Exploiting Deep Representations for Neural Machine Translation

Zi-Yi Dou
Nanjing University
douzy@smail.nju.edu.cn

Zhaopeng Tu*
Tencent AI Lab
zptu@tencent.com

Xing Wang
Tencent AI Lab
brightxwang@tencent.com

Shuming Shi
Tencent AI Lab
shumingshi@tencent.com

Tong Zhang
Tencent AI Lab
bradymzhang@tencent.com

Abstract

Advanced neural machine translation (NMT) models generally implement encoder and decoder as multiple layers, which allows systems to model complex functions and capture complicated linguistic structures. However, only the top layers of encoder and decoder are leveraged in the subsequent process, which misses the opportunity to exploit the useful information embedded in other layers. In this work, we propose to simultaneously expose all of these signals with layer aggregation and multi-layer attention mechanisms. In addition, we introduce an auxiliary regularization term to encourage different layers to capture diverse information. Experimental results on widely-used WMT14 English⇒German and WMT17 Chinese⇒English translation data demonstrate the effectiveness and universality of the proposed approach.

1 Introduction

Neural machine translation (NMT) models have advanced the machine translation community in recent years (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014). NMT models generally consist of two components: an *encoder* network to summarize the input sentence into sequential representations, based on which a *decoder* network generates target sentence word by word with an attention model (Bahdanau et al., 2015; Luong et al., 2015).

Nowadays, advanced NMT models generally implement encoder and decoder as multiple layers, regardless of the specific model architectures such as RNN (Zhou et al., 2016; Wu et al., 2016), CNN (Gehring et al., 2017), or Self-Attention Network (Vaswani et al., 2017; Chen et al., 2018). Several researchers have revealed that different layers are able to capture different types of syntax and semantic information (Shi et al., 2016; Peters et al., 2018; Anastasopoulos and Chiang, 2018). For example, Shi et al. (2016) find that both local and global source syntax are learned by the NMT encoder and different types of syntax are captured at different layers.

However, current NMT models only leverage the top layers of encoder and decoder in the subsequent process, which misses the opportunity to exploit useful information embedded in other layers. Recently, aggregating layers to better fuse semantic and spatial information has proven to be of profound value in computer vision tasks (Huang et al., 2017; Yu et al., 2018). In natural language processing community, Peters et al. (2018) have proven that simultaneously exposing all layer representations outperforms methods that utilize just the top layer for transfer learning tasks.

Inspired by these findings, we propose to exploit deep representations for NMT models. Specifically, we investigate two types of strategies to better fuse information across layers, ranging from layer aggregation to multi-layer attention. While layer aggregation strategies combine hidden states at the same position across different layers, multi-layer attention allows the model to combine information in different positions. In addition, we introduce an auxiliary objective to encourage different layers to capture diverse information, which we believe would make the deep representations more meaningful.

We evaluated our approach on two widely-used WMT14 English⇒German and WMT17 Chinese⇒English translation tasks. We employed TRANSFORMER (Vaswani et al., 2017) as the baseline system since it has proven to outperform other architectures on the two tasks (Vaswani et al., 2017; Hassan et al., 2018). Experimen-

* Zhaopeng Tu is the corresponding author of the paper. This work was conducted when Zi-Yi Dou was interning at Tencent AI Lab.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4253–4262
Brussels, Belgium, October 31 - November 4, 2018.

tal results show that exploiting deep representations consistently improves translation performance over the vanilla TRANSFORMER model across language pairs. It is worth mentioning that TRANSFORMER-BASE with deep representations exploitation outperforms the vanilla TRANSFORMER-BIG model with only less than half of the parameters.

2 Background: Deep NMT

Deep representations have proven to be of profound value in machine translation (Meng et al., 2016; Zhou et al., 2016). Multiple-layer encoder and decoder are employed to perform the translation task through a series of nonlinear transformations from the representation of input sequences to final output sequences. The layer can be implemented as RNN (Wu et al., 2016), CNN (Gehring et al., 2017), or Self-Attention Network (Vaswani et al., 2017). In this work, we take the advanced Transformer as an example, which will be used in experiments later. However, we note that the proposed approach is generally applicable to any other type of NMT architectures.

