Hint-Based Training for Non-Autoregressive Machine Translation

Zhuohan Li[1] Zi Lin[2,3] Di He[2] Fei Tian[4] Tao Qin[5] Liwei Wang[2] Tie-Yan Liu[5]
[1]UC Berkeley [2]Peking University [3]Google AI [4]Facebook [5]Microsoft Research
zhuohan@cs.berkeley.edu {zi.lin, di_he, wanglw}@pku.edu.cn
feitia@fb.com {taoqin, tyliu}@microsoft.com

Abstract

Due to the unparallelizable nature of the autoregressive factorization, AutoRegressive Translation (ART) models have to generate tokens sequentially during decoding and thus suffer from high inference latency. Non-AutoRegressive Translation (NART) models were proposed to reduce the inference time, but could only achieve inferior translation accuracy. In this paper, we proposed a novel approach to leveraging the hints from hidden states and word alignments to help the training of NART models. The results achieve significant improvement over previous NART models for the WMT14 En-De and De-En datasets and are even comparable to a strong LSTM-based ART baseline but one order of magnitude faster in inference.

1 Introduction

Neural machine translation has attracted much attention in recent years (Bahdanau et al., 2014, 2016; Kalchbrenner et al., 2016; Gehring et al., 2016). Given a sentence $x = (x_1, \dots, x_{T_x})$ from the source language, the straight-forward way for translation is to generate the target words $y = (y_1, \dots, y_{T_y})$ one by one from left to right. This is also known as the *AutoRegressive* Translation (ART) models, in which the joint probability is decomposed into a chain of conditional probabilities:

$$P(y|x) = \Pi_{t=1}^{T_y} P(y_t|y_{<t}, x), \tag{1}$$

While the ART models have achieved great success in terms of translation quality, the time consumption during inference is still far away from satisfactory. During training, the predictions at different positions can be estimated in parallel since the ground truth pair (x, y) is exposed to the model. However, during inference, the model has to generate tokens sequentially as $y_{<t}$ must be inferred on the fly. Such autoregressive behavior becomes the bottleneck of the computational time (Wu et al., 2016).

In order to speed up the inference process, a line of works begin to develop non-autoregressive translation models. These models break the autoregressive dependency by decomposing the joint probability with

$$P(y|x) = P(T_y|x)\Pi_{t=1}^{T_y} P(y_t|x). \tag{2}$$

The lost of autoregressive dependency largely hurt the consistency of the output sentences, increase the difficulty in the learning process and thus lead to a low quality translation. Previous works mainly focus on adding different components into the NART model to improve the expressiveness of the network structure to overcome the loss of autoregressive dependency (Gu et al., 2017; Lee et al., 2018; Kaiser et al., 2018). However, the computational overhead of new components will hurt the inference speed, contradicting with the goal of the NART models: to parallelize and speed up neural machine translation models.

To tackle this, we proposed a novel hint-based method for NART model training. We first investigate the causes of the poor performance of the NART model. Comparing with the ART model, we find that: (1) the positions where the NART model outputs incoherent tokens will have very high hidden states similarity; (2) the attention distributions of the NART model are more ambiguous than those of ART model. Therefore, we design two kinds of hints from the hidden states and attention distributions of the ART model to help the training of the NART model. The experimental results show that our model achieves significant improvement over the NART baseline models and is even comparable to a strong ART baseline in Wu et al. (2016).

The work was performed at Microsoft Research Asia.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5708–5713,
Hong Kong, China, November 3–7, 2019.

