Predicting and Using Target Length in Neural Machine Translation

Zijian Yang **Yingbo Gao** **Weiyue Wang** **Hermann Ney**
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{zyang|gao|wwang|ney}@i6.informatik.rwth-aachen.de

Abstract

Attention-based encoder-decoder models have achieved great success in neural machine translation tasks. However, the lengths of the target sequences are not explicitly predicted in these models. This work proposes length prediction as an auxiliary task and set up a sub-network to obtain the length information from the encoder. Experimental results show that the length prediction sub-network brings improvements over the strong baseline system and that the predicted length can be used as an alternative to length normalization during decoding.

1 Introduction

In recent years, neural network (NN) models have achieved great improvements in machine translation (MT) tasks. Sutskever et al. (2014) introduced the encoder-decoder network, Bahdanau et al. (2015) developed the attention-based architecture, and Vaswani et al. (2017) proposed the transformer model with self-attentions, which delivers state-of-the-art performances.

Despite the success achieved in neural machine translation (NMT), current NMT systems do not model the length of the output explicitly, and thus various length normalization approaches are often used in decoding. Length normalization is a common technique used in the beam search of NMT systems to enable a fair comparison of partial hypotheses with different lengths. Without any form of length normalization, regular beam searches will prefer shorter hypotheses to longer ones on average, as a negative logarithmic probability is added at each step, resulting in lower (more negative) scores for longer sentences. The simplest way is to normalize the score of the current partial hypothesis (e_1^i) by its length ($|i|$):

$$s(e_1^i, f_1^J) = \frac{\log p(e_1^i|f_1^J)}{|i|} \quad (1)$$

where f_1^J is the source sequence. To use a softer approach, the denominator $|i|$ can also be raised to the power of a number between 0 and 1 or replaced by more complex functions, as proposed in Wu et al. (2016). Moreover, a constant word reward is used in He et al. (2016) as an alternative to length normalization. All of these approaches tackle the length problem in decoding, and all NMT systems use at least one of them to ensure the performance.

In addition to investigating various types of length normalization, their rationality is rarely explored. Although length normalization appears to be simple and effective, it is still an additional technique to help a "weak" machine translation model that cannot handle the hypothesis length properly. In this work it is proposed to model the target length using the neural network itself in a multi-task learning way. The estimated length information can either be implicitly included in the network to "guide" translation, or it can be used explicitly as an alternative to length normalization during decoding. The experimental results on various datasets show that the proposed system achieves improvements compared to the baseline model and the predicted length can easily be used to replace the length normalization.

2 Related Work

Multi-task learning is an important training strategy that aims to improve the generalization performance of the main task with some other related tasks (Luong et al., 2016; Martínez Alonso and Plank, 2017). With regard to deep learning, multi-task learning is applied successfully in many areas, such as natural language processing (Liu et al., 2015), computer vision (Donahue et al., 2014), and speech processing (Heigold et al., 2013). In this work, the prediction of the target length while generating translation hypotheses can be seen as a

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 389–395
December 4 - 7, 2020.

multi-task learning application.

Murray and Chiang (2018) and Stahlberg and Byrne (2019) attribute the fact that beam search prefers shorter candidates due to the local normalization of NMT. To address this problem, in addition to the standard length normalization technique, Wu et al. (2016) propose a more complicated correction with a hyperparameter that can be adjusted for different language pairs. In He et al. (2016), a word reward function is proposed that simulates the coverage vector in statistical machine translation so that the decoder prefers a long translation. Huang et al. (2017) and Yang et al. (2018) suggest variations of this reward that provide better guarantees during search. There are also works on target vocabulary prediction in the encoder-decoder model that implicitly predicts the target length (Weng et al., 2017; Suzuki and Nagata, 2017). In our work, the target length is explicitly modeled by the neural network itself, which indicates that the entire system relies more on statistics rather than heuristics.