Specifically, the encoder is composed of a stack of L identical layers, each of which has two sub-layers. The first sub-layer is a self-attention network, and the second one is a position-wise fully connected feed-forward network. A residual connection (He et al., 2016) is employed around each of the two sub-layers, followed by layer normalization (Ba et al., 2016). Formally, the output of the first sub-layer $\mathbf{C}_e^l$ and the second sub-layer $\mathbf{H}_e^l$ are calculated as

$$\begin{aligned}\mathbf{C}_e^l &= \textsc{Ln}\big(\textsc{Att}(\mathbf{Q}_e^l, \mathbf{K}_e^{l-1}, \mathbf{V}_e^{l-1}) + \mathbf{H}_e^{l-1}\big),\\ \mathbf{H}_e^l &= \textsc{Ln}\big(\textsc{Ffn}(\mathbf{C}_e^l) + \mathbf{C}_e^l\big),\end{aligned} \quad (1)$$

where $\textsc{Att}(\cdot)$, $\textsc{Ln}(\cdot)$, and $\textsc{Ffn}(\cdot)$ are self-attention mechanism, layer normalization, and feed-forward networks with ReLU activation in between, respectively. $\{\mathbf{Q}_e^l, \mathbf{K}_e^{l-1}, \mathbf{V}_e^{l-1}\}$ are query, key and value vectors that are transformed from the (*l-1*)-th encoder layer $\mathbf{H}_e^{l-1}$.

The decoder is also composed of a stack of L identical layers. In addition to two sub-layers in each decoder layer, the decoder inserts a third sub-layer $\mathbf{D}_d^l$ to perform attention over the output of the encoder stack $\mathbf{H}_e^L$:

$$\begin{aligned}\mathbf{C}_d^l &= \textsc{Ln}\big(\textsc{Att}(\mathbf{Q}_d^l, \mathbf{K}_d^{l-1}, \mathbf{V}_d^{l-1}) + \mathbf{H}_d^{l-1}\big),\\ \mathbf{D}_d^l &= \textsc{Ln}\big(\textsc{Att}(\mathbf{C}_d^l, \mathbf{K}_e^{L}, \mathbf{V}_e^{L}) + \mathbf{C}_d^l\big),\\ \mathbf{H}_d^l &= \textsc{Ln}\big(\textsc{Ffn}(\mathbf{D}_d^l) + \mathbf{D}_d^l\big),\end{aligned} \quad (2)$$

where $\{\mathbf{Q}_d^l, \mathbf{K}_d^{l-1}, \mathbf{V}_d^{l-1}\}$ are transformed from the (*l-1*)-th decoder layer $\mathbf{H}_d^{l-1}$, and $\{\mathbf{K}_e^L, \mathbf{V}_e^L\}$ are transformed from the top layer of the encoder. The top layer of the decoder $\mathbf{H}_d^L$ is used to generate the final output sequence.

Multi-layer network can be considered as a strong feature extractor with extended receptive fields capable of linking salient features from the entire sequence (Chen et al., 2018). However, one potential problem about the vanilla Transformer, as shown in Figure 1a, is that both the encoder and decoder stack layers in sequence and only utilize the information in the top layer. While studies have shown deeper layers extract more semantic and more global features (Zeiler and Fergus, 2014; Peters et al., 2018), these do not prove that the last layer is the ultimate representation for any task. Although residual connections have been incorporated to combine layers, these connections have been "shallow" themselves, and only fuse by simple, one-step operations (Yu et al., 2018). We investigate here how to better fuse information across layers for NMT models.

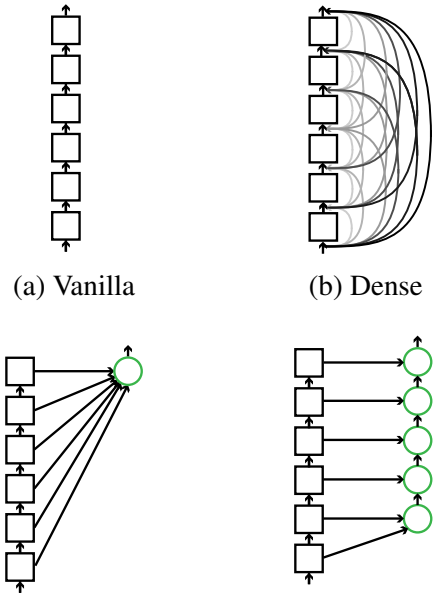
In the following sections, we simplify the equations to $\mathbf{H}^l = \textsc{Layer}(\mathbf{H}^{l-1})$ for brevity.