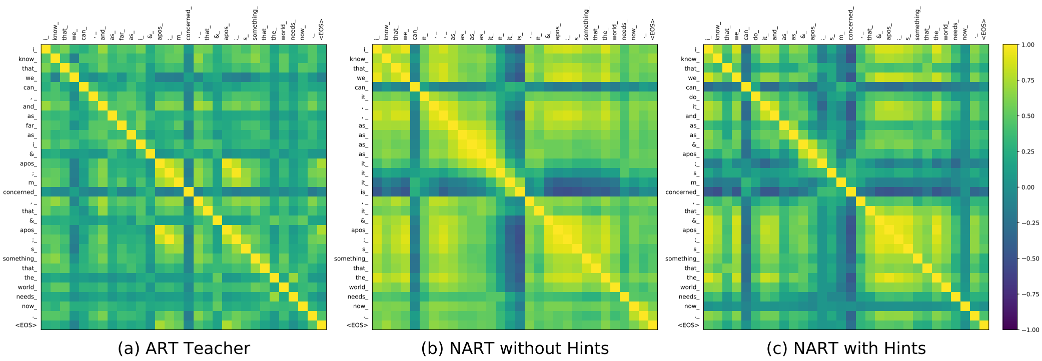

Figure 1: Case study: the above three figures visualize the hidden state cosine similarities of different models. The axes correspond to the generated target tokens. Each pixel shows the cosine similarities $\cos_{ij}$ between the last layer hidden states of the i-th and j-th generated tokens, where the diagonal pixel will always be 1.0.

2 Approach

In this section, we first describe the observations on the ART and NART models, and then discuss what kinds of information can be used as hints to help the training of the NART model. We follow the network structure in Vaswani et al. (2017), use a copy of the source sentence as decoder input, remove the attention masks in decoder self-attention layers and add a positional attention layer as suggested in Gu et al. (2017). We provide a visualization of ART and NART models we used in Figure 3 and a detailed description of the model structure in Appendix.

2.1 Observation: Illed States and Attentions

According to the case study in Gu et al. (2017), the translations of the NART models contain incoherent phrases (e.g. repetitive words) and miss meaningful tokens on the source side, while these patterns do not commonly appear in ART models. After some empirical study, we find two non-obvious facts that lead to this phenomenon.

First, we visualize the cosine similarities between decoder hidden states of a certain layer in both ART and NART models for sampled cases. Mathematically, for a set of hidden states $r_1, \ldots, r_T$, the pairwise cosine similarity can be derived by $\cos_{ij} = \langle r_i, r_j \rangle / (\|r_i\| \cdot \|r_j\|)$. We then plot the heatmap of the resulting matrix $\cos$. A typical example is shown in Figure 1, where the cosine similarities in the NART model are larger than those of the ART model, indicating that the hidden states across positions in the NART model are "similar". Positions with highly-correlated hidden states tend to generate the same word and make the NART model output repetitive tokens, e.g., the yellow area on the top-left of Figure 1(b), while this does not happen in the ART model (Figure 1(a)). According to our statistics, 70% of the cosine similarities between hidden states in the ART model are less than 0.25, and 95% are less than 0.5.

Second, we visualize the encoder-decoder attentions for sampled cases, shown in Figure 2. Good attentions between the source and target sentences are usually considered to lead to accurate translation while poor ones may cause wrong output tokens (Bahdanau et al., 2014). In Figure 2(b), the attentions of the ART model almost covers all source tokens, while the attentions of the NART model do not cover "farm" but with two "morning". This directly makes the translation result worse in the NART model. These phenomena inspire us to use the intermediate hidden information in the ART model to guide the learning process of the NART model.

2.2 Hints from the ART teacher Model

Our study motivates us to leverage the intermediate hidden information from an ART model to improve the NART model. We focus on how to define hints from a well-trained ART *teacher* model and use it to guide the training process of a NART *student* model. We study layer-to-layer hints and assume both the teacher and student models have an M-layer encoder and an N-layer decoder, despite the difference in stacked components.

Without the loss of generality, we discuss our method on a given paired sentence (x, y). In real experiments, losses are averaged over all training data. For the teacher model, we use $a^{tr}_{t,l,h}$ as the encoder-to-decoder attention distribution of h-th head in the l-th decoder layer at position t, and

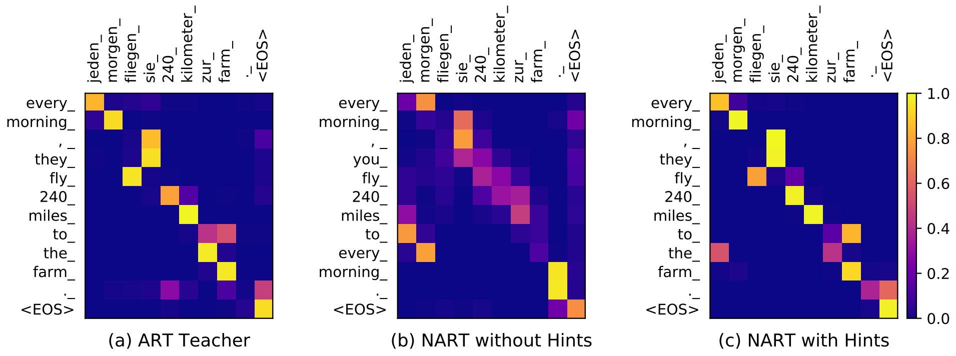

Figure 2: Case study: the above three figures visualize the encoder-decoder attention weights of different models. The x-axis and y-axis correspond to the source and generated target tokens respectively. The attention distribution is from a single head of the third layer encoder-decoder attention, which is the most informative one according to our observation. Each pixel shows attention weights α_{ij} between the i-th source token and j-th target token.

