

Controllable Text Simplification with Lexical Constraint Loss

Daiki Nishihara[†], Tomoyuki Kajiwara[‡], Yuki Arase^{†*}

[†]Graduate School of Information Science and Technology, Osaka University

[‡]Institute for Datability Science, Osaka University

^{*}Artificial Intelligence Research Center (AIRC), AIST

[†]{nishihara.daiki, arase}@ist.osaka-u.ac.jp

[‡]kajiwara@ids.osaka-u.ac.jp

Abstract

We propose a method to control the level of a sentence in a text simplification task. Text simplification is a monolingual translation task translating a complex sentence into a simpler and easier to understand the alternative. In this study, we use the grade level of the US education system as the level of the sentence. Our text simplification method succeeds in translating an input into a specific grade level by considering levels of both sentences and words. Sentence level is considered by adding the target grade level as input. By contrast, the word level is considered by adding weights to the training loss based on words that frequently appear in sentences of the desired grade level. Although existing models that consider only the sentence level may control the syntactic complexity, they tend to generate words beyond the target level. Our approach can control both the lexical and syntactic complexity and achieve an aggressive rewriting. Experiment results indicate that the proposed method improves the metrics of both BLEU and SARI.

1 Introduction

Text simplification (Shardlow, 2014) is the task of rewriting a complex text into a simpler form while preserving its meaning. Its applications include reading comprehension assistance and language education support. Because each target user has different reading abilities and/or knowledge, we need a text simplification system that translates an input sentence into a sentence of an appropriate difficulty level for each user. According to the input hypothesis (Krashen, 1985), educational materials slightly beyond the learner’s level effectively improve their reading abilities. On the contrary, materials that are too difficult for learners deteriorate their learning motivation. In the context of language education, teachers manually simplify

Grade	Examples
12	<u>According to</u> the Pentagon , 152 female troops have been killed while serving in Iraq and Afghanistan .
7	The Pentagon <u>says</u> 152 female troops have been killed while serving in Iraq and Afghanistan .
5	The military says 152 female have died .

Table 1: Example sentences with different grade levels. To control the sentence level, syntactic (underline) and/or lexical (bold) paraphrasing is performed.

sentences for each learner. To reduce the burden on teachers, automatic text simplification systems are desired (Petersen and Ostendorf, 2007).

As mentioned, text simplification translates a complex sentence into a simpler alternative. The transformation allows entailment and omission/replacement of phrases and words. Table 1 shows sentences in different grade levels. Sentence level depends on both the syntactic and lexical complexities. When simplifying a sentence of grade level 12 into grade level 7¹, paraphrasing “According to ~,” to “~ says” reduces the syntactic complexity. In addition, when simplifying the sentence from the grade levels 12 to 5, paraphrasing “Pentagon” to “military” reduces the lexical complexity. Assuming an application to language education, we aim at automatically rewriting the input sentence to accommodate the level of difficulty appropriate for each grade level, as shown in Table 1.

Many previous studies (Specia, 2010; Wubben et al., 2012; Xu et al., 2016; Nisioi et al., 2017; Zhang and Lapata, 2017; Vu et al., 2018;

¹In this study, we use grades K-12.

Guo et al., 2018; Zhao et al., 2018) in text simplification have trained machine translators on a monolingual parallel corpus consisting of complex-simple sentence pairs without considering the level of each sentence. Therefore, these text simplification models are ignorant regarding the sentence level. Scarton and Specia (2018) developed a pioneering text simplification model that can control the sentence level. They trained a text simplification model on a parallel corpus by attaching tags specifying 11 grade levels to each sentence (Xu et al., 2015). The trained model allows the generation of a sentence of a desired level specified by a tag attached to the input. This model may control the syntactic complexity such as the sentence length; however, it often outputs overly difficult words beyond the target grade level. To control the lexical complexity in text simplification, we propose a method for add weights to a training loss according to levels of words on top of (Scarton and Specia, 2018), and thus output only words under the desired level.

Experiment results indicate that the proposed method improves the BLEU and SARI scores by 1.04 and 0.15 compared to Scarton and Specia (2018). Moreover, our detailed analysis indicates that our method controls both the lexical and syntactic complexities and promotes an aggressive rewriting.

