Towards More Diverse Input Representation for Neural Machine Translation

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Tiejun Zhao, Muyun Yang, and Hai Zhao

***Abstract*—Source input information plays a very important role in the Transformer-based translation system. In practice, word embedding and positional embedding of each word are added as the input representation. Then self-attention networks are used to encode the global dependencies in the input representation to generate a source representation. However, this processing on the source representation only adopts a single source feature and excludes richer and more diverse features such as recurrence features, local features, and syntactic features, which results in tedious representation and thereby hinders the further translation performance improvement. In this paper, we introduce a simple and efficient method to encode more diverse source features into the input representation simultaneously, and thereby learning an effective source representation by self-attention networks. In particular, the proposed grouped strategy is only applied to the input representation layer, to keep the diversity of translation information and the efficiency of the self-attention networks at the same time. Experimental results show that our approach improves the translation performance over the state-of-the-art baselines of Transformer in regard to WMT14 English-to-German and NIST Chinese-to-English machine translation tasks.**

***Index Terms*—Neural machine translation, source features, diverse input representation.**

I. Introduction

RECENTLY, the Transformer-based translation system [1], which solely relies on self-attention networks (**SANs**) to learn a source representation, has attracted increasing attention in the machine translation community. To encode the source input information, the word vectors for the source sentence are added with the corresponding position vectors in turn to form an input representation. The input representation is then fed to the SANs to parallel perform (multi-head) and stack (multi-layer) self-attentive functions (i.e., dot-product) to learn the source representation. In particular, the multi-head self-attention mechanism is adopted to diversify the source representation from different representation subspaces, achieving state-of-the-art translation performance in several language pairs [1], [2].

Although the multi-head self-attention encourages diversity among multiple attention heads, these head representations computed by the dot-product function only attend to a single global dependency feature in their respective vector spaces and do not take into account other source features (i.e., recurrence features, local features, syntax features, etc). Recent studies reveal that the lack of these diverse source features hinders further improvement of the translation capacity of Transformer [3]–[10]. For example, source recurrence features are captured by additional multiple layers RNN to improve the performance of Transformer model [3], [5], [8]. In addition, source local features are learned, respectively, by a Gaussian bias [6] and convolutional self-attention networks [9] to enhance the Transformer-based translation systems. More recently, source-side linearized constituency tree knowledge is incorporated into the Transformer-based NMT by a multi-task framework [10]. Although these studies successfully improved the translation performance, they tend to focus on the capture of a single additional source feature by a specific-feature model architecture and did not consider whether more diverse features can work together to improve the performance of the Transformer-based NMT model.

Inspired by the finding that not all global head representations are necessary for predicting translation [11], [12], we introduce a simple and efficient multi-group strategy to fuse more diverse source features into the input representation, and thus learn an effective source representation using SANs. To this end, we first split the word vector into different groups, each of which is used to capture a source feature. Consider four groups as an example. The first group uses the original addition of the positional embeddings and the word vectors to obtain its input representation, thereby learning the global features using SANs; the second group focuses on encoding the local features using the convolutional neural network [13] to its input representation; the third group focuses on encoding the recurrence features under an RNN [14] to its input representation; and in the

Manuscript received November 24, 2019; revised March 17, 2020; accepted May 10, 2020. Date of publication May 20, 2020; date of current version June 5, 2020. The work of R. Wang was supported in part by JSPS grant-in-aid for early-career scientists (19K20354): "Unsupervised Neural Machine Translation in Universal Scenarios" and in part by NICT tenure-track researcher startup fund "Toward Intelligent Machine Translation." The work of M. Utiyama was supported by JSPS KAKENHI under Grant 19H05660. The work of T. Zhao and M. Yang were supported in part by the National Key Research and Development Program of China (No. 2018YFC0830700) and in part by the National Natural Science Foundation of China (No. 61806075). The work of H. Zhao was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0304100 and in part by the Key Projects of National Natural Science Foundation of China under Grants U1836222 and 61733011. The associate editor coordinating the review of this manuscript and approving it for publication was Taro Watanabe. *(Corresponding author: Rui Wang.)*

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita are with the Advanced Speech Translation Research and Development Promotion Center, National Institute of Information and Communications Technology, Kyoto 619-0289, Japan (e-mail: khchen@nict.go.jp; wangrui@nict.go.jp; mutiyama@nict.go.jp; eiichiro.sumita@nict.go.jp).

Tiejun Zhao and Muyun Yang are with the Machine Intelligence and Translation Laboratory, School of Computer Science of Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: tjzhao@hit.edu.cn; yangmuyun@hit.edu.cn).

Hai Zhao is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhaohai@cs.sjtu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2020.2996077

fourth group, we add part-of-speech (**POS**) tag vectors of words to the existing input representation pattern to capture syntax features of the source sentence. The four input representations are then concatenated to provide the input of the encoder to learn the source representation with more diverse features for the decoder. Experiments on the WMT14 English-to-German and NIST Chinese-to-English translation tasks show that the proposed models can improve the performance of NMT over the state-of-the-art Transformer baseline systems.

This paper primarily makes the following contributions:

- This work enables the Transformer model to capture more diverse source features simultaneously instead of a single additional source feature in a unified grouped strategy.
- The proposed grouped strategy is only applied to the input representation layer of the SAN-based encoder, which keeps the diversity of translation information and the efficiency of SANs at the same time.
- Our NMT models can significantly improve the translation performance with slight additional training and decoding costs on the two widely used translation tasks.
- We showed the effect of each source feature through quantitative analysis and verified that diverse source features contributed to the improvement of translation performance.

II. Background

A. Input Representation

In the Transformer-based network architecture [1], given a sequence of word vectors $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_J\}$ for the source sentence, the positional embedding of each word is computed initially based on its position:

$$\begin{aligned} \boldsymbol{pe}_{(j,2i)} &= \sin(j/10000^{2i/d_{model}}), \\ \boldsymbol{pe}_{(j,2i+1)} &= \cos(j/10000^{2i/d_{model}}), \end{aligned} \tag{1}$$

where j is the word's numerical position index in the sentence and i is the dimension of the position index. The word vector $\boldsymbol{x}_j$ is then added with $\boldsymbol{pe}_j$ to yield a combined embedding $\boldsymbol{v}_j$:

$$\boldsymbol{v}_j = \boldsymbol{x}_j + \boldsymbol{pe}_j. \tag{2}$$

As a result, there is an input representation $H = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_J\}$. In summary, the above process is formally denoted by the function Func_{pe} that generates the input representation:

$$\boldsymbol{H} = \text{Func}_{pe}(\boldsymbol{X}). \tag{3}$$

In Transformer, *H* serves as the input of the SAN-based encoder to learn the source representation.

B. Multi-Head Self-Attention

In Transformer [1], multi-head self-attention is often used to learn the source representation from multiple individual attention functions instead of a single attention function. Specifically, the input representation $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_J\}$ is packed into a query matrix $\boldsymbol{Q}$, a key matrix $\boldsymbol{K}$, and a value matrix $\boldsymbol{V}$. The multi-head self-attention is performed over $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$:

$$\begin{aligned} \text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) &= \text{Concat}(\boldsymbol{O}_1 : \boldsymbol{O}_2 : \cdots : \boldsymbol{O}_H)\boldsymbol{W}^O, \\ \boldsymbol{O}_h &= \text{softmax}\left(\frac{\boldsymbol{Q}_h {\boldsymbol{K}_h}^T}{\sqrt{d_{model}}}\right)\boldsymbol{V}_h, \\ \boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h &= \boldsymbol{Q}\boldsymbol{W}_h^Q, \boldsymbol{K}\boldsymbol{W}_h^K, \boldsymbol{V}\boldsymbol{W}_h^V, \end{aligned} \tag{4}$$

where $\boldsymbol{Q}_h$, $\boldsymbol{K}_h$, and $\boldsymbol{K}_h$ indicate the query, key, and value matrices of the h-th head. $\{\boldsymbol{W}_h^Q, \boldsymbol{W}_h^K, \boldsymbol{W}_h^V\} \in \mathbb{R}^{d_{model} \times d_k}$ represent parameter matrices, d_{model} and d_k denote the dimensions of the model and the head. Therefore, H self-attention functions are applied in parallel to produce the output states.

