

# Very Deep Transformers for Neural Machine Translation

Xiaodong Liu<sup>†</sup>, Kevin Duh<sup>‡</sup>, Liyuan Liu<sup>§</sup> and Jianfeng Gao<sup>†</sup>

<sup>†</sup> Microsoft Research    <sup>‡</sup> Johns Hopkins University

<sup>§</sup> University of Illinois at Urbana-Champaign

{xiaodl, jfgao}@microsoft.com

kevinduh@cs.jhu.edu, ll2hanj@illinois.edu

## Abstract

We explore the application of very deep Transformer models for Neural Machine Translation (NMT). Using a simple yet effective initialization technique that stabilizes training, we show that it is feasible to build standard Transformer-based models with up to 60 encoder layers and 12 decoder layers. These deep models outperform their baseline 6-layer counterparts by as much as 2.5 BLEU, and achieve new state-of-the-art benchmark results on WMT14 English-French (43.8 BLEU) and WMT14 English-German (30.1 BLEU). The code and trained models will be publicly available at: <https://github.com/namisan/exdeep-nmt>.

## 1 Introduction

The capacity of a neural network influences its ability to model complex functions. In particular, it has been argued that *deeper* models are conducive to more expressive features (Bengio, 2009). Very deep neural network models have proved successful in computer vision (He et al., 2016; Srivastava et al., 2015) and text classification (Conneau et al., 2017; Minaee et al., 2020). In neural machine translation (NMT), however, current state-of-the-art models such as the Transformer typically employ only 6-12 layers (Bawden et al., 2019; Junczys-Dowmunt, 2019; Ng et al., 2019).

Previous work has shown that it is difficult to train deep Transformers, such as those over 12 layers (Bapna et al., 2018). This is due to optimization challenges: the variance of the output at each layer compounds as they get deeper, leading to unstable gradients and ultimately diverged training runs.

In this empirical study, we re-investigate whether deeper Transformer models are useful for NMT. We apply a recent initialization technique called ADMIN (Liu et al., 2020a), which remedies the variance problem. This enables us train Transformers

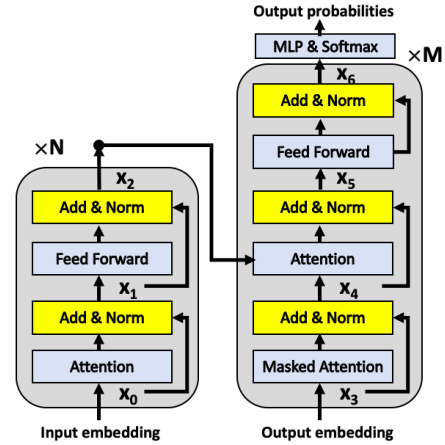


Figure 1: Transformer model

that are significantly deeper, e.g. with 60 encoder layers and 12 decoder layers.<sup>1</sup>

In contrast to previous research, we show that it is indeed feasible to train the standard<sup>2</sup> Transformer (Vaswani et al., 2017) with many layers. These deep models significantly outperform their 6-layer baseline, with up to 2.5 BLEU improvement. Further, they obtain state-of-the-art on the WMT'14 EN-FR and WMT'14 EN-DE benchmarks.

## 2 Background

We focus on the Transformer model (Vaswani et al., 2017), shown in Figure 1. The encoder consists of  $N$  layers/blocks of attention + feed-forward components. The decoder consists of  $M$  layers/blocks of masked-attention, attention, and feed-forward components. To illustrate, the input tensor  $x_{i-1}$  at the encoder is first transformed

<sup>1</sup>We choose to focus on this layer size since it results in the maximum model size that can fit within a single GPU system. The purpose of this study is to show that it is feasible for most researchers to experiment with very deep models; access to massive GPU budgets is not a requirement.

