Rewiring the Transformer with Depth-Wise LSTMs

Hongfei Xu[1,2], Yang Song[1], Qiuhui Liu[3], Josef van Genabith[2], Deyi Xiong[4]
[1]Zhengzhou University, Henan, China
[2]DFKI and Saarland University, Informatics Campus, Saarland, Germany
[3]China Mobile Online Services, Henan, China
[4]College of Intelligence and Computing, Tianjin University, Tianjin, China
{hfxunlp, ysongnlp, liuqhano}@foxmail.com, josef.van_genabith@dfki.de, dyxiong@tju.edu.cn

Abstract

Stacking non-linear layers allows deep neural networks to model complicated functions, and including residual connections in Transformer layers is beneficial for convergence and performance. However, residual connections may make the model "forget" distant layers and fail to fuse information from previous layers effectively. Selectively managing the representation aggregation of Transformer layers may lead to better performance. In this paper, we present a Transformer with depth-wise LSTMs connecting cascading Transformer layers and sub-layers. We show that layer normalization and feed-forward computation within a Transformer layer can be absorbed into depth-wise LSTMs connecting pure Transformer attention layers. Our experiments with the 6-layer Transformer show significant BLEU improvements in both WMT 14 English-German / French tasks and the OPUS-100 many-to-many multilingual NMT task, and our deep Transformer experiments demonstrate the effectiveness of depth-wise LSTM on the convergence and performance of deep Transformers.

Keywords: Transformer, Depth-wise LSTM, Neural Machine Translation

1. Introduction

The multi-layer structure together with non-linear activation functions allow neural networks to model complicated functions. Increasing the depth of models can increase their capacity and benefit their performance if optimization difficulties (Mhaskar et al., 2017; Telgarsky, 2016; Eldan and Shamir, 2016; He et al., 2016; Bapna et al., 2018) can be properly addressed.

For machine translation, the performance of the Transformer translation model (Vaswani et al., 2017) benefits from including residual connections (He et al., 2016) in stacked layers and sub-layers (Bapna et al., 2018; Wu et al., 2019b; Wei et al., 2020; Zhang et al., 2019; Xu et al., 2020a; Li et al., 2020; Huang et al., 2020; Xiong et al., 2020; Mehta et al., 2021; Li et al., 2021; Xu et al., 2021d). However, the residual connections within each layer only fuse information through simple, one-step operations (Yu et al., 2018), which may make the model "forget" distant layers, and aggregating layers is of profound value to better fuse linguistic information at different levels of representation (Peters et al., 2018; Shen et al., 2018; Wang et al., 2018, 2019; Dou et al., 2018, 2019). Selectively aggregating different layer representations of the Transformer may further improve the performance.

In this paper, we propose to train Transformers with depth-wise LSTMs which regard outputs of stacked Transformer layers as steps in a time series and manage representation aggregation in and across layers. Our general motivation is that complex cross-layer information management offered by depth-wise LSTMs may bring about additional benefits over simple residual connections: LSTMs (Hochreiter and Schmidhuber, 1997) have been shown to (i) avoid gradient explosion and vanishing, (ii) selectively learn what to remember and what to forget while ensuring convergence.

We explore the use of LSTMs to connect layers in stacked deep architectures for Transformers: we show how residual connections can be replaced by LSTMs connecting self-, cross- and masked self-attention layers. In contrast to standard LSTMs that process token sequences, we refer to the use of LSTMs in connecting stacked layers of deep architectures as "depth-wise LSTMs".

Our contributions are as follows:

- We suggest that selectively aggregating different layer representations of the Transformer may improve the performance, and propose to use depth-wise LSTMs to connect stacked (sub-) layers of Transformers. We show how Transformer layer normalization and feed-forward sub-layers can be absorbed by depth-wise LSTMs, while connecting pure Transformer attention layers by depth-wise LSTMs (for Transformer encoder and decoder blocks), replacing residual connections.
- We show that the 6-layer Transformer using depth-wise LSTM can bring significant improvements in both WMT tasks and the challenging OPUS-100 multilingual NMT task. We

Corresponding author: Josef van Genabith.

LREC-COLING 2024, pages 14122–14133
20-25 May, 2024.

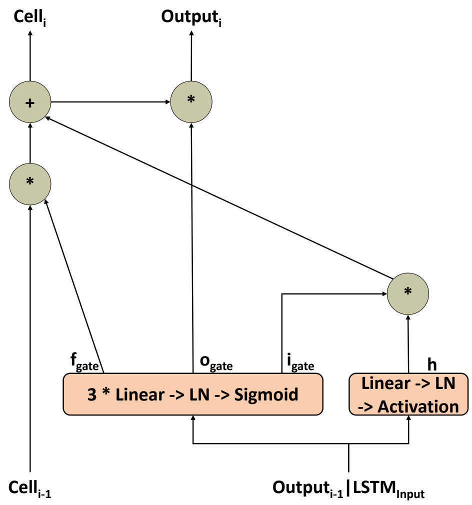

Figure 1: Depth-wise LSTM computation.

show that depth-wise LSTM also has the ability to support deep Transformers with up to 24 layers, and that the 12-layer Transformer using depth-wise LSTM already performs at the level of the 24-layer vanilla Transformer.