3 Neural Length Model

To predict the target length based on the standard transformer architecture (Vaswani et al., 2017), we build a multi-layer sub-network that only requires information from the source sequence (or the encoder). In this work the length prediction task is considered as a classification task for different lengths. Other methods, such as directly generating a real number, binarizing the length, or performing multiple binary classification tasks, are also being tested, but the classification method performs best.

3.1 Modeling

We predict the length of the target sequence by a classifier in the range of $[0, 200]$, the input of which is a single vector without time dimension, which is extracted from the encoder. To obtain this vector, we first concatenate the encoder output and the embedding of the source tokens, followed by a linear layer with an activation function to map the vectors to the same dimension as the original encoder output. Then we set the length of the concatenated vectors to 200 by clipping or zero padding, in order to have a fixed length of time dimension, which could be compressed to a single vector by convolution and max-pooling. Then, the vectors run through a convolutional layer with an activation function and a max-pooling layer. A linear layer is then used to project the max-pooled vector into a single vector.

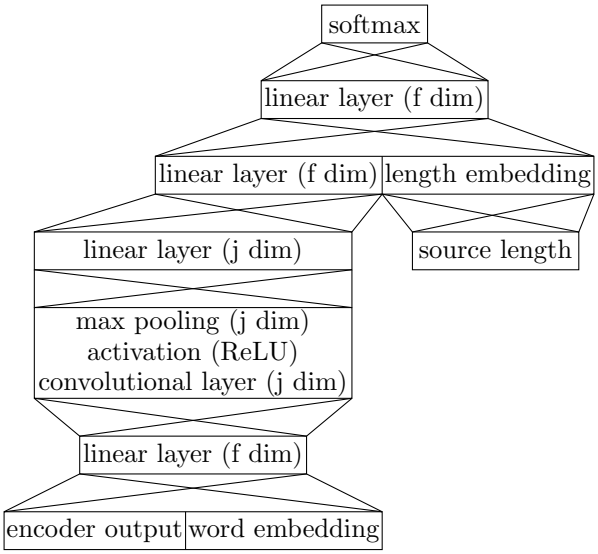

Figure 1: The architecture of the length prediction sub-network.

We also embed the length of source sequence into a 201 dimension vector with a length embedding matrix, which is initialized by the empirical distribution of the length. This length embedding is then concatenated with the output logit of the length prediction sub-network. Again, this concatenated vector is projected through a linear layer onto a vector s with 201 dimensions. Finally, the length distribution q_l is given by a softmax over s. And the predicted length l_{pred} is the l with the highest probability. The complete structure of the proposed length prediction sub-network is illustrated in Figure 1.

When we train the model with the translation and length prediction tasks jointly, the gradient of the length model will propagate to the translation model (referred to as no-connection in this paper). Thus, these two models will influence each other during multi-task training. In addition, the translation model could benefit from concatenating the length prediction output vector s to the outputs of each decoder layer (referred to as cross-concat in this paper). After the vector is concatenated, a linear projection is run through to maintain the feature dimension of the vector as the original one, so that it can be used without modifying the rest of the original transformer model. Here we detach s from the backpropagation graph so that the length prediction is not affected by this connection. In this method, we think that with the concatenation, the

length information could be passed to the decoder and used implicitly.

3.2 Training

During training, Kullback–Leibler (KL) divergence (Kullback and Leibler, 1951) is used as the loss of the length prediction task:

$$\text{Loss}_{\text{length}} = D_{\text{KL}}(P||Q) = \sum_{l} p_l \log \frac{p_l}{q_l} \quad (2)$$

where q_l is the probability from model output. Suppose l_{target} is the actual length of the target sequence, p_l is the target distribution given by a Gaussian function added with a neighborhood reward $d(l, l_{\text{target}})$. Formally, p_l is given as:

$$p_l = \frac{a_l}{\sum_{l'} a_{l'}} \quad (3)$$

where

$$a_l = \exp\left(-\left(\frac{l - l_{\text{target}}}{\sigma}\right)^2\right) + d(l, l_{\text{target}}) \quad (4)$$