<sup>2</sup>Note there are architectural variants that enable deeper models (Wang et al., 2019; Nguyen and Salazar, 2019), discussed in Sec 2. We focus on the standard architecture here.

by a multi-head attention mechanism to generate the tensor  $f_{ATT}(\mathbf{x}_{i-1})$ . This result is added back with  $\mathbf{x}_{i-1}$  as a residual connection, then layer-normalization ( $f_{LN}(\cdot)$ ) is applied to generate the output:  $\mathbf{x}_i = f_{LN}(\mathbf{x}_{i-1} + f_{ATT}(\mathbf{x}_{i-1}))$ . Continuing onto the next component,  $\mathbf{x}_i$  is passed through a feed-forward network  $f_{FF}(\cdot)$ , and is again added and layer-normalized to generate the output tensor:  $\mathbf{x}_{i+1} = f_{LN}(\mathbf{x}_i + f_{FF}(\mathbf{x}_i))$ . Abstractly, the output tensor at each Add+Norm component in the Transformer (Figure 1) can be expressed as:

$$\mathbf{x}_i = f_{LN}(\mathbf{x}_{i-1} + f_i(\mathbf{x}_{i-1})) \quad (1)$$

where  $f_i$  represents a attention, masked-attention, or feed-forward subnetwork. This process repeats  $2 \times N$  times for a  $N$ -layer encoder and  $3 \times M$  times for a  $M$ -layer decoder. The final output of the decoder is passed through a softmax layer which predicts the probabilities of output words, and the entire network is optimized via back-propagation.

Optimization difficulty has been attributed to vanishing gradient, despite layer normalization (Xu et al., 2019) providing some mitigation. The lack of gradient flow between the decoder and the *lower* layers of the encoder is especially problematic; this can be addressed with short-cut connections (Bapna et al., 2018; He et al., 2018). An orthogonal solution is to swap the positions of layerwise normalization  $f_{LN}$  and subnetworks  $f_i$  within each block (Nguyen and Salazar, 2019; Domhan, 2018; Chen et al., 2018) by:  $\mathbf{x}_i = f_i(\mathbf{x}_{i-1} + f_{LN}(\mathbf{x}_{i-1}))$ . This is known as pre-LN (contrasted with post-LN in Eq. 1), and has been effective in training networks up to 30 layers (Wang et al., 2019).<sup>3</sup>

However, it has been shown that post-LN, if trained well, can outperform pre-LN (Liu et al., 2020a). Ideally, we hope to train a standard Transformer without additional architecture modifications. In this sense, our motivation is similar to that of Wu et al. (2019b), which grows the depth of a standard Transformer in a stage-wise fashion.

### 3 Initialization Technique

The initialization technique ADMIN (Liu et al., 2020a) we will apply here reformulates Eq. 1 as:

$$\mathbf{x}_i = f_{LN}(\mathbf{x}_{i-1} \cdot \omega_i + f_i(\mathbf{x}_{i-1})) \quad (2)$$

where  $\omega_i$  is a constant vector that is element-wise multiplied to  $\mathbf{x}_{i-1}$  in order to balance the contribution against  $f_i(\mathbf{x}_{i-1})$ . The observation is that in

addition to vanishing gradients, the unequal magnitudes in the two terms  $\mathbf{x}_{i-1}$  and  $f_i(\mathbf{x}_{i-1})$  is the main cause of instability in training. Refer to (Liu et al., 2020a) for theoretical details.<sup>4</sup>

ADMIN initialization involves two phases: At the **Profiling** phase, we randomly initialize the model parameters using default initialization, set  $\omega_i = 1$ , and perform one step forward pass in order to compute the output variance of the residual branch  $Var[f(\mathbf{x}_{i-1})]$  at each layer.<sup>5</sup> In the **Training** phase, we fix  $\omega_i = \sqrt{\sum_{j < i} Var[f(\mathbf{x}_{j-1})]}$ , and then train the model using standard back-propagation. After training finishes,  $\omega_i$  can be folded back into the model parameters to recover the standard Transformer architecture. This simple initialization method is effective in ensuring that training does not diverge, even in deep networks.

## 4 Experiments

Experiments are conducted on standard WMT’14 English-French (FR) and English-German (DE) benchmarks. For FR, we mimic the setup<sup>6</sup> of (Ott et al., 2018), with 36M training sentences and 40k subword vocabulary. We use the provided ‘valid’ file for development and newstest14 for test. For DE, we mimic the setup<sup>7</sup> of (So et al., 2019), with 4.5M training sentences, 32K subword vocabulary, newstest2013 for dev, and newstest2014 for test.

We adopt the hyper-parameters of the Transformer-based model (Vaswani et al., 2017) as implemented in FAIRSEQ (Ott et al., 2019), i.e. 512-dim word embedding, 2048 feed-forward model size, and 8 heads, but vary the number of layers. RAdam (Liu et al., 2019) is our optimizer.<sup>8</sup>

**Main Result:** Our goal is to explore whether very deep Transformers are feasible and effective. We compare: (a) **6L-6L**: a baseline Transformer Base with 6 layer encoder and 6 layer decoder, vs. (b) **60L-12L**: A deep transformer with 60 encoder

<sup>4</sup>Note that paper presents results of 18-layer Transformers on the WMT’14 En-De, which we also use here. Our contribution is a more comprehensive evaluation.