2. Transformer with Depth-Wise LSTM

2.1. Depth-Wise LSTM

The computation of depth-wise LSTM is the same as the conventional LSTM except that depth-wise LSTM connects stacked Transformer layers instead of tokens in a token sequence as in conventional LSTMs. The gate mechanisms in the original LSTM are to enhance its ability in capturing long-distance relations and to address the gradient vanishing/exploding issue in sequence modeling. In our work, we regard the outputs of stacked layers as a "vertical" sequence, and utilize the same gate mechanisms to selectively aggregate information from stacked Transformer layer outputs and to address the gradient vanishing issue of deep Transformers. LSTMs are able to capture long-distance relationships: they can learn to selectively use the representations of distant tokens in the processing of a current input token in a sequence. In a sense, the layer-by-layer computations in Transformer encoder and decoder stacks are just such sequences where information from a Transformer layer $n-1$ is passed on to layer n. Our depth-wise LSTMs connect layers of multi-head attention information instead of token embeddings. Because of the different types of attention (self, cross and masked), we develop tailored ways of connecting (sub-) layers in encoder stacks and decoder stacks with depth-wise LSTMs.

We equip our depth-wise LSTM with layer normalization. This has shown better performance as an LSTM-based NMT decoder (Chen et al., 2018; Xu et al., 2021b) than vanilla LSTM. The computation graph of our depth-wise LSTM is shown in Figure 1.

The depth-wise LSTM concatenates the input from the current Transformer layer $LSTM_{Input}$ to the LSTM with the output of the LSTM from the previous layer $Output_{i-1}$:

$$c = Output_{i-1}|LSTM_{Input} \tag{1}$$

where "|" indicates concatenation.

Next, the depth-wise LSTM computes three gates (input gate i_{gate}, forget gate f_{gate} and output gate o_{gate}) and the hidden representation h from the concatenated representation c:

$$i_{gate} = \sigma(\mathrm{LN}(W_i c + b_i)) \tag{2}$$

$$f_{gate} = \sigma(\mathrm{LN}(W_f c + b_f)) \tag{3}$$

$$o_{gate} = \sigma(\mathrm{LN}(W_o c + b_o)) \tag{4}$$

$$h = \mathrm{GeLU}(\mathrm{LN}(W_h c + b_h)) \tag{5}$$

where W_* and b_* are weight and bias parameters, σ is the sigmoid activation function, LN is the layer normalization.

We consider the role of the computation of the hidden state (Equation 5) similar to the position-wise feed-forward sub-layer in each of the original Transformer encoder and decoder layers, and remove the feed-forward sub-layer from the original encoder and decoder layers when we connect them by our depth-wise LSTMs. The original Transformer uses a 2-layer feed-forward network. In an additional set of experiments we model these two layers in the hidden state of the depth-wise LSTM in terms of two weight matrices W_{h1} and W_{h2} but use the GLU activation function (Shazeer, 2020) for parameter efficiency, as shown in Equation 6 (compare Equation 5):

$$h = W_{h2}\mathrm{GLU}(\mathrm{LN}(W_{h1} c + b_{h1})) + b_{h2} \tag{6}$$

After the computation of the hidden state, the cell state and the output of the LSTM unit are computed as:

$$Cell_i = Cell_{i-1} * f_{gate} + h * i_{gate} \tag{7}$$

$$Output_i = Cell_i * o_{gate} \tag{8}$$

where $*$ indicates element-wise multiplication.

As the depth-wise LSTM is computed across stacked Transformer layers and the token embeddings are already produced before computing the

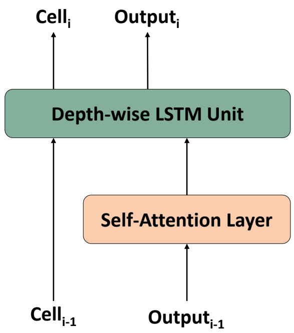

Figure 2: Encoder layer with depth-wise LSTM.

first encoder/decoder layer, we use the token embeddings as $Cell_0$ and $Output_0$.

The gate mechanisms (Equations 2, 3, 4) of the depth-wise LSTM can selectively learn to treat representations from different Transformer levels differently while guarding against vanishing and exploding gradients (Table 2).

We use depth-wise LSTM rather than a depth-wise multi-head attention network (Dou et al., 2018) with which we can build the NMT model solely based on the attention mechanism for two reasons: 1) we have to compute the stacking of Transformer layers sequentially as in sequential token-by-token decoding, and compared to the use of depth-wise LSTM of $O(n)$ complexity, depth-wise multi-head attention networks suffer from $O(n^2)$ complexity and they cannot be parallelized at the depth level. 2) the attention mechanism linearly combines representations with attention weights. Thus, it lacks the ability to provide the non-linearity compared to the LSTM, which we suggest is important.