use $r^{tr}_{t,l}$ as the output of the l-th decoder layer after feed forward network at position t. Correspondingly, $a^{st}_{t,l,h}$ and $r^{st}_{t,l}$ are used for the student model. We propose a hint-based training framework that contains two kinds of hints:

Hints from hidden states The discrepancy of hidden states motivates us to use hidden states of the ART model as a hint for the learning process of the NART model. One straight-forward method is to regularize the L_1 or L_2 distance between each pair of hidden states in ART and NART models. However, since the network components are quite different in ART and NART models, applying the straight-forward regression on hidden states hurts the learning process and fails. Therefore, we design a more implicit loss to help the student refrain from the incoherent translation results by acting towards the teacher in the hidden-state level:

$$\mathcal{L}_{hid} = \frac{2}{(T_y-1)T_yN}\sum_{s=1}^{T_y-1}\sum_{t=s+1}^{T_y}\sum_{l=1}^{N}\phi(d_{st}, d_{tr}),$$

where $d_{st} = \cos(r^{st}_{s,l}, r^{st}_{t,l})$, $d_{tr} = \cos(r^{tr}_{s,l}, r^{tr}_{t,l})$, and ϕ is a penalty function. In particular, we let

$$\phi(d_{st}, d_{tr}) = \begin{cases} -\log(1-d_{st}), & \text{if } d_{st} \geq \gamma_{st} \\ & \text{and } d_{tr} \leq \gamma_{tr}; \\ 0, & \text{else}, \end{cases}$$

where $-1 \leq \gamma_{st}, \gamma_{tr} \leq 1$ are two thresholds controlling whether to penalize or not. We design this loss since we only want to penalize hidden states that are highly similar in the NART model, but not similar in the ART model. We have tested several choices of $-\log(1-d_{st})$, e.g., $\exp(d_{st})$, from which we find similar experimental results.

Hints from word alignments We observe that meaningful words in the source sentence are sometimes untranslated by the NART model, and the corresponding positions often suffer from ambiguous attention distributions. Therefore, we use the word alignment information from the ART model to help the training of the NART model.

In particular, we minimize KL-divergence between the per-head encoder-to-decoder attention distributions of the teacher and the student to encourage the student to have similar word alignments to the teacher model, i.e.

$$\mathcal{L}_{align} = \frac{1}{T_yNH}\sum_{t=1}^{T_y}\sum_{l=1}^{N}\sum_{h=1}^{H} D_{\mathrm{KL}}(a^{tr}_{t,l,h} \| a^{st}_{t,l,h}).$$

Our final training loss $\mathcal{L}$ is a weighted sum of two parts stated above and the negative log-likelihood loss $\mathcal{L}_{nll}$ defined on bilingual sentence pair (x, y), i.e.

$$\mathcal{L} = \mathcal{L}_{nll} + \lambda\mathcal{L}_{hid} + \mu\mathcal{L}_{align}, \tag{3}$$

where λ and μ are hyperparameters controlling the weight of different loss terms.

3 Experiments

3.1 Experimental Settings

The evaluation is on two widely used public machine translation datasets: IWSLT14 German-to-English (De-En) (Huang et al., 2017; Bahdanau et al., 2016) and WMT14 English-to-German (En-De) dataset (Wu et al., 2016; Gehring et al., 2017). To compare with previous works, we also reverse WMT14 English-to-German dataset and obtain WMT14 German-to-English dataset.

