its predecessor unless the previous token is </s>, in which case the tag increases by 1. The tags serve as a locality constraint, encouraging target-to-source attention to concentrate on the current source sentence being translated.

Formally, for a source document X and a target document Y, the probability model of Transformer can be written as

$$\hat{Y} = \arg\max_{Y} P(Y|X), \tag{1}$$

and G-Transformer extends it by having

$$\hat{Y} = \arg_Y \max_{Y,G_Y} P(Y, G_Y|X, G_X), \tag{2}$$

where G_X and G_Y denotes the two sequences of group tags

$$\begin{aligned} G_X &= \{g_i = k \text{ if } w_i \in sent_k^X \text{ else } 0\}|_{i=1}^{|X|}, \\ G_Y &= \{g_j = k \text{ if } w_j \in sent_k^Y \text{ else } 0\}|_{j=1}^{|Y|}, \end{aligned} \tag{3}$$

where $sent_k$ represents the k-th sentence of X or Y. For the example shown in Figure 6, $G_X = \{1, ..., 1, 2, ..., 2, 3, ..., 3, 4, ...\}$ and $G_Y = \{1, ..., 1, 2, ..., 2, 3, ..., 3, 4, ...\}$.

Group tags influence the auto-regressive translation process by interfering with the attention mechanism, which we show in the next section. In G-Transformer, we use the group-tag sequence G_X and G_Y for representing the alignment between X and Y, and for generating the *localized* contextual representation of X and Y.

4.1 Group Attention

An attention module can be seen as a function mapping a query and a set of key-value pairs to an output (Vaswani et al., 2017). The query, key, value, and output are all vectors. The output is computed by summing the values with corresponding attention weights, which are calculated by matching the query and the keys. Formally, given a set of queries, keys, and values, we pack them into matrix Q, K, and V, respectively. We compute the matrix outputs

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{4}$$

where d_k is the dimensions of the key vector.

Attention allows a model to focus on different positions. Further, multi-head attention (MHA) allows a model to gather information from different representation subspaces

$$\begin{aligned} \text{MHA}(Q, K, V) &= \text{Concat}(head_1, ..., head_h)W^O, \\ head_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \end{aligned} \tag{5}$$

where the projections of W^O, W_i^Q, W_i^K, and W_i^V are parameter matrices.

We update Eq 4 using group-tags, naming it group attention (GroupAttn). In addition to inputs Q, K, and V, two sequences of group-tag inputs are involved, where G_Q corresponds to Q and G_K corresponds to K. We have

$$\begin{aligned} args &= (Q, K, V, G_Q, G_K), \\ \text{GroupAttn}(args) &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M(G_Q, G_K)\right)V, \end{aligned} \tag{6}$$

where function $M(\cdot)$ works as an attention mask, excluding all tokens outside the sentence. Specifically, $M(\cdot)$ gives a big negative number γ to make *softmax* close to 0 for the tokens with a different group tag compared to current token

$$M(G_Q, G_K) = \min(1, \text{abs}(G_Q I_K^T - I_Q G_K^T)) * \gamma, \tag{7}$$

where I_K and I_Q are constant vectors with value 1 on all dimensions, that I_K has dimensions equal to the length of G_K and I_Q has dimensions equal to the length of G_Q. The constant value γ can typically be $-1e8$.

Similar to Eq 5, we use group multi-head attention

$$\begin{aligned} args &= (Q, K, V, G_Q, G_K), \\ \text{GroupMHA}(args) &= \text{Concat}(head_1, ..., head_h)W^O, \end{aligned} \tag{8}$$

where

$$head_i = \text{GroupAttn}(QW_i^Q, KW_i^K, VW_i^V, G_Q, G_K), \tag{9}$$

and the projections of W^O, W_i^Q, W_i^K, and W_i^V are parameter matrices.

Encoder. For each layer a group multi-head attention module is used for self-attention, assigning the same group-tag sequence for the key and the value that $G_Q = G_K = G_X$.