2.2. Encoder Layers Connected via Depth-Wise LSTMs

Directly replacing residual connections with LSTM units will introduce a large amount of additional parameters and computation. Given that the task of computing the LSTM hidden state is similar to the feed-forward sub-layer in the original Transformer layers, we propose to replace the feed-forward sub-layer with the newly introduced LSTM unit, which only introduces one LSTM unit per layer, and the parameters of the LSTM can be shared across layers.

The original Transformer encoder layer only contains two sub-layers: the self-attention sub-layer based on the multi-head attention network and the 2-layer feed-forward network sub-layer.

The encoder layer with the depth-wise LSTM unit, as shown in Figure 2, first performs the self-attention computation, then the depth-wise LSTM unit takes the self-attention results and the output and the cell state of the previous layer to compute the output and the cell state of the current layer.

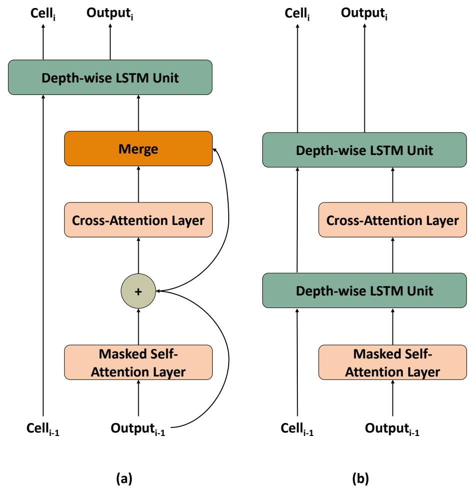

Figure 3: Decoder layer with depth-wise LSTM.

2.3. Decoder Layers Connected via Depth-Wise LSTMs

Different from encoder layers, decoder layers involve two multi-head attention sub-layers: a masked self-attention sub-layer to attend the decoding history and a cross-attention sub-layer to attend information from the source side. Given that the depth-wise LSTM unit only takes one input, we introduce a merging layer to merge the outputs of these two sub-layers into one as the input to the LSTM unit. The architecture is shown in Figure 3 (a).

Specifically, the decoder layer with depth-wise LSTM first computes the masked self-attention sub-layer and the cross-attention sub-layer as in the original decoder layer, then it merges the outputs of these two sub-layers and feeds the merged representation into the depth-wise LSTM unit which also takes the cell and the output of the previous layer to compute the output of the current decoder layer and the cell state of the LSTM. We examine both element-wise addition and concatenation as merging operation.

Another way to take care of the outputs of these two sub-layers in the decoder layer is to replace their residual connections with two depth-wise LSTM sub-layers, as shown in Figure 3 (b). This leads to better performance (as shown in Table 4), but at the costs of more parameters and decoder depth in terms of sub-layers.

3. Experiments

We implemented our approach based on the Neutron implementation of the Transformer (Xu and Liu, 2019). To show the effects of depth-wise LSTMs on the 6-layer Transformer, we first conducted experiments on the WMT 14 English to German and English to French news translation tasks to compare with the Transformer baseline Vaswani et al. (2017). Additionally, we also examined the impact of our approach on deep Transformers and in a multilingual NMT task. The deep Transformer experiments were conducted on the WMT 14 English to German task and the WMT 15 Czech to English task following Bapna et al. (2018); Xu et al. (2020a), and the multilingual NMT experiments were performed on the challenging OPUS-100 dataset following Zhang et al. (2020). The concatenation of newstest 2012 and newstest 2013 was used for validation and newstest 2014 as test set for the WMT 14 English to German and English to French news translation tasks, and newstest 2013 as validation set for the WMT 15 Czech to English task. Newstest 2014 provided the test sets for both the WMT 14 English to German and the English to French task, and newstest 2015 was the test set for the Czech to English task.

3.1. Settings

We applied joint Byte-Pair Encoding (Sennrich et al., 2016) with $32k$ merging operations on all data sets to address the unknown word issue. We only kept sentences with a maximum of 256 subword tokens for training. For fair comparison, we did not tune any hyperparameters but followed Vaswani et al. (2017) for all experiment settings.

Though Zhang et al. (2019); Xu et al. (2020b) suggest using a large batch size which may lead to improved performance, we only used a batch size of $25k$ target tokens (through gradient accumulation of small batches) to fairly compare with previous work (Vaswani et al., 2017; Xu et al., 2020a).

We used a beam size of 4 for decoding, and evaluated tokenized case-sensitive BLEU with the averaged model of the last 5 checkpoints for the Transformer Base setting and 20 checkpoints for the Transformer Big setting saved at intervals of $1,500$ training steps. We also conducted significance tests (Koehn, 2004). To measure the efficiency of different settings, we sorted the WMT 14 En-De test set of 3003 sentences by the number of tokens of the input sentence to reduce the number of padding tokens during batching, and tested the inference speed on a single RTX 4090 GPU (matrix multiplications were computed in FP16 precision for faster decoding), and reported the beam decoding speed (number of sentences per second).