where

$$d(l, l_{\text{target}}) = \begin{cases} 1 & \textbf{if} \quad l = l_{\text{target}} \\ 0.1 & \textbf{if} \quad |l - l_{\text{target}}| = 1 \\ 0 & \textbf{others} \end{cases} \quad (5)$$

here σ is a constant and is used to control the shape of the distribution. In contrast to cross entropy with label smoothing, in which there is only one true label with a high probability and others are treated equally, the probability p_l becomes smaller if l is further away from l_{target}, which creates the desired relationship between each class in the classifier.

We use cross entropy with label smoothing as the training loss for the translation task. We linearly combine the translation loss with the length loss, so that the training loss is given by

$$\text{Loss}_{\text{all}} = \lambda_1 \text{Loss}_{\text{translation}} + \lambda_2 \text{Loss}_{\text{length}} \quad (6)$$

3.3 Decoding

Besides using the length information implicitly (as the two methods mentioned above), we can also guide the decoding step with the length prediction explicitly. With the help of the length prediction, we have a mathematically reasonable control of the output length in comparison to the length normalization in beam search. Since the predicted target length cannot be 100% accurate and a source sentence can have multiple possible translations of different lengths, we control the length of the inference by penalizing the score (logarithmic probability) of the end-of-sentence (EOS) token during beam search, rather than forcing the length of the inference to match the predicted length. More specifically, if the length of the hypothesis is shorter than the predicted length, the EOS token score is penalized; if the hypothesis is longer than the predicted length, the EOS token score is rewarded to facilitate the selection of the EOS token in beam search to finalize the hypothesis. A logarithmic linear penalty is introduced, which is added to the score of EOS token at each time step during beam search:

$$P = \alpha \log \frac{L_{\text{hyp}}}{L_{\text{pred}}} \quad (7)$$

where L_{hyp} is the length of the hypothesis, L_{pred} is the predicted length of the target sentence, and α is a hyperparameter to control the penalty.

4 Experiments

4.1 Experimental Setup

We first conduct experiments on a relatively small dataset, IWSLT2014 German→English (160k sentence pairs) (Cettolo et al., 2014), to tune hyperparameters and analyze the performance. Then we train our model on other four different language pairs, which are Spanish-English (es-en), Italian-English (it-en), Dutch-English (nl-en) and Romanian-English (ro-en). At last, the experiments are carried out on the WMT (Barrault et al., 2019) German↔English (4M sentence pairs) datasets in order to compare our system with the baseline model. All datasets used in this work are preprocessed by *fairseq*[1] (Ott et al., 2019). Data statistics can be found in Table 1.

data set	language pair	number of sentence pairs		
		train	valid	test
IWSLT	de-en	160k	7.3k	6.8k
	es-en	169k	7.7k	5.6k
	it-en	167k	7.6k	6.6k
	nl-en	154k	7.0k	5.4k
	ro-en	168k	7.6k	5.6k
WMT	en↔de	4.5M	3.0k	3.0k

Table 1: Data statistics of IWSLT and WMT datasets.

We employ the transformer base architecture (Vaswani et al., 2017) as the baseline model and

[1] https://github.com/pytorch/fairseq

this work is implemented in *fairseq*. All model hyperparameters of the baseline model for IWSLT match the settings in *fairseq*. For the WMT experiments, the settings are the same as for the original base transformer model. The sub-network used for the length prediction only increases the number of free parameters by less than 10%, the influence on the training and decoding speed is also marginal. Experimental performance is measured using BLEU (Papineni et al., 2002; Post, 2018) and CHARACTER (Wang et al., 2016) (CTER) metrics.

4.2 Experimental Results

For the length prediction task, the inference length does not have to correspond exactly to the reference length, since there can be multiple correct translations with different lengths. Therefore, we consider the predictions that fulfill $|l_{\text{predict}} - l_{\text{target}}|/l_{\text{target}} \leq T$ to be accurate, where T is a threshold.