2 Related Work

2.1 Text Simplification

Text simplification can be regarded as a monolingual machine translation problem. Previous studies have trained a model to translate complex sentences into simpler sentences on parallel corpora between Wikipedia and Simple Wikipedia (W-SW) (Zhu et al., 2010; Coster and Kauchak, 2011). As in the field of machine translation, early studies (Specia, 2010; Wubben et al., 2012; Xu et al., 2016) were mainly based on a statistical machine translation (Koehn et al., 2007; Post et al., 2013). Inspired by the success of neural machine translation (Bahdanau et al., 2015), recent studies (Nisioi et al., 2017; Zhang and Lapata, 2017; Vu et al., 2018; Guo et al., 2018; Zhao et al., 2018) use the encoder-decoder model with the attention mechanism. These studies do not consider the level of each sentence.

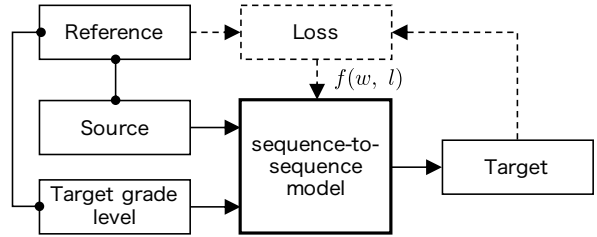


Figure 1: Our method adds a weight to the training loss based on levels of words w and target level l .

2.2 Controllable Text Simplification

In addition to W-SW, Newsela (Xu et al., 2015) is a famous dataset available for text simplification. Newsela is a parallel corpus with 11 grade levels. Scarton and Specia (2018) trained a level-controllable text simplification model on Newsela. Although their model is a standard attentional encoder-decoder model similar to (Nisioi et al., 2017), a special token `<grade>` indicating the grade level of the target sentence is attached to the beginning of the input sentence. This is a promising approach that has been successful in similar tasks (Johnson et al., 2017; Niu et al., 2018). As expected regarding the task of text simplification, this approach has improved both BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016) compared to a baseline model (Nisioi et al., 2017) that does not consider the target level at all. This model allows the syntactic complexity to be controlled; however, it tends to output overly difficult words beyond the target grade level.

3 Loss Function with Word Level

To control the lexical complexity, our model weighs a training loss of a text simplification model considering words that frequently appear in the sentences of a specific grade level, as shown in Figure 1. Here, the weight $f(w, l)$ corresponds to the relevance of the word w at grade level l .

A sequence-to-sequence model commonly uses the cross-entropy loss. When a model outputs $\mathbf{o} = [o_1, \dots, o_N]$ (where N is the size of the vocabulary) at a certain time step, the cross-entropy loss is as follows:

$$L(\mathbf{o}, \mathbf{y}) = -\mathbf{y} \log \mathbf{o}^\top = -\log o_c \quad (1)$$

where $\mathbf{y} = [y_1, \dots, y_N]$ is a one-hot vector in which only the c -th element of a correct word is 1 and others are all 0. Our model adds weights to the loss function (Equation 1) based on the level of

words such that the model learns to output words of the desired level:

$$L'(\mathbf{o}, \mathbf{y}, w, l) = -f(w, l) \cdot \log o_c. \quad (2)$$

As $f(\cdot, \cdot)$, we use TFIDF or PPMI assuming that words frequently appear in sentences of level l also have the same level l .

TFIDF We compute the TFIDF regarding sentences of the same level as a document:

$$\text{TFIDF}(w, l) = P(w | l) \cdot \log \frac{D}{\text{DF}(w)} \quad (3)$$

where $P(w | l)$ is a probability that word w appears in a set of sentences of grade level l , D is the number of grade levels², and $\text{DF}(w)$ is the number of grade levels in which w appears. By so doing, TFIDF provides more weights to words that uniquely appear in the sentences of a specific level.

