Look Harder: A Neural Machine Translation Model with Hard Attention

Sathish Indurthi **Insoo Chung** **Sangha Kim**
Samsung Research, Seoul, South Korea
{s.indurthi, insooo.chung, sangha01.kim}@samsung.com

Abstract

Soft-attention based Neural Machine Translation (NMT) models have achieved promising results on several translation tasks. These models attend all the words in the source sequence for each target token, which makes them ineffective for long sequence translation. In this work, we propose a hard-attention based NMT model which selects a subset of source tokens for each target token to effectively handle long sequence translation. Due to the discrete nature of the hard-attention mechanism, we design a reinforcement learning algorithm coupled with reward shaping strategy to efficiently train it. Experimental results show that the proposed model performs better on long sequences and thereby achieves significant BLEU score improvement on English-German (EN-DE) and English-French (EN-FR) translation tasks compared to the soft-attention based NMT.

1 Introduction

In recent years, soft-attention based neural machine translation models (Bahdanau et al., 2015; Gehring et al., 2017; Hassan et al., 2018) have achieved state-of-the-art results on different machine translation tasks. The soft-attention mechanism computes the context (encoder-decoder attention) vector for each target token by weighting and combining all the tokens of the source sequence, which makes them ineffective for long sequence translation (Lawson et al., 2017). Moreover, weighting and combining all the tokens of the source sequence may not be required – a few relevant tokens are sufficient for each target token.

Different attention mechanisms have been proposed to improve the quality of the context vector. For example, Luong et al. (2015); Yang et al. (2018) proposed a local-attention mechanism to selectively focus on a small window of source tokens to compute the context vector. Even though local-attention has improved the translation quality, it is not flexible enough to focus on relevant tokens when they fall outside the specified window size.

To overcome the shortcomings of the above approaches, we propose a hard-attention mechanism for a deep NMT model (Vaswani et al., 2017). The proposed model solely selects a few relevant tokens across the entire source sequence for each target token to effectively handle long sequence translation. Due to the discrete nature of the hard-attention mechanism, we design a Reinforcement Learning (RL) algorithm with reward shaping strategy (Ng et al., 1999) to train it. The proposed hard-attention based NMT model consistently outperforms the soft-attention based NMT model (Vaswani et al., 2017), and the gap grows as the sequence length increases.

2 Background

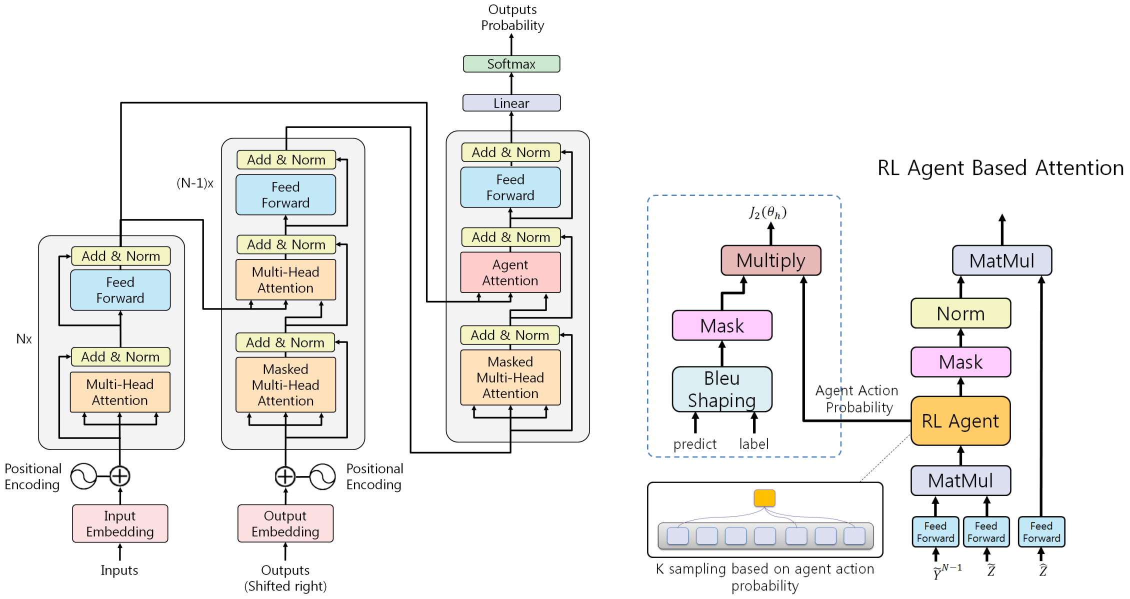
A typical NMT model based on encoder-decoder architecture generates a target sequence $\boldsymbol{y} = \{y_1, \cdots, y_n\}$ given a source sequence $\boldsymbol{x} = \{x_1, \cdots, x_m\}$ by modeling the conditional probability $p(\boldsymbol{y}|\boldsymbol{x}, \theta)$. The encoder ($\theta_e$) computes a set of representations $\boldsymbol{Z} = \{\boldsymbol{z_1}, \cdots, \boldsymbol{z_m}\} \in \mathbb{R}^{m \times d}$ corresponding to $\boldsymbol{x}$ and the decoder (θ_d) generates one target word at a time using the context vector computed using $\boldsymbol{Z}$. It is trained on a set of D parallel sequences to maximize the log likelihood:

