ruBTS: Russian Sentence Simplification Using Back-translation

Farit Galeev
Innopolis University
Innopolis, Russia

Marina Leushina
Innopolis University
Innopolis, Russia

Vladimir Ivanov
Innopolis University,
Innopolis, Russia
Kazan Federal University,
Kazan, Russia

{f.galeev,m.leushina,v.ivanov}@innopolis.ru

Abstract

Automatic text simplification is a crucial task enabling to reduce text complexity while preserving meaning. This paper presents our solution to the Russian Sentence Simplification Shared Task (RSSE) based on a back-translation technique. We show that applying the simple back-translation approach for sentence simplification can give competitive results with the other methods without fine-tuning or training.

Keywords: sentence simplification, Russian language, back-translation
DOI: 10.28995/2075-7182-2021-20-259-267

ruBTS: Упрощение предложений с использованием обратного перевода для русского языка

Фарит Галеев
Университет Иннополис

Марина Леушина
Университет Иннополис

Владимир Иванов
Университет Иннополис,
Казанский Федеральный Университет

f.galeev@innopolis.ru
m.leushina@innopolis.ru
v.ivanov@innopolis.ru

Аннотация

Автоматическое упрощение текста является важной задачей, позволяющей снизить сложность текста при сохранении его смысла. В статье представлено наше решение общей задачи по упрощению предложения на русском языке (RSSE), основанное на методе обратного перевода. Мы показываем, что применение простого обратного перевода для упрощения предложений может дать конкурентные результаты с другими методами без какой-либо тонкой настройки или обучения.

Ключевые слова: упрощение предложений, русский язык, обратный перевод

1 Introduction

Text simplification is used to reduce text complexity, either as part of natural language processing or as a stand-alone study area. Simplified texts are more accessible to people with reading and understanding issues, especially non-native speakers, people with related disorders. Text simplification implies modifications at different levels, including stylistic, grammatical, and lexical levels, preserving the text's original meaning. One of the most studied settings is simplifying a text sentence by sentence (sentence-level simplification), traditionally included paraphrasing as a subtask. However, with recent advances in natural language generation, sentence-level simplification combines transformer-based architectures and paraphrases and typically implies text generation as well [16].

Evaluation of modern methods for sentence simplification carried out using English corpora. Evaluation sentence simplification methods on Russian texts is a novel task that was proposed as a part of the

Dialog 2021 Evaluation initiative[1]. The organizers of the Russian Sentence Simplification Shared Task (RuSimpleSentEval, or RSSE) [14] prepared both train and test sets using a crowd-sourcing platform and translated texts from Simple Wikipedia. This paper presents our solution[2] to the proposed task based on a back-translation technique that does not require any fine-tuning and has shown competitive results.

2 Related Works

2.1 Text Simplification: Approaches

The most recent survey on text simplification [16] points out the connection between summarization and simplification and classifies text simplification approaches by the same categories: extractive and abstractive. Extractive approaches are based on selecting information from the text to preserve the most significant parts and drop less informative details. It is mainly used to simplify a significant amount of text and essentially solve text summarization tasks.

Abstractive approaches, contrary to extractive ones, use text generation for creating simplified text. These approaches can be divided into sentence-level and text-level simplification, or both. The main difference is that simplification can happen on only the lexical level by identifying complex words or phrases and replacing them with the more simple substitute. It can involve syntactic simplification, which may split complex grammatical constructions into simpler ones, delete or add information in the text.

However, these techniques can be used simultaneously. One can achieve it by learning simplification from data directly. A way to do that is the sequence to sequence modelling, the method for text-to-text generation, which was first applied for Text Simplification in 2017 by Nisioi et al. [12]. One of the best simplifications works, ACCESS [11], solves sentence simplification task using both lexical and syntactic approaches. AudienCe-CEntric Sentence Simplification aims to control attributes, which correspond to the text complexity: the amount of compression, amount of paraphrasing, lexical complexity, and syntactic complexity. Their solution is based on the transformer model [19], which is trained in a sequence-to-sequence manner.