<sup>5</sup>We estimate the variance with one batch of 8k tokens.

<sup>6</sup><https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-WMT'14en2fr.sh>

<sup>7</sup>[https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/data\\_generators/translate\\_ende.py](https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/data_generators/translate_ende.py)

<sup>8</sup>For FR, #warmup steps is 8000, max #epochs is 50, and peak learning rate is 0.0007. For DE, #warmup steps is 4000, max #epochs is 50, and learning rate is 0.001. Max #tokens in each batch is set to 3584 following (Ott et al., 2019).

<sup>3</sup>The 96-layer GPT-3 (Brown et al., 2020) uses pre-LN.

Model	WMT'14 English-French (FR)					WMT'14 English-German (DE)				
	#param	T↓	M↑	BLEU↑	Δ	#param	T↓	M↑	BLEU↑	Δ
6L-6L Default	67M	42.2	60.5	41.3	-	61M	54.4	46.6	27.6	-
6L-6L ADMIN	67M	41.8	60.7	41.5	0.2	61M	54.1	46.7	27.7	0.1
60L-12L Default	262M	diverge				256M	diverge			
60L-12L ADMIN	262M	<b>40.3</b>	<b>62.4</b>	<b>43.8</b>	2.5	256M	<b>51.8</b>	<b>48.3</b>	<b>30.1</b>	2.5

Table 1: Test results on WMT'14 benchmarks, in terms of TER (T↓), METEOR (M↑), and BLEU. Δ shows difference in BLEU score against baseline 6L-6L Default. Best results are boldfaced. 60L-12L ADMIN outperforms 6L-6L in all metrics with statistical significance ( $p < 0.05$ ). Following convention, BLEU is computed by `multi-bleu.perl` via the standardized tokenization of the publicly-accessible dataset.

BLEU via <code>multi-bleu.perl</code>	FR	DE
60L-12L ADMIN	<b>43.8</b>	<b>30.1</b>
(Wu et al., 2019b)	43.3	29.9
(Wang et al., 2019)	-	29.6
(Wu et al., 2019a)	43.2	29.7
(Ott et al., 2018)	43.2	29.3
(Vaswani et al., 2017)	41.8	28.4
(So et al., 2019)	41.3	29.8
(Gehring et al., 2017)	40.5	25.2
BLEU via <code>sacreBLEU.py</code>	FR	DE
60L-12L ADMIN	<b>41.8</b>	<b>29.5</b>
(Ott et al., 2018)	41.4	28.6
(So et al., 2019)	n/a	29.2

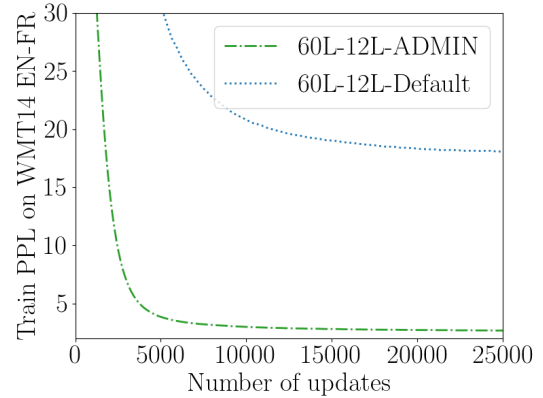
Table 2: State-of-the-Art on WMT'14 EN-FR/EN-DE

layers and 12 decoder layers.<sup>9</sup> For each architecture, we train with either default initialization (Glorot and Bengio, 2010) or ADMIN initialization.