$$J_1(\theta) = \frac{1}{N} \sum_{i=1}^{D} \log p\left(\boldsymbol{y}^i | \boldsymbol{x}^i; \theta\right), \qquad (1)$$

where $\theta = \{\theta_e, \theta_d\}$.

In recent years, among all the encoder-decoder architectures for NMT, Transformer (Vaswani et al., 2017) has achieved the best translation quality (Wu et al., 2018). The encoder and decoder blocks of the Transformer are composed of a stack

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3037–3043
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics



(a) (b)

Figure 1: (a) Overview of hard-attention based Transformer network. (b) Overview of RL agent based hard-attention and objective function.

of N (=6) identical layers as shown in Figure 1a. Each layer in the encoder contains two sub-layers, a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Each decoder layer consists of three sub-layers; the first and third sub-layers are similar to the encoder sub-layers, and the additional second sub-layer is used to compute the encoder-decoder attention (context) vector based on the soft-attention based approaches (Bahdanau et al., 2015; Gehring et al., 2017). Here we briefly describe the soft computation of encoder-decoder attention vector in the Transformer architecture. Please refer Vaswani et al. (2017) for the detailed architecture.

For each target word $\hat{\boldsymbol{y}}_t$, the second sub-layer in the decoder computes encoder-decoder attention $\boldsymbol{a}_t$ based on the encoder representations, $\boldsymbol{Z}$. In practice we compute the attention vectors simultaneously for all the time steps by packing $\hat{\boldsymbol{y}}_t$'s and $\boldsymbol{z}_i$'s in to matrices. The soft attention of the encoder-decoder, $\boldsymbol{A}^i$, for all the decoding steps is computed as follows:

$$\boldsymbol{A}^i(\hat{\boldsymbol{Y}}^{i-1}, \boldsymbol{Z}) = \text{softmax}\left(\frac{\hat{\boldsymbol{Y}}^{i-1}\boldsymbol{Z}}{\sqrt{d}}\right)\boldsymbol{Z}, \forall i \in N, \tag{2}$$

where d is the dimension and $\hat{\boldsymbol{Y}}^{i-1} \in \mathbb{R}^{n \times d}$ is the decoder output from the previous layer.

3 Proposed Model

Section 3.1 introduces our proposed hard-attention mechanism to compute the context vector for each target token. We train the proposed model by designing a RL algorithm with reward shaping strategy - described in Section 3.2.

3.1 Hard Attention

Instead of computing the weighted average over all the encoder output as shown in Eq. 2, we specifically select a subset of encoder outputs ($\boldsymbol{z}_i$'s) for the last layer (N) of the decoder using the hard-attention mechanism as shown in Figure 1a. This allows us to efficiently compute the encoder-decoder attention vector for long sequences. To compute the hard-attention between the last layers of the Transformer encoder-decoder blocks, we replace the second sub-layer of the decoder block's last layer with the RL agent based attention mechanism. Overview of the proposed RL agent based attention mechanism is shown in Figure 1b and computed as follows:

First, we learn the projections $\tilde{\boldsymbol{Y}}^{N-1}, \tilde{\boldsymbol{Z}}$ for $\hat{\boldsymbol{Y}}^{N-1}$ and $\boldsymbol{Z}$ as,

$$\begin{aligned}\tilde{\boldsymbol{Y}}^{N-1} &= \tanh(\boldsymbol{W}_2^d(\boldsymbol{W}_1^d\hat{\boldsymbol{Y}}^{N-1} + \boldsymbol{b}_1^d) + \boldsymbol{b}_2^d),\\ \tilde{\boldsymbol{Z}} &= \tanh(\boldsymbol{W}_2^e(\boldsymbol{W}_1^e\boldsymbol{Z} + \boldsymbol{b}_1^e) + \boldsymbol{b}_2^e).\end{aligned}$$

