Controllable Text Simplification with Deep Reinforcement Learning

Daiki Yanamoto†**, Tomoki Ikawa**†**, Tomoyuki Kajiwara**†**,**
Takashi Ninomiya†**, Satoru Uchida**‡**, Yuki Arase***
† Ehime University, ‡ Kyushu University, * Osaka University
{yanamoto@ai., ikawa@ai., kajiwara@, ninomiya@}cs.ehime-u.ac.jp
uchida@flc.kyushu-u.ac.jp, arase@ist.osaka-u.ac.jp

Abstract

We propose a method for controlling the difficulty of a sentence based on deep reinforcement learning. Although existing models are trained based on the word-level difficulty, the sentence-level difficulty has not been taken into account in the loss function. Our proposed method generates sentences of appropriate difficulty for the target audience through reinforcement learning using a reward calculated based on the difference between the difficulty of the output sentence and the target difficulty. Experimental results of English text simplification show that the proposed method achieves a higher performance than existing approaches. Compared to previous studies, the proposed method can generate sentences whose grade-levels are closer to those of human references estimated using a fine-tuned pre-trained model.

1 Introduction

Text simplification (Alva-Manchego et al., 2020) is a task of rewriting complex sentences into simpler versions while preserving the meaning. This technology assists people with language disabilities (Carroll et al., 1998), language learners (Petersen and Ostendorf, 2007), and children (De Belder and Moens, 2010) in reading texts and learning a language.

To maximize the effectiveness of text simplification, rewrites should be appropriate for the language ability of the target audience. Therefore, controllable text simplification (Scarton and Specia, 2018; Nishihara et al., 2019; Agrawal et al., 2021), which is controlled to match the target difficulty level, has been actively studied. Controllable text simplification models are trained on a parallel corpus of complex and simple sentences with labels for the target difficulty level, such as Newsela (Xu et al., 2015). Although studies have focused on word-level difficulties (Nishihara et al., 2019; Agrawal et al., 2021), sentence-level difficulties were not taken into account. Therefore, while these methods are effective for local editing, such as word substitution, there is room for improvement for global editing, for example, controlling the sentence length and structure. Such global editing is crucial to improve the controllability of the sentence difficulty.

To address this problem, we propose a controllable text simplification model based on deep reinforcement learning to take advantage of sentence-level objectives. Although deep reinforcement learning has also been used in traditional text simplification[1] (Zhang and Lapata, 2017; Nakamachi et al., 2020), in this study, a novel reward function for difficulty control is designed. Our reward is calculated based on the difference between the difficulty of the generated sentence and the target difficulty level.

Experimental results using Newsela-Auto (Jiang et al., 2020) show that the proposed method can generate sentences whose grade-levels are closer to those of human references estimated using a fine-tuned pre-trained model than the previous methods.

2 Related Work

Scarton and Specia (2018) first tackled controllable text simplification by applying a language control method in multilingual machine translation (Johnson et al., 2017). These methods control the output sentence, *i.e.*, its language and difficulty level, by adding a special token at the beginning of the input sentence. Subsequent studies focusing on controllable text simplification, including the present study, have used special tokens that indicate the target difficulty level.

Nishihara et al. (2019) proposed a training method that, in addition to the special tokens, takes into account the word-level difficulty. They estimate the difficulty of a word based on the target

[1] A task to simplify input sentences freely without setting a target difficulty level.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 398–404

difficulty and word frequency in the training corpus, and weight the cross-entropy loss to promote the generation of words appropriate to the target difficulty level. Agrawal et al. (2021) similarly estimated the word difficulty and edited sentences with a non-autoregressive model to avoid generating difficult words. In contrast to previous studies that considered the word-level difficulty, we improve the controllability of the sentence difficulty by employing the sentence-level difficulty.