3 Proposed Approaches

In this section, we first introduce how to exploit deep representations by simultaneously exposing all of the signals from all layers (Sec 3.1). Then, to explicitly encourage different layers to incorporate various information, we propose one way to measure the diversity between layers and add a regularization term to our objective function to maximize the diversity across layers (Sec 3.2).

3.1 Deep Representations

To exploit deep representations, we investigate two types of strategies to fuse information across layers, from layer aggregation to multi-layer attention. While layer aggregation strategies combine hidden states at the same position across different layers, multi-layer attention allows the model to combine information in different positions.



(a) Vanilla (b) Dense

(c) Linear (d) Iterative

Figure 1: Illustration of (a) vanilla model without any aggregation and (b,c,d) layer aggregation strategies. Aggregation nodes are represented by green circles.

3.1.1 Layer Aggregation

While the aggregation strategies are inspired by previous work, there are several differences since we have simplified and generalized from the original model, as described below.

Dense Connection. The first strategy is to allow all layers to directly access previous layers:

$$\mathbf{H}^l = f(\mathbf{H}^1, \ldots, \mathbf{H}^{l-1}). \quad (3)$$

In this work, we mainly investigate whether densely connected networks work for NMT, which have proven successful in computer vision tasks (Huang et al., 2017). The basic strategy of densely connected networks is to connect each layer to every previous layer with a residual connection:

$$\mathbf{H}^l = \text{Layer}(\mathbf{H}^{l-1}) + \sum_{i=1}^{l-1} \mathbf{H}^i. \quad (4)$$

Figure 1b illustrates the idea of this approach. Our implementation differs from (Huang et al., 2017) in that we use an addition instead of a concatenation operation in order to keep the state size constant. Another reason is that concatenation operation is computationally expensive, while residual connections are more efficient.

While dense connection directly feeds previous layers to the subsequent layers, the following mechanisms maintain additional layers to aggregate standard layers, from shallow linear combination, to deep non-linear aggregation.

Linear Combination. As shown in Figure 1c, an intuitive strategy is to linearly combine the outputs of all layers:

$$\widehat{\mathbf{H}} = \sum_{l=1}^{L} \mathbf{W}_l \mathbf{H}^l, \quad (5)$$

where $\{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$ are trainable matrices. While the strategy is similar in spirit to (Peters et al., 2018), there are two main differences: (1) they use normalized weights while we directly use parameters that could be either positive or negative numbers, which may benefit from more modeling flexibility. (2) they use a scalar that is shared by all elements in the layer states, while we use learnable matrices. The latter offers a more precise control of the combination by allowing the model to be more expressive than scalars (Tu et al., 2017).

We also investigate strategies that *iteratively* and *hierarchically* merge layers by incorporating more depth and sharing, which have proven effective for computer vision tasks (Yu et al., 2018).

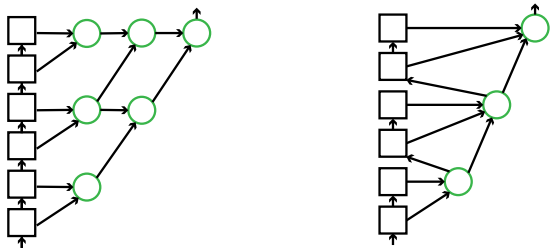
Iterative Aggregation. As illustrated in Figure 1d, iterative aggregation follows the iterated stacking of the backbone architecture. Aggregation begins at the shallowest, smallest scale and then iteratively merges deeper, larger scales. The iterative deep aggregation function I for a series of layers $\mathbf{H}_1^l = \{\mathbf{H}^1, \cdots, \mathbf{H}^l\}$ with increasingly deeper and semantic information is formulated as

$$\widehat{\mathbf{H}}^l = I(\mathbf{H}_1^l) = \text{AGG}(\mathbf{H}^l, \widehat{\mathbf{H}}^{l-1}), \quad (6)$$

where we set $\widehat{\mathbf{H}}^1 = \mathbf{H}^1$ and $\text{AGG}(\cdot, \cdot)$ is the aggregation function:

$$\text{AGG}(x, y) = \text{LN}(\text{FF}([x; y]) + x + y). \quad (7)$$

As seen, in this work, we first concatenate x and y into $z = [x; y]$, which is subsequently fed to a feed-forward network with a sigmoid activation in between. Residual connection and layer normalization are also employed. Specifically, both x and y have residual connections to the output. The choice of the aggregation function will be further studied in the experiment section.



(a) CNN-like tree (b) Hierarchical

Figure 2: Hierarchical aggregation (b) that aggregates layers through a tree structure (a).