PPMI Pointwise mutual information (PMI) allows estimating the strength of a co-occurrence between w and l :

$$\text{PMI}(w, l) = \log \frac{P(w | l)}{P(w)}. \quad (4)$$

where $P(w)$ is a probability of word w being within the entire training corpus, whereas $P(w | l)$ is the same as Equation 3. Words with negative PMI scores have a negative correlation against l that means w tends to appear across different sentence levels. Hence, we ignore w with a negative PMI using a positive-PPMI (PPMI) function:

$$\text{PPMI}(w, l) = \max(\text{PMI}(w, l), 0). \quad (5)$$

Both TFIDF and PPMI have a range of $[0, \infty)$, and thus we apply the Laplace smoothing:

$$f(w, l) = \text{Func}(w, l) + 1 \quad (6)$$

$$\text{Func} \in \{\text{PPMI}, \text{TFIDF}\} \quad (7)$$

4 Experiment

4.1 Dataset

We evaluated whether our method can control the grade levels in a text simplification using the Newsela corpus. The Newsela corpus provides

Grade	#Sentences	#Words	S-length
2	953	9,882	10.37
3	3,865	47,211	12.22
4	43,971	618,184	14.06
5	31,918	526,769	16.50
6	19,535	367,319	18.80
7	17,322	356,307	20.57
8	15,446	376,678	24.39
9	7,897	200,242	25.36
10	1,018	30,693	30.15
11	104	2,844	27.35
12	50,799	1,484,625	29.23
All	192,828	4,020,754	20.85

Table 2: Statistics for the Newsela corpus, where S-length shows the average number of words in a sentence.

news articles of different levels, which have been manually rewritten by human experts. It conforms to the grade levels in the US education system, where the levels range from 2 to 12.

We use the publicly available version of the Newsela corpus³ that has been sentence-aligned by Xu et al. (2015) and divided into 94k, 1k, and 1k sentences for the training, development, and test, respectively, by Zhang and Lapata (2017). As in previous studies, we regard each sentence in an article as sharing the same level as the entire article. Zhang and Lapata (2017) first divided the set of articles and then extracted sentence pairs to avoid the same sentences appearing in both the training and test sets. Note that the Newsela corpus used in (Scarton and Specia, 2018) is different from the present corpus, and is preprocessed differently. Due to these differences, the training, development, and test sets used in (Scarton and Specia, 2018) are unreproducible. Therefore, we reimplemented (Scarton and Specia, 2018) and compared it to our method using our public corpus.

Table 2 shows statistics for the Newsela corpus, which clearly present the tendency that lower grade sentences are significantly shorter than those of higher grades. This indicates that aggressive omission of phrases is required to simplify sentences of grade 8 to 12 into those of grade 2 to 7.

²Here, $D = 11$ because we use grade levels 2 to 12.

³<https://newsela.com/data/>

	BLEU \uparrow	SARI \uparrow	BLEU _{ST} \downarrow	MAE _{LEN} \downarrow	MPMI \uparrow
source	21.37	2.82	100.0	10.73	0.08
reference	100.0	70.13	18.30	0.00	0.23
s2s	20.43	28.21	37.60	4.38	0.12
s2s+grade	20.82	29.44	31.96	3.77	0.15
s2s+grade+TFIDF	21.00	29.58	31.56	3.75	0.15
s2s+grade+PPMI	21.86	29.59	31.38	3.69	0.19

Table 3: Results on the Newsela test set.

4.2 Methods for Comparison

During this experiment, the following four methods were compared.

1. **s2s** is a baseline, plain sequence-to-sequence model based on the attention mechanism.
2. **s2s+grade** is our re-implementation of [Scarton and Specia \(2018\)](#), which is a state-of-the-art controllable text simplification.
3. **s2s+grade+TFIDF** is our model (Sec. 3) implemented on s2s+grade, which adds TFIDF-based word weighing to the loss function. TFIDF scores were pre-computed using the training data.
4. **s2s+grade+PPMI** is our other model (Sec. 3) implemented on s2s+grade, which adds PPMI-based word weighing in the loss function. PPMI scores were pre-computed using the training data.

4.3 Implementation Details

In this study, we implemented our model using Marian ([Junczys-Dowmunt et al., 2018](#)).⁴ Both the encoder and decoder consist of 2 layers of Bi-LSTM with the 1,024-dimensions of hidden layers and 512-dimensions of the embedding layer shared by the encoder and decoder including its output layer. Word embedding was randomly initialized. A dropout rate of 0.2 was applied to the hidden layer, and a dropout rate of 0.1 was applied to the embedding layer. Adam was used as an optimizer. Training was stopped when the perplexity measured on the development set stopped improving for 8 epochs.⁵ All scores reported in this experiment are the averages of 3 trials with random initialization.

⁴<https://github.com/marian-nmt/marian/commit/02f4af4>

⁵48.7 epochs on average to train s2s+grade+PPMI.