In traditional text simplification (Zhang and Lapata, 2017; Nakamachi et al., 2020), deep reinforcement learning has been used to improve the simplicity of the sentences generated using LSTM-based simplification models (Luong et al., 2015). Zhang and Lapata (2017) improved the simplification performance with SARI (Xu et al., 2016), an evaluation metric for text simplification, as a reward. Nakamachi et al. (2020) trained reward models for the grammaticality, synonymity, and simplicity through supervised learning using BERT (Devlin et al., 2019). We also use deep reinforcement learning based on a reward estimated by BERT. However, our approach differs from that of Nakamachi et al. (2020) in two ways. First, we target *controlling* the difficulty of the output sentences. Second, we use a powerful Transformer-based (Vaswani et al., 2017) simplification model, which has been the mainstream in recent years (Zhao et al., 2018; Kajiwara, 2019; Martin et al., 2020; Maddela et al., 2021).

3 Proposed Method

We improve the controllability of the sentence difficulty through reinforcement learning using a reward based on the sentence-level difficulty on a previous controllable text simplification model (Scarton and Specia, 2018). Our model consists of a difficulty estimation model and a simplification model. The former model estimates the difficulty of the generated sentence, and the latter model is trained through reinforcement learning to minimize the difference between the estimated and target difficulties.[2]

3.1 Training Difficulty Estimation Model

Our difficulty estimation is based on a regression model that predicts the difficulty of a sentence. We finetune BERT (Devlin et al., 2019), a Transformer-based (Vaswani et al., 2017) masked language model, to develop a difficulty estimation model.

The loss function is the mean squared error (MSE) of the target difficulty $\mathbf{g} = (g_1, g_2, \dots, g_N)$ and the estimated difficulty $\hat{\mathbf{g}} = (\hat{g}_1, \hat{g}_2, \dots, \hat{g}_N)$:

$$L = \frac{1}{N} \sum_{n=1}^{N} (g_n - \hat{g}_n)^2, \tag{1}$$

where N denotes the batch size.

3.2 Training Simplification Model

Our simplification model is a Transformer-based sequence-to-sequence model (Vaswani et al., 2017). Following Scarton and Specia (2018), we include information regarding the target difficulty level in the input sentence. For example, if the target difficulty level is specified as “3”, a special token “<3>” is attached at the beginning of the input sentence.

We train the simplification model in two steps. First, we train a controllable text simplification model corresponding to Scarton and Specia (2018) during the pretraining step. We then improve the controllability of the sentence difficulty during the reinforcement learning step.

3.2.1 Pretraining

Following Nakamachi et al. (2020), we apply a pretraining with cross-entropy loss to stabilize the reinforcement learning. Letting $\mathbf{x}$ be a complex source sentence and $\mathbf{y} = (y_1, y_2, \dots, y_M)$ be a simple target sentence of length M, the loss function is as follows:

$$L_c = -\frac{1}{M} \sum_{m=1}^{M} \log p(y_m | \mathbf{y}_{<m}, \mathbf{x}). \tag{2}$$

3.2.2 Reinforcement Learning

We finetune the pretrained simplification model through deep reinforcement learning using the REINFORCE algorithm (Williams, 1992). Our reward is calculated based on the estimated difficulty of the generated sentence by the simplification model and the target difficulty assigned to the input sentence. It was designed such that a smaller difference between these difficulties results in a larger reward.

First, the difficulty estimation model receives the sentences generated by the simplification model and outputs the estimated difficulty $\hat{g}$. Based on this estimated difficulty $\hat{g}$ and target difficulty g, the squared error $e = (g - \hat{g})^2$ is calculated.

[2] We employed the K-12 grade levels in Newsela (Xu et al., 2015). Following previous studies (Scarton and Specia, 2018; Nishihara et al., 2019), we assume that the level of a sentence is equal to the level of the document containing that sentence.

Next, based on the maximum and minimum values of error e, *i.e.*, $e_{\max}$ and $e_{\min}$, we transform e into a reward r by applying the following normalization:

$$r = \frac{r_{\max} - r_{\min}}{e_{\min} - e_{\max}}(e - e_{\max}) + r_{\min}, \quad (3)$$

where $r_{\min}$ and $r_{\max}$ are the lower and upper bounds of the reward, respectively. This normalization gives a larger reward close to the maximum reward $r_{\max}$ as the squared error e decreases.