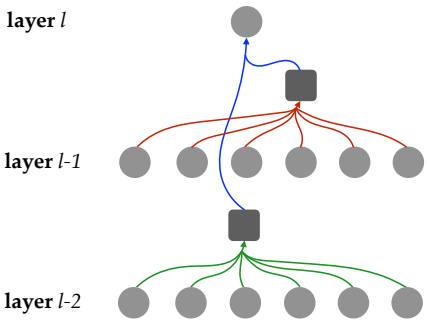


Figure 3: Multi-layer attention allows the model to attend multiple layers to construct each hidden state. We use two-layer attention for illustration, while the approach is applicable to any layers lower than l.

Hierarchical Aggregation. While iterative aggregation deeply combines states, it may still be insufficient to fuse the layers for its sequential architecture. Hierarchical aggregation, on the other hand, merges layers through a tree structure to preserve and combine feature channels, as shown in Figure 2. The original model proposed by Yu et al. (2018) requires the number of layers to be the power of two, which limits the applicability of these methods to a broader range of NMT architectures (e.g. six layers in (Vaswani et al., 2017)). To solve this problem, we introduce a CNN-like tree with the filter size being two, as shown in Figure 2a. Following (Yu et al., 2018), we first merge aggregation nodes of the same depth for efficiency so that there would be at most one aggregation node for each depth. Then, we further feed the output of an aggregation node back into the backbone as the input to the next sub-tree, instead of only routing intermediate aggregations further up the tree, as shown in Figure 2b. The interaction between aggregation and backbone nodes allows the model to better preserve features.

Formally, each aggregation node $\widehat{\mathbf{H}}^i$ is calculated as

$$\widehat{\mathbf{H}}^i = \begin{cases} \textsc{Agg}(\mathbf{H}^{2i-1}, \mathbf{H}^{2i}), & i = 1 \\ \textsc{Agg}(\mathbf{H}^{2i-1}, \mathbf{H}^{2i}, \widehat{\mathbf{H}}^{i-1}), & i = 2, 3 \end{cases}$$

where $\textsc{Agg}(\mathbf{H}^{2i-1}, \mathbf{H}^{2i})$ is computed via Eqn. 7, and $\textsc{Agg}(\mathbf{H}^{2i-1}, \mathbf{H}^{2i}, \widehat{\mathbf{H}}^{i-1})$ is computed as

$$\textsc{Agg}(x, y, z) = \textsc{Ln}(\textsc{Ff}([x; y; z]) + x + y + z).$$

The aggregation node at the top layer $\widehat{\mathbf{H}}^{L/2}$ serves as the final output of the network.

3.1.2 Multi-Layer Attention

Partially inspired by Meng et al. (2016), we also propose to introduce a multi-layer attention mechanism into deep NMT models, for more power of transforming information across layers. In other words, for constructing each hidden state in any layer-l, we allow the self-attention model to attend any layers lower than l, instead of just layer *l-1*:

$$\begin{aligned} \mathbf{C}^l_{-1} &= \textsc{Att}(\mathbf{Q}^l, \mathbf{K}^{l-1}, \mathbf{V}^{l-1}), \\ \mathbf{C}^l_{-2} &= \textsc{Att}(\mathbf{Q}^l, \mathbf{K}^{l-2}, \mathbf{V}^{l-2}), \\ &\cdots \\ \mathbf{C}^l_{-k} &= \textsc{Att}(\mathbf{Q}^l, \mathbf{K}^{l-k}, \mathbf{V}^{l-k}), \\ \mathbf{C}^l &= \textsc{Agg}(\mathbf{C}^l_{-1}, \ldots, \mathbf{C}^l_{-k}), \end{aligned} \quad (8)$$

where $\mathbf{C}^l_{-i}$ is sequential vectors queried from layer *l-i* using a separate attention model, and $\textsc{Agg}(\cdot)$ is similar to the pre-defined aggregation function to transform k vectors $\{\mathbf{C}^l_{-1}, \ldots, \mathbf{C}^l_{-k}\}$ to a d-dimension vector, which is subsequently used to construct the encoder and decoder layers via Eqn. 1 and 2 respectively. Note that multi-layer attention only modifies the self-attention blocks in both encoder and decoder, while does not revises the encoder-decoder attention blocks.

