# GRET: Global Representation Enhanced Transformer

**Rongxiang Weng,**[1,2] **Haoran Wei,**[2] **Shujian Huang,**[1*] **Heng Yu,**[2]
**Lidong Bing,**[2] **Weihua Luo,**[2] **Jiajun Chen**[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Machine Intelligence Technology Lab, Alibaba Group, Hangzhou, China
{wengrx, funan.whr, yuheng.yh, l.bing, weihua.luowh}@alibaba-inc.com, {huangsj, chenjj}@nju.edu.cn

**Abstract**

Transformer, based on the encoder-decoder framework, has achieved state-of-the-art performance on several natural language generation tasks. The encoder maps the words in the input sentence into a sequence of hidden states, which are then fed into the decoder to generate the output sentence. These hidden states usually correspond to the input words and focus on capturing local information. However, the global (sentence level) information is seldom explored, leaving room for the improvement of generation quality. In this paper, we propose a novel global representation enhanced Transformer (GRET) to explicitly model global representation in the Transformer network. Specifically, in the proposed model, an external state is generated for the global representation from the encoder. The global representation is then fused into the decoder during the decoding process to improve generation quality. We conduct experiments in two text generation tasks: machine translation and text summarization. Experimental results on four WMT machine translation tasks and LCSTS text summarization task demonstrate the effectiveness of the proposed approach on natural language generation[1].

## 1 Introduction

Transformer (Vaswani et al. 2017) has outperformed other methods on several neural language generation (NLG) tasks, like machine translation (Deng et al. 2018), text summarization (Chang, Huang, and Hsu 2018), *etc*. Generally, Transformer is based on the *encoder-decoder framework* which consists of two modules: an encoder network and a decoder network. The encoder encodes the input sentence into a sequence of hidden states, each of which corresponds to a specific word in the sentence. The decoder generates the output sentence word by word. At each decoding time-step, the decoder performs attentive read (Luong, Pham, and Manning 2015; Vaswani et al. 2017) to fetch the input hidden states and decides which word to generate.

As mentioned above, the decoding process of Transformer only relies on the representations contained in these hidden states. However, there is evidence showing that hidden states from the encoder in Transformer only contain local representations which focus on word level information. For example, previous work (Vaswani et al. 2017; Devlin et al. 2018; Song et al. 2020) showed that these hidden states pay much attention to the word-to-word mapping; and the weights of attention mechanism, determining which target word will be generated, is similar to word alignment.

As Frazier (1987) pointed, the global information, which is about the whole sentence in contrast to individual words, should be involved in the process of generating a sentence. Representation of such global information plays an import role in neural text generation tasks. In the recurrent neural network (RNN) based models (Bahdanau, Cho, and Bengio 2014), Chen (2018) showed on text summarization task that introducing representations about global information could improve quality and reduce repetition. Lin et al. (2018b) showed on machine translation that the structure of the translated sentence will be more correct when introducing global information. These previous work shows global information is useful in current neural network based model. However, different from RNN (Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Bahdanau, Cho, and Bengio 2014) or CNN (Gehring et al. 2016; 2017), although self-attention mechanism can achieve long distance dependence, there is no explicit mechanism in the Transformer to model the global representation of the whole sentence. Therefore, it is an appealing challenge to provide Transformer with such a kind of global representation.

In this paper, we divide this challenge into two issues that need to be addressed: 1). *how to model the global contextual information?* and 2). *how to use global information in the generation process?*, and propose a novel global representation enhanced Transformer (GRET) to solve them. For the first issue, we propose to generate the global representation based on local word level representations by two complementary methods in the encoding stage. On one hand, we adopt a modified *capsule network* (Sabour, Frosst, and Hinton 2017) to generate the global representation based the features extracted from local word level representations. The local representations are generally related to the word-to-word mapping, which may be redundant or noisy. Using them to

*Corresponding author

[1]Source code is available at: https://github.com/wengrx/GRET

generate the global representation directly without any filtering is inadvisable. Capsule network, which has a strong ability of feature extraction (Zhao et al. 2018), can help to extract more suitable features from local states. Comparing with other networks, like CNN (Krizhevsky, Sutskever, and Hinton 2012), it can see all local states at one time, and extract feature vectors after several times of deliberation.

On the other hand, we propose a *layer-wise recurrent structure* to further strengthen the global representation. Previous work shows the representations from each layer have different aspects of meaning (Peters et al. 2018; Dou et al. 2018), e.g. lower layer contains more syntactic information, while higher layer contains more semantic information. A complete global context should have different aspects of information. However, the global representation generated by the capsule network only obtain intra-layer information. The proposed layer-wise recurrent structure is a helpful supplement to combine inter-layer information by aggregating representations from all layers. These two methods can model global representation by fully utilizing different grained information from local representations.