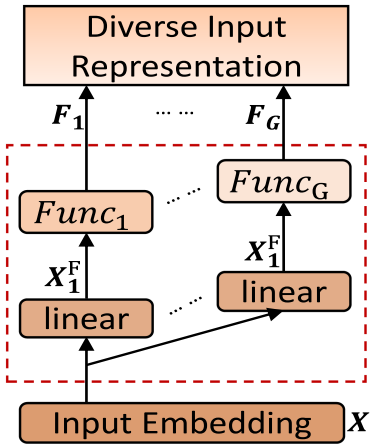


Fig. 1. Proposed multi-group strategy for learning the diverse input representation.

III. Diverse Input Representation

Next, we introduce a novel multi-group strategy to simultaneously encode more diverse translation features into the input representation (see Fig. 1) in the different ways. Specifically, we divide each word vector into multiple groups, each of which is used to model a kind of source feature. These diverse feature representations are together the input to the SAN-based encoder for learning the source representation.

Formally, given a sequence of word vectors $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_J\}$ for source sentence, we divide *X* into several groups of diverse features with a different, learnable linear projections, namely,

$$\boldsymbol{X}_g^F = \boldsymbol{X}\boldsymbol{W}_g^F, \tag{5}$$

where $\boldsymbol{X}_g^F$ is the representation of the g-th group for learning a specific feature F. $\boldsymbol{W}_g^F \in \mathbb{R}^{d_{model} \times d_g}$ denotes the parameter matrix. d_{model}, which is the dimension of the Transformer model, is equal to $\sum_1^G d_g$. As a result, there are G groups $\{\boldsymbol{X}_1^F, \boldsymbol{X}_2^F, \ldots, \boldsymbol{X}_G^F\}$.

We then apply a specific method of learning its input representation $\boldsymbol{F}_g$ over each group $\boldsymbol{X}_g^F$,

$$\boldsymbol{F}_g = \text{Func}_g(\boldsymbol{X}_g^F), \tag{6}$$

where Func_g (i.e., Func_{pe} is introduced in Section II-A) is a function for learning the input representation of the specific feature F. Finally, these feature representations $\{\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_G\}$ are concatenated to produce an input representation $\boldsymbol{H}$ with diverse source features,

$$\boldsymbol{F} = \text{Concat}(\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_G). \tag{7}$$

Naturally, the proposed strategy allows more diverse source features to be encoded into the input representation. Therefore, in addition to the existing global feature, we shall explore three additional source features, namely, recurrence features, local features, and syntactic (POS-tagger) features, to verify the effectiveness of our method.

A. Recurrence Feature Representation

For recurrence features, we apply a bidirectional RNN [15] to learn its recurrent input representation for a source sentence. Formally, given one group $\boldsymbol{X}_g^R = \{\boldsymbol{x}_1^R, \boldsymbol{x}_2^R, \ldots, \boldsymbol{x}_J^R\}$, $\boldsymbol{x}_j^R \in \mathbb{R}^{d_r}$, the forward RNN $f_{\overrightarrow{\text{RNN}}}$ and the backward RNN $f_{\overleftarrow{\text{RNN}}}$ are used to learn a forward annotation vector and a backward annotation vector, respectively,

$$\begin{aligned} \overrightarrow{\boldsymbol{r}}_j &= f_{\overrightarrow{\text{RNN}}}(\boldsymbol{x}_j^g, \overrightarrow{\boldsymbol{r}}_{j-1}), \\ \overleftarrow{\boldsymbol{r}}_j &= f_{\overleftarrow{\text{RNN}}}(\boldsymbol{x}_j^g, \overleftarrow{\boldsymbol{r}}_{j+1}). \end{aligned} \tag{8}$$

Both $\overrightarrow{\boldsymbol{r}}_j$ and $\overleftarrow{\boldsymbol{r}}_j$ are concatenated as a hidden state $[\overrightarrow{\boldsymbol{r}}_j; \overleftarrow{\boldsymbol{r}}_j]$, and we feed it into a linear projection layer to map the $2{\cdot}d_r$-dimensions $[\overrightarrow{\boldsymbol{r}}_j; \overleftarrow{\boldsymbol{r}}_j]$ into a d_r-dimensions vector $\boldsymbol{r}_j$. As a result, $\boldsymbol{R} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_J\}$ is the recurrence feature representations for the source sentence. In summary, the above process is formally denoted by the following function Func_{Rec}:

$$\boldsymbol{R} = \text{Func}_{Rec}(\boldsymbol{X}_g^R). \tag{9}$$

B. Local Feature Representation

For local features, we use a one-dimensional convolution [13] to perform a nonlinear transformation over the given group $\boldsymbol{X}_g^L = \{\boldsymbol{x}_1^L, \boldsymbol{x}_2^L, \ldots, \boldsymbol{x}_J^L\}$, $\boldsymbol{x}_j^L \in \mathbb{R}^{d_l}$. First, the convolution operations are performed,

$$\boldsymbol{l}_j = \phi([\boldsymbol{x}_{j-m/2}^L, \ldots, \boldsymbol{x}_{j+m/2}^L]\boldsymbol{W}_j) + \boldsymbol{x}_j^L, \tag{10}$$

where ϕ is the activation function (i.e., ReLU), $\boldsymbol{W}_j \in \mathbb{R}^{m \times d_l}$ is a weight matrix, and m is the width of the convolution kernel (m is 5 in our experiments). After the convolution kernel has traversed the $\boldsymbol{X}_g^L$ (the stride is one), there is an input representation with local features $\boldsymbol{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_J\}$, $\boldsymbol{l}_j \in \mathbb{R}^{d_l}$. This processing sequence is formally expressed as a following function Func_{Loc}, satisfying

$$\boldsymbol{L} = \text{Func}_{Loc}(\boldsymbol{X}_g^L). \tag{11}$$

In this paper, we only use one layer of convolution to learn the representation of local features.

C. Syntactic Feature Representation

Inspired by the linguistic input features [16], we consider for simplicity the POS-tag syntactic feature. Given one group $\boldsymbol{X}_g^S = \{\boldsymbol{x}_1^S, \boldsymbol{x}_2^S, \ldots, \boldsymbol{x}_J^S\}, \boldsymbol{x}_j^S \in \mathbb{R}^{d_s}$, each POS-tag is initialized to a vector $\boldsymbol{p}_j$ with the same dimensions as $\boldsymbol{x}_j^S$. We then append the POS-tag vector and the positional embedding to $\boldsymbol{x}_j^S$,

$$\boldsymbol{vp}_j = \mathbf{x}_j^S + \boldsymbol{pe}_j + \boldsymbol{p}_j. \tag{12}$$

As a result, there is an input representation with the POS-tag feature information $\boldsymbol{P} = \{\boldsymbol{vp}_1, \boldsymbol{vp}_2, \ldots, \boldsymbol{vp}_J\}$. This processing sequence is formally denoted by the following function Func_{Syn}, satisfying

$$\boldsymbol{P} = \text{Func}_{Syn}(\boldsymbol{X}_g^S). \tag{13}$$

Note that we only take the POS-tag feature as an example in this paper and the POS-tags of the source sentence are gained through the Stanford CoreNLP toolkit [17]. Generally, we can also capture other syntactic feature information in the source sentence, such as the linearized parse tree [18], the long-distance dependence [19], and the neural syntactic distance [20].