Finally, we use reward r to weigh the cross-entropy loss in Equation (2):

$$L_r = -r \cdot \frac{1}{M} \sum_{m=1}^{M} \log p(y_m | \mathbf{y}_{<m}, \mathbf{x}). \quad (4)$$

4 Experiments

4.1 Dataset

We used a parallel corpus for English controllable text simplification, Newsela-Auto[3] (Jiang et al., 2020). Following the official setup, we used this dataset for the training, validation, and test sets shown in Table 1. The difficulty estimation model uses pairs of sentences and difficulty labels rather than parallel sentence pairs. We used both the source and target sentences, removing the sentence overlap[4].

	Train	Valid	Test
Difficulty Estimation Model	236,773	28,921	29,381
Simplification Model	394,300	43,317	44,067

Table 1: Number of sentences in the training, validation, and test sets. Note that the difficulty estimation model is trained using sentences, whereas the simplification model is trained using sentence pairs.

4.2 Implementation Details

For the difficulty estimation model, we used BERT[5] (Devlin et al., 2019). We used HuggingFace Transformers (Wolf et al., 2020) to fine-tune it for 5 epochs with a batch size of 32 sentences, and Adam (Kingma and Ba, 2015) optimizer. The learning rate was set to $5e-5$ and decreased linearly to zero at the end of the training. The model with the smallest MSE was selected after every $1,000$ steps of the evaluation conducted using the validation set. Although we also trained RoBERTa[6] (Liu et al., 2019) and ALBERT[7] (Lan et al., 2020) under the same settings, we chose BERT, which achieved the lowest MSE[8] in our preliminary experiments.

For the simplification model, we used Transformer (Vaswani et al., 2017) with Reinforce-Joey[9] (Kiegeland and Kreutzer, 2021) for reinforcement learning. This model consists of 6 layers, 8 attention heads, 512 dimensions for the embedding layers, $2,048$ dimensions for the feed forward layers, and a Dropout rate of 0.1. We shared the weights of all embedding layers. As a preprocessing step, we tokenized the corpus using SentencePiece[10] (Kudo and Richardson, 2018) with a vocabulary size of $30,000$.

We pretrained the model for 20 epochs with a minibatch of $6,000$ tokens, and Adam optimizer. We set the learning rate to $1e-8$ and used the learning scheduling applied by Vaswani et al. (2017) with $4,000$ warmup steps. The model with the largest SARI (Xu et al., 2016) was selected after every $1,000$ steps of evaluation using the validation set.

We then conducted reinforcement learning for 10 epochs with a minibatch of 240 tokens and the Adam optimizer whose learning rate was fixed at $1e-8$. The model was selected in the same way as for the pretraining, using $6,000$ steps. Following Kiegeland and Kreutzer (2021), in Equation (3), we set $r_{\min} = -0.5$ and $r_{\max} = 0.5$, respectively.

4.3 Comparative Methods

We compare four types of Transformer-based simplification models: a model without the target difficulty (`base`), a controllable model with the target difficulty level attached to the beginning of the input sentence (`base+grade`) (Scarton and Specia, 2018), a controllable model trained while taking into account the word-level difficulty (`base+grade+word`) (Nishihara et al., 2019), and the proposed model (`base+grade+sent`).

[3] https://github.com/chaojiang06/wiki-auto

[4] The deduplication process reduced the training, validation, and test sets for the difficulty estimation model.

[5] https://huggingface.co/bert-base-cased

[6] https://huggingface.co/roberta-base

[7] https://huggingface.co/albert-base-v2

[8] In our test set, BERT, RoBERTa, and ALBERT had MSE of 3.32, 3.37, and 3.36, respectively.