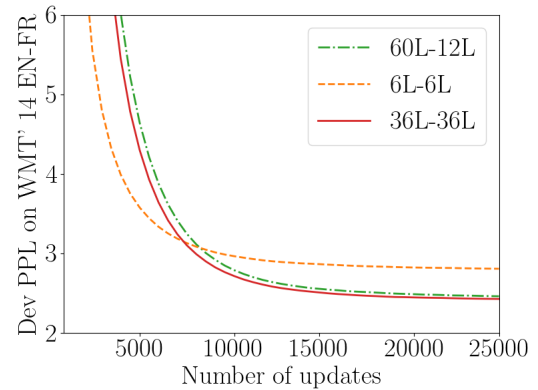
The results in terms of BLEU (Papineni et al., 2002), TER (Snover et al., 2006), and METEOR (Lavie and Agarwal, 2007) are reported in Table 1. Similar to previous work (Bapna et al., 2018), we observe that deep 60L-12L Default diverges during training. But the same deep model with ADMIN successfully trains and impressively achieves 2.5 BLEU improvement over the baseline 6L-6L Default in both datasets. The improvements are also seen in terms of other metrics: in EN-FR, 60L-12L ADMIN outperforms the 6L-6L models in TER (40.3 vs 42.2) and in METEOR (62.4 vs 60.5). All results are statistically significant ( $p < 0.05$ ) with a 1000-sample bootstrap test (Clark et al., 2011).

These results indicate that it is feasible to

<sup>9</sup>We use “(N)L-(M)L” to denote that a model has N encoder layers and M decoder layers. N & M are chosen based on GPU (16G) memory constraint. For reproducibility and simplicity, we focused on models that fit easily on a single GPU system. Taking FR as an example, it takes 2.5 days to train 60L-12L using one DGX-2 (16 V100's), 2 days to train a 6L-6L using 4 V100's.



(a) Train set perplexity: Default vs ADMIN



(b) Dev set perplexity: different ADMIN models

Figure 2: Learning curve

train standard (post-LN) Transformers that are very deep.<sup>10</sup> These models achieve state-of-the-art results in both datasets. The top results in the literature are compared in Table 2.<sup>11</sup> We list BLEU scores computed with `multi-bleu.perl` on the tokenization of the downloaded data (commonly done in previ-

<sup>10</sup>Note: the pre-LN version does train successively on 60L-12L and achieves 29.3 BLEU in DE & 43.2 in FR. It is better than 6L-6L but worse than 60L-12L ADMIN.

<sup>11</sup>The table does not include systems that use extra data.

Model	BLEU	a	b	c	d	e	f	g
a:6L-6L	41.5		-	-	-	-	-	-
b:12L-12L	42.6	+		-	-	-	-	-
c:24L-12L	43.3	+	+		=	-	=	=
d:48L-12L	43.6	+	+	=		=	=	+
e:60L-12L	43.8	+	+	+	=		=	+
f:36L-36L	43.7	+	+	=	=	=		+
g:12L-60L	43.1	+	+	=	-	-	-	

Table 3: BLEU comparison of different encoder and decoder layers (using ADMIN initialization, on WMT’14 EN-FR). In the matrix, each element (i,j) indicates if the model in row i significantly outperforms the model in column j (+), under-performs j (-), or has no statistically significant difference (=).

ous work), and with `sacrebleu.py` (version: tok.13a+version.1.2.10). which allows for a safer token-agnostic evaluation (Post, 2018).

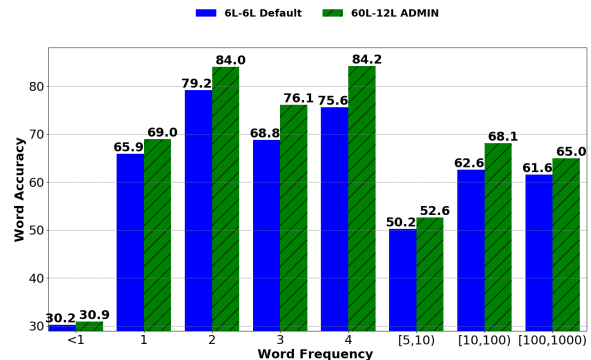
**Learning Curve:** We would like to understand why 60L-12L ADMIN is doing better from the optimization perspective. Figure 2 (a) plots the learning curve comparing ADMIN to Default initialization. We see that Default has difficulty decreasing the training perplexity; its gradients hit NaN, and the resulting model is not better than a random model. In Figure 2 (b), we see that larger models (60L-12L, 36L-36L) are able obtain lower dev perplexities than 6L-6L, implying that the increased capacity does lead to better generalization.