For the second issue, we propose to use *a context gating mechanism* to dynamically control how much information from the global representation should be fused into the decoder at each step. In the generation process, every decoder states should obtain global contextual information before outputting words. And the demand from them for global information varies from word to word in the output sentence. The proposed gating mechanism could utilize the global representation effectively to improve generation quality by providing a customized representation for each state.

Experimental results on four WMT translation tasks, and LCSTS text summarization task show that our GRET model brings significant improvements over a strong baseline and several previous researches.

## 2 Approach

Our GRET model includes two steps: modeling the global representation in the encoding stage and incorporating it into the decoding process. We will describe our approach in this section based on Transformer (Vaswani et al. 2017).

### 2.1 Modeling Global Representation

In the encoding stage, we propose two methods for modeling the global representation at different granularity. We firstly use capsule network to extract features from local word level representations, and generate global representation based on these features. Then, a layer-wise recurrent structure is adopted subsequently to strengthen the global representation by aggregating the representations from all layers of the encoder. The first method focuses on utilizing word level information to generate a sentence level representation, while the second method focuses on combining different aspects of sentence level information to obtain a more complete global representation.

**Intra-layer Representation Generation** We propose to use *capsules with dynamic routing* to extract the specific and suitable features from the local representations for stronger global representation modeling, which is an effective and strong feature extraction method (Sabour, Frosst, and Hinton 2017; Zhang, Liu, and Song 2018)[2]. Features from hidden states of the encoder are summarized into several capsules, and the weights (routes) between hidden states and capsules are updated by dynamic routing algorithm iteratively.

**Algorithm 1** Dynamic Routing Algorithm

**procedure**: ROUTING($\mathbf{H}$, $r$)
**for** $i$ in input layer and $k$ in output layer **do**
  $b_{ki} \leftarrow 0$;
**end for**
**for** $r$ iterations **do**
  **for** $k$ in output layer **do**
    $\mathbf{c}_k \leftarrow \text{softmax}(\mathbf{b}_k)$;
  **end for**
  **for** $k$ in output layer **do**
    $\mathbf{u}_k \leftarrow q(\sum_i^I c_{ki}\mathbf{h}_i)$;
    ▷ $\mathbf{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_i, \cdots\}$
  **end for**
  **for** $i$ in input layer and $k$ in output layer **do**
    $b_{ki} \leftarrow b_{ki} + \mathbf{h}_i \cdot \mathbf{u}_k$;
  **end for**
**end for**
**return** $\mathbf{U}$; ▷ $\mathbf{U} = \{\mathbf{u}_1, \cdots, \mathbf{u}_k, \cdots\}$

Formally, given an encoder of the Transformer which has $M$ layers and an input sentence $\mathbf{x} = \{x_1, \cdots, x_i, \cdots, x_I\}$ which has $I$ words. The sequence of hidden states $\mathbf{H}^m = \{\mathbf{h}_1^m, \cdots, \mathbf{h}_i^m, \cdots, \mathbf{h}_I^m\}$ from the $m^{\text{th}}$ layer of the encoder is computed by

$$\mathbf{H}^m = \text{LN}(\text{SAN}(\mathbf{Q}_e^m, \mathbf{K}_e^{m-1}, \mathbf{V}_e^{m-1})), \quad (1)$$

where the $\mathbf{Q}_e^m$, $\mathbf{K}_e^{m-1}$ and $\mathbf{V}_e^{m-1}$ are query, key and value vectors which are same as $\mathbf{H}^{m-1}$, the hidden states from the $m-1^{\text{th}}$ layer. The $\text{LN}(\cdot)$ and $\text{SAN}(\cdot)$ are layer normalization function (Ba, Kiros, and Hinton 2016) and self-attention network (Vaswani et al. 2017), respectively. We omit the residual network here.

Then, the capsules $\mathbf{U}^m$ with size of $K$ are generated by $\mathbf{H}^m$. Specifically, the $k^{\text{th}}$ capsule $\mathbf{u}_k^m$ is computed by

$$\mathbf{u}_k^m = q(\sum_i^I c_{ki}\hat{\mathbf{h}}_i^m), \ c_{ki} \in \mathbf{c}_k, \quad (2)$$

$$\hat{\mathbf{h}}_i^m = \mathbf{W}_k\mathbf{h}_i^m, \quad (3)$$

where $q(\cdot)$ is non-linear squash function (Sabour, Frosst, and Hinton 2017):

$$\text{squash}(\mathbf{t}) = \frac{||\mathbf{t}||^2}{1+||\mathbf{t}||^2}\frac{\mathbf{t}}{||\mathbf{t}||}, \quad (4)$$

and $\mathbf{c}_k$ is computed by

$$\mathbf{c}_k = \text{softmax}(\mathbf{b}_k), \ \mathbf{b}_k \in \mathbf{B}, \quad (5)$$

[2]Other details of the Capsule Network are shown in Sabour, Frosst, and Hinton (2017) .