3.2 Layer Diversity

Intuitively, combining layers would be more meaningful if different layers are able to capture diverse information. Therefore, we explicitly add a *regularization* term to encourage the diversities between layers:

$$\mathcal{L} = \mathcal{L}_{\text{likelihood}} + \lambda \mathcal{L}_{\text{diversity}}, \quad (9)$$

where λ is a hyper-parameter and is set to 1.0 in this paper. Specifically, the regularization term measures the average of the distance between ev-

ery two adjacent layers:

$$\mathcal{L}_{diversity} = \frac{1}{L-1}\sum_{l=1}^{L-1} D(\mathbf{H}^l, \mathbf{H}^{l+1}). \quad (10)$$

Here $D(\mathbf{H}^l, \mathbf{H}^{l+1})$ is the averaged *cosine-squared distance* between the states in layers $\mathbf{H}^l = \{\mathbf{h}_1^l, \ldots, \mathbf{h}_N^l\}$ and $\mathbf{H}^{l+1} = \{\mathbf{h}_1^{l+1}, \ldots, \mathbf{h}_N^{l+1}\}$:

$$D(\mathbf{H}^l, \mathbf{H}^{l+1}) = \frac{1}{N}\sum_{n=1}^{N}(1 - \cos^2(\mathbf{h}_n^l, \mathbf{h}_n^{l+1})).$$

The cosine-squared distance between two vectors is maximized when two vectors are linearly independent and minimized when two vectors are linearly dependent, which satisfies our goal.[1]

4 Experiments

4.1 Setup

Dataset. To compare with the results reported by previous work (Gehring et al., 2017; Vaswani et al., 2017; Hassan et al., 2018), we conducted experiments on both Chinese⇒English (Zh⇒En) and English⇒German (En⇒De) translation tasks. For the Zh⇒En task, we used all of the available parallel data with maximum length limited to 50, consisting of about 20.62 million sentence pairs. We used newsdev2017 as the development set and newstest2017 as the test set. For the En⇒De task, we trained on the widely-used WMT14 dataset consisting of about 4.56 million sentence pairs. We used newstest2013 as the development set and newstest2014 as the test set. Byte-pair encoding (BPE) was employed to alleviate the Out-of-Vocabulary problem (Sennrich et al., 2016) with 32K merge operations for both language pairs. We used 4-gram NIST BLEU score (Papineni et al., 2002) as the evaluation metric, and *sign-test* (Collins et al., 2005) to test for statistical significance.

Models. We evaluated the proposed approaches on advanced Transformer model (Vaswani et al., 2017), and implemented on top of an open-source toolkit – THUMT (Zhang et al., 2017). We followed Vaswani et al. (2017) to set the configurations and train the models, and have reproduced their reported results on the En⇒De task. The parameters of the proposed models were initialized by the pre-trained model. We tried $k = 2$ and $k = 3$ for the multi-layer attention model, which allows to attend to the lower two or three layers.

We have tested both *Base* and *Big* models, which differ at hidden size (512 vs. 1024), filter size (2048 vs. 4096) and the number of attention heads (8 vs. 16).[2] All the models were trained on eight NVIDIA P40 GPUs where each was allocated with a batch size of 4096 tokens. In consideration of computation cost, we studied model variations with *Base* model on En⇒De task, and evaluated overall performance with *Big* model on both Zh⇒En and En⇒De tasks.

4.2 Results

Table 1 shows the results on WMT14 En⇒De translation task. As seen, the proposed approaches improve the translation quality in all cases, although there are still considerable differences among different variations.

Model Complexity Except for dense connection, all other deep representation strategies introduce new parameters, ranging from 14.7M to 33.6M. Accordingly, the training speed decreases due to more efforts to train the new parameters. Layer aggregation mechanisms only marginally decrease decoding speed, while multi-layer attention decreases decoding speed by 21% due to an additional attention process for each layer.

Layer Aggregation (Rows 2-5): Although dense connection and linear combination only marginally improve translation performance, iterative and hierarchical aggregation strategies achieve more significant improvements, which are up to +0.99 BLEU points better than the baseline model. This indicates that deep aggregations outperform their shallow counterparts by incorporating more depth and sharing, which is consistent with the results in computer vision tasks (Yu et al., 2018).

Multi-Layer Attention (Rows 6-7): Benefiting from the power of attention models, multi-layer attention model can also significantly outperform baseline, although it only attends to one or two additional layers. However, increasing the number of lower layers to be attended from $k = 2$ to

[1] We use cosine-squared distance instead of cosine distance, since the latter is maximized when two vectors are in opposite directions. In such case, the two vectors are in fact linearly dependent, while we aim at encouraging the vectors independent from each other.