IV. Transformer Model With Diverse Input Representation

Based on the proposed diverse multi-group strategy, we first divide the source input $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_J\}$ into two groups $\boldsymbol{X}_1^H = \{\boldsymbol{x}_1^H, \boldsymbol{x}_2^H, \ldots, \boldsymbol{x}_J^H\}$ and $\boldsymbol{X}_2^F = \{\boldsymbol{x}_1^F, \boldsymbol{x}_2^F, \ldots, \boldsymbol{x}_J^F\}$, which are used to learn the existing global features and other (diverse) features, respectively. In particular, the dimension of each group's vector is the time of the head in the multi-head self-attention. For example, if d_{model} is 512 and H is 8, the dimension of each head is 64, and hence $\boldsymbol{x}_j^H \in \mathbb{R}^{2*64}$ and $\boldsymbol{x}_j^F \in \mathbb{R}^{6*64}$. This ensures that the introduction of a new feature does not require any change to the original multi-head self-attention.

Initially, the feature function Func_{pe} is applied to the first group $\boldsymbol{X}_1^H$ for learning the input representation $\boldsymbol{H}$ of global features. For other source features, there are three methods of learning the source representation for Transformer NMT:

Rec_Diverse: The recurrence feature representation $\boldsymbol{R}$ is learned over the $\boldsymbol{X}_2^F$ using Eq.(9) (Section III-A). $\boldsymbol{R}$ and $\boldsymbol{H}$ are then concatenated as the input of SAN-based encoder to learn the final source representation.

Loc_Diverse: The local feature representation $\boldsymbol{L}$ is learned over the $\boldsymbol{X}_2^F$ using Eq.(10) (Section III-B. $\boldsymbol{L}$ and $\boldsymbol{H}$ are then concatenated with as the input of SAN-based encoder to learn the final source representation.

Syn_Diverse: The POS-tag feature representation $\boldsymbol{P}$ is learned over the $\boldsymbol{X}_2^F$ using Eq.(13) (Section III-C). $\boldsymbol{P}$ and $\boldsymbol{H}$ are then concatenated as the input of the SAN-based encoder to learn the final source representation.

To enhance the translation performance further, we divide the source input $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_J\}$ into four groups: $\boldsymbol{X}_1^H = \{\boldsymbol{x}_1^H, \boldsymbol{x}_2^H, \ldots, \boldsymbol{x}_J^H\}$, $\boldsymbol{X}_2^R = \{\boldsymbol{x}_1^R, \boldsymbol{x}_2^R, \ldots, \boldsymbol{x}_J^R\}$, $\boldsymbol{X}_3^L = \{\boldsymbol{x}_1^L, \boldsymbol{x}_2^L, \ldots, \boldsymbol{x}_J^L\}$, and $\boldsymbol{X}_4^P = \{\boldsymbol{x}_1^P, \boldsymbol{x}_2^P, \ldots, \boldsymbol{x}_J^P\}$, which are used to learn global features, recurrence features, local features, and the POS-tag features in turn, called **Fusing_Diverse** (see Fig. 2). In our experiments, their dimensions[1] are $\boldsymbol{x}_j^H \in \mathbb{R}^{2*64}$, $\boldsymbol{x}_j^R \in$

[1]A detailed analysis regarding the dimensional ratios between the various features is presented in Section V-C.

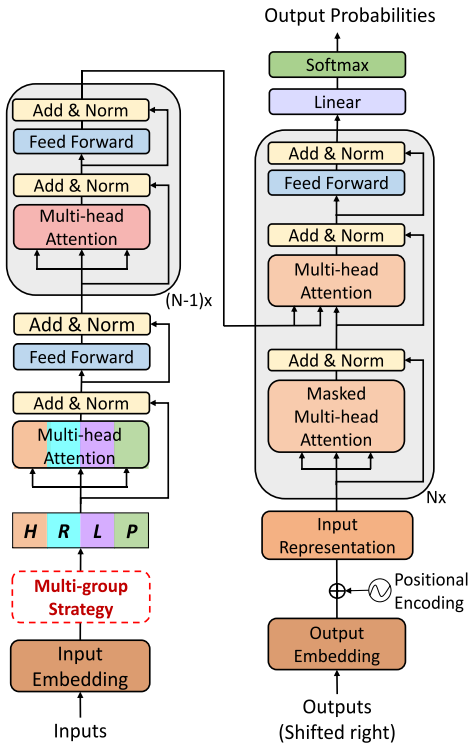


Fig. 2. Proposed Fusing_Diverse model. Note that *H*, *R*, *L*, and *L* denote the input representation of global features, the input representation of recurrence features, the input representation of local features, and the input representation of POS-tag features. They are concatenated to form our diverse input representation.

$\mathbb{R}^{2*64}$, $\boldsymbol{x}_j^L \in \mathbb{R}^{2*64}$, and $\boldsymbol{x}_j^P \in \mathbb{R}^{2*64}$, respectively. Similarly, the global feature function Func_{pe} is applied to the first group $\boldsymbol{X}_1^H$ for learning its input representation $\boldsymbol{H}$. Moreover, $\boldsymbol{H}$ together with the other three feature representations (i.e, $\boldsymbol{R}$, $\boldsymbol{L}$, and $\boldsymbol{P}$) are concatenated as the input of the SAN-based encoder to learn the final source representation.

V. Experiments

A. Data Sets

The proposed methods were evaluated using two translation data sets, namely the WMT14 English to German (EN-DE) and the NIST Chinese to English (ZH-EN). The EN-DE training set includes 4.43 million bilingual sentence pairs from the WMT14 corpora,[2] where the newstest2013 and newstest2014 data sets were used as development and test set, respectively. The ZH-EN training set includes 1.25 million bilingual sentence pairs from the LDC corpora,[3] where the NIST06 and NIST03/NIST04/NIST05 data-sets were used as the development set and test set, respectively.

[2]Common Crawl, News Commentary, and Europarl v7.

[3]LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08, and LDC2005T06.

B. System Setting

The byte pair encoding (**BPE**) algorithm [21] was applied to encode all sentences, and limited the vocabulary size to 32K. The dimension of all input and output layers was set to 512, and that of the inner FFN layer was set to 2048. The total number of heads in all multi-head modules was set to 8 in both the encoder and decoder layers. Each training batch consisted of a set of sentence pairs containing approximately 4096*4 source tokens and 4096*4 target tokens. During training, the label smoothing value was set to 0.1, and then the attention dropout and residual dropout rates were set to 0.1. The Adam optimizer was used to tune the model parameters. The learning rate was varied under 8000 warm-up steps. We validated the model at intervals of 1000 training steps on the development sets. After finishing the training of 200k batches, the model with the highest BLEU score (dev set) was selected to evaluate the test sets. The beam size of decoding was set to 4. Also, other settings are the same in the experiment setting of the original Transformer [1]. All models were trained and evaluated on a single P100 GPU. The tokenized case-sensitive 4-gram BLEU score [22] is used as the evaluation metric.

Furthermore, there are some additional specific settings. For the +Rec_Diverse model, given the 512-dimensions of word vector, the recurrent features have the 128-dimensions of two heads, that is, d_r is equal to 128, and the hidden dimension of RNN is 256. For the +Syn_Diverse model, the POS-tags were gained through the Stanford CoreNLP toolkit [17] and we chose the dimensions of three heads to capture POS tag features, that is, d_s is equal to 192. BPE units from the same word have the same POS-tag as this word. For the +Loc_Diverse and +Fusing_Diverse models, local features have the 128-dimensions of two heads, that is, d_l is equal to 128. For the baseline Transformer (Base) and our proposed NMT models, their embedding sizes are as shown in Table I. When the convolution kernel traverses the sequence of word vectors including the start and end symbols of the input sentence. The width of the convolution kernel m is set as 5 and the stride is set as 1 empirically.