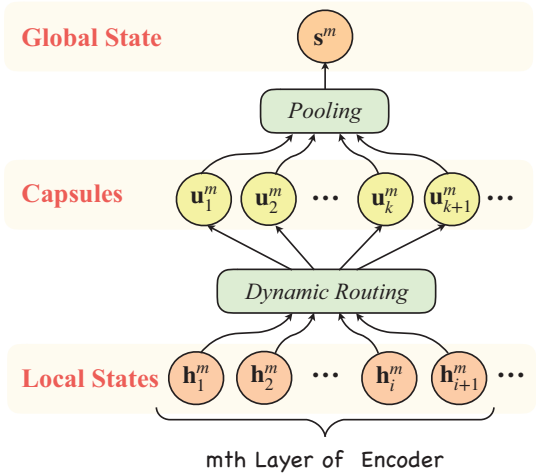


Figure 1: The overview of generating the global representation with capsule network.

where the matrix $\mathbf{B}$ is initialized by zero and whose row and column are $K$ and $I$, respectively. This matrix will be updated when all capsules are produced.

$$\mathbf{B} = \mathbf{B} + \mathbf{U}^{m\top} \cdot \mathbf{H}^{m}. \tag{6}$$

The algorithm is shown in Algorithm 1. The sequence of capsules $\mathbf{U}^m$ could be used to generate the global representation.

Different from the original capsules network which use a concatenation method to generate the final representation, we use an *attentive pooling method* to generate the global representation[3]. Formally, in the $m^{\text{th}}$ layer, the global representation is computed by

$$\mathbf{s}^m = \text{FFN}(\sum_{k=1}^{K} a_k \mathbf{u}_k^m), \tag{7}$$

$$a_k = \frac{\exp(\hat{\mathbf{s}}^m \cdot \mathbf{u}_k^m)}{\sum_{t=1}^{K} \exp(\hat{\mathbf{s}}^m \cdot \mathbf{u}_t^m)}, \tag{8}$$

where $\text{FFN}(\cdot)$ is a feed-forward network and the $\hat{\mathbf{s}}^m$ is computed by

$$\mathbf{s}^m = \text{FFN}(\frac{1}{K}\sum_{k=1}^{K} \mathbf{u}_k^m). \tag{9}$$

This *attentive* method can consider the different roles of the capsules and better model the global representation. The overview of the process of generating the global representation are shown in Figure 1.

**Inter-layer Representation Aggregation** Traditionally, the Transformer model only fed the last layer's hidden states

[3]Typically, the concatenation and other pooling methods, e.g. mean pooling, could be used here easily, but they will decrease 0.1∼0.2 BLEU in machine translation experiment.

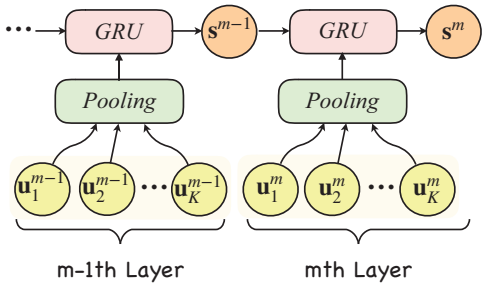


Figure 2: The overview of the layer-wise recurrent structure.

$\mathbf{H}^M$ as representations of input sentence to the decoder to generate the output sentence. Following this, we can feed the last layer's global representation $\mathbf{s}^M$ into the decoder directly. However, current global representation only contain the intra-layer information, the other layers' representations are ignored, which were shown to have different aspects of meaning in previous work (Wang et al. 2018b; Dou et al. 2018). Based on this intuition, we propose a *layer-wise recurrent structure* to aggregate the representations generated by employing the capsule network on all layers of the encoder to model a complete global representation.

The layer-wise recurrent structure aggregates each layer's intra global state by a gated recurrent unit (Cho et al. 2014, GRU) which could achieve different aspects of information from the previous layer's global representation. Formally, we adjust the computing method of $\mathbf{s}^m$ by

$$\mathbf{s}^m = \text{GRU}(\text{ATP}(\mathbf{U}^m), \mathbf{s}^{m-1}), \tag{10}$$

where the $\text{ATP}(\cdot)$ is the attentive pooling function computed by Eq 7-9. The GRU unit can control the information flow by forgetting useless information and capturing suitable information, which can aggregate previous layer's representations usefully. The layer-wise recurrent structure could achieve a more exquisite and complete representation. Moreover, the proposed structure only need one more step in the encoding stage which is not time-consuming. The overview of the aggregation structure is shown in Figure 2.