[2] Here "filter size" refers to the hidden size of the feed-forward network in the Transformer model.

#	Model	# Para.	Train	Decode	BLEU	△
1	TRANSFORMER-BASE	88.0M	1.82	1.33	27.64	–
2	+ Dense Connection	+0.0M	1.69	1.28	27.94	+0.30
3	+ Linear Combination	+14.7M	1.59	1.26	28.09	+0.45
4	+ Iterative Aggregation	+31.5M	1.32	1.22	28.61	+0.97
5	+ Hierarchical Aggregation	+23.1M	1.46	1.25	28.63	+0.99
6	+ Multi-Layer Attention (k=2)	+33.6M	1.19	1.05	28.58	+0.94
7	+ Multi-Layer Attention (k=3)	+37.8M	1.12	1.00	28.62	+0.98
8	+ Iterative Aggregation + $\mathcal{L}_{diversity}$	+31.5M	1.28	1.22	28.76	+1.12
9	+ Hierarchical Aggregation + $\mathcal{L}_{diversity}$	+23.1M	1.41	1.25	28.78	+1.14
10	+ Multi-Layer Attention (k=2)+ $\mathcal{L}_{diversity}$	+33.6M	1.12	1.05	28.75	+1.11

Table 1: Evaluation of translation performance on WMT14 English⇒German ("En⇒De") translation task. "# Para." denotes the number of parameters, and "Train" and "Decode" respectively denote the training speed (steps/second) and decoding speed (sentences/second) on Tesla P40.

System	Architecture	En⇒De		Zh⇒En	
		# Para.	BLEU	# Para.	BLEU
Existing NMT systems					
(Wu et al., 2016)	RNN with 8 layers	N/A	26.30	N/A	N/A
(Gehring et al., 2017)	CNN with 15 layers	N/A	26.36	N/A	N/A
(Vaswani et al., 2017)	TRANSFORMER-BASE	65M	27.3	N/A	N/A
	TRANSFORMER-BIG	213M	28.4	N/A	N/A
(Hassan et al., 2018)	TRANSFORMER-BIG	N/A	N/A	N/A	24.2
Our NMT systems					
this work	TRANSFORMER-BASE	88M	27.64	108M	24.13
	+ Deep Representations	111M	28.78†	131M	24.76†
	TRANSFORMER-BIG	264M	28.58	304M	24.56
	+ Deep Representations	356M	29.21†	396M	25.10†

Table 2: Comparing with existing NMT systems on WMT14 English⇒German and WMT17 Chinese⇒English tasks. "+ Deep Representations" denotes "+ Hierarchical Aggregation + $\mathcal{L}_{diversity}$". "†" indicates statistically significant difference ($p < 0.01$) from the TRANSFORMER baseline.

$k = 3$ only gains marginal improvement, at the cost of slower training and decoding speeds. In the following experiments, we set set $k = 2$ for the multi-layer attention model.

Layer Diversity (Rows 8-10): The introduced diversity regularization consistently improves performance in all cases by encouraging different layers to capture diverse information. Our best model outperforms the vanilla Transformer by +1.14 BLEU points. In the following experiments, we used hierarchical aggregation with diversity regularization (Row 8) as the default strategy.

Main Results Table 2 lists the results on both WMT17 Zh⇒En and WMT14 En⇒De translation tasks. As seen, exploiting deep representations consistently improves translation performance across model variations and language pairs, demonstrating the effectiveness and universality of the proposed approach. It is worth mentioning that TRANSFORMER-BASE with deep representations exploitation outperforms the vanilla TRANSFORMER-BIG model, with only less than half of the parameters.

4.3 Analysis

We conducted extensive analysis from different perspectives to better understand our model. All results are reported on the En⇒De task with TRANSFORMER-BASE.

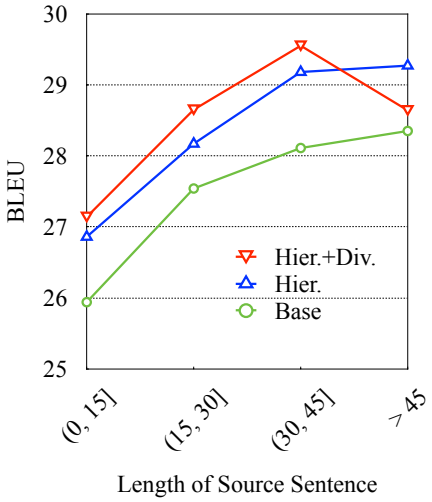


Figure 4: BLEU scores on the En⇒De test set with respect to various input sentence lengths. "Hier." denotes hierarchical aggregation and "Div." denotes diversity regularization.