**Fine-grained error analysis:** We are also interested in understanding how BLEU improvements are reflected in terms of more nuanced measures. For example, do the deeper models particularly improve translation of low frequency words? Do they work better for long sentences? The answer is that the deeper models appear to provide improvements generally across the board (Figure 3).<sup>12</sup>

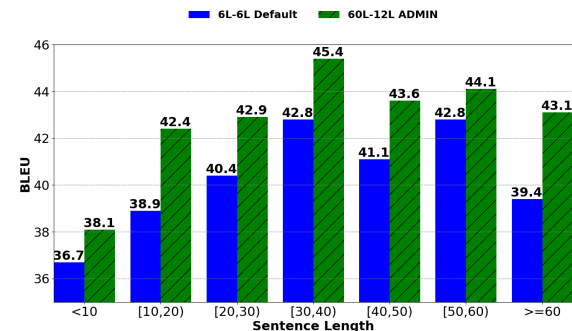
**Ablation Studies:** We experimented with different number of encoder and decoder layers, given the constraint of a 16GB GPU. Table 3 shows the pairwise comparison of models. We observe that 60L-12L, 48L-12L, and 36L-36L are statistically tied for best BLEU performance. It appears that deeper encoders are more worthwhile than deeper decoders, when comparing 60L-12L to 12L-60L, despite the latter having more parameters.<sup>13</sup>

<sup>12</sup>Computed by `compare-mt` (Neubig et al., 2019).

<sup>13</sup>Recall from Figure 1 that each encoder layer has 2 subnetwork components and each decoder layer has 3 components.



(a) Word accuracy according to frequency in the training data



(b) BLEU scores according to sentence length.

Figure 3: Fine-grained Error Analysis: note the deep model performs better across the board, indicating that it helps translation in general.

We also experiment with wider networks, starting with a 6L-6L Transformer-Big (1024-dim word embedding, 4096 feed-forward size, 16 heads) and doubling its layers to 12L-12L. The BLEU score on EN-FR improved from 43.2 to 43.6 (statistically significant,  $p < 0.05$ ). A 24L-12L Transformer with BERT-Base like settings (768-dim word embedding, 3072 feed-forward size, 12 heads) obtain **44.0** BLEU score on WMT’14 EN-FR. This shows that increased depth also helps models that are already relatively wide.

## 5 Conclusion

We show that it is feasible to train Transformers at a depth that was previously believed to be difficult. Using ADMIN initialization, we build Transformer-based models of 60 encoder layers and 12 decoder layers. On WMT’14 EN-FR and WMT’14 EN-EN, these deep models outperform the conventional 6-layer Transformers by up to 2.5 BLEU, and obtain state-of-the-art results.

We believe that the ability to train very deep models may open up new avenues of research in NMT, including: (a) Training on extremely large



but noisy data, e.g. back-translation (Edunov et al., 2018) and adversarial training (Cheng et al., 2019; Liu et al., 2020b), to see if it can be exploited by the larger model capacity. (b) Analyzing the internal representations, to see if deeper networks can indeed extract higher-level features in syntax and semantics (Belinkov and Glass, 2019). (c) Compressing the very deep model via e.g. knowledge distillation (Kim and Rush, 2016), to study the trade-offs between size and translation quality.

## Acknowledgments

We thank Hao Cheng, Akiko Eriguchi and Hany Hassan Awadalla for valuable discussions.

## References

- Ankur Bapna, Mia Xu 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.
- Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of edinburgh submissions to the wmt19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.
- Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Yoshua Bengio. 2009. *Learning Deep Architectures for AI*, volume Foundations and Trends in Machine Learning. NOW Publishers.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, 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)*, Melbourne, Australia. Association for Computational Linguistics.
- Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333.
- Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.
- Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808, Melbourne, Australia. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, page 1243–1252. JMLR.org.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *Advances in Neural Information Processing Systems*, pages 7944–7954.
- Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume*

- 2: *Shared Task Papers, Day 1*), pages 225–233, Florence, Italy. Association for Computational Linguistics.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.
- A. Lavie and A. Agarwal. 2007. METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Workshop on Statistical Machine Translation*.
- Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.
- Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020a. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*.
- Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020b. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.
- Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2020. Deep learning based text classification: a comprehensive review. *arXiv preprint arXiv:2004.03705*.
- Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. [compare-mt: A tool for holistic comparison of language generation systems](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. [Facebook fair’s wmt19 news translation task submission](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.
- Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. [Scaling neural machine translation](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*.
- 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.
- M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*.
- David So, Quoc Le, and Chen Liang. 2019. [The evolved transformer](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5877–5886, Long Beach, California, USA. PMLR.
- Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385.
- 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 *Advances in neural information processing systems*, pages 5998–6008.
- 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 Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.
- Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019a. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- 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.
- Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. 2019. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems 32*, pages 4381–4391. Curran Associates, Inc.