## 2.2 Incorporating into the Decoding Process

Before generating the output word, each decoder state should consider the global contextual information. We combine the global representation in decoding process with an additive operation to the last layer of the decoder guiding the states output true words. However, the demand for the global information of each target word is different. Thus, we propose *a context gating mechanism* which can provide specific information according to each decoder hidden state.

Specifically, given an decoder which has $N$ layers and the target sentence $\mathbf{y}$ which has $J$ words in the training stage, the hidden states $\mathbf{R}^N = \{\mathbf{r}_1^N, \cdots, \mathbf{r}_j^N, \cdots, \mathbf{r}_J^N\}$ from the $N^{\text{th}}$ layer of the decoder is computed by

$$\begin{aligned}\mathbf{R}^N = \text{LN}(&\text{SAN}(\mathbf{Q}_d^N, \mathbf{K}_d^{N-1}, \mathbf{V}_d^{N-1}) \\ &+ \text{SAN}(\mathbf{Q}_d^N, \mathbf{K}_e^M, \mathbf{V}_e^M)),\end{aligned} \tag{11}$$

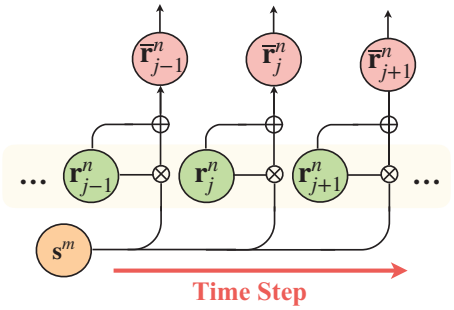


Figure 3: The context gating mechanism of fusing the global representation into decoding stage.

where $\mathbf{Q}_d^N$, $\mathbf{K}_d^{N-1}$ and $\mathbf{V}_d^{N-1}$ are hidden states $\mathbf{R}^{N-1}$ from $N-1^{\text{th}}$ layer. The $\mathbf{K}_e^M$ and $\mathbf{V}_e^M$ are same as $\mathbf{H}^M$. We omit the residual network here.

For each hidden state $\mathbf{r}_j^N$ from $\mathbf{R}^N$, the context gate is calculated by:

$$\mathbf{g}_j = \text{sigmoid}(\mathbf{r}_j^N, \mathbf{s}^M). \tag{12}$$

The new state, which contains the needed global information, is computed by:

$$\bar{\mathbf{r}}_j^N = \mathbf{r}_j^N + \mathbf{s}_j^M * \mathbf{g}. \tag{13}$$

Then, the output probability is calculated by the output layer's hidden state:

$$P(y_j|y_{<j}, \mathbf{x}) = \text{softmax}(\text{FFN}(\bar{\mathbf{r}}_j^N)). \tag{14}$$

This method enables each state to achieve it's customized global information. The overview is shown in Figure 3.

### 2.3 Training

The training process of our GRET model is same as the standard Transformer. The networks is optimized by maximizing the likelihood of the output sentence $\mathbf{y}$ given input sentence $\mathbf{x}$, denoted by $\mathcal{L}_{\text{trans}}$.

$$\mathcal{L}_{\text{trans}} = \frac{1}{J}\sum_{j=1}^{J} \log P(y_j|y_{<j}, \mathbf{x}), \tag{15}$$

where $P(y_j|y_{<j}, \mathbf{x})$ is defined in Equation 14.

## 3 Experiment

### 3.1 Implementation Detail

**Data-sets** We conduct experiments on machine translation and text summarization tasks. In machine translation, we employ our approach on four language pairs: Chinese to English (ZH→EN), English to German (EN→DE), German to English (DE→EN), and Romanian to English (RO→EN) [4]. In text summarization, we use LCSTS (Hu, Chen, and Zhu 2015) [5] to evaluate the proposed method. These data-sets are public and widely used in previous work, which will make other researchers replicate our work easily.

In machine translation, on the ZH→EN task, we use WMT17 as training set which consists of about 7.5M sentence pairs. We use `newsdev2017` as validation set and `newstest2017` as test set which have 2002 and 2001 sentence pairs, respectively. On the EN→DE and DE→EN tasks, we use WMT14 as training set which consists of about 4.5M sentence pairs. We use `newstest2013` as validation set and `newstest2014` as test set which have 2169 and 3000 sentence pairs, respectively. On the RO→EN task, we use WMT16 as training set which consists of about 0.6M sentence pairs. We use `newstest2015` as validation set and `newstest2016` as test set which has 3000 and 3002 sentence pairs, respectively.

In text summarization, following in Hu, Chen, and Zhu (2015) , we use PART I as training set which consists of 2M sentence pairs. We use the subsets of PART II and PART III scored from 3 to 5 as validation and test sets which consists of 8685 and 725 sentence pairs, respectively.