4.4 Automatic Evaluation Metrics

Following previous studies on text simplification, *e.g.*, [Scarton and Specia \(2018\)](#), BLEU⁶ ([Papineni et al., 2002](#)) and SARI⁷ ([Xu et al., 2016](#)) were used to evaluate the overall performance.

In addition, we investigate the scores of BLEU_{ST}, mean absolute error (MAE) of sentence length (MAE_{LEN}), and mean PMI (MPMI) for a detailed analysis. BLEU_{ST} computes a BLEU score by taking the source and output sentences as input, which allows evaluating the degree of rewrites made by a model. The lower BLEU_{ST} is, the more actively the model rewrites the source sentence.

In addition, MAE_{LEN} approximately evaluates the syntactic complexity of the output based on its length:

$$\text{MAE}_{\text{Len}} = \frac{1}{N} \sum_{\substack{s_R \in \text{Reference} \\ s_T \in \text{Target}}} |\text{Len}(s_R) - \text{Len}(s_T)|, \quad (8)$$

where N is the number of sentences in the test set, and $\text{Len}(\cdot)$ provides the number of words in a sentence. The lower the MAE_{LEN} is, the more appropriate the length of the output.

MPMI evaluates to what extent the levels of the output words match with the target level:

$$\text{MPMI} = \frac{1}{W} \sum_{s \in \text{Target}} \sum_{w \in s} \text{PMI}(w, l_s), \quad (9)$$

where W is the number of words appearing in the output and l_s is the grade level of sentence s . PMI scores were pre-computed using the training data. The higher the MPMI is, the more words of the target level are generated by the model.

⁶We use the multi-bleu-detok.perl script from <https://github.com/moses-smt/ MosesDecoder>

⁷<https://github.com/cocoxu/simplification>

Grade	Examples
Source 12	In its original incarnation during the ‘ 60s , African-American ” freedom songs ” aimed to motivate protesters to march into harm ’s way <u>and , on a broader scale , spread news of the struggle to a mainstream audience .</u>
7	s2s+grade: In the 1960s , African-American ” freedom songs are aimed to motivate protesters to march into harm ’s way . s2s+grade+PPMI: In its original people in the 1960s , African-American ” freedom songs are aimed to inspire protesters to march into harm ’s way .
4	s2s+grade: In the 1960s , African-American ” freedom songs are aimed to motivate protesters to march into harm ’s way . s2s+grade+PPMI: African-American ” freedom songs are aimed to inspire protesters to march into harm ’s way .

Table 4: Example of model outputs. Here, s2s+grade+PPMI successfully simplified some complex words (highlighted in bold) and deleted the underlined phrases.

5 Results and Analysis

5.1 Overall Results

Table 3 shows the experiment results. The first two rows show the performances when the source sentence itself or the reference sentence is regarded as the model output, which sets the standard to interpret the scores.

Our method outperforms the state-of-the-art baseline in both the BLEU and SARI metrics. In particular, s2s+grade+PPMI improved the BLEU and SARI scores by 1.04 and 0.15 compared to s2s+grade, respectively.

An evaluation in BLEU_{ST} shows that our proposed models conduct an aggressive rewriting. In addition, s2s+grade+PPMI, which has the highest performance in both the BLEU and BLEU_{ST} metrics, conducts many appropriate rewrites that are far from the source and close to the reference. The s2s baseline, which does not consider the target level, applies conservative rewriting, whereas the proposed model, which considers it more properly conducts more aggressive rewriting.

The evaluations of MAE_{LEN} and MPMI show that s2s+grade+PPMI can best the control both syntactic and lexical complexities. From these results, we confirmed the effectiveness of the text simplification model that takes the word level into account.

Table 4 shows examples of the model outputs. Here, s2s+grade+PPMI paraphrases a complex word “incarnation” into “people” for grade level 7. In addition, the complex word “motivate” is simplified to “inspire” for grade level 4. Although

Grade	FKGL			MPMI	
	prev.	prop.	diff.	prev.	prop.
<8>	4.92	5.33	+0.41	0.11	0.12
<7>	4.87	5.25	+0.38	0.10	0.12
<6>	4.47	4.56	+0.09	0.12	0.14
<5>	3.51	3.71	+0.20	0.13	0.15
<4>	2.68	2.69	+0.01	0.16	0.19
<3>	2.06	1.89	-0.17	0.18	0.23
<2>	1.81	1.44	-0.37	0.20	0.24
MAE	1.52	1.45	-	-	-

Table 5: FKGL and MPMI of s2s+grade (prev.) and s2s+grade+PPMI (prop.) for each grade level. Models suitable for the target level are highlighted in bold.

both models can remove unimportant phrases “and , on ~”, s2s+grade+PPMI successfully summarized shorter sentences for grade level 4.