4.3.1 Length Analysis

Following Bahdanau et al. (2015) and Tu et al. (2016), we grouped sentences of similar lengths together and computed the BLEU score for each group, as shown in Figure 4. Generally, the performance of TRANSFORMER-BASE goes up with the increase of input sentence lengths, which is superior to the performance of RNN-based NMT models on long sentences reported by (Bentivogli et al., 2016). We attribute this to the strength of self-attention mechanism to model global dependencies without regard to their distance.

Clearly, the proposed approaches outperform the baseline model in all length segments, while there are still considerable differences between the two variations. Hierarchical aggregation consistently outperforms the baseline model, and the improvement goes up on long sentences. One possible reason is that long sentences indeed require deep aggregation mechanisms. Introducing diversity regularization further improves performance on most sentences (e.g. ≤ 45), while the improvement degrades on long sentences (e.g. > 45). We conjecture that complex long sentences may need to store duplicate information across layers, which conflicts with the diversity objective.

4.3.2 Effect on Encoder and Decoder

Both encoder and decoder are composed of a stack of L layers, which may benefit from the proposed approach. In this experiment, we investigated how our models affect the two components, as shown in Table 3. Exploiting deep representations of encoder or decoder individually consistently outperforms the vanilla baseline model, and exploiting both components further improves the performance. These results provide support for the claim that exploiting deep representations is useful for both understanding input sequence and generating output sequence.

Model	Applied to		BLEU
	Encoder	*Decoder*	
BASE	N/A	N/A	26.13
OURS	✓	×	26.32
	×	✓	26.41
	✓	✓	**26.69**

Table 3: Experimental results of applying hierarchical aggregation to different components on En⇒De validation set.

4.3.3 Impact of Aggregation Choices

Model	RESIDUAL	AGGREGATE	BLEU
BASE	N/A	N/A	26.13
OURS	None	SIGMOID	25.48
	Top		26.59
	All		**26.69**
		RELU	26.56
		ATTENTION	26.54

Table 4: Impact of residual connections and aggregation functions for hierarchical layer aggregation.

As described in Section 3.1.1, the function of hierarchical layer aggregation is defined as

$$\text{AGG}(x, y, z) = \text{LN}(\text{FF}([x; y; z]) + x + y + z),$$

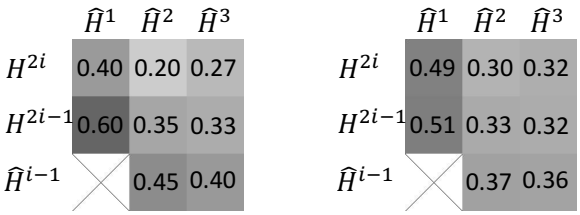
where $\text{FF}(\cdot)$ is a feed-forward network with a sigmoid activation in between. In addition, all the input layers $\{x, y, z\}$ have residual connections to the output. In this experiment, we evaluated the impact of residual connection options, as well as different choices for the aggregation function, as shown in Table 4.

Concerning residual connections, if none of the input layers are connected to the output layer ("None"), the performance would decrease. The translation performance is improved when the output is connected to only the top level of the input layers ("Top"), while connecting to all input layers ("All") achieves the best performance. This indi-

cates that cross-layer connections are necessary to avoid the gradient vanishing problem.

Besides the feed-forward network with sigmoid activation, we also tried two other aggregation functions for $\text{FF}(\cdot)$: (1) A feed-forward network with a RELU activation in between; and (2) multi-head self-attention layer that constitutes the encoder and decoder layers in the TRANSFORMER model. As seen, all the three functions consistently improve the translation performance, proving the robustness of the proposed approaches.

4.4 Visualization of Aggregation



(a) No regularization. (b) With regularization.

Figure 5: Visualization of the exploitation of input representations for hierarchical aggregation. x-axis is the aggregation node and y-axis is the input representation. $\widehat{H}^i$ denotes the i-th aggregation layer, and H^i denotes the i-th encoder layer. The rightmost and topmost position in x-axis and y-axis respectively represent the highest layer.