**Settings** In machine translation, we apply byte pair encoding (BPE) (Sennrich, Haddow, and Birch 2016) to all language pairs and limit the vocabulary size to 32K. In text summarization, we limit the vocabulary size to 3500 based on the character level. Out-of-vocabulary words and chars are replaced by the special token *UNK*.

For the Transformer, we set the dimension of the input and output of all layers as 512, and that of the feed-forward layer to 2048. We employ 8 parallel attention heads. The number of layers for the encoder and decoder are 6. Sentence pairs are batched together by approximate sentence length. Each batch has 50 sentence and the maximum length of a sentence is limited to 100. We set the value of dropout rate to 0.1. We use the Adam (Kingma and Ba 2014) to update the parameters, and the learning rate was varied under a warm-up strategy with 4000 steps (Vaswani et al. 2017). Other details are shown in Vaswani et al. (2017) . The number of capsules is set 32 and the default time of iteration is set 3. The training time of the Transformer is about 6 days on the DE→EN task. And the training time of the GRET model is about 12 hours when using the parameters of baseline as initialization.

After the training stage, we use beam search for heuristic decoding, and the beam size is set to 4. We measure translation quality with the NIST-BLEU (Papineni et al. 2002) and summarization quality with the ROUGE (Lin 2004).

### 3.2 Main Results

**Machine Translation** We employ the proposed GRET model on four machine translation tasks. All results are summarized in Table 1. For fair comparison, we reported several Transformer baselines with same settings reported by previous work (Vaswani et al. 2017; Hassan et al. 2018; Gu et al. 2018) and researches about enhancing local word level representations (Dou et al. 2018; Yang et al. 2018; Shaw, Uszkoreit, and Vaswani 2018; Yang et al. 2019).

The results on the WMT17 ZH→EN task are shown in the second column of Table 1. The improvement of our GRET

[4] http://www.statmt.org/wmt17/translation-task.html
[5] http://icrc.hitsz.edu.cn/Article/show/139.html

| Model | ZH→EN | EN→DE | DE→EN | RO→EN |
|---|---|---|---|---|
| Transformer* (Vaswani et al. 2017) | – | 27.3 | – | – |
| Transformer* (Hassan et al. 2018) | 24.13 | – | – | – |
| Transformer* (Gu et al. 2018) | – | 27.02 | – | 31.76 |
| DeepRepre* (Dou et al. 2018) | 24.76 | 28.78 | – | – |
| Localness* (Yang et al. 2018) | 24.96 | 28.54 | – | – |
| RelPos* (Shaw, Uszkoreit, and Vaswani 2018) | 24.53 | 27.94 | – | – |
| Context-aware* (Yang et al. 2019) | 24.67 | 28.26 | – | – |
| GDR* (Zheng et al. 2019) | – | 28.10 | – | – |
| Transformer | 24.31 | 27.20 | 32.34 | 32.17 |
| GRET | 25.53‡ | 28.46† | 33.79‡ | 33.06‡ |

Table 1: The comparison of our GRET , Transformer baseline and related work on the WMT17 Chinese to English (ZH→EN), WMT14 English to German (EN→DE) and German to English (DE→EN), and WMT16 Romania to English (RO→EN) tasks (* indicates the results came from their paper, †/‡ indicate significantly better than the baseline ($p < 0.05/0.01$)).

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L |
|---|---|---|---|
| RNNSearch* (Hu, Chen, and Zhu 2015) | 30.79 | – | – |
| CopyNet* (Gu et al. 2016) | 34.4 | 21.6 | 31.3 |
| MRT* (Ayana, Liu, and Sun 2016) | 37.87 | 25.43 | 35.33 |
| AC-ABS* (Li, Bing, and Lam 2018) | 37.51 | 24.68 | 35.02 |
| CGU* (Lin et al. 2018a) | 39.4 | 26.9 | 36.5 |
| Transformer* (Chang, Huang, and Hsu 2018) | 42.35 | 29.38 | 39.23 |
| Transformer | 43.14 | 29.26 | 39.72 |
| GRET | 44.77 | 30.96 | 41.21 |

Table 2: The comparison of our GRET , Transformer baseline and related work on the LCSTS text summarization task (* indicates the results came from their paper).

model could be up to 1.22 based on a strong baseline system, which outperforms all previous work we reported. To our best knowledge, our approach attains the state-of-the-art in relevant researches.

Then, the results on the WMT14 EN→DE and DE→EN tasks, which is the most widely used data-set recently, are shown in the third and fourth columns. The GRET model could attain 28.46 BLEU (+1.26) on the EN→DE and 33.79 BLEU (+1.45) on the DE→EN, which are competitive results compared with previous studies.