[9] https://github.com/samuki/reinforce-joey

[10] https://github.com/google/sentencepiece

Models	Automatic Evaluation SARI	add	keep	del	Human Evaluation Grammar	Meaning	Simplicity (↓)
`base`	37.51	3.04	38.64	70.85	3.53	**2.54***	0.046
`base+grade`	41.10	3.35	42.90	77.04	3.53*	2.32*	0.087*
`base+grade+word`	41.50	**3.44**	**42.97**	78.07	**3.62**	2.34*	0.030
`base+grade+sent` (ours)	**41.96**	3.41	42.22	**80.24**	3.59	2.08	**−0.013**

Table 2: Results on the Newsela-Auto test set. Here, add, keep, and del are the F1 scores for each adding, keeping, and deletion operations of word 4-grams that comprise SARI. (*: significant at $p < 0.05$ between `base+grade+sent` and others for paired-sample t-test.)

Grade level	`base`	`base+grade`	`base+grade+word`	`base+grade+sent`	References
8	6.79 (1.83)	**7.75 (1.47)**	8.02 (1.52)	7.67 (1.54)	7.98 (0.92)
7	6.15 (1.71)	7.43 (1.43)	7.40 (1.42)	**6.90 (1.34)**	6.84 (1.26)
6	5.90 (1.41)	6.61 (1.39)	6.52 (1.34)	**6.04 (1.25)**	6.12 (1.14)
5	5.81 (1.39)	5.73 (1.21)	5.66 (1.18)	**5.24 (1.00)**	5.23 (0.90)
4	5.38 (1.53)	4.70 (0.93)	4.54 (0.82)	**4.40 (0.73)**	4.56 (0.78)
3	5.15 (2.15)	4.04 (1.04)	4.00 (1.00)	**3.87 (0.87)**	4.07 (1.07)
2	4.93 (2.93)	3.85 (1.85)	3.79 (1.79)	**3.74 (1.75)**	3.78 (1.78)
All	5.48 (1.74)	5.07 (1.15)	4.98 (1.09)	**4.73 (0.98)**	4.84 (1.01)

Table 3: Average estimated difficulty of the sentences generated for each target difficulty. The numbers in parentheses are the MAE between the target and estimated difficulties. The lowest errors are highlighted in bold, except for the references.

4.4 Automatic Evaluation

Table 2 shows the automatic evaluation results. For the overall simplification quality, we evaluated SARI (Xu et al., 2016) using the EASSE toolkit[11] (Alva-Manchego et al., 2019). The proposed method achieved the best performance with SARI. The F1 scores evaluating the addition, keeping, and deletion operations of word 4-grams show that the proposed method improves the addition (add) and deletion (del) compared to the `base+grade` model. This result implies that the proposed method actively paraphrases complex expressions into simpler versions.

4.5 Human Evaluation

To assess the quality of the generated sentences and the controllability of the sentence difficulty, we conducted a human evaluation for 100 sentences randomly selected from the test set. Grammaticality (grammar) and meaning preservation (meaning) were evaluated on a 4-point scale according to Xu et al. (2016). For evaluating controllability of the sentence difficulty, the output and reference sentences were compared and ranked in terms of their simplicity. Here, we allowed the same ranking between sentences with no clear difference in simplicity. Note that a simpler sentence, *i.e.*, having a lower ranking, does not necessarily mean a better sentence. We evaluated the difference between the rank of the reference sentence and that of the output sentence. That is, the smaller the difference in the simplicity ranking, the better the model successfully controls the difficulty of the sentence. We hired five human evaluators through a crowdsourcing service.[12] The evaluators were master workers, US residents, and had a minimum approval rate of 95%.[13]

The right side of Table 2 shows the average scores of the human evaluations. The proposed method achieved the best controllability of sentence difficulty with some cost in meaning.