Decoder. We use one group multi-head attention module for self-attention and another group multi-head attention module for cross-attention. Similar to the encoder, we assign the same group-tag sequence to the key and value of the self-attention, that $G_Q = G_K = G_Y$, but use different group-tag sequences for cross-attention that $G_Q = G_Y$ and $G_K = G_X$.

Method	TED		News		Europarl	
	s-BLEU	d-BLEU	s-BLEU	d-BLEU	s-BLEU	d-BLEU
SENTNMT (Vaswani et al., 2017)	23.10	-	22.40	-	29.40	-
HAN (Miculicich et al., 2018b)	24.58	-	25.03	-	28.60	-
SAN (Maruf et al., 2019)	24.42	-	24.84	-	29.75	-
Hybrid Context (Zheng et al., 2020)	25.10	-	24.91	-	30.40	-
Flat-Transformer (Ma et al., 2020)	24.87	-	23.55	-	30.09	-
Transformer on sent (baseline)	24.82	-	25.19	-	31.37	-
Transformer on doc (baseline)	-	0.76	-	0.60	-	33.10
G-Transformer random initialized (ours)	23.53	25.84*	23.55	25.23*	32.18*	33.87*
G-Transformer fine-tuned on sent Transformer (ours)	**25.12**	**27.17***	**25.52**	**27.11***	**32.39***	**34.08***
Fine-tuning on Pre-trained Model						
Flat-Transformer+BERT (Ma et al., 2020)	26.61	-	24.52	-	31.99	-
G-Transformer+BERT (ours)	**26.81**	-	**26.14**	-	**32.46**	-
Transformer on sent fine-tuned on BART (baseline)	27.78	-	29.90	-	31.87	-
Transformer on doc fine-tuned on BART (baseline)	-	28.29	-	30.49	-	34.00
G-Transformer fine-tuned on BART (ours)	**28.06**	**30.03***	**30.34***	**31.71***	**32.74***	**34.31***

Table 2: Case-sensitive BLEU scores on En-De translation. "*" indicates statistically significant at $p < 0.01$ compared to the Transformer baselines.

Complexity. Consider a document with M sentences and N tokens, where each sentence contains N/M tokens on average. The complexities of both the self-attention and cross-attention in Transformer are $O(N^2)$. In contrast, the complexity of group attention in G-Transformer is $O(N^2/M)$ given the fact that the attention is restricted to a local sentence. Theoretically, since the average length N/M of sentences tends to be constant, the time and memory complexities of group attention are approximately $O(N)$, making training and inference on very long inputs feasible.

4.2 Combined Attention

We use only group attention on lower layers for local sentence representation, and combined attention on top layers for integrating local and global context information. We use the standard multi-head attention in Eq 5 for global context, naming it global multi-head attention (GlobalMHA). Group multi-head attention in Eq 8 and global multi-head attention are combined using a gate-sum module (Zhang et al., 2016; Tu et al., 2017)

$$\begin{aligned} H_L &= \text{GroupMHA}(Q, K, V, G_Q, G_K), \\ H_G &= \text{GlobalMHA}(Q, K, V), \\ g &= \text{sigmoid}([H_L, H_G]W + b), \\ H &= H_L \odot g + H_G \odot (1 - g), \end{aligned} \tag{10}$$

where W and b are linear projection parameters, and $\odot$ denotes element-wise multiplication.

Previous study (Jawahar et al., 2019) shows that the lower layers of Transformer catch more local syntactic relations, while the higher layers represent longer distance relations. Based on these findings, we use combined attention only on the top layers for integrating local and global context. By this design, on lower layers, the sentences are isolated from each other, while on top layers, the cross-sentence interactions are enabled. Our experiments show that the top 2 layers with global attention are sufficient for document-level NMT, and more layers neither help nor harm the performance.

4.3 Inference

During decoding, we generate group-tag sequence G_Y according to the predicted token, starting with 1 at the first <s> and increasing 1 after each </s>. We use beam search and apply the maximum length constraint on each sentence. We generate the whole document from start to end in one beam search process, using a default beam size of 5.