To investigate the impact of diversity regularization, we visualized the exploitation of the input representations for hierarchical aggregation in encoder side, as shown in Figure 5. Let $\mathbf{H}^i = \{H^{2i}, H^{2i-1}, \widehat{H}^{i-1}\}$ be the input representations, we calculated the exploitation of the j-th input as

$$s_j = \frac{\sum_{w \in W_j} |w|}{\sum_{H_{j'} \in \mathbf{H}} \{\sum_{w' \in W_{j'}} |w'|\}}, \qquad (11)$$

where W_j is the parameter matrix associated with the input H_j. The score s_j is a rough estimation of the contribution of H_j to the aggregation $\widehat{H}^i$.

We have two observations. First, the model tends to utilize the bottom layer more than the top one, indicating the necessity of fusing information across layers. Second, using the diversity regularization in Figure 5(b) can encourage each layer to contribute more equally to the aggregation. We hypothesize this is because of the diversity regularization term encouraging the different layers to contain diverse and equally important information.

5 Related Work

Representation learning is at the core of deep learning. Our work is inspired by technological advances in representation learning, specifically in the field of *deep representation learning* and *representation interpretation*.

Deep Representation Learning Deep neural networks have advanced the state of the art in various communities, such as computer vision and natural language processing. One key challenge of training deep networks lies in how to transform information across layers, especially when the network consists of hundreds of layers.

In response to this problem, ResNet (He et al., 2016) uses skip connections to combine layers by simple, one-step operations. Densely connected network (Huang et al., 2017) is designed to better propagate features and losses through skip connections that concatenate all the layers in stages. Yu et al. (2018) design structures iteratively and hierarchically merge the feature hierarchy to better fuse information in a deep fusion.

Concerning machine translation, Meng et al. (2016) and Zhou et al. (2016) have shown that deep networks with advanced connecting strategies outperform their shallow counterparts. Due to its simplicity and effectiveness, skip connection becomes a standard component of state-of-the-art NMT models (Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017). In this work, we prove that deep representation exploitation can further improve performance over simply using skip connections.

Representation Interpretation Several researchers have tried to visualize the representation of each layer to help better understand what information each layer captures (Zeiler and Fergus, 2014; Li et al., 2016; Ding et al., 2017). Concerning natural language processing tasks, Shi et al. (2016) find that both local and global source syntax are learned by the NMT encoder and different types of syntax are captured at different layers. Anastasopoulos and Chiang (2018) show that higher level layers are more representative than lower level layers. Peters et al. (2018) demonstrate that higher-level layers capture context-dependent aspects of word meaning while lower-level layers model aspects of syntax. Inspired by these observations, we propose to expose all of these representations to better

fuse information across layers. In addition, we introduce a regularization to encourage different layers to capture diverse information.

6 Conclusion

In this work, we propose to better exploit deep representations that are learned by multiple layers for neural machine translation. Specifically, the hierarchical aggregation with diversity regularization achieves the best performance by incorporating more depth and sharing across layers and by encouraging layers to capture different information. Experimental results on WMT14 English⇒German and WMT17 Chinese⇒English show that the proposed approach consistently outperforms the state-of-the-art TRANSFORMER baseline by +0.54 and +0.63 BLEU points, respectively. By visualizing the aggregation process, we find that our model indeed utilizes lower layers to effectively fuse the information across layers.

Future directions include validating our approach on other architectures such as RNN (Bahdanau et al., 2015) or CNN (Gehring et al., 2017) based NMT models, as well as combining with other advanced techniques (Shaw et al., 2018; Shen et al., 2018; Yang et al., 2018; Li et al., 2018) to further improve the performance of TRANSFORMER.

Acknowledgments

We thank the anonymous reviewers for their insightful comments.

References

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *NAACL*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *EMNLP*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, Goerge Foster, Llion Jones, Parmar Niki, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In *ACL*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *ACL*.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *ACL*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *CVPR*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. In *EMNLP*.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E Hopcroft. 2016. Convergent learning: Do different neural networks learn the same representations? In *ICLR*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Fandong Meng, Zhengdong Lu, Zhaopeng Tu, Hang Li, and Qun Liu. 2016. A deep memory-based architecture for sequence-to-sequence learning. In *ICLR Workshop*.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *NAACL*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: directional self-attention network for RNN/CNN-free language understanding. In *AAAI*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *EMNLP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *TACL*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *ACL*.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *EMNLP*.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. 2018. Deep layer aggregation. In *CVPR*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, and Yang Liu. 2017. THUMT: An Open Source Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1706.06415*.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *TACL*.