4.6 Analysis: Difficulty of Output Sentences

For a detailed analysis of the sentence-level difficulty, Table 3 shows the difficulty of the generated sentences for each target difficulty level.[14] To obtain the average difficulty of the generated sen-

[11] `https://github.com/feralvam/easse`

[12] `https://www.mturk.com/`

[13] We estimated the hourly rate to be about \$8 and paid a total of \$150 for crowdworkers.

[14] The Newsela-Auto test set (Jiang et al., 2020) does not include target difficulty levels higher than 8.

source	The burning of fossil fuels, such as coal, oil and gas, creates greenhouse gases that heat up the Earth and change the climate.
`base+grade+sent`	The burning of oil and gas makes the Earth warm.
reference	These gases are getting trapped in the air and heating up the Earth.
source	"It's more of a family than living outside the base," said Jessica Konczal, 33, whose husband is Sergeant Matthew Konczal.
`base+grade+sent`	"It's more of a family than living outside the base," said Jessica Konczal.
reference	Jessica Konczal is 33 and lives on the base.

Table 4: Example output sentences.

tences, we used the difficulty estimation model described in Section 3.1. The numbers in parentheses are the MAE between the target and estimated difficulties. Among them, our `base+grade+sent` model achieved the lowest MAE for all target difficulties except the most difficult level of 8.

4.7 Analysis: Quality of Output Sentences

We analyze the trade-off between synonymity and simplicity in the human evaluation of our model. Example output sentences from the proposed method are shown in Table 4. Our model tends to output shorter sentences by reducing the content from the input sentences to gain simplicity. In other words, our output sentences do not guarantee a "perfect" semantic correspondence with the input sentences. However, such semantic omissions are often found even in references made by professional writers at Newsela. Specifically, 70% of the reference sentences omit more than one quarter of the words of the input sentence, and 44% delete more than half of the words. As the examples in Table 4 show, our output sentences remove supplemental details but preserve the main content.

5 Conclusion

We proposed a deep reinforcement learning method for controllable text simplification that takes into account the sentence-level difficulty. We designed a reward based on the difference between the target difficulty and that of the generated sentence. Experimental results show that our method is evaluated highly owing to its overall simplification in an automatic evaluation, and for its controllability of the sentence difficulty in a manual evaluation.

Acknowledgement

This work was supported by JSPS KAKENHI, Grant Number JP21H03564.

References

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A Non-Autoregressive Edit-Based Approach to Controllable Text Simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier Automatic Sentence Simplification Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 49–54.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-Driven Sentence Simplification: Survey and Benchmark. *Computational Linguistics*, 46(1):135–187.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical Simplification of English Newspaper Text to Assist Aphasic Readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.

Jan De Belder and Marie-Francine Moens. 2010. Text Simplification for Children. In *Proceedings of the SIGIR 2010 Workshop on Accessible Search Systems*, pages 19–26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF Model for Sentence Alignment in Text Simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Tomoyuki Kajiwara. 2019. Negative Lexically Constrained Decoding for Paraphrase Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052.

Samuel Kiegeland and Julia Kreutzer. 2021. Revisiting the Weaknesses of Reinforcement Learning for Neural Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1673–1681.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Taku Kudo and John Richardson. 2018. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the 8th International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*.

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.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable Text Simplification with Explicit Paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable Sentence Simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698.

Akifumi Nakamachi, Tomoyuki Kajiwara, and Yuki Arase. 2020. Text Simplification with Reinforcement Learning Using Supervised Rewards on Grammaticality, Meaning Preservation, and Simplicity. In *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: Student Research Workshop*, pages 153–159.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable Text Simplification with Lexical Constraint Loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266.

Sarah E Petersen and Mari Ostendorf. 2007. Text Simplification for Language Learners: A Corpus Analysis. In *Proceedings of the Workshop on Speech and Language Technology in Education*, pages 69–72.

Carolina Scarton and Lucia Specia. 2018. Learning Simplifications for Specific Target Audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 712–718.

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.

Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Xingxing Zhang and Mirella Lapata. 2017. Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating Transformer and Paraphrase Rules for Sentence Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173.