We then compute the attention scores $\boldsymbol{S}$ as,

$$\boldsymbol{S}(\tilde{\boldsymbol{Y}}^{N-1}, \tilde{\boldsymbol{Z}}) = \tilde{\boldsymbol{Y}}^{N-1}\tilde{\boldsymbol{Z}}. \tag{3}$$

We apply the hard-attention mechanism on attention scores ($\boldsymbol{S}$) to dynamically choose multiple relevant encoder tokens for each decoding token. Given $\boldsymbol{S}$, this mechanism generates an equal length of binary random-variables, $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_m\}$ for each target token, where $\beta_i = 1$ indicates that $\boldsymbol{z_i}$ is relevant whereas $\beta_i = 0$ indicates that $\boldsymbol{z_i}$ is irrelevant. The relevant tokens are sampled using *bernoulli* distribution over each β_i for all the target tokens. This hard selection of encoder outputs introduces discrete latent variables and estimating them requires RL algorithms. Hence, we design the following reinforcement learner policy for the hard-attention for each decoding step t.

$$\boldsymbol{\pi_t}(r|\boldsymbol{s}^t, \boldsymbol{\theta_h}) = \beta_i^t \tag{4}$$

where $\beta_i^t \in \boldsymbol{\beta}$ represents the probability of a encoder output (agent's action) being selected at time t, and $\boldsymbol{s}^t \in \boldsymbol{S}$ is the state of the environment. Now, the hard encoder-decoder attention, $\tilde{A}$, is calculated as, follows:

$$\hat{\boldsymbol{Z}} = \tanh(\boldsymbol{W}_{\boldsymbol{2}}^{\hat{e}}(\boldsymbol{W}_{\boldsymbol{1}}^{\hat{e}}\boldsymbol{Z} + \boldsymbol{b}_{\boldsymbol{1}}^{\hat{e}}) + \boldsymbol{b}_{\boldsymbol{2}}^{\hat{e}}) \tag{5}$$

$$\tilde{\boldsymbol{A}} = \beta\hat{\boldsymbol{Z}} \tag{6}$$

Unlike the soft encoder-decoder attention A in Eq. 2, which contains the weighted average of entire encoder outputs, the hard encoder-decoder attention $\tilde{A}$ in Eq. 6 contains information from only relevant encoder outputs for each decoding step.

3.2 Strategies for RL training

The model parameters come from the encoder, decoder blocks and reinforcement learning agent, which are denoted as $\boldsymbol{\theta_e}, \boldsymbol{\theta_d}$ and $\boldsymbol{\theta_h}$ respectively. Estimation of $\boldsymbol{\theta_e}$ and $\boldsymbol{\theta_d}$ is done by using the objective J_1 in Eq. 1 and gradient descent algorithm. However, estimating $\boldsymbol{\theta_h}$ is difficult given their discrete nature. Therefore, we formulate the estimation of $\boldsymbol{\theta_h}$ as a reinforcement learning problem and design a reward function over it. An overveiw of the proposed RL training is given in Algorithm 1.

We use BLEU (Papineni et al., 2002) score between the predicted sequence and the target sequence as our reward function, denoted as $R(\boldsymbol{y}', \boldsymbol{y})$, where $\boldsymbol{y}'$ is the predicted output sequence. The objective is to maximize the reward with respect to the agent's action in Eq. 4 and defined as,

$$J_2(\boldsymbol{\theta_h}) = \sum_{t=1}^{n} \mathrm{log}p(r|\boldsymbol{s}^t, \boldsymbol{\theta_h})R(\boldsymbol{y}', \boldsymbol{y}) \tag{7}$$

Algorithm 1: Hard Attention based NMT

```
Input: Training examples, ${\boldsymbol{x_i}, \boldsymbol{y_i}}_{i=1}^{L}$,
  hyperparameters such as learning rate ($\alpha$), $\lambda$
Initialize model parameters $\theta = \{\theta_e, \theta_h, \theta_d\}$
while not done do
    Sample $k$ training examples
    Compute attention scores $S$ using Eq. 3
    for each decoding step do
        Compute the policy using Eq. 4 to
          select the relevant source sequence
          tokens
        Compute the reward $r_t$ using Eq. 8
    end
    Compute $J(\theta) = -(J_1(\theta_e, \theta_d) + J_2(\theta_h))$
      using Eq. 1 and Eq. 9
    Update the parameters $\theta$ with gradient
      descent: $\theta' = \theta - \alpha\nabla J(\theta)$
end
Return: $\theta$
```