Based on the OpenNMT toolkit [23], we implemented the following baseline methods: the vanilla Transformer model [1], a multi-head attention with disagreement regularization (**DisReg**) model [24], a Transformer model with relative positional representation (**Rel_PE**) [25], and a Transformer model with directional SAN (**DiSAN**).

Moreover, we reported the results of known related studies:

- **RNMT+** [5] combining the structures of RNN and SANs to model the translation processing.
- **+Localness** [6] introducing a Gaussian bias to tune the output of the SAN for modeling local information in the source sentence.
- **+BIRNN** [8] modeling directly the recurrence feature for the Transformer with an additional recurrence encoder.
- **+CSAN** [9] introducing a convolutional SANs to model local information of the source sentence for Transformer.
- **+LinguisticFeature** [16] simply concatenated the vector representations of source word and its syntactic and dependency labels as the source input. We re-implemented

TABLE I
EMBEDDING SIZE FOR DIFFERENT FEATURES IN THE EACH MODEL

Models	Embeding size			
	Global features	Recurrence features	Local features	Syntactic features
Transformer (base)	512	N/A	N/A	N/A
+Rec_Diverse	384	128	N/A	N/A
+Local_Diverse	384	N/A	128	N/A
+Syn_Diverse	320	N/A	N/A	192
+Fusing_Diverse	128	128	128	128

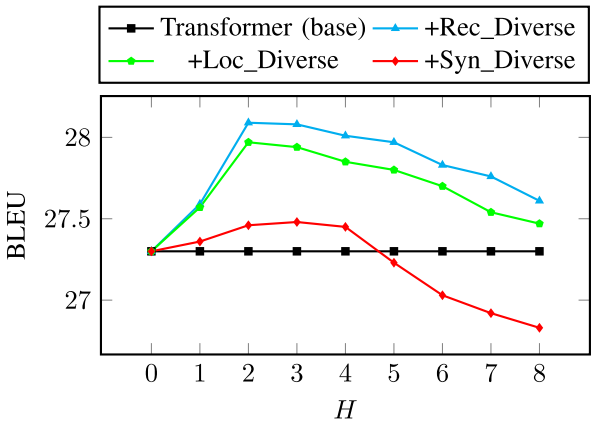


Fig. 3. BLEU scores of the EN-DE newstest2014 test set for the Transformer (Base), +Rec_Diverse, +Loc_Diverse, and +Syn_Diverse models with different numbers of heads.

and applied their method to the Transformer-based NMT model.

Note that our method slightly modified the input representation of the existing encoder to enhance the source representation, but did not make any modifications to the decoder. Therefore, we only report the results of the above comparison methods over the encoder of the Transformer framework in the subsequent experiments.

C. Effect of Source Features

Fig. 3 shows the BLEU scores of the EN-DE newstest 2014 test set on different numbers of heads for this source feature, to explore the relation between diverse source features and translation performance. +Rec_Diverse and +Loc_Diverse models are superior to the baseline Transformer (Base) model in all the number of head, and +Syn_Diverse model outperforms the baseline Transformer (Base) model when H is lower than 4. This means that these diverse source features are beneficial for the Transformer-based NMT model. Moreover, BLEU scores of +Rec_Diverse and +Loc_Diverse models begin to decrease when H is more than 2. In particular, BLEU scores of +Syn_Diverse model begin to decrease when H is more than 3, and even BLEU scores of +Syn_Diverse model are lower than that of the baseline Transformer (Base) model when H is more than 4. Meanwhile, each of the three source features is a different contribution to the improvement of translation performance. For example, BLEU scores of +Rec_Diverse model are better than that of +Loc_Diverse model at each number of heads. Therefore, considering the feature diversity and the training efficiency, we set the number of heads for each source feature to two heads in the proposed +Fusing_Diverse model.

TABLE II
RESULTS FOR THE WMT14 EN-DE TRANSLATION TASK

System	newstest2014	#Param
Existing NMT Systems		
Transformer (Base) [1]	27.31	88.0M
+Localness [6]	27.81	88.8M
+BIRNN [8]	27.54	97.4M
+CSAN [9]	27.86	88.0M
Transformer (Big) [1]	28.4	213.4M
RNMT+ [5]	28.49	378.9M
Our NMT Systems		
Transformer (Base)	27.30 (26.7)	97.35M
+DisReg [24]	27.45 (26.8)	97.35M
+Rel_PE [25]	27.63 (27.0)	97.35M
+DiSAN [4]	27.66 (27.0)	97.39M
+LinguisticFeatures [16]	27.58 (26.8)	97.35M
+Rec_Diverse	28.11++ (27.4++)	97.88M
+Loc_Diverse	27.97++ (27.4++)	97.35M
+Syn_Diverse	27.46+ (26.8)	97.35M
+Fusing_Diverse	28.58++ (27.8++)	97.88M
Transformer (Big)	28.25 (27.6)	272.6M
+Fusing_Diverse	29.32++ (28.1++)	274.2M

BLEU (in parentheses: sacreBLEU [26]) scores on EN-DE newstest2014 test set. Note: "++/+" after the BLEU and sacreBLEU scores indicate that the proposed method was significantly better than the corresponding baseline Transformer (base or big) at significance level $p < 0.01/0.05$.

D. Main Results

Table II presents the results of the newstest2014 test set for the EN-DE translation task. We make the following observations:

1) Our baseline Transformer (Base) model achieves similar performances with that of the original Transformer (Base) [1]. +Rec_Diverse, +Loc_Diverse, and +Syn_Diverse obtain 0.81/0.67/0.16 BLEU points improvements over the Transformer (Base). This result indicates that the various source features are beneficial for machine translation in addition to the original global feature.

2) By introducing the same feature, +Rec_Diverse outperforms the existing +BIRNN, and our +Loc_Diverse also outperforms the existing +Localness and +CSAN. This indicates the superiority of our approach.

3) Among the proposed features, +Rec_Diverse and +Loc_Diverse outperform +Syn_Diverse by 0.6 and 0.49 BLEU points, respectively. Compared with the POS-tag features, both the recurrence and local features work better together with the SANs.

TABLE III
RESULTS FOR THE ZH-EN TRANSLATION TASK

System	NIST03	NIST04	NIST05	#Param
Trans (Base)	46.51	45.91	45.81	77.9M
+DisReg [24]	46.81	46.28	45.72	77.9M
+Rel_PE [25]	46.91	46.47	46.32	77.9M
+DiSAN [4]	47.01	46.23	46.36	78.3M
+LinguisticFeature [16]	46.61	45.18	46.07	78.3M
+Rec_Diverse	47.07++	46.27+	46.32++	78.0M
+Loc_Diverse	46.89+	46.21+	46.23++	77.9M
+Syn_Diverse	46.48	46.08+	46.37+	77.9M
+Fusing_Diverse	47.36++	46.48++	47.29++	78.1M
Trans (Big)	47.83	47.21	47.37	243.7M
+Fusing_Diverse	48.57++	47.96++	48.02++	245.3M

Note: "++/+" after the BLEU score indicate that the proposed method was significantly better than the corresponding baseline Transformer (base or big) at significance level $p < 0.01/0.05$.

4) +Syn_Diverse is slightly better than Transformer (Base). The reason may be that the neural network itself has encoded the same type of translation knowledge as POS-tag features. +LinguisticFeature with many syntactic features gains 0.12 BLEU score improvement over the proposed +Syn_Diverse with only POS-tag features. We think that there is a ceiling on the improvement of translation performance for the same type of translation knowledge.