3.2. Main Results

We first examine the effects of our approach on the 6-layer Transformer on the WMT 14 English-German and English-French task to compare with Vaswani et al. (2017), and results are shown in Table 1.

Models	En-De	En-Fr
Transformer Base	27.55	39.54
with depth-wise LSTM	**28.53**†	**40.10**†
Transformer Big	28.83	41.92
with depth-wise LSTM	**29.58**†	**43.11**†

Table 1: Results on WMT 14 En-De and En-Fr. † indicates $p < 0.01$ in the significance test.

Approaches	BLEU	Para.(M)	Speed
Transformer	27.55	62.37	750.58
Depth-wise RNN	23.24	68.67	737.60
Depth-wise LSTM	**28.53**	70.25	674.96

Table 2: Ablation study of depth-wise approaches on WMT 14 En-De.

In our approach ("with depth-wise LSTM"), we used the 2-layer neural network for the computation of the LSTM hidden state (Equation 6) and shared LSTM parameters across stacked encoder layers and different shared parameters across decoder layers for computing the LSTM gates (Equations 2, 3, 4). Details are provided in our ablation study.

Table 1 shows that our approach based on the depth-wise LSTM can obtain significant improvements on both tasks over the original Transformer with both the Transformer Base setting and the Transformer Big setting. In particular, significant improvements ($+1.19$ BLEU) obtained by our approach on the En-Fr task (trained on $\sim$36M sentence pairs) with the Transformer Big support the effectiveness of our approach in large-scale and challenging settings.

Our approach with the Transformer base setting brings about more improvements on the English-German task than that on the English-French task. We conjecture that maybe because the performance on the English-French task using a large dataset ($\sim$36M sentence pairs) may rely more on the capacity of the model (i.e. the number of parameters) than on the complexity of the modeling function (i.e. depth of the model, non-linearity strength per-layer, etc.). With the Transformer Big model which contains more parameters than the Transformer Base, the improvement on En-Fr ($+1.19$) is larger than that on En-De ($+0.75$), with $\sim$4.5M sentence pairs.

3.3. Ablation Study

We conducted ablation studies on the WMT 14 En-De task with the Base setting.

Considering that the layer stacks of the 6-layer Transformer are not that deep and vanilla RNNs

LSTM FFN	Hidden size	BLEU	Para.(M)	Speed
1-layer (Eq. 5)	512	27.84	45.05	742.19
2-layer (Eq. 6)	2048	**28.53**	70.25	674.96
	1586	28.20	62.37	683.67

Table 3: Ablation study of LSTM hidden computation on WMT 14 En-De.

Merging	BLEU	Para.(M)	Speed
Concat	28.26	78.90	649.27
Add	28.53	70.25	674.96
2 Depth-wise LSTMs	**28.81**	100.18	581.13

Table 4: Results of merging operations for decoder layer on WMT 14 En-De.

Sharing	BLEU	Para.(M)
All	26.94	44.00
Gate	**28.53**	70.25
None	28.25	87.59

Table 5: Results of sharing LSTM parameters on WMT 14 En-De.

can play a similar role as LSTMs, is it possible to train the model with a depth-wise RNN rather than the depth-wise LSTM? We first study using different approaches (Transformer, the depth-wise RNN and the depth-wise LSTM) for the 6-layer Transformer, and results are shown in Table 2.

When using the depth-wise RNN, the architecture is quite similar to the standard Transformer layer without residual connections but using the concatenation of the input to the encoder/decoder layer with the output(s) of attention layer(s) as the input to the last FFN sub-layer. Table 2 shows that the 6-layer Transformer with the depth-wise RNN is able to converge, but its performance is much worse than the model with the depth-wise LSTM (and also much worse than the vanilla Transformer) with depth-wise LSTM outperforming the vanilla Transformer, suggesting the importance of the gating mechanisms of the depth-wise LSTM. The decoding speed of our baseline vanilla Transformer implementation (750.58 sentences/s) is quite fast, and is 1.12 times as fast as the depth-wise LSTM approach, but our approach leads to a higher BLEU score than the baseline, and as shown in Table 6, our approach indeed requires fewer parameters and brings about faster decoding speed than the vanilla Transformer for a comparable BLEU score.

Next, we study the effects of two types of computations for the LSTM hidden state in Equations 5 and 6 on the performance on the WMT 14 En-De task. Results are shown in Table 3.