Reward Shaping To generate the complete target sentence, the agent needs to take actions at each target word, but only one reward is available for all these tens of thousands of actions. This makes RL training inefficient since the same terminal reward is applied to all the intermediate actions. To overcome this issue we adopt reward shaping strategy of Ng et al. (1999). This strategy assigns distinct rewards to each intermediate action taken by the agent. The intermediate reward, denoted as $r_t(y_t', y)$, for the agent action at decoding step t is computed as:

$$r_t(\boldsymbol{y_t'}, \boldsymbol{y}) = R(\boldsymbol{y_{1..t}'}, \boldsymbol{y}) - R(\boldsymbol{y_{1..t-1}'}, \boldsymbol{y}) \tag{8}$$

During training, we use cumulative reward $\left(\sum_{t=1}^{n} r_t(\boldsymbol{y_t'}, \boldsymbol{y})\right)$ achieved from the decoding step t to update the agent's policy.

Entropy Bonus We add entropy bonus to avoid policy to collapse too quickly. The entropy bonus encourages an agent to take actions more unpredictably, rather than less so. The RL objective function in Eq. 7 becomes,

$$\hat{J}_2(\boldsymbol{\theta_h}) = J_2(\boldsymbol{\theta_h}) + \lambda H(\boldsymbol{\pi_t}(r|\boldsymbol{s}^t, \boldsymbol{\theta_h})) \tag{9}$$

We approximate the gradient $\Delta_{\boldsymbol{\theta_h}}\hat{J}_2(\boldsymbol{\theta_h})$ by using REINFORCE (Williams, 1992) algorithm which allows us to jointly train $J_1(\boldsymbol{\theta_e}, \boldsymbol{\theta_d})$ and $\hat{J}_2(\boldsymbol{\theta_h})$.

Architecture	Model	BLEU	
		EN-DE	EN-FR
Vaswani et al. (2017)	Transformer_big	28.40	41.00
Wu et al. (2018)	Transformer_big + sequence-loss	28.75	41.47
Yang et al. (2018)	Transformer_big + localness	28.89	n/a
this work	Transformer_big + hard-attention	**29.29**	**42.26**

Table 1: Performance of various models on EN-DE and EN-FR translation tasks.

	# Sequences in each group						
	1-10	11-20	21-30	31-40	41-50	51-60	$\geq$ 61
EN-DE	469	1148	796	383	160	40	8
EN-FR	235	765	796	596	396	165	90

Table 2: Number of sequences present in each group (based on sequence length) of EN-DE and EN-FR testsets.

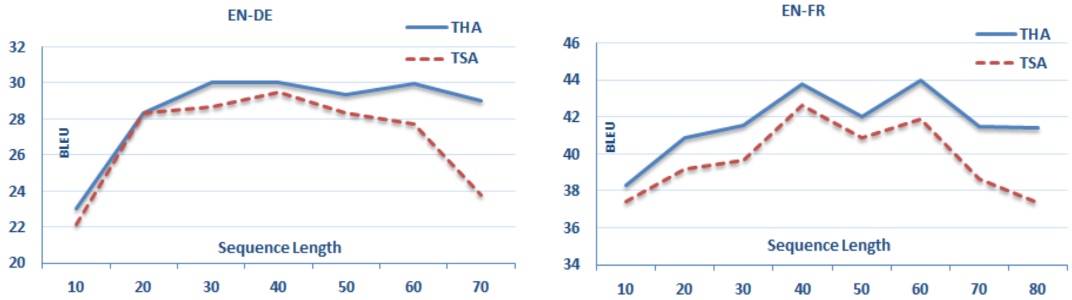


Figure 2: Performance of Transformer with Soft-Attention (TSA), and Transformer with Hard Attention (THA) for various sequence lengths on EN-DE and EN-FR translation tasks.

4 Experimental Results

4.1 Datasets

We conduct experiments on WMT 2014 English-German (EN-DE) and English-French (EN-FR) translation tasks. The approximate number of training pairs in EN-DE and EN-FR datasets are 4.5M and 36M respectively; *newstest2013* and *newstest2014* are used as the dev and test sets. We follow the similar preprocessing steps as described in Vaswani et al. (2017) for both the datasets. We encode the sentences using word-piece vocabulary (Wu et al., 2016) and the shared source-target vocabulary size is set to 32000 tokens.

4.2 Results

4.3 Implementation Details

We adopt the implementation of the Transformer (Vaswani et al., 2018) with *transformer_big* settings. All the models are trained using 8 NVIDIA Tesla P40 GPUs on a single machine. The BLEU score used in the reward shaping is calculated similarly to Bahdanau et al. (2017); all the n-gram counts start from 1, and the resulting score is multiplied by the length of the target reference sentence. The beam search width (=4) is set empirically based on the dev set performance and λ in Eq. 9 is set to 1e-3 in all the experiments.