5) +Fusing_Diverse is substantially better than +Rec_Diverse, +Loc_Diverse, and +Syn_Diverse. This means that the proposed multi-group strategy enables multiple diverse source features to complement each other to realize an enhanced translation performance from the source sentence.

6) In addition, +Fusing_Diverse outperformed the recent advanced methods (both reported and re-implemented). This indicates that the proposed method achieved improvement over state-of-the-art baselines.

7) The proposed models seldom introduced additional model parameters, indicating that the improvement was from the diverse input feature information rather than larger neural networks.

Furthermore, Table III also shows a similar improvement regarding the ZH-EN translation task. This means that the proposed approach is a universal method for improving the translation of other language pairs.

E. Diversity Analysis

We investigated the diversity of the proposed +Fusing_Diverse empirically in terms of the learned head representations. To show the effect of source features, following the evaluation method in Li *et al.* [24], we first computed the averaged cosine distance (Sim) of the head representations between two feature groups G_1, G_2:

$$\mathrm{Sim}(G_1, G_2) = \frac{1}{H_1 * H_2} \sum_{i=1}^{H_1} \sum_{j=1}^{H_2} \frac{O_1^i \cdot O_2^j}{\| O_1^i \| \| O_2^j \|}, \qquad (14)$$

where H_1 and H_2 are the total head numbers of G_1 and G_2; O_1^i and O_2^j are the final feature representations for each token of $head_i$ and $head_j$, respectively. We then accumulated the similarity scores for the newstest2014 test set sentences and divided by the total number of tokens included in the newstest2014 test set. Table IV presents the similarity scores between the heads of different feature groups.

TABLE IV
SIMILARITY SCORES BETWEEN THE HEADS OF DIFFERENT FEATURE GROUPS ON THE NEWSTEST2014 TEST SET FOR THE PROPOSED +FUSING_DIVERSE MODEL

Model	Feature Group	Feature Group			
		Global	Rec	Loc	Syn
Transformer (Base)	Global	0.826	n/a	n/a	n/a
	Average	0.826			
+Fusing_Diverse	Global	0.817	0.415	0.483	0.765
	Rec	0.415	0.853	0.382	0.612
	Loc	0.483	0.382	0.912	0.717
	Syn	0.765	0.612	0.717	0.932
	Average	0.641			

TABLE V
TRAINING SPEED AND DECODING SPEED MEASURED IN SOURCE TOKENS PER SECOND

System	#Speed1	#Speed2
Transformer (Base)	9910	181
+Rec_Diverse	9839	176
+Loc_Diverse	9866	179
+Syn_Diverse	9900	181
+Fusing_Diverse	9811	175

Note: "#Speed1" and "#Speed2 d (tokens/second) denote the training and decoding on the EN-DE translation task.

1) The similarity scores of the heads in the same group were significantly higher than those of the heads between different groups; for example, Sim(Global, Global) vs Sim(Global, Rec). This indicates that the same type of source feature information tends to have similar head representations which are weak in capturing more diverse feature information of the source sentence for the Transformer-based NMT.

2) The average similarity of the proposed +Fusing_Diverse (0.641) was much less than the average similarity using the baseline Transformer (Base) global information (0.826). This indicates that the proposed method is helpful for encoding more diverse source features into the source representation.

F. Training Speed and Decoding Speed

Table V lists the training speeds and decoding speeds of the vanilla Transformer (Base) and the proposed models on the EN-DE training set and the newstest2014 test set. The training and decoding speeds of our models were slightly lower than those of the Transformer (Base). These results together indicate that the proposed method is highly efficient.

G. Analysis on Recurrence Features

Typically, RNN is good at capturing order dependencies between words in NMT [5], [8], [27]–[30]. Fig. 4 showed experiment results with source sentences of different lengths which are corresponding to different recurrent steps on the Transformer (Base)+Rec_Diverse model, to verify the effect of the introduced local features. For example, "(30, 40]" indicates that the length

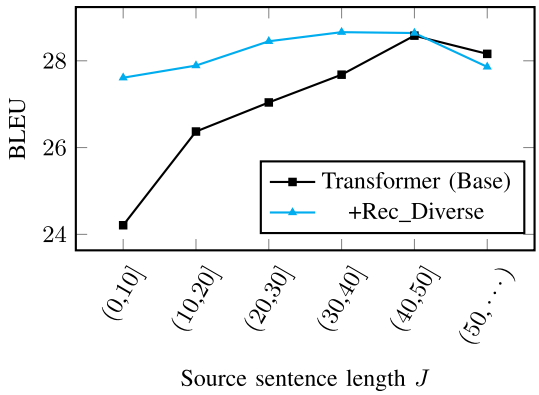


Fig. 4. Effect of recurrent steps through different sentence length over the EN-DE newstest2014 test set.

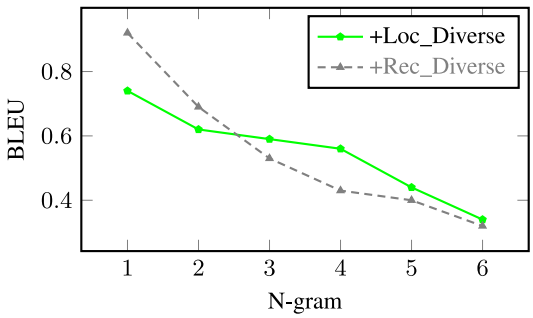


Fig. 5. Performance improvement of +Loc_Diverse model according to N-gram on the EN-DE newstest2014 test set. Y-axis denotes the gaps of different granularities of n-grams between +Loc_Diverse (or comparison +Rec_Diverse) and Transformer (Base).

of sentences is between thirty and forty, including forty. The BLEU score typically goes up with the increase of the sentence length until fifty. We think that the increase of the sentence length gives more context information, from which the self-attention mechanism can encode more global translation knowledge into the sentence representation. Thus, the trend does not hold when the source sentence length J is more than fifty, indicating which the advantage of the self-attention mechanism is an upper limit. Meanwhile, the performance of the proposed +Rec_Diverse is superior to that of the Transformer (Base) between ten and fifty of sentence length. The difference may be that the order information for recurrence features is different from that captured by positional encoding in the existing encoder. In other words, the proposed +Rec_Diverse enables the self-attention mechanism to model global dependency over the recurrence features, thus encoding another diverse order dependencies for recurrence features into the final source representation.

H. Analysis on Local Features

Generally, modeling local features is to capture useful phrase patterns in NMT [6], [9], [13], [31]. To evaluate the phrase pattern, we calculated the accuracy of different n-grams for the proposed +Loc_Diverse model and the baseline Transformer (Base) model, as shown in Fig. 5. The Transformer (Base)+Loc_Diverse model consistently outperforms the baseline Transformer (Base) model on each n-gram, indicating that local features are beneficial to encode more translation information from the source sentence. In particular, when n-grams are greater than 2, the accuracies of n-grams in +Loc_Diverse are higher than one in +Rec_Diverse. This indicates that local features are good at capturing phrasal information in the source sentence compared to recurrence features.

I. Linguistic Analyses

In this section, we selected eight probing tasks [32] (See Table VI) to study what syntactic and semantic properties are captured by the encoders. Specifically, we used the encoders of the trained Transformer (Base) model and our four models to generate the sentence representations of input, which are used to carry out the above eight probing tasks, and the results are as shown in Table VII. We make the following observations:

1) For syntactic properties, +Rec_Diverse, and +Loc_Diverse outperformed the baseline Transformer (Base). Recurrence and local features encoded more source syntactic information into the learned source representations. Meanwhile, +Syn_Diverse model was superior to the baseline Transformer (Base) model while +Syn_Diverse model was slightly less than the baseline Transformer (Base) model on the "TrDep" and "BShif" tasks. In other words, POS-tag features provided syntactic constituent information for the encoder.