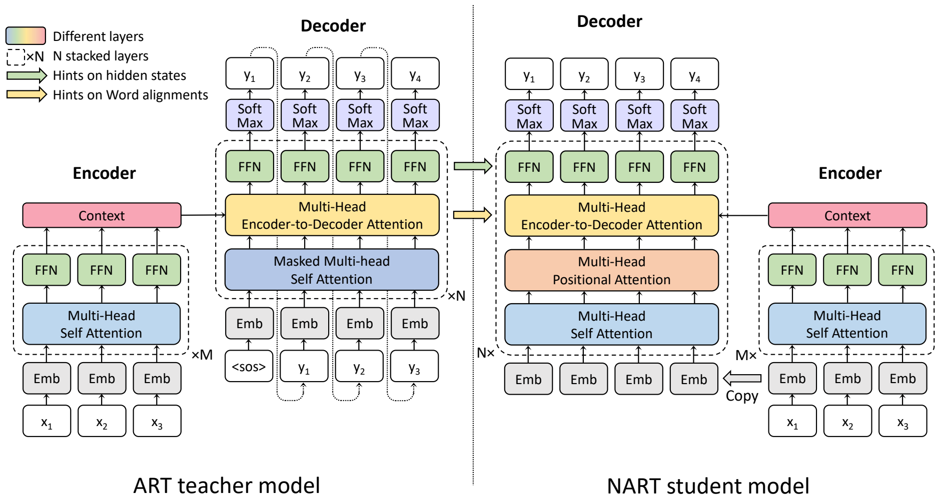

Figure 3: Hint-based training from ART model to NART model.

We pretrain Transformer (Vaswani et al., 2017) as the teacher model on each dataset, which achieves 33.26/27.30/31.29 in terms of BLEU (Papineni et al., 2002) in IWSLT14 De-En, WMT14 En-De and De-En test sets. The student model shares the same number of layers in encoder/decoder, size of hidden states/embeddings and number of heads as the teacher models (Figure 3). Following Gu et al. (2017); Kim and Rush (2016), we replace the target sentences by the decoded output of the teacher models.

Hyperparameters $(\gamma_{st}, \gamma_{tr}, \lambda, \mu)$ for hint-based learning are determined to make the scales of three loss components similar after initialization. We also employ label smoothing of value $\epsilon_{ls} = 0.1$ (Szegedy et al., 2016) in all experiments. We use Adam optimizer and follow the setting in Vaswani et al. (2017). Models for WMT14/IWSLT14 tasks are trained on 8/1 NVIDIA M40 GPUs respectively. The model is based on the open-sourced `tensor2tensor` (Vaswani et al., 2018).[1] More settings can be found in Appendix.

3.2 Inference

During training, T_y does not need to be predicted as the target sentence is given. During testing, we have to predict the length of the target sentence for each source sentence. In many languages, the length of the target sentence can be roughly estimated from the length of the source sentence. We choose a simple method to avoid the computational overhead, which uses input length to determine target sentence length: $T_y = T_x + C$, where C is a constant bias determined by the average length differences between the source and target training sentences. We can also predict the target length ranging from $[(T_x+C)-B, (T_x+C)+B]$, where B is the halfwidth. By doing this, we can obtain multiple translation results with different lengths. Note that we choose this method only to show the effectiveness of our proposed method and a more advanced length estimation method can be used to further improve the performance.

Once we have multiple translation results, we additionally use our ART teacher model to evaluate each result and select the one that achieves the highest probability. As the evaluation is fully parallelizable (since it is identical to the parallel training of the ART model), this rescoring operation will not hurt the non-autoregressive property of the NART model.

3.3 Experimental Results

We compare our model with several baselines, including three ART models, the fertility based (FT) NART model (Gu et al., 2017), the deterministic iterative refinement based (IR) NART model (Lee et al., 2018), and the Latent Transformer (LT; Kaiser et al., 2018) which is not fully non-autoregressive by incorporating an autoregressive sub-module in the NART model architecture.

The results are shown in the Table 1.[2] Across

[1] Open-source code can be found at `https://github.com/zhuohan123/hint-nart`

[2] ‡ and † indicate that the latency is measured on our own platform or by previous works, respectively. Note that the latencies may be affected by hardware settings and such absolute values are not fair for direct comparison, so we also list the speedup of the works compared to their ART baselines.