5 G-Transformer Results

We compare G-Transformer with Transformer baselines and previous document-level NMT models on both non-pretraining and pre-training settings. The detailed descriptions about these training settings are in Appendix C.1. We make statistical significance test according to Collins et al. (2005).

5.1 Results on Non-pretraining Settings

As shown in Table 2, the sentence-level Transformer outperforms previous document-level models on News and Europarl. Compared to this strong baseline, our randomly initialized model of G-Transformer improves the s-BLEU by 0.81 point on the large dataset Europarl. The results on the small datasets TED and News are worse, indicating overfitting with long inputs. When G-Transformer is trained by fine-tuning the sentence-

level Transformer, the performance improves on the three datasets by 0.3, 0.33, and 1.02 s-BLEU points, respectively.

Different from the baseline of document-level Transformer, G-Transformer can be successfully trained on small TED and News. On Europarl, G-Transformer outperforms Transformer by 0.77 d-BLEU point, and G-Transformer fine-tuned on sentence-level Transformer enlarges the gap to 0.98 d-BLEU point.

G-Transformer outperforms previous document-level MT models on News and Europarl with a significant margin. Compared to the best recent model Hyrbid-Context, G-Transformer improves the s-BLEU on Europarl by 1.99. These results suggest that in contrast to previous short-context models, sequence-to-sequence model taking the whole document as input is a promising direction.

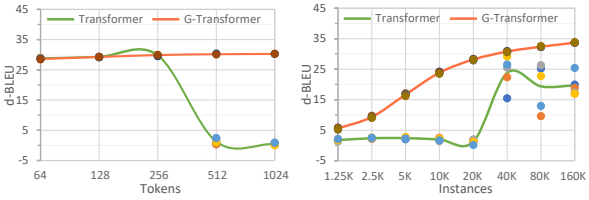
5.2 Results on Pre-training Settings

There is relatively little existing work about document-level MT using pre-training. Although Flat-Transformer+BERT gives a state-of-the-art scores on TED and Europarl, the score on News is worse than previous non-pretraining model HAN (Miculicich et al., 2018b). G-Transformer+BERT improves the scores by margin of 0.20, 1.62, and 0.47 s-BLEU points on TED, News, and Europarl, respectively. It shows that with a better contextual representation, we can further improve document-level MT on pretraining settings.

We further build much stronger Transformer baselines by fine-tuning on mBART25 (Liu et al., 2020). Taking advantage of sequence-to-sequence pre-training, the sentence-level Transformer gives much better s-BLEUs of 27.78, 29.90, and 31.87, respectively. G-Transformer fine-tuned on mBART25 improves the performance by 0.28, 0.44, and 0.87 s-BLEU, respectively. Compared to the document-level Transformer baseline, G-Transformer gives 1.74, 1.22, and 0.31 higher d-BLEU points, respectively. It demonstrates that even with well-trained sequence-to-sequence model, the locality bias can still enhance the performance.

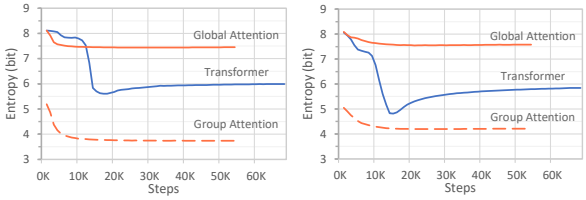
5.3 Convergence

We evaluate G-Transformer ad Transformer on various input length, data scale, and model size to better understand that to what extent it has solved the convergence problem of Transformer.



(a) Input Length (b) Data Scale

Figure 7: G-Transformer compared with Transformer.



(a) Cross-Attention (b) Encoder Self-Attention

Figure 8: Comparison on the development of cross-attention and encoder self-attention.

Input Length. The results are shown in Figure 7a. Unlike Transformer, which fails to train on long input, G-Transformer shows stable scores for inputs containing 512 and 1024 tokens, suggesting that with the help of locality bias, a long input does not impact the performance obviously.