5.2 Analysis for Each Grade Level

To analyze the level control in detail, we simplified each source sentence in the test set to all simpler grade levels⁸. This analysis does not allow an evaluation based on references such as BLEU because references are only given for some levels for each source sentence.

Table 5 shows FKGL (Kincaid et al., 1975) and MPMI for each target grade level for s2s+grade (prev.) and s2s+grade+PPMI (prop.). FKGL is

⁸We omitted grade levels <9>-<12> because the sentences with these grade levels do not exist in the reference sentences of the training set.

an automatic evaluation metric that estimates the textual readability. The FKGL scores correspond to grade levels of K-12.

An analysis of the FKGL revealed that both models were oversimplified. However, MAE with the target grade level shows that the proposed model is superior to the baseline model. Focusing on the FKGL differences, the proposed model generates simpler sentences for the simpler target grade levels than the baseline model, and vice versa. These results show that incorporating word levels into the model contributes to a level control in text simplification.

In the evaluation of MPMI, the proposed method consistently outperforms the state-of-the-art baseline at all target levels. As expected, we confirmed that the proposed method for weighting the cross-entropy losses based on PPMI encourages the use of words suitable for the target grade level.

6 Conclusion

We proposed a text simplification method that controls not only the sentence level but also the word level. Our method controls the word level by weighing words in the loss function, which frequently appear in text of a specific grade level. The evaluation results confirmed that our method improved both the BLEU and SARI scores, and achieved an aggressive rewriting compared to [Scarton and Specia \(2018\)](#). A detailed analysis indicated that our method achieved an accurate control of the level in converting the sentences into those of the target level.

In this study, we regard a document and the sentences contained within it to have the same grade level as in previous studies. In practice, however, this assumption may not hold. Although the readability and level in the units of document ([Kincaid et al., 1975](#)) and phrase ([Pavlick and Callison-Burch, 2016](#); [Maddela and Xu, 2018](#)) have been studied, there have been no previous works focusing on the level of the sentences. This direction is an area of our future work.

Acknowledgments

This project is funded by Microsoft Research Asia, Microsoft Japan Co., Ltd., JST ACT-I Grant Number JPMJPR18UB, and JSPS KAKENHI Grant Number JP18K11435.

References

- Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of International Conference on Learning Representations*.
- William Coster and David Kauchak. 2011. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 665–669.
- Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic Multi-Level Multi-Task Learning for Sentence Simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476.
- 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.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 116–121.
- J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Technical report.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Demo and Poster session*, pages 177–180.
- S. D. Krashen. 1985. The Input Hypothesis: Issues and implications. *London: Longman*.
- Mounica Maddela and Wei Xu. 2018. A Word-Complexity Lexicon and A Neural Readability Ranking Model for Lexical Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760.

- Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring Neural Text Simplification Models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 85–91.
- Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-Task Neural Models for Translating Between Styles Within and Across Languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021.
- 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 of the Association for Computational Linguistics*, pages 311–318.
- Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A Paraphrase Database for Simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 143–148.
- Sarah E Petersen and Mari Ostendorf. 2007. Text Simplification for Language Learners: A Corpus Analysis. In *Proceedings of Workshop on Speech and Language Technology in Education*, pages 69–72.
- Matt Post, Juri Ganitkevitch, Luke Orland, Jonathan Weese, Cao Yuan, and Chris Callison-Burch. 2013. Joshua 5.0: Sparser, better, faster, server. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 206–212.
- 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.
- Matthew Shardlow. 2014. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing 2014*, pages 58–70.
- Lucia Specia. 2010. Translating from Complex to Simplified Sentences. In *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language*, pages 30–39.
- Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Sentence Simplification with Memory-Augmented Neural Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 79–85.
- Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence Simplification by Monolingual Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 1015–1024.
- 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, Saptono Andi, and Parmanto Bambang. 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.
- Zhemina Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1353–1361.