To verify the generality of our approach, we also experiment it on low resource language pair of the WMT16 RO→EN task. Results are shown in the last column. The improvement of the GRET is 0.89 BLEU, which is a material improvement in low resource language pair. And it shows that proposed methods could improve translation quality in low resource scenario.

Experimental results on four machine translation tasks show that modeling global representation in the current Transformer network is a general approach, which is not limited by the language or size of training data, for improving translation quality.

**Text Summarization** Besides machine translation, we also employ proposed methods in text summarization, a monolingual generation task, which is an important and typical task in natural language generation.

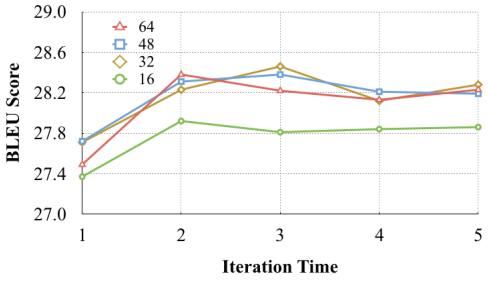


Figure 4: The comparison of the GTR with different number of capsules at different iteration times on the EN→DE task.

The results are shown in Table 2, we also reports several popular methods in this data-set as a comparison. Our approach achieves considerable improvements in ROUGE-1/2/L (+1.63/+1.70/+1.49) and outperforms other work with same settings. The improvement on text summarization is even more than machine translation. Compared with machine translation, text summarization focuses more on extracting suitable information from the input sentence, which is an advantage of the GRET model.

Experiments on the two tasks also show that our approach could work on different types of language generation task and may improve the performance of other text generation tasks.

| Model | Capsule | Aggregate | Gate | #Param | Inference | BLEU | Δ |
|---|---|---|---|---|---|---|---|
| Transformer | – | – | – | 61.9M | 1.00x | 27.20 | – |
| *Our Approach* | | | | 61.9M | 0.99x | 27.39 | +0.19 |
| | ✓ | | | 63.6M | 0.87x | 28.02 | +0.82 |
| | ✓ | ✓ | | 68.1M | 0.82x | 28.32 | +1.02 |
| | ✓ | | ✓ | 63.6M | 0.86x | 28.23 | +1.03 |
| | | ✓ | | 66.6M | 0.95x | 27.81 | +0.61 |
| | | ✓ | ✓ | 66.8M | 0.93x | 27.76 | +0.56 |
| | | | ✓ | 62.1M | 0.98x | 27.53 | +0.33 |
| | ✓ | ✓ | ✓ | 68.3M | 0.81x | 28.46 | +1.26 |

Table 3: Ablation study on the WMT14 English to German (EN→DE) machine translation task.

| Model | #Param | Inference | BLEU |
|---|---|---|---|
| Transformer-Base | 61.9M | 1.00x | 27.20 |
| GTR-Base | 68.3M | 0.81x | 28.46 |
| Transformer-Big | 249M | 0.59x | 28.47 |
| GRET-Big | 273M | 0.56x | 29.33 |

Table 4: The comparison of GRET and Transformer with *big* setting (Vaswani et al. 2017) on the EN→DE task.

| Model | Precision | | |
|---|---|---|---|
| | Top-200 | Top-500 | Top-1000 |
| *Last* | 43% | 52% | 64% |
| *Average* | 49% | 57% | 69% |
| GRET | 63% | 74% | 81% |

Table 5: The precision from the bag-of-words predictor based on GRET , last encoder state (*Last*) and averaging all local states (*Average*) on the EN→DE task.

### 3.3 Ablation Study

To further show the effectiveness and consumption of each module in our GRET model, we make ablation study in this section. Specifically, we investigate how the *capsule network*, *aggregate structure* and *gating mechanism* affect the performance of the global representation.

The results are shown in Table 3. Specifically, without the capsule network, the performance decreases 0.7 BLEU , which means extracting features from local representations iteratively could reduce redundant information and noisy. This step determines the quality of global representation directly. Then, aggregating multi-layers' representations attains 0.61 BLEU improvement. The different aspects of information from each layer is an excellent complement for generating the global representation. Without the gating mechanism, the performance decreases 0.24 BLEU score which shows the context gating mechanism is important to control the proportion of using the global representation in each decoding step. While the GRET model will take more time, we think it is worthwhile to improve generation quality by reducing a bit of efficiency in most scenario.

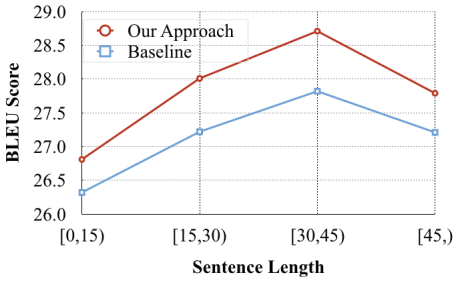


Figure 5: The comparison of the GTR with different number of capsules at different iteration times on the EN→DE task.