Data Scale. As shown in Figure 7b, overall G-Transformer has a smooth curve of performance on the data scale from 1.25K to 160K. The variances of the scores are much lower than Transformer, indicating stable training of G-Transformer. Additionally, G-Transformer outperforms Transformer by a large margin on all the settings.

Model Size. Unlike Transformer, which fails to train on Big and Large model settings, G-Transformer shows stable scores on different model sizes. As shown in Appendix C.2, although performance on small datasets TED and News drops largely for Big and Large model, the performance on large dataset Europarl only decreases by 0.10 d-BLEU points for the Big model and 0.66 for the Large model.

Loss. Looking into the training process of the above experiments, we see that both the training and validation losses of G-Transformer converge much faster than Transformer, using almost half time to reach the same level of loss. Furthermore, the validation loss of G-Transformer converges to much lower values. These observations demonstrate that G-Transformer converges faster and better.

Attention Distribution. Benefiting from the separate group attention and global attention, G-Transformer avoids the oscillation of attention

Method	TED	News	Europarl	Drop
G-Transformer (fnt.)	25.12	25.52	32.39	-
- target-side context	25.05	25.41	32.16	-0.14
- source-side context	24.56	24.58	31.39	-0.70

Table 3: Impact of source-side and target-side context reporting in s-BLEU. Here, fnt. denotes the model fine-tuned on sentence-level Transformer.

Method	deixis	el.infl.	el.VP
CADec (Voita et al., 2019b)	81.6	72.2	80.0
LSTM-Tran (Zhang et al., 2020)	**91.0**	82.2	78.2
sent (Voita et al., 2019b)	50.0	53.0	28.4
concat (Voita et al., 2019b)	83.5	76.2	76.6
G-Transformer	89.9	**84.8**	**82.4**

Table 4: Impact on discourse by the source-side context, in accuracy of correctly identifying the discourse phenomena. Here, el. means ellipsis. LSTM-Tran denotes LSTM-Transformer.

Method	TED	News	Europarl	Drop
G-Transformer (rnd.)	25.84	25.23	33.87	-
- word-dropout	25.49	24.65	33.70	-0.37
- language locality	22.47	22.41	33.63	-1.78
- translation locality	0.76	0.60	33.10	-14.68

Table 5: Contribution of locality bias and word-dropout reporting in d-BLEU. Here, rnd. denotes the model trained using randomly initialized parameters.

Method	TED	News	Europarl	Drop
G-Transformer (rnd.)				
Combined attention	25.84	25.23	33.87	-
Only group attention	25.62	25.14	33.12	-0.35
Only global attention	25.00	24.54	32.87	-0.84

Table 6: Separate effect of group and global attention reporting in d-BLEU. Here, rnd. denotes the model trained using randomly initialized parameters.

range, which happens to Transformer. As shown in Figure 8a, Transformer sticks at the plateau area for about 13K training steps, but G-Transformer shows a quick and monotonic convergence, reaching the stable level using about 1/4 of the time that Transformer takes. Through Figure 8b, we can find that G-Transformer also has a smooth and stable curve for the convergence of self-attention distribution. These observations imply that the potential conflict of local sentence and document context can be mitigated by G-Transformer.

5.4 Discussion of G-Transformer

Document Context. We study the contribution of the source-side and target-side context by removing the cross-sentential attention in Eq 10 from the encoder and the decoder gradually. The results are shown in Table 3. We take the G-Transformer fine-tuned on the sentence-level Transformer as our starting point. When we disable the target-side context, the performance decreases by 0.14 s-BLEU point on average, which indicates that the target-side context does impact translation performance significantly. When we further remove the source-side context, the performance decrease by 0.49, 0.83, and 0.77 s-BLEU point on TED, News, and Europarl, respectively, which indicates that the source-side context is relatively more important for document-level MT.