Models We compare the proposed model with the soft-attention based Transformer model(Vaswani et al., 2017). To check whether the performance improvements are coming from the hard-attention mechanism (Eq. 4) or from the sequence reward incorporated in the objective function (Eq. 7), we compare our work with previously proposed sequence loss based NMT method (Wu et al., 2018). This NMT method is built on top of the Transformer model and trained by combing cross-entropy loss and sequence reward (BLEU score). We also compare our model with the recently proposed Localness Self-Attention network (Yang et al., 2018) which incorporates a localness bias into the Transformer attention distribution to capture useful local context.

Main Results Table 1 shows the performance of various models on EN-DE and EN-FR translation tasks. These test set case sensitive BLEU scores are obtained using SacreBLEU toolkit[1] (Post, 2018). The BLEU score difference between our hard-attention based Transformer model and the original soft-attention based Transformer model indicates the effectiveness of selecting a few relevant source tokens for each target token. The performance gap between our method and sequence loss based Transformer (Wu et al., 2018) shows that the improvements are indeed coming from the hard-attention mechanism. Our approach of incorporating hard-attention into decoder's top self-attention layer to select relevant tokens yielded better results compared to the Localness Self-Attention (Yang et al., 2018) approach of incorporating localness bias only to lower self-attention layers. It can be noted that our model achieved 29.29 and 42.26 BLEU points on EN-DE and EN-FR tasks respectively – surpassing the previously published models.

Analysis To see the effect of the hard-attention mechanism for longer sequences, we group the sequences in the test set based on their length and compute the BLEU score for each group. Table 2 shows the number of sequences present in each group. Figure 2 shows that Transformer with hard attention is more effective in handling the long sequences. Specifically, the performance gap between our model (THA) and the original Transformer model (TSA) grows bigger as sequences become longer.

5 Related Work

Even though RL based models are difficult to train, in recent years, multiple works (Mnih et al., 2014; Choi et al., 2017; Yu et al., 2017; Narayan et al., 2018; Sathish et al., 2018; Shen et al., 2018) have shown to improve the performance of several natural language processing tasks. Also, it has been used in NMT (Edunov et al., 2018; Wu et al., 2017; Bahdanau et al., 2017) to overcome the inconsistency between the token level objective function and sequence-level evaluation metrics such as BLEU. Our approach is also related to the method proposed by Lei et al. (2016) to explain the decision of text classifier. However, here we focus on selecting a few relevant tokens from a source sequence in a translation task.

Recently, several innovations are proposed on top of the Transformer model to improve performance and training speed. For example, Shaw et al. (2018) incorporated relative positions and Ott et al. (2018) proposed efficient training strategies. These improvements are complementary to the proposed method. Incorporating these techniques will further improve the performance of the proposed method.

6 Conclusion

In this work, we proposed a hard-attention based NMT model which focuses solely on a few relevant source sequence tokens for each target token to effectively handle long sequence translation. We train our model by designing an RL algorithm with the reward shaping strategy. Our model sets new state-of-the-art results on EN-DE and EN-FR translation tasks.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *Proceedings of the Fifth International Conference on Learning Representations, ICLR-2017*.

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

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. 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 355–364. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming

[1] https://github.com/mjpost/sacrebleu

Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Dieterich Lawson, George Tucker, Chung-Cheng Chiu, Colin Raffel, Kevin Swersky, and Navdeep Jaitly. 2017. Learning hard alignments with variational inference. *CoRR*, abs/1705.05524.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2204–2212, Cambridge, MA, USA. MIT Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759. Association for Computational Linguistics.

Andrew Y Ng, Daishi Harada, and Stuart J Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. pages 278–287.

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, Belgium, Brussels. 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*, ACL '02, pages 311–318, Stroudsburg, PA, USA. 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. Association for Computational Linguistics.

Indurthi Sathish, Seunghak Yu, Seohyun Back, and Heriberto Cuayahuitl. 2018. Cut to the chase: A context zoom-in network for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 570–575. 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.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced self-attention network: A hybrid of hard and soft attention for sequence modeling. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 4345–4352. AAAI Press.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

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.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621. Association for Computational Linguistics.

Lijun Wu, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Sequence prediction with unlabeled data by reward function learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3098–3104.

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.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458. Association for Computational Linguistics.

Adams Wei Yu, Hongrae Lee, and Quoc Le. 2017. Learning to skim text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1880–1890. Association for Computational Linguistics.