Table 3 shows that a 2-layer feed-forward neural network (Equation 6) in the depth-wise LSTM outperforms the original computation of the LSTM hidden state which uses only one layer (Equation 5), which is consistent with intuition. However, even with only one layer for the hidden state computation and with 27.77% fewer parameters (45.05M against 62.37M), depth-wise LSTM (Equation 5) still slightly outperforms the vanilla Transformer baseline in BLEU (27.84 against 27.55), suggesting that the improvements from using depth-wise LSTMs are not just due to the increased amount of parameters. The 1-layer LSTM FFN model also archieves a comparable decoding speed compared to the baseline (742.19 v.s. 750.58). When we reduce the hidden dimension of Equation 6 to 1586, which results in approximately the same number of parameters as the standard Transformer, depth-wise LSTM still outperforms the baseline by $+0.65$ BLEU.

We also study the merging operations, concatenation, element-wise addition, and the use of 2 depth-wise LSTM sub-layers, to combine the masked self-attention sub-layer output and the cross-attention sub-layer output in decoder layers. Results are shown in Table 4.

Table 4 shows that, even though this is counter-intuitive, element-wise addition (with fewer parameters) empirically results in slightly higher BLEU than the concatenation operation. Furthermore, even though using 2 depth-wise LSTM sub-layers connecting cross- and masked self-attention sub-layers leads to the highest BLEU score, showing the advantage of fully replacing residual connections with depth-wise LSTMs, it also introduces more parameters and increases the decoder depth in terms of sub-layers. For fair comparison, we use the simpler element-wise addition operation in our experiments by default.

As the number of Transformer layers is pre-specified, the parameters of the depth-wise LSTM can either be shared across layers or be independent. Table 3 documents the importance of the capacity of the module for the hidden state computation, and sharing the module is likely to hurt its capacity. We additionally study to share only parameters for gate computation (Equations 2, 3, 4) and to share all parameters (i.e. parameters for both the computation of gates and of the hidden state). Results are shown in Table 5.

Table 5 shows that: 1) Sharing parameters for the computation (Equation 6) of the depth-wise LSTM

Models	Layers		En-De	Cs-En	Para.(M)	Speed
	Encoder	Decoder				
Transformer Base						
TA (Bapna et al., 2018)*	16		28.39	29.36	93.87	711.78
DLCL (Wang et al., 2019)	30	6	29.3		137.97	577.30
ODE (Li et al., 2022a)	24		30.29		119.17	565.86
Layer Aggregation (Dou et al., 2018)	6		28.63		111.10	667.57
EM Routing (Dou et al., 2019)	6		28.81	None	144.80	561.28
SDU (Chai et al., 2020)*	6		28.22		78.13	664.20
Luna (Ma et al., 2021)	6		27.8		77.60	None
DSI (Zhang et al., 2019)	20		28.67		149.54	298.50
LCPI (Xu et al., 2020a)	24		29.20	29.88	194.66	229.90
Transformer Big						
Layer Aggregation (Dou et al., 2018)	6		29.21		356.38	264.55
EM Routing (Dou et al., 2019)	6		28.97	None	490.38	221.70
MC (Wei et al., 2020)	18		30.56		798.23	70.37
ODE (Li et al., 2022a)	12	6	30.77		288.46	315.91
Transformer Base	3		26.36	27.91	40.33	1209.62
	6		27.55	28.40	62.37	750.58
	12		28.12	29.38	106.47	429.00
	18		28.60	29.61	150.57	299.81
	24		29.02	29.73	194.66	229.90
Transformer Base with depth-wise LSTM	3		27.38	28.26	46.63	1121.16
	6		28.53	29.15	70.25	674.96
	12		29.26	29.64	122.23	379.83
	18		**29.41**	**30.27**	172.63	277.21
	24		29.18	30.02	223.02	202.40
Transformer Big with depth-wise LSTM	12		**30.69**	**30.57**	452.04	181.58
+ experiment settings of Li et al. (2022a)	12	6	**31.12**	**31.25**	338.75	316.15
+ 1-layer LSTM FFN (Eq. 5)			30.83	30.96	288.41	363.60

Table 6: Results of Deep Transformers. "*" indicates reproduction of the approach.

hidden state significantly hampers performance, which is consistent with our conjecture. 2) Sharing parameters for the computation of gates (Equations 2, 3, 4) leads to slightly higher BLEU with fewer parameters introduced than without sharing them ("None" in Table 5). Thus, in the other experiments, we bind parameters for the computation of LSTM gates across stacked layers by default.

3.4. Deep Transformers

We examine whether depth-wise LSTM has the ability to ensure the convergence of deep Transformers and measure performance on the WMT 14 English to German task and the WMT 15 Czech to English task following Bapna et al. (2018); Xu et al. (2020a), and compare our approach with the pre-norm Transformer in which residual connections are not normalized by layer normalization. To compare with the previous studies, we replace the English to French task with the Czech to English task with ~ 15M sentence pairs. The 4.5M dataset of the En-De task is not small, and the Cs-En data that has more than 15M sentence pairs is even larger, and can be considered a large-scale dataset. Together with the English-French experiment (Table 1), this allows us to assess the effectiveness of our approach with large datasets and deep Transformers. For fairness and reliable comparisons across all our experiments, we strictly followed the experiment settings of Vaswani et al. (2017) by default, without using relative positional encoding (Shaw et al., 2018), dense connections, larger number of warm up steps, and larger batch sizes, although several previous studies (Wang et al., 2019; Zhang et al., 2019; Li et al., 2020, 2022a) employ some or all of these different settings for higher BLEU scores. Results are shown in Table 6.