To further understand the impact of the source-side context, we conduct an experiment on automatic evaluation on discourse phenomena which rely on source context. We use the human labeled evaluation set (Voita et al., 2019b) on English-Russion (En-Ru) for deixis and ellipsis. We follow the Transformer concat baseline (Voita et al., 2019b) and use both 6M sentence pairs and 1.5M document pairs from OpenSubtitles2018 (Lison et al., 2018) to train our model. The results are shown in Table 4. G-Transformer outperforms Transformer baseline concat (Voita et al., 2019b) with a large margin on three discourse features, indicating a better leverage of the source-side context. When compared to previous model LSTM-T, G-Transformer achieves a better ellipsis on both infl. and VP. However, the score on deixis is still lower, which indicates a potential direction that we can investigate in further study.

Word-dropout. As shown in Table 5, word-dropout (Appendix C.1) contributes about 0.37 d-BLEU on average. Its contribution to TED and News is obvious in 0.35 and 0.58 d-BLEU, respectively. However, for large dataset Europarl, the contribution drops to 0.17, suggesting that with sufficient data, word-dropout may not be necessary.

Locality Bias. In G-Transformer, we introduce locality bias to the language modeling of source and target, and locality bias to the translation between source and target. We try to understand these biases by removing them from G-Transformer. When all the biases removed, the model downgrades to a document-level Transformer. The results are shown in Table 5. Relatively speaking, the contribution of language locality bias is about 1.78 d-BLEU on average. While the translation locality bias contributes for about 14.68 d-BLEU on average, showing critical impact on the model convergence on small datasets. These results suggest that the locality bias may be the key to train

whole-document MT models, especially when the data is insufficient.

Combined Attention. In G-Transformer, we enable only the top K layers with combined attention. On Europarl7, G-Transformer gives 33.75, 33.87, and 33.84 d-BLEU with top 1, 2, and 3 layers with combined attention, respectively, showing that $K = 2$ is sufficient. Furthermore, we study the effect of group and global attention separately. As shown in Table 6, when we replace the combined attention on top 2 layers with group attention, the performance drops by 0.22, 0.09, and 0.75 d-BLEU on TED, News, and Europarl, respectively. When we replace the combined attention with global attention, the performance decrease is enlarged to 0.84, 0.69, and 1.00 d-BLEU, respectively. These results demonstrate the necessity of combined attention for integrating local and global context information.

6 Related Work

The unit of translation has evolved from word (Brown et al., 1993; Vogel et al., 1996) to phrase (Koehn et al., 2003; Chiang, 2005, 2007) and further to sentence (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014) in the MT literature. The trend shows that larger units of translation, when represented properly, can lead to improved translation quality.

A line of document-level MT extends translation unit to multiple sentences (Tiedemann and Scherrer, 2017; Agrawal et al., 2018; Zhang et al., 2020; Ma et al., 2020). However, these approaches are limited within a short context of maximum four sentences. Recent studies extend the translation unit to whole document (Junczys-Dowmunt, 2019; Liu et al., 2020), using large augmented dataset or pretrained models. Liu et al. (2020) shows that Transformer trained directly on document-level dataset can fail, resulting in unreasonably low BLEU scores. Following these studies, we also model translation on the whole document. We solve the training challenge using a novel locality bias with group tags.

Another line of work make document-level machine translation sentence by sentence, using additional components to represent the context (Maruf and Haffari, 2018; Zheng et al., 2020; Zhang et al., 2018; Miculicich et al., 2018b; Maruf et al., 2019; Yang et al., 2019). Different from these approaches, G-Transformer uses a generic design for both source and context, translating whole document in one beam search instead of sentence-by-sentence. Some methods use a two-pass strategy, generating sentence translation first, integrating context information through a post-editing model (Voita et al., 2019a; Yu et al., 2020). In contrast, G-Transformer uses a single model, which reduces the complexity for both training and inference.