### 3.4 Effectiveness on Different Model Settings

We also experiment the GRET model with *big* setting on the EN→DE task. The *big* model is far larger than above *base* model and get the state-of-the-art performance in previous work (Vaswani et al. 2017).

The results are shown in Table 4, Transformer-Big outperforms Transformer-Base, while the GRET-Big improves 0.86 BLEU score comparing with the Transformer-Big. It is worth to mention that our model with base setting could achieve a similar performance to the Transformer-Big, which reduces parameters by almost 75% (68.3M VS. 249M) and inference time by almost 27% (0.81x VS. 0.56x).

### 3.5 Analysis of the Capsule

The number of capsules and the iteration time from dynamic routing algorithm may affect the performance of the proposed model. We evaluate the GRET model with different number of capsules at different iteration times on the EN→DE task. The results are shown in Figure 4.

We can get two empirical conclusions in this experiment. First, the first three iterations can significantly improve the performance, while the results of more iterations (4 and 5) tend to stabilize. Second, the increase of capsule number (48 and 64) doesn't get a further gain. We think the reason is that most sentences are shorter than 50, just the suitable amount of capsules can extract enough features.

### 3.6 Probing Experiment

What does the global representation learn is an interesting question. Following Weng et al. (2017) , we do a probing

| | |
|---|---|
| Input | 出台或者即将出台楼市调控政策的二线城市不止苏州，还包括合肥、南京等城市。 |
| Reference | In addition to Suzhou, other second-tier cities including Hefei and Nanjing will also introduce property market regulations and control policies. |
| Transformer | The second-tier cities, including Hefei and Nanjing, are not only Suzhou, *but also the cities of Hefei.* |
| GRET | The second-tier cities, not only Suzhou, but also Hefei and Nanjing **will also introduce property market regulations and control policies.** |
| Input | 莫斯科旅游警察的人员招录、警务模式、装备配备给我们带来了很多启发。 |
| Reference | The recruiting, police mode and equipment of Moscow tourism police officers have inspired us a lot. |
| Transformer | We have a lot of inspiration from the *Moscow Travel Police*, the police *model*, and the equipment. |
| GRET | We have a lot of inspiration by the **Moscow Travel Police's recruiting, police mode**, and equipment. |

Figure 6: Translation cases from Transformer and our GRET model on the ZH→EN task.

experiment here. We train a *bag-of-words predictor* by maximizing $P(\mathbf{y}_{bow}|\mathbf{s}^M)$, where $\mathbf{y}_{bow}$ is an unordered set containing all words in the output sentence. The structure of the predictor is a simple feed-forward network which maps the global state to the target word embedding matrix.

Then, we compare the precision of target words in the top-*K* words which are chosen through the predicted probability distribution[6]. The results are shown in Table 5, the global state from GRET can get higher precision in all conditions, which shows that the proposed method can obtain more information about the output sentence and partial answers why the GRET model could improve the generation quality.

### 3.7 Analysis of Sentence Length

To see the effectiveness of the global representation, we group the EN→DE test set by the length of the input sentences to re-evaluate the models. The set is divided into 4 sets. Figure 5 shows the results. We find that our model outperforms the baseline in all categories, especially in the longer sentences, which shows that fusing the global representation may help the generation of longer sentences by providing more complete information.

### 3.8 Case Study

We show two real-cases on the ZH→EN task to see the difference between the baseline and our model. These cases are shown in Figure 6. The "Source" indicates the source sentence and the "Reference" indicates the human translation. The **bold font** indicates improvements of our model; and the *italic font* indicates translation errors.

Each output from GRET is decided by previous state and the global representation. So, it can avoid some common translation errors like over/under translation, caused by the strong language model of the decoder which ignores some translation information. For example, the over translation of "*the cities of Hefei*" in case 1 is corrected by the GRET model. Furthermore, providing global information can avoid current state only focuses on the word-to-word mapping. In case 2, the vanilla Transformer translates the "Moscow Travel Police" according to the source input "mosike lvyou jingcha"', but omits the words "de renyuan zhaolu", which leads it fails to translate the target word "*recruiting*".

[6]Experiment details are shown in Weng et al. (2017) .

## 4 Related Work

Several work also try to generate global representation. In machine translation, Lin et al. (2018b) propose a deconvolutional method to obtain global information to guide the translation process in RNN-based model. However, the limitation of CNN can not model the global information well and there methods can not employ on the Transformer. In text summarization, Chen (2018) also propose to incorporate global information in RNN-based model to reduce repetition. They use an additional RNN to model the global representation, which is time-consuming and can not get the long-dependence relationship, which hinders the effectiveness of the global representation.