Table 6 shows that though the BLEU improvements start saturating with deep depth-wise LSTM Transformers of more than 12 layers, depth-wise LSTM is able to ensure convergence of up to 24 layer Transformers. The experiments also show that the size differences between these datasets did not lead to differences in optimization.

Notably, on the En-De task, the 12-layer Transformer with depth-wise LSTM already outperforms the 24-layer vanilla Transformer, suggesting efficient use of layer parameters. On the Cs-En task, the 12-layer model with depth-wise LSTM performs on a par with the 24-layer baseline. Unlike in the En-De task, increasing depth over the 12-layer Transformer can still achieve some BLEU improvements, with the 18-layer model resulting in the best performance. We conjecture that this is probably because the data set of the Cs-En task ($\sim$15M) is larger than that of the En-De task ($\sim$4.5M), and increasing the depth of the model for the Cs-En task also increases its number of parameters and capacity. For the En-De task, the 12-layer Transformer with depth-wise LSTM may already provide both sufficient complexity and capacity for the data set.

It is a common problem that increasing the depth does not always lead to better performance, whether with residual connections (Li et al., 2022b) or other previous studies on deep Transformers (Bapna et al., 2018; Wang et al., 2019; Li et al., 2022a), and the use of wider models is the usual method of choice for further improvements. Although for the Base Transformer model our approach does not lead to significant improvements for models deeper than 18 layers, we argue that the 18-layer Transformer Base is not the performance limit of our approach, because we may increase the width of the model in addition to the depth. As shown in Table 6, the 12-layer Transformer Big with depth-wise LSTM is able to achieve further improvements over Transformer Base models, and using fewer layers and parameters achieves performance on par with Wei et al. (2020). Using relative positional encoding, larger batch sizes, etc., following the experiment settings of Li et al. (2022a) can also lead to better performance with our approach.

As for the costs, the decoder depth has a strong impact on inference speed, as the decoder has to be computed once for each decoding step during auto-regressive decoding (Kasai et al., 2021; Xu et al., 2021c), and the use of only deep encoders (Bapna et al., 2018; Wang et al., 2019; Li et al., 2022a; Chai et al., 2020) normally leads to faster inference speed than using both a deep encoder and a deep decoder. But in general, Table 6 shows that our approach uses fewer parameters and leads to faster decoding speed than the baselines to obtain a comparable BLEU score, showing the efficiency of our method.

3.5. Multilingual NMT

Multilingual translation uses a single model to translate between multiple language pairs (Firat et al., 2016; Johnson et al., 2017; Aharoni et al., 2019). Model capacity has been found crucial for massively multilingual NMT to support language pairs with varying typological characteristics (Zhang et al., 2020; Xu et al., 2021a). Using model layers efficiently with depth-wise LSTMs is likely to benefit multilingual NMT.

To test the effectiveness of depth-wise LSTMs in the multilingual setting, we conducted experiments on the challenging massively many-to-many translation task on the OPUS-100 corpus (Tiedemann, 2012; Aharoni et al., 2019; Zhang et al., 2020). We tested the performance of 6-layer models following the experiment settings of Zhang et al. (2020) for fair comparison. We adopted BLEU (Papineni et al., 2002) for translation evaluation with the SacreBLEU toolkit (Post, 2018). [1] We report average BLEU over 94 language pairs BLEU$_{94}$, win ratio WR (%) compared to Zhang et al. (2020), average BLEU over 4 selected typologically different target languages with varied training data sizes (de, zh, br, te) BLEU$_4$. Results are shown in Table 7.

Compared to the baseline (Zhang et al., 2020), Table 7 shows that: 1) our approach can lead to $+3.02$ and $+3.38$ BLEU improvements on average in the En$\rightarrow$xx and xx$\rightarrow$En directions respectively in the evaluation over 4 typologically different languages, and 2) using depth-wise LSTM is able to bring about $+2.57$ and $+1.19$ BLEU improvements on average when translating English to 94 languages and translating them into English respectively. Given that the one-to-many translation task requires more model capacity than the many-to-one translation task (Arivazhagan et al., 2019), the larger average BLEU improvements and a higher win ratio of 98.94% (93 of 94 languages) in the En$\rightarrow$xx direction than in the xx$\rightarrow$En direction demonstrate the effectiveness of our approach especially when model capacity is crucial, suggesting the more effective use of model parameters with depth-wise LSTMs than vanilla Transformer.

3.6. Efficiency Discussion

Despite the depth-wise LSTM Transformers having more non-linear operations than the standard Transformer, we suggest that it is more efficient.