Models	WMT14 En-De	WMT14 De-En	IWSLT14 De-En	Latency	Speedup
Autoregressive models					
LSTM-based S2S (Wu et al., 2016; Bahdanau et al., 2016)	24.60	/	28.53	/	/
ConvS2S (Gehring et al., 2017; Edunov et al., 2017)	26.43	/	32.84	/	/
Transformer (Vaswani et al., 2017)	27.30	31.29	33.26	784 ms‡	1.00×
Non-autoregressive models					
FT (Gu et al., 2017)	17.69	20.62	/	39 ms†	15.6×
FT (rescoring 10 candidates)	18.66	22.41	/	79 ms†	7.68×
FT (rescoring 100 candidates)	19.17	23.20	/	257 ms†	2.36×
IR (Lee et al., 2018, adaptive refinement steps)	21.54	25.43	/	/	2.39×
LT (Kaiser et al., 2018)	19.8	/	/	105 ms†	5.78×
LT (rescoring 10 candidates)	21.0	/	/	/	/
LT (rescoring 100 candidates)	22.5	/	/	/	/
NART w/ hints	**21.11**	**25.24**	**25.55**	**26 ms‡**	**30.2×**
NART w/ hints ($B = 4$, 9 candidates)	**25.20**	**29.52**	**28.80**	**44 ms‡**	**17.8×**

Table 1: Performance on WMT14 En-De, De-En and IWSLT14 De-En tasks. "/" means non-reportable.

different datasets, our method achieves significant improvements over previous non-autoregressive models. Specifically, our method outperforms fertility based NART model with 6.54/7.11 BLEU score improvements on WMT En-De and De-En tasks in similar settings and achieves comparable results with state-of-the-art LSTM-based model on WMT En-De task. Furthermore, our model achieves a speedup of 30.2 (output a single sentence) or 17.8 (teacher rescoring) times over the ART counterparts. Note that our speedups significantly outperform all previous works, because of our lighter design of the NART model: without any computationally expensive module trying to improve the expressiveness.

We also visualize the hidden state cosine similarities and attention distributions for the NART model with hint-based training, as shown in Figure 1(c) and 2(c). With hints from hidden states, the hidden states similarities of the NART model decrease in general, and especially for the positions where the original NART model outputs incoherent phrases. The attention distribution of the NART model after hint-based training is more similar to the ART teacher model and less ambiguous comparing to the NART model without hints.

According to our empirical analysis, the percentage of repetitive words drops from 8.3% to 6.5% by our proposed methods on the IWSLT14 De-En test set, which is a 20%+ reduction. This shows that our proposed method effectively improve the quality of the translation outputs. We also provide several case studies in Appendix.

Finally, we conduct an ablation study on IWSLT14 De-En task. As shown in Table 2, the hints from word alignments provide an improvement of about 1.6 BLEU points, and the hints from hidden states improve the results by about 0.8 BLEU points. We also test these models on a subsampled set whose source sentence lengths are at least 40. Our model outperforms the baseline model by more than 3 BLEU points (20.63 v.s. 17.48).

Model	$\mathcal{L}_{nll}$	$\mathcal{L}_{nll} + \mathcal{L}_{align}$	$\mathcal{L}_{nll} + \mathcal{L}_{align} + \mathcal{L}_{hid}$
BLEU	23.08	24.76	**25.55**
Long-sentence BLEU	17.48	19.24	**20.63**

Table 2: Ablation studies on IWSLT14 De-En. Results are BLEU scores without teacher rescoring.

4 Conclusion

In this paper, we proposed to use hints from a well-trained ART model to enhance the training of NART models. Our results on WMT14 En-De and De-En significantly outperform previous NART baselines, with one order of magnitude faster in inference than ART models. In the future, we will focus on designing new architectures and training methods for NART models to achieve comparable accuracy as ART models.

Acknowledgment

This work is supported by National Key R&D Program of China (2018YFB1402600), NSFC (61573026) and BJNSF (L172037) and a grant from Microsoft Research Asia. We would like to thank the anonymous reviewers for their valuable comments on our paper.

References

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

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2017. Classical structured prediction losses for sequence to sequence learning. *arXiv preprint arXiv:1711.04956*.

Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Minghao Hu, Yuxing Peng, Furu Wei, Zhen Huang, Dongsheng Li, Nan Yang, and Ming Zhou. 2018. Attention-guided answer distillation for machine reading comprehension. *arXiv preprint arXiv:1808.07644*.

Po-Sen Huang, Chong Wang, Dengyong Zhou, and Li Deng. 2017. Neural phrase-based machine translation. *arXiv preprint arXiv:1706.05565*.

Łukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Pamar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. *arXiv preprint arXiv:1803.03382*.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

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 6000–6010.

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*.