The locality bias we introduce to G-Transformer is different from the ones in Longformer (Beltagy et al., 2020) and Reformer (Kitaev et al., 2020) in the sense that we discuss locality in the context of representing the alignment between source sentences and target sentences in document-level MT. Specifically, Longformer introduces locality only to self-attention, while G-Transformer also introduces locality to cross-attention, which is shown to be the key for the success of G-Transformer. Reformer, basically same as Transformer, searches for attention targets in the whole sequence, while G-Transformer mainly restricts the attention inside a local sentence. In addition, the motivations are different. While Longformer and Reformer focus on the time and memory complexities, we focus on attention patterns in cases where a translation model fails to converge during training.

7 Conclusion

We investigated the main reasons for Transformer training failure in document-level MT, finding that target-to-source attention is a key factor. According to the observation, we designed a simple extension of the standard Transformer architecture, using group tags for attention guiding. Experiments show that the resulting G-Transformer converges fast and stably on small and large data, giving the state-of-the-art results compared to existing models under both pre-training and random initialization settings.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. We thank Westlake University High-Performance Computing Center for supporting on GPU resources. This work is supported by grants from Alibaba Group Inc. and Sichuan Lan-bridge Information Technology Co.,Ltd.

References

Ruchit Rajeshkumar Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *21st Annual Conference of the European Association for Machine Translation*, pages 11–20.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Guangsheng Bao and Yue Zhang. 2021. Contextualized rewriting for text summarization. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*.

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

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05)*, pages 531–540.

Eva Martínez Garcia, Cristina España-Bonet, and Lluís Màrquez. 2015. Document-level machine translation with word vector models. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 59–66, Antalya, Turkey.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Christian Hardmeier. 2014. *Discourse in statistical machine translation*. Ph.D. thesis, Acta Universitatis Upsaliensis.

Christian Hardmeier, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. 2013. Docent: A document-level decoder for phrase-based statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 193–198, Sofia, Bulgaria. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods*

in Natural Language Processing, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284, Melbourne, Australia. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018a. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018b. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. 2014. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 78–85, Doha, Qatar. Association for Computational Linguistics.

Luigi Rizzi. 2013. Locality. *Lingua*, 130:169–186.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. 2019. Analysing concatenation approaches to document-level NMT in two different domains. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 51–61, Hong Kong, China. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

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.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, 5:87–99.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. 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.

Zhengxin Yang, Jinchao Zhang, Fandong Meng, Shuhao Gu, Yang Feng, and Jie Zhou. 2019. Enhancing context modeling with a query-guided capsule network for document-level translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1527–1537, Hong Kong, China. Association for Computational Linguistics.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Better document-level machine translation with Bayes' rule. *Transactions of the Association for Computational Linguistics*, 8:346–360.

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*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Pei Zhang, Boxing Chen, Niyu Ge, and Kai Fan. 2020. Long-short term masking transformer: A simple but effective baseline for document-level neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1087, Online. Association for Computational Linguistics.

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-20*, pages 3983–3989.

A Evaluation Metrics

Following Liu et al. (2020), we use sentence-level BLEU score (s-BLEU) as the major metric for our evaluation. However, when document-level Transformer is compared, we use document-level BLEU score (d-BLEU) since the sentence-to-sentence alignment is not available.

s-BLEU. To calculate sentence-level BLEU score on document translations, we first split the translations into sentences, mapping to the corresponding source sentences. Then we calculate the BLEU score on pairs of translation and reference of the same source sentence.

d-BLEU. When the alignments between translation and source sentences are not available, we calculate the BLEU score on document-level, matching n-grams in the whole document.

B Transformer

B.1 Model

Transformer (Vaswani et al., 2017) has an encoder-decoder structure, using multi-head attention and feed-forward network as basic modules. In this paper, we mainly concern about the attention module.

Attention. An attention module works as a function, mapping a query and a set of key-value pairs to an output, that the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a matching function of the query with the corresponding key. Formally, for matrix inputs of query Q, key K, and value V,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (11)$$

where d_k is the dimensions of the key vector.

Multi-Head Attention. Build upon single-head attention module, multi-head attention allows the model to attend to different positions of a sequence, gathering information from different representation subspaces by heads.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O, \quad (12)$$

where

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \quad (13)$$

that the projections of W^O, W_i^Q, W_i^K, and W_i^V are parameter matrices.