In our deep Transformer experiments, Table 6 shows that our depth-wise LSTM Transformer with fewer layers, parameters and computations can lead to competitive/better performance and faster decoding speed than vanilla Transformers with more layers but a similar BLEU score, and the depth-wise LSTM Transformer is in fact more efficient as we need fewer layers to achieve comparable performance.

In the multilingual NMT task which relies heavily on the model capacity, Table 7 shows that the use of depth-wise LSTM can bring about $+2.52$ BLEU improvements on average when translating English to 94 languages.

In Table 3, we reduce the 2-layer FFN of the Transformer with depth-wise LSTM to only one layer

[1] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.4.1

Models	Direction	BLEU_{94}	WR	BLEU_4
Transformer	En→xx	18.75	-	14.73
	xx→En	27.02		22.50
Transformer + LALN + LALT (Zhang et al., 2020)	En→xx	20.81	-	17.45
	xx→En	27.22		23.30
Depth-wise LSTM	En→xx	**23.38**	98.94	**20.47**
	xx→En	**28.41**	79.79	**26.68**

Table 7: Results of multilingual NMT.

with significantly fewer hidden units ($2048 \rightarrow 512$), this saves a large number of parameters and computations, and our approach with 45.05M parameters still slightly outperforms the baseline residual Transformer with 62.37M parameters (Table 2).

Our depth-wise LSTM Transformer Base performs on a par with the previous layer aggregation work (Dou et al., 2018) on the WMT 14 En-De task. However, our model only contains 70.25M parameters while Dou et al. (2018) involves 111M parameters.

4. Related Work

He et al. (2016) present the residual learning framework to ease the training of deep neural networks. Srivastava et al. (2015) propose the highway network which contains a transform gate and a carry gate to control the produced output and the input. Chai et al. (2020) propose a highway Transformer with a self-gating mechanism for language models. However, our work is significantly different from theirs in two aspects. First, residual connections are still kept in their model. Second, their architecture does not use any mechanisms to track long-distance dependencies between stacked layers compared to depth-wise LSTM in our work.

Layer Aggregation Yu et al. (2018) suggest that skip connections are "shallow" themselves, and only fuse by simple, one-step operations, and therefore Yu et al. (2018) augment standard architectures with deeper aggregation to better fuse information across layers to improve recognition and resolution. Shen et al. (2018) propose a densely connected NMT architecture to create new features with dense connections. Wang et al. (2018) propose a multi-layer representation fusion approach to learning a better representation from the layer stack. Dou et al. (2018) simultaneously expose all layer representations with layer aggregation. Dou et al. (2019) propose to use routing-by-agreement strategies to aggregate layers dynamically.

Deep NMT Zhou et al. (2016) introduce fast-forward connections and an interleaved bi-directional architecture for stacking LSTM layers. Wang et al. (2017) propose a Linear Associative Unit to reduce the gradient propagation path inside the recurrent unit.

Deep Transformers For the convergence of deep Transformers, Bapna et al. (2018) propose the Transparent Attention mechanism which allows each decoder layer to attend weighted combinations of all encoder layer outputs. Wang et al. (2019) present the Dynamic Linear Combination of Layers approach that additionally aggregates shallow layers' outputs for each encoder layer. Wu et al. (2019b) propose a two-stage approach. Wei et al. (2020) introduce a depth-wise GRU to additionally aggregate outputs of all encoder layers for the top decoder layer, but residual connections are still kept. Zhang et al. (2019) and Xu et al. (2020a) propose the layer-wise Depth-Scaled Initialization approach and the Lipschitz constrained parameter initialization approach, respectively, to reduce the standard deviation of layer normalization inputs and to ensure the functionality of residual connection. Kasai et al. (2021); Xu et al. (2021c) propose to accelerate decoding by using deep encoders and shallower decoders. Li et al. (2022a) design an ODE Transformer which is analogous to the Runge-Kutta method. Hao et al. (2022) present approaches to exploring hyperparameters of deep Transformers for low-resource NMT with shallow Transformers.

Regarding parameter efficiency for NMT, Wu et al. (2019a) present lightweight and dynamic convolutions. Ma et al. (2021) approximate softmax attention with two nested linear attention functions. These methods are orthogonal to our work and it should be possible to combine them with our approach.

5. Conclusion

In this paper, we replace residual connections of the Transformer with depth-wise LSTMs, to selectively manage the representation aggregation of layers benefiting performance while ensuring convergence of the Transformer. Specifically, we show

how to integrate the computation of multi-head attention networks and feed-forward networks with the depth-wise LSTM for the Transformer.

Our experiments with the 6-layer Transformer show that our approach using depth-wise LSTM can achieve significant BLEU improvements in both WMT news translation tasks and the very challenging OPUS-100 many-to-many multilingual translation task over baselines. Our deep Transformer experiments demonstrate that: 1) the depth-wise LSTM approach ensures that deep Transformers with up to 24 layers converge, 2) the 12-layer Transformer using depth-wise LSTM already performs on a par with the 24-layer vanilla Transformer, suggesting more efficient usage of per-layer parameters with our depth-wise LSTM approach than the baseline.