2) Concerning semantic properties, +Rec_Diverse gained the highest scores on the "SubN" and "ObjN" tasks. Recurrence features can capture more long-distance clause information in the source sentence. Also, +Loc_Diverse and +Syn_Diverse gave the highest score on the "Tense" and "SoMo' tasks, respectively. This indicates that POS-tag features are beneficial to distinguish the tense of the (main clause verb while local features provide a local (or phrasal) context information to encode translation information related to important noun or verb.

3) For +Fusing_Diverse, although the highest scores were gained over only three probing tasks ("TrDep", "BShif", and "Coln"), its scores over other five tasks were slightly inferior to the highest score of the corresponding task, indicating that +Fusing_Diverse fuses four features to some extent and allows them to complement each other to improve the performance of NMT as shown in Table II and Table III.

J. Translation Case

Fig. 6 shows the translation outputs of the proposed +Fusing_Diverse and the Transformer (Base) on the ZH-EN translation task. Intuitively, compared with the reference translations, two obvious translations that are inconsistent with the semantics of the source language sentence in the translation output generated by the Transformer (Base) model. First, the red phrase "have worked" may mean that North and South Korea have worked together in actual affairs rather than in the spirit of diplomacy in the source sentence. Second, there are two different Summer Olympics in the source sentence, including the 2000 Sydney Summer Olympics and the 2004 Athens Summer Olympics. However, there is only one Summer Olympics held in the error

TABLE VI
SELECTED EIGHT PROBING TASKS [32] TO STUDY WHAT SYNTACTIC AND SEMANTIC PROPERTIES ARE CAPTURED BY THE ENCODERS

Probing Tasks		Content
Syntactic	TrDep	Checking whether an encoder infers the hierarchical structure of sentence
	ToCo	Sentences should be classified in terms of the sequence of top constituents immediately below the sentence node
	BShif	Testing whether two consecutive tokens within the sentence have been inverted
Semantic	Tense	Asking for the tense of the main clause verb
	SubN	Focusing on the number of the main clause's subject
	ObjN	Testing for the number of the direct object of the main clause
	SoMo	Some sentences are modified by replacing a random noun or verb with another one and the classifier should tell whether a sentence has been modified
	CoIn	Containing sentences made of two coordinate clauses

TABLE VII
CLASSIFICATION ACCURACIES ON EIGHT PROBING TASKS OF EVALUATING LINGUISTICS EMBEDDED IN THE ENCODER OUTPUTS

Model	Syntactic			Semantic				
	TrDep	ToCo	BShif	Tense	SubN	ObjN	SoMo	CoIn
Transformer (Base)	45.67	78.41	73.14	89.02	83.13	83.62	54.16	63.57
+Rec_Diverse	46.89	79.17	73.84	88.29	**84.71**	**83.79**	54.31	63.65
+Loc_Diverse	46.46	79.06	73.47	89.28	83.49	83.41	**54.94**	63.52
+Syn_Diverse	45.61	**79.24**	73.08	**89.96**	82.64	83.16	54.59	63.54
+Fusing_Diverse	**47.96**	79.21	**74.03**	89.92	84.59	83.71	54.82	**63.69**

year in the translation output generated by the Transformer (Base) model, that is, "the Sydney Summer Olympics, which was held in 2004 at the Athens Summer Games". In contrast, the translation output generated by the +Fusing_Diverse model faithfully expresses the meaning of the source language sentence. We think that the +Fusing_Diverse model is to encode diverse source translation knowledge to generate a better target translation.

From an evaluation perspective, the number of matched n-grams in +Fusing_Diverse is much more than that of matched n-grams in Transformer (Base). In particular, there are many successfully matched 3-gram and 4-gram fragments, for example, "between the two sides", "the past few years", "marched together", and "the delegations". Intuitively, these matched n-grams for our +Fusing_Diverse are more coherent than ones for the Transformer (Base), which is consistent with the results in Fig. 4. Moreover, these successfully matched higher n-grams are consistent with the results in Fig. 5. Therefore, we think that the improvement of +Fusing_Diverse may be mainly attributed to the introduced recurrent features and local features, thus leading to a more faithful and fluent target translation.

VI. RELATED WORK

In this section, we briefly review previous studies that are related to our work. Here we divide previous work into three categories:

A. Multi-Features in Traditional SMT

In traditional SMT, many translation knowledge has been proved to be useful for machine translation, for example, reordering knowledge [33]–[35], structural knowledge [36]–[39], syntactic knowledge [40]–[42], long-distance dependency features [43]–[45]. The successful introduction of these translation features is mainly due to the log-linear model, in which machine translation is treated as a multi-features system each feature of which is to capture the relevant knowledge of the translation task. SMT with the log-linear model allows any other translation feature to be easily introduced into the existing SMT system. In this paper, we proposed a unified method to explore more diverse source features for the NMT, and hope that the proposed method can help researchers easily find and verify the effect of other potentially useful features in NMT.

B. Source Features in Traditional NMT

In the traditional NMT, there has been a substantial amount of research works on source translation knowledge, which used important translation features to enhance the performance of NMT. These source features fall into three groups:

Multi-source features: Zoph *et al.* [46] used a German sentence with equivalent meaning to the French sentence as the source input to generate an English translation. Firat *et al.* [47] further proposed a shared attention mechanism where each target language has one attention shared by all source languages to translate between many different source and target languages. Moreover, image information, which corresponds to the content described by the source language sentence, is also an additional translation feature to build the multi-model NMT for enhancing the representation of the source sentence [48]–[50].

Syntactic features: Generally, the syntax tree of the input sentence is linearized as a sequence of syntactic labels to encode syntax information explicitly so as to improve the performance of NMT. For example, the vector representations of syntactic features are used to augment the neural network layer of NMT, for example, the word vectors of the input layer [16], [18], [19] and the states of the hidden layer [18], [51]. Moreover, the long-distant constraint from the dependence tree is used to learn an additional syntax context vector for predicting translations [45], [52]–[54].