λ_1	λ_2	model	acc. [%]	BLEU[%]
1	0	baseline	-	34.8
0	1	length model	83.4	-
1	1	no-connection	86.7	35.3
		cross-concat	86.1	35.3

Table 2: Accuracy rate and BLEU scores of the proposed system with the length model on the IWSLT German→English task. The accuracy of the length prediction task is reported on the validation dataset.

language pair	es-en	it-en	nl-en	ro-en
baseline	41.2	32.6	37.8	38.4
no-connection	41.3	32.8	37.8	38.8
cross-concat	41.3	32.7	38.3	38.7

Table 3: Performance (in BLEU[%] scores) using different methods for different language pairs.

Table 2 shows the experiments carried out with the standard translation model ($\lambda_1 = 1$ and $\lambda_2 = 0$), the pure length model ($\lambda_1 = 0$ and $\lambda_2 = 1$) and the combination of the two models ($\lambda_1 = 1$ and $\lambda_2 = 1$). For the accuracy, here we choose the threshold $T = 20\%$. It is observed that the joint training of the two models performs better for both the translation and the length prediction task. Due to the multi-task learning, although the translation task does not explicitly influence the length prediction, it helps to bring model parameters to a better local optimum.

We use no-connection, cross-concat model to train on other language pairs with the same hyperparameters as on IWSLT de-en to test the performance, as shown in Table 3. For cross-concat, the BLEU score of the nl-en system is improved by 0.5%. For other language pairs, the results of two methods are almost the same, all of which are better than the baseline.

Figure 2 shows the relative differences Δ_l between the predicted and actual lengths for different target sequence lengths. When l_{target} is between about 10 and 40, the prediction is pretty good: for most l_{target} in $[10, 40]$, Δ_l is less than 15%. When l_{target} is in $[40, 100]$, the prediction becomes worse, but most of them are still less than 20%. After 100, the prediction is pretty bad. There are two reasons for this: first, the length of most target sequences in training data is between 10 and 40, so the model does not often see the cases that the sequence is too long; second, there are very few long sequences in the validation dataset, so the results for these points lack statistical meaning.

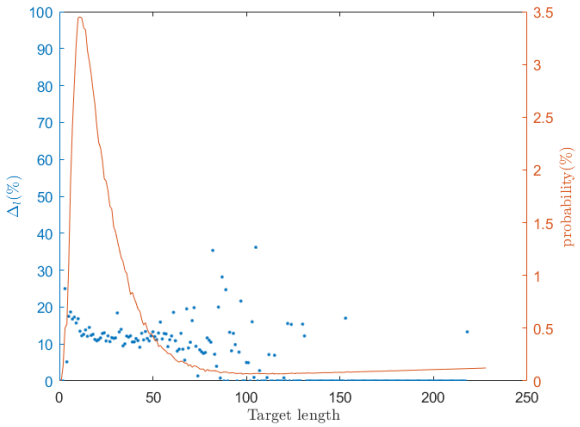

Figure 2: The left y-axis shows the relative difference between the target and the predicted length and the right y-axis is for the empirical distribution of l_{target}.

Table 4 shows the comparison between the proposed approach and the baseline model. Here we set $\alpha = 10$ according to the experiments that are carried out on the IWSLT dataset. The additional sub-network for length prediction improves the BLEU score by up to 0.9% over the strong baseline model. Moreover, the predicted length successfully serves as an alternative to length normalization. Regardless of whether the inference tends to be longer or shorter than the reference, the ratio when using the predicted length is slightly better than using the length normalization, which shows better control of the length.