6. Acknowledgements

We thank anonymous reviewers for their insightful comments. Hongfei Xu and Yang Song acknowledge the support of the National Natural Science Foundation of China (Grant No. 62306284), China Postdoctoral Science Foundation (Grant No. 2023M743189), and the Natural Science Foundation of Henan Province (Grant No. 232300421386). Josef van Genabith and Hongfei Xu are supported by the German Federal Ministry of Education and Research (BMBF) under funding code 01IW20010 (CORA4NLP). Deyi Xiong is partially supported by the Key Research and Development Program of Yunnan Province (No. 202203AA080004).

7. Bibliographical References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033. Association for Computational Linguistics.

Yekun Chai, Shuo Jin, and Xinwen Hou. 2020. Highway transformer: Self-gating enhanced self-attentive networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6887–6900, Online. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262, Brussels, Belgium. Association for Computational Linguistics.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Longyue Wang, Shuming Shi, and Tong Zhang. 2019. Dynamic layer aggregation for neural machine translation with routing-by-agreement. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 86–93.

Ronen Eldan and Ohad Shamir. 2016. The power of depth for feedforward neural networks. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 907–940, Columbia University, New York, New York, USA. PMLR.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Wenjie Hao, Hongfei Xu, Lingling Mu, and Hongying Zan. 2022. Optimizing deep transformers for chinese-thai low-resource translation. In *Machine Translation*, pages 117–126, Singapore. Springer Nature Singapore.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. 2020. Improving transformer optimization through better initialization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483. PMLR.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Bei Li, Quan Du, Tao Zhou, Yi Jing, Shuhan Zhou, Xin Zeng, Tong Xiao, JingBo Zhu, Xuebo Liu, and Min Zhang. 2022a. ODE transformer: An ordinary differential equation-inspired model for sequence generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8335–8351, Dublin, Ireland. Association for Computational Linguistics.

Bei Li, Ziyang Wang, Hui Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2021. Learning light-weight translation models from deep transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13217–13225.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 995–1005, Online. Association for Computational Linguistics.

Zuchao Li, Yiran Wang, Masao Utiyama, Eiichiro Sumita, Hai Zhao, and Taro Watanabe. 2022b. What works and doesn't work, a deep decoder for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 459–471, Dublin, Ireland. Association for Computational Linguistics.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. 2021. Luna: Linear unified nested attention. In *Advances in Neural Information Processing Systems*, volume 34, pages 2441–2453. Curran Associates, Inc.

Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Delight: Deep and light-weight transformer. In *International Conference on Learning Representations*.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. 2017. When and why are deep networks better than shallow ones? In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2343–2348.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

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

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Noam Shazeer. 2020. GLU variants improve transformer. *CoRR*, abs/2002.05202.

Yanyao Shen, Xu Tan, Di He, Tao Qin, and Tie-Yan Liu. 2018. Dense information flow for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1294–1303, New Orleans, Louisiana. Association for Computational Linguistics.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Matus Telgarsky. 2016. benefits of depth in neural networks. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539, Columbia University, New York, New York, USA. PMLR.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017. Deep neural machine translation with linear associative unit. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Vancouver, Canada. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Qiang Wang, Fuxue Li, Tong Xiao, Yanyang Li, Yinqiao Li, and Jingbo Zhu. 2018. Multi-layer representation fusion for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3015–3026, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xiangpeng Wei, Heng Yu, Yue Hu, Yue Zhang, Rongxiang Weng, and Weihua Luo. 2020. Multi-scale collaborative deep models for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 414–426, Online. Association for Computational Linguistics.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019a. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019b. Depth growing for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5558–5563, Florence, Italy. Association for Computational Linguistics.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR.

Hongfei Xu and Qiuhui Liu. 2019. Neutron: An Implementation of the Transformer Translation Model and its Variants. *arXiv preprint arXiv:1903.07402*.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, and Deyi Xiong. 2021a. Modeling task-aware MIMO cardinality for efficient multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 361–367, Online. Association for Computational Linguistics.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. 2020a. Lipschitz constrained parameter initialization for deep transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 397–402, Online. Association for Computational Linguistics.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Meng Zhang. 2021b. Multi-head highly parallelized LSTM decoder for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 273–282, Online. Association for Computational Linguistics.

Hongfei Xu, Josef van Genabith, Qiuhui Liu, and Deyi Xiong. 2021c. Probing word translations in the transformer and trading decoder for encoder layers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–85, Online. Association for Computational Linguistics.

Hongfei Xu, Josef van Genabith, Deyi Xiong, and Qiuhui Liu. 2020b. Dynamically adjusting transformer batch size by monitoring gradient direction change. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3519–3524, Online. Association for Computational Linguistics.

Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince, and Yanshuai Cao. 2021d. Optimizing deeper transformers on small datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2089–2102, Online. Association for Computational Linguistics.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. 2018. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383.