Zhang, Liu, and Song (2018) propose a sentence-state LSTM for text representation. Our method shows an alternative way of obtaining the representation, on the implementation of the Transformer.

Many previous researches notice the importance of the representations generated by the encoder and focus on making full use of them. Wang et al. (2018a) propose to use Capsule network to generate hidden states directly, which inspire us to use capsules with dynamic routing algorithm to extract specific and suitable features from these hidden states. Wang et al.; Dou et al. (2018b; 2018) propose to utilize the hidden states from multiple layers which contain different aspects of information to model more complete representations, which inspires us to use the states in multiple layers to enhance the global representation.

## 5 Conclusion

In this paper, we address the problem that Transformer doesn't model global contextual information which will decrease generation quality. Then, we propose a novel GRET model to generate an external state by the encoder containing global information and fuse it into the decoder dynamically. Our approach solves the both issues of how to model and how to use the global contextual information. We compare the proposed GRET with the state-of-the-art Transformer model. Experimental results on four translation tasks and one text summarization task demonstrate the effectiveness of the approach. In the future, we will do more analysis and combine it with the methods about enhancing local representations to further improve generation performance.

## Acknowledgements

We would like to thank the reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by the National Key R&D Program of China (No. 2019QY1806), the National Science Foundation of China (No. 61672277), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074).

## References

Ayana, S. S.; Liu, Z.; and Sun, M. 2016. Neural headline generation with minimum risk training. *arXiv preprint arXiv:1604.01904*.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.

Chang, C.-T.; Huang, C.-C.; and Hsu, J. Y.-j. 2018. A hybrid word-character model for abstractive summarization. *CoRR*.

Chen, G. 2018. Chinese short text summary generation model combining global and local information. In *NCCE*.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Deng, Y.; Cheng, S.; Lu, J.; Song, K.; Wang, J.; Wu, S.; Yao, L.; Zhang, G.; Zhang, H.; Zhang, P.; et al. 2018. Alibaba's neural machine translation systems for wmt18. In *Conference on Machine Translation: Shared Task Papers*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

Dou, Z.-Y.; Tu, Z.; Wang, X.; Shi, S.; and Zhang, T. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Frazier, L. 1987. Sentence processing: A tutorial review.

Gehring, J.; Auli, M.; Grangier, D.; and Dauphin, Y. N. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*.

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; and Socher, R. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Hassan, H.; Aue, A.; Chen, C.; Chowdhary, V.; Clark, J.; Federmann, C.; Huang, X.; Junczys-Dowmunt, M.; Lewis, W.; Li, M.; et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Hu, B.; Chen, Q.; and Zhu, F. 2015. Lcsts: A large scale chinese short text summarization dataset. In *EMNLP*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Li, P.; Bing, L.; and Lam, W. 2018. Actor-critic based training framework for abstractive summarization. *arXiv preprint arXiv:1803.11070*.

Lin, J.; Sun, X.; Ma, S.; and Su, Q. 2018a. Global encoding for abstractive summarization. In *ACL*.

Lin, J.; Sun, X.; Ren, X.; Ma, S.; Su, J.; and Su, Q. 2018b. Deconvolution-based global decoding for neural machine translation. In *ACL*.

Lin, C.-Y. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL*.

Luong, M.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic routing between capsules. *CoRR*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-attention with relative position representations. In *NAACL*.

Song, K.; Wang, K.; Yu, H.; Zhang, Y.; Huang, Z.; Luo, W.; Duan, X.; and Zhang, M. 2020. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In *AAAI*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wang, M.; Xie, J.; Tan, Z.; Su, J.; et al. 2018a. Towards linear time neural machine translation with capsule networks. *arXiv*.

Wang, Q.; Li, F.; Xiao, T.; Li, Y.; Li, Y.; and Zhu, J. 2018b. Multi-layer representation fusion for neural machine translation. In *COLING*.

Weng, R.; Huang, S.; Zheng, Z.; Dai, X.; and Chen, J. 2017. Neural machine translation with word predictions. In *EMNLP*.

Yang, B.; Tu, Z.; Wong, D. F.; Meng, F.; Chao, L. S.; and Zhang, T. 2018. Modeling localness for self-attention networks. In *EMNLP*.

Yang, B.; Li, J.; Wong, D. F.; Chao, L. S.; Wang, X.; and Tu, Z. 2019. Context-aware self-attention networks. In *AAAI*.

Zhang, Y.; Liu, Q.; and Song, L. 2018. Sentence-state lstm for text representation. In *ACL*.

Zhao, W.; Ye, J.; Yang, M.; Lei, Z.; Zhang, S.; and Zhao, Z. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

Zheng, Z.; Huang, S.; Tu, Z.; DAI, X.-Y.; and CHEN, J. 2019. Dynamic past and future for neural machine translation. In *EMNLP-IJCNLP*.