model		English→German newstest2014			German→English newstest2017		
		BLEU[%] ↑	CTER[%] ↓	len. ratio	BLEU[%] ↑	CTER[%] ↓	len. ratio
baseline		27.3	45.8	1.024	33.0	41.8	**0.974**
+ len. model	no-connection	**27.6**	**45.5**	1.020	33.4	41.5	0.972
	cross-concat	27.4	45.7	1.024	33.4	41.3	0.970
- len. norm.	no-connection	**27.6**	45.6	**1.018**	**33.9**	**40.9**	0.973
	cross-concat	27.3	45.8	1.021	33.7	41.2	**0.974**

Table 4: Comparison between the proposed system and the baseline model. "+ len. model" indicates that the length prediction sub-network is added to the baseline architecture. "- len. normalization" denotes that the predicted length is used during decoding as an alternative to the length normalization as described in Section 3.3. "len. ratio" gives the length ratio between the hypothesis length and the reference length: the closer to 1, the better.

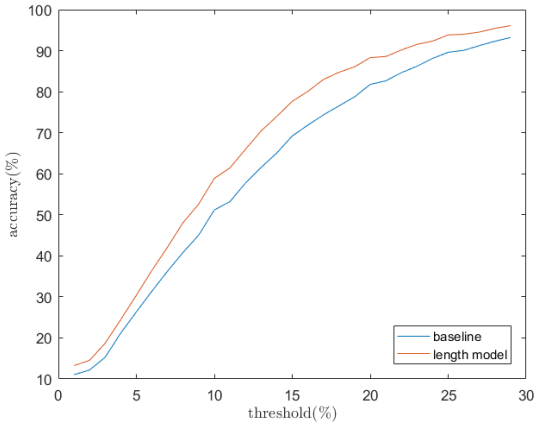

Figure 3: Length prediction model outperforms the baseline model in length prediction test accuracy.

Figure 3 shows the relationship between the length prediction accuracy of the baseline model and the length prediction model, and the threshold T for calculating accuracy. Since the transformer baseline model does not predict the target length, the length prediction of baseline is obtained from the average ratio of source sentence length to target sentence length. For the length prediction task, the accuracy of our model is always better than the baseline, which indicates that on WMT data, our model can still predict the target length well.

5 Conclusion

In this paper, we propose a length prediction sub-network based on the transformer architecture, and a method of using the length prediction information on the decoder side, namely cross-concat. In decoding, we use the predicted length to calculate a logarithmic linear penalty in the beam search in order to replace the length normalization. Experimental results show that the sub-network can predict target length well and further improve translation quality. In addition, the predicted length can be used to replace the length normalization with a better and more mathematically explainable control of the output length. For future work, the use of length prediction in positional encoding (Lakew et al., 2019; Takase and Okazaki, 2019) and non-autoregressive (or partially autoregressive) NMT (Gu et al., 2017; Lee et al., 2018; Stern et al., 2019) could be further investigated.

Acknowledgements

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project "SEQCLAS") and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project "CoreTec"). The GPU computing cluster was supported by DFG (Deutsche Forschungsgemeinschaft) under grant INST 222/1168-1 FUGG.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared*

Task Papers, Day 1), pages 1–61, Florence, Italy. Association for Computational Linguistics.

Mauro Cettolo, an Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, US.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *CoRR*, abs/1711.02281.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with SMT features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 151–157. AAAI Press.

Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean. 2013. Multilingual acoustic models using distributed deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8619–8623. IEEE.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2134–2139, Copenhagen, Denmark. Association for Computational Linguistics.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

Surafel Melaku Lakew, Mattia Antonino Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. *CoRR*, abs/1910.10408.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921, Denver, Colorado. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain. Association for Computational Linguistics.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Belgium, Brussels. Association for Computational Linguistics.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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 *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. 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, Belgium, Brussels. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? 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 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019,*

Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 291–297, Valencia, Spain. Association for Computational Linguistics.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. 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 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Rongxiang Weng, Shujian Huang, Zaixiang Zheng, Xinyu Dai, and Jiajun Chen. 2017. Neural machine translation with word predictions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 136–145, Copenhagen, Denmark. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.