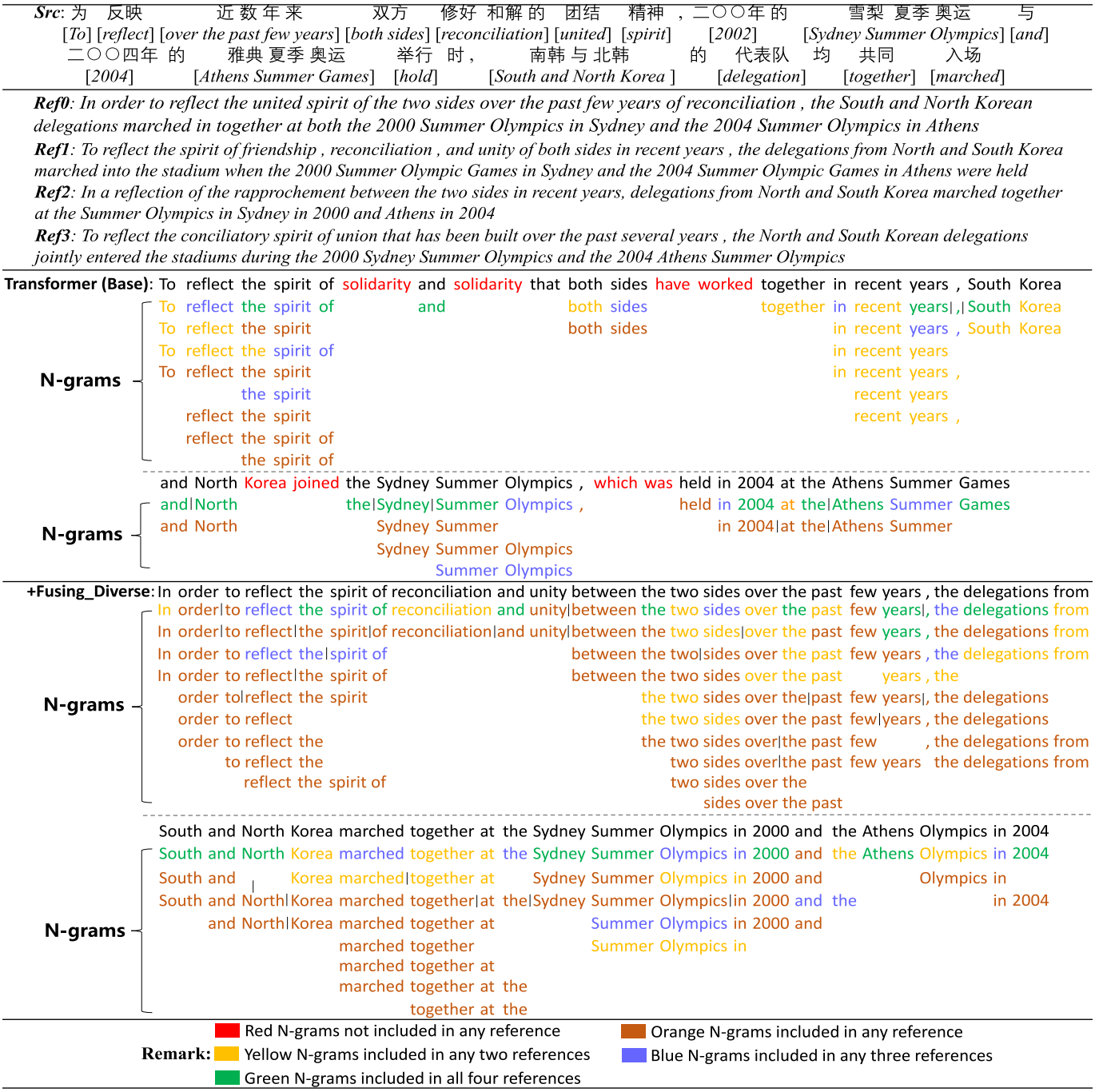


Fig. 6. Chinese-to-English translation outputs of Transformer (Base) and +Fusing_Diverse models. In the "N-grams", one word or multiple consecutive words of the same color represent an n-gram unit. Also, Multiple groups of words of the same color are separated by a black separator "|", where each group represents an n-gram unit.

Structural features: In addition to linearizing the parse tree, the structural neural networks were adopted to encode source translation information. For example, Eriguchi *et al.* [55] first used a tree-based encoder to learn the representation of the source sentence following its parse tree instead of the sequential encoder. It was extended further using the bidirectional tree encoder which learned both the sequential and tree-structured representations [56]. Gū *et al.*[57] exploited a top-down tree-structured model to combine a sequential encoder with tree-structured decoding augmented with a syntax-aware attention model.

C. Source Features in Transformer

Recent studies reveal that the lack of these diverse source features (i.e, structural features) hinder further improvement of

the translation capacity of Transformer [3], [5]–[9], [58]–[60]. For example, Chen *et al.* [5] first attempted to merge the representations learned by the SAN-based and RNN-based encoders in a unified framework and reported an exciting improvement in translation performance. Song *et al.* [3] introduced a sequence-to-sequence model for the double path network, including a convolutional-based NMT path and a SAN-based NMT path, to model translation processing. Hao *et al.* [8] integrated a novel attentive recurrent network into the existing Transformer model to capture the recurrence feature in the source sentence. Yang *et al.* [6] introduced a Gaussian bias to revise the original attention distribution for capturing local context features. Yang *et al.* [9] further proposed convolutional self-attention networks to model locality for the self-attention mechanism and interactions between features learned by different attention heads. Ma *et al.* [59] introduced neural syntactic distance [20] into the existing Transformer model to capture syntactic features of source sentence. This paper focused on capturing more diverse source features simultaneously by a general method instead of a specific-feature model architecture.

In addition, Voita *et al.* [11] evaluated the contribution of individual attention heads, and found that specialized heads are last to be pruned without seriously affecting performance. Michel et al [12] surprisingly observed that some layers can even be reduced to a single head. These findings provided a potential opportunity for introducing more diverse source features into the existing SAN-encoder.

VII. Conclusion

In this paper, we investigated the effect of diverse source features in the neural machine translation network architecture, namely, Transformer. The proposed multi-group strategy allows these diverse source features to be encoded into the input representation in a unified way, leading thereby to an efficient source representation for the decoder of the Transformer-base NMT. Experimental results obtained from WMT14 EN-DE and NIST ZH-EN translation tasks show that the proposed models can effectively improve the performance of Transformer NMT model.

In the future, we will be exploring more translation features, including both source and target features, to enhance the Transformer translation system. Moreover, we will also apply our method to other natural language processing tasks.

References

[1] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., Dec. 2017, pp. 5998–6008.

[2] Z.-Y. Dou, Z. Tu, X. Wang, S. Shi, and T. Zhang, "Exploiting deep representations for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 4253–4262.

[3] K. Song, X. Tan, D. He, J. Lu, T. Qin, and T.-Y. Liu, "Double path networks for sequence to sequence learning," in *Proc. 27th Int. Conf. Comput. Linguist.*, Aug. 2018, pp. 3064–3074.

[4] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for RNN/CNN-free language understanding," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 5446–5455.

[5] M. X. Chen *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2018, pp. 76–86.

[6] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang, "Modeling localness for self-attention networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 4449–4458.

[7] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, "Universal transformers," in *Proc. Int. Conf. Learn. Representations*, Mar. 2019.

[8] J. Hao, X. Wang, B. Yang, L. Wang, J. Zhang, and Z. Tu, "Modeling recurrence for transformer," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2019, pp. 1198–1207.

[9] B. Yang, L. Wang, D. F. Wong, L. S. Chao, and Z. Tu, "Convolutional self-attention networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2019, pp. 4040–4045.

[10] A. Currey and K. Heafield, "Incorporating source syntax into transformer-based neural machine translation," in *Proc. 4th Conf. Mach. Transl.*, Aug. 2019, pp. 24–33.

[11] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2019, pp. 5797–5808.

[12] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2019, pp. 14 014–14 024.

[13] J. Gehring, M. Auli, D. Grangier, and Y. Dauphin, "A convolutional encoder model for neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2017, pp. 123–135.

[14] K. Cho *et al.*, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2014, pp. 1724–1734.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 3104–3112.

[16] R. Sennrich and B. Haddow, "Linguistic input features improve neural machine translation," in *Proc, 1st Conf. Mach. Translation*, Aug. 2016, pp. 83–91.

[17] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguist.: Syst. Demonstrations*, Jun. 2014, pp. 55–60.

[18] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, and G. Zhou, "Modeling source syntax for neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2017, pp. 688–697.

[19] K. Chen *et al.*, "Neural machine translation with source dependency representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2017, pp. 2846–2852.

[20] Y. Shen, Z. Lin, A. P. Jacob, A. Sordoni, A. Courville, and Y. Bengio, "Straight to the tree: Constituency parsing with neural syntactic distance," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2018, pp. 1171–1180.

[21] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist.*, Aug. 2016, pp. 1715–1725.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2002, pp. 311–318.

[23] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. Syst. Demonstrations*, Jul. 2017, pp. 67–72.

[24] J. Li, Z. Tu, B. Yang, M. R. Lyu, and T. Zhang, "Multi-head attention with disagreement regularization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 2897–2903.

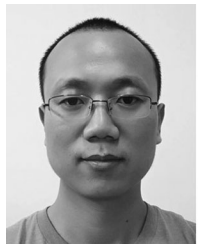
[25] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2018, pp. 464–468.