Encoder. The encoder consists of a stack of N identical layers. Each layer has a multi-head self-attention, stacked with a feed-forward network. A residual connection is applied to each of them.

Decoder. Similar as the encoder, the decoder also consists of a stack of N identical layers. For each layer, a multi-head self-attention is used to represent the target itself, and a multi-head cross-attention is used to attend to the encoder outputs. The same structure of feed-forward network and residual connection as the encoder is used.

B.2 Training Settings

We build our experiments based on Transformer implemented by Fairseq (Ott et al., 2019). We use shared dictionary between source and target, and use a shared embedding table between the encoder and the decoder. We use the default setting proposed by Transformer (Vaswani et al., 2017), which uses Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, a learning rate of $5e-4$, and an inverse-square schedule with warmup steps of 4000. We apply label-smoothing of 0.1 and dropout of 0.3 on all settings. To study the impact of input length, data scale, and model size, we take the learning rate and other settings as controlled variables that are fixed for all experiments. We determine the number of updates/steps automatically by early stop on validation set. We train base and big models on 4 GPUs of Navidia 2080ti, and large model on 4 GPUs of v100.

C G-Transformer

C.1 Training Settings

We generate the corresponding group tag sequence dynamically in the model according to the special sentence-mark tokens <s> and </s>. Taking a document "<s> there is no public transport . </s> <s> local people struggle to commute . </s>" as an example, a group-tag sequence $G = \{1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2\}$ is generated according to Eq 3, where 1 starts on the first <s> and ends on the first </s>, 2 the second, and so on. The model can be trained either randomly initialized or fine-tuned.

Randomly Initialized. We use the same settings as Transformer to train G-Transformer, using label-smoothing of 0.1, dropout of 0.3, Adam optimizer, and a learning rate of $5e-4$ with 4000 warmup steps. To encourage inferencing the translation from the context, we apply a word-dropout

Method	TED		News		Europarl	
	s-BLEU	d-BLEU	s-BLEU	d-BLEU	s-BLEU	d-BLEU
G-Transformer random initialized (Base)	23.53	25.84	23.55	25.23	32.18	33.87
G-Transformer random initialized (Big)	23.29	25.48	22.22	23.82	32.04	33.77
G-Transformer random initialized (Large)	6.23	8.95	13.68	15.33	31.51	33.21

Table 7: G-Transformer on different model size.

(Bowman et al., 2016) with a probability of 0.3 on both the source and the target inputs.

Fine-tuned on Sentence-Level Transformer. We use the parameters of an existing sentence-level Transformer to initialize G-Transformer. We copy the parameters of the multi-head attention in Transformer to the group multi-head attention in G-Transformer, leaving the global multi-head attention and the gates randomly initialized. For the global multi-head attention and the gates, we use a learning rate of $5e-4$, while for other components, we use a smaller learning rate of $1e-4$. All the parameters are jointly trained using Adam optimizer with 4000 warmup steps. We apply a word-dropout with a probability of 0.1 on both the source and the target inputs.

Fine-tuned on mBART25. Similar as the fine-tuning on sentence-level Transformer, we also copy parameters from mBART25 (Liu et al., 2020) to G-Transformer, leaving the global multi-head attention and the gates randomly initialized. We following the settings (Liu et al., 2020) to train the model, using Adam optimizer with a learning rate of $3e-5$ and 2500 warmup steps. Here, we do not apply word-dropout, which empirically shows a damage to the performance.

C.2 Results on Model Size

As shown in Table 7, G-Transformer has a relatively stable performance on different model size. When increasing the model size from Base to Big, the performance drops for about 0.24, 1.33, and 0.14 s-BLEU points, respectively. Further to Large model, the performance drops further for about 17.06, 8.54, and 0.53 s-BLEU points, respectively. Although the performance drop on small dataset is large since overfitting on larger model, the drop on large dataset Europarl is relatively small, indicating a stable training on different model size.