[26] M. Post, "A call for clarity in reporting BLEU scores," in *Proc. 3rd Conf. Mach. Translation: Res. Papers*, Oct. 2018, pp. 186–191.

[27] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2013, pp. 1700–1709.

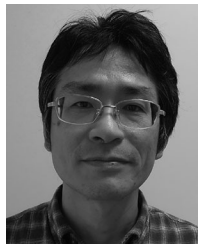
[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Representations*, May 2015.

[29] Y. Ding, Y. Liu, H. Luan, and M. Sun, "Visualizing and understanding neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2017, pp. 1150–1159.
[30] K. Chen, R. Wang, M. Utiyama, and E. Sumita, "Recurrent positional embedding for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, Nov. 2019, pp. 1361–1367.
[31] J. Hao, X. Wang, S. Shi, J. Zhang, and Z. Tu, "Multi-granularity self-attention for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, Nov. 2019, pp. 887–897.
[32] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single vector: Probing sentence embeddings for linguistic properties," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2018, pp. 2126–2136.
[33] D. Xiong, Q. Liu, and S. Lin, "Maximum entropy based phrase reordering model for statistical machine translation," in *Proc. 21st Int. Conf. Comput. Linguist. 44th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2006, pp. 521–528.
[34] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2008, pp. 848–856.
[35] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2011, pp. 1045–1054.
[36] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. Human Lang. Technol. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2003, pp. 127–133.
[37] D. Chiang, "Hierarchical phrase-based translation," *Comput. Linguist.*, vol. 33, no. 2, pp. 201–228, 2007.
[38] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proc. 21st Int. Conf. Comput. Linguist. 44th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2006, pp. 609–616.
[39] M. Zhang, H. Jiang, A. Aw, H. Li, C. L. Tan, and S. Li, "A tree sequence alignment-based tree-to-tree translation model," in *Proc. ACL-08: HLT*, Jun. 2008, pp. 559–567.
[40] P. Koehn and H. Hoang, "Factored translation models," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Jun. 2007, pp. 868–876.
[41] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proc. Human Lang. Technol.: Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, Jun. 2009, pp. 218–226.
[42] R. Sennrich, "Modelling and optimizing on syntactic N-grams for statistical machine translation," *Trans. Assoc. Comput. Linguist.*, vol. 3, pp. 169–182, 2015.
[43] J. Xie, H. Mi, and Q. Liu, "A novel dependency-to-string model for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Jul. 2011, pp. 216–226.
[44] L. Li, A. Way, and Q. Liu, "Dependency graph-to-string translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 33–43.
[45] K. Chen *et al.*, "A neural approach to source dependence based context model for statistical machine translation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 2, pp. 266–280, Feb. 2018.
[46] B. Zoph and K. Knight, "Multi-source neural translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2016, pp. 30–34.
[47] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proc. Conf. North Amer. Chapter. Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2016, pp. 866–875.
[48] I. Calixto, Q. Liu, and N. Campbell, "Doubly-attentive decoder for multi-modal neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2017, pp. 1913–1924.
[49] I. Calixto and Q. Liu, "Incorporating global visual features into attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2017, pp. 992–1003.
[50] I. Calixto, M. Rios, and W. Aziz, "Latent variable model for multi-modal translation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2019, pp. 6392–6405.
[51] S. Wu, M. Zhou, and D. Zhang, "Improved neural machine translation with source syntax," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4179–4185.
[52] K. Chen, R. Wang, M. Utiyama, E. Sumita, and T. Zhao, "Syntax-directed attention for neural machine translation," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 4792–4799.
[53] A. Currey and K. Heafield, "Multi-source syntactic neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 2961–2966.
[54] S. Wu, D. Zhang, Z. Zhang, N. Yang, M. Li, and M. Zhou, "Dependency-to-dependency neural machine translation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 11, pp. 2132–2141, Nov. 2018.
[55] A. Eriguchi, K. Hashimoto, and Y. Tsuruoka, "Tree-to-sequence attentional neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist.*, Aug. 2016, pp. 823–833.
[56] H. Chen, S. Huang, D. Chiang, and J. Chen, "Improved neural machine translation with a syntax-aware encoder and decoder," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2017, pp. 1936–1945.
[57] J. Gū, H. S. Shavarani, and A. Sarkar, "Top-down tree structured decoding with syntactic connections for neural machine translation and parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 401–413.
[58] K. Ahmed, N. S. Keskar, and R. Socher, "Weighted transformer network for machine translation," 2017, *arXiv: abs/1711.02132*.
[59] C. Ma, A. Tamura, M. Utiyama, E. Sumita, and T. Zhao, "Improving neural machine translation with neural syntactic distance," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, Jun. 2019, pp. 2032–2037.
[60] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang, and J. Xie, "A hierarchy-to-sequence attentional neural machine translation model," *IEEE/ACM Trans. Audio, Speech Lang. Proc.*, vol. 26, no. 3, pp. 623–632, Mar. 2018.



Kehai Chen received the B.S. degree from the Xi'an University of Technology, in 2010, the M.S. degree from the University of Chinese Academy of Sciences, in 2013, and the Ph.D. degree from the Harbin Institute of Technology, in 2018, all in computer science. He is a Researcher in National Institute of Information and Communications Technology, Japan since 2018. His research interests include machine translation and natural language processing.

Rui Wang received the B.S. degree from the Harbin Institute of Technology, in 2009, the M.S. degree from the Chinese Academy of Sciences, in 2012 and the Ph.D. degree in Shanghai Jiao Tong University, in 2016, all in computer science. He was a Joint Ph.D. in Centre Nationnal de la Recherche Scientifique, France in 2014. He is a Researcher in National Institute of Information and Communications Technology, Japan since 2016. His research interests include machine translation and natural language processing.



Masao Utiyama is a Research Manager of the National Institute of Information and Communications Technology, Japan. He completed his doctoral dissertation at the University of Tsukuba in 1997. His main research field is machine translation.

Eiichiro Sumita received the Bachelor and Master degree in computer science from The University of Electro-Communications, Japan, in 1980 and 1982, respectively, and the Ph.D degree in engineering from Kyoto University, Japan, in 1999. He is currently Director of Multilingual Translation Laboratory of National Institute of Information and Communication Technology from 2006. He worked at Advanced Telecomunications Research Institute International from 1992 to 2009 and IBM Research-Tokyo from 1980 to 1991. His research interests include Machine Translation and e-Learning.

Tiejun Zhao is a Professor at the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include: natural language understanding, content-based web information processing, and applied artificial intelligence. He has published three academic books and 60 papers in journals and conference proceedings in the last 3 years. He has been a PC member at ACL, COLING over the last 5 years and was also appointed as an MT Track Co-Chair for COLING 2014.

Muyun Yang is an Associate Professor of School of Computer Science and Technology, Harbin Institute of Technology. His research interests include: machine translation, computational linguistics and artificial intelligence. He has published one academic book and 20 papers on journals and conferences in recent 3 years.

Hai Zhao received the B.Eng. degree in sensor and instrument engineering and the M.Phil. degree in control theory and engineering from Yanshan University, Qinhuangdao, China, in 1999 and 2000, respectively, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2005. He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University after he joined the university in 2009. He was a Research Fellow with the City University of Hong Kong from 2006 to 2009, a Visiting Scholar in Microsoft Research Asia in 2011, a Visiting Expert in NICT, Japan in 2012. His research interests include natural language processing and related machine learning, data mining and artificial intelligence. He is an ACM Professional member, and served as Area Co-Chair in ACL 2017 on Tagging, Chunking, Syntax and Parsing, (senior) area chairs in ACL 2018, 2019 on Phonology, Morphology and Word Segmentation.