The transformer is the model that was originally presented for Neural Machine Translation (NMT) [19]. Text Simplification can be considered Monolingual Translation, where the source text would be translated to the more straightforward text. Experiments with NMT techniques and Text Simplification first suggested [21] and conducted [20] in 2016. In the paper, Wang Tong et al. identify several differences between NMT and Text Simplification, which should be addressed while using NMT directly for simplification. It includes differences in vocabulary sizes, shared words in aligned sentences, and difficulties when splitting sentences in two. The LSTM-based model (Long Short-Term Memory) learned how to perform reversing, sorting, and replacement operations (for lexical and grammatical simplification).

Zhang et al. [25] suggest another model based on the monolingual translation and sequence to sequence approach. However, authors use reinforcement learning algorithms to encourage a variety of simplification tricks by rewarding simplicity, relevance, and fluency. The motivation behind this is that most used datasets for simplification contain many copies of the original text as simplifications, which creates an imbalance in the applied simplifications.

Another way to better comprehend simplification with the monolingual translation is to include external knowledge bases to increase learned simplification rules. Zhao et al. [26] suggest two modifications, one of which takes advantage of Simple PPDB (A Paraphrase Database for Simplification) [13], by encouraging the model to apply simplification rules, presented in Simple PPDB.

2.2 Text Simplification: Datasets

One downside of these approaches is that a lot of paired data is required to achieve good results. There are several widely used datasets for English:

[1] http://www.dialog-21.ru/evaluation/; https://github.com/dialogue-evaluation/RuSimpleSentEval

[2] Source code of solution is available at `https://github.com/HiGal/RSSE`

- Simple English Wikipedia: several datasets (Wikipedia - Simple Wikipedia [6], PWKP [27], SS Corpus [5]) constructed by parsing Simple English Wikipedia in pair with regular English Wikipedia to obtained paired sentences.
- Xu et al. [23] point at the problems in Simple Wikipedia, such as not aligned sentences between corresponding articles, target sentences that are not simpler than the source, or just noisy sentences. Thus, they present the Newsela dataset, which contains the data of 1130 news articles, where each article contains five versions (one original and four simplified versions), re-written by editors from Newsela company.
- Xu et al. also presented Turk dataset [24] in 2016, which was collected through a crowdsourced rewriting of English Wikipedia sentences on Amazon Mechanical Turk.

Nevertheless, for other languages, it presents a problem in obtaining such a dataset. For Russian, recent work by Gudkov et al. [2] presents a method for paraphrase generating based on the denoising procedure and resulting ParaPhraser Plus corpus. Authors show that automatically aligned and ranked datasets can generate paraphrasing, especially in low-resource languages. Another way to solve the Text Simplification problem for such languages is to turn to multilingual models of various transformer architectures, allowing them to join datasets of different languages to enlarge the training data.

2.3 Text Simplification Evaluation: Metrics and Tools

The primary evaluation metric used for the translation problem is BLEU (bilingual evaluation understudy). However, several works [18], [24] showed that this is not the best choice for text simplification due to its low correlation with grammaticality and meaning preservation and human evaluation of simplification. In 2016, Wei Xu et al. [24] use paraphrasing as the primary tool for Text Simplification and suggest two new metrics that stated solve these problems. New metrics, FK-BLEU and SARI, measure readability and goodness of word choice, respectively.

FK-BLEU represents a combination of the Flesch-Kincaid Index (FK)[7] and BLEU, allowing this metric to measure readability (from FK) and adequacy (from BLEU). Flesch-Kincaid Index is a readability metric that was suggested back in 1975. It is calculated based on the number of words in sentences and the number of syllables in words. Although this metric is easy to compute since it relies on average lengths of sentences, it does not reflect adequacy. It also does not reflect on meaning preservation since it does not compare sentences with any references of possible simplifications.

SARI, in turn, uses multiple references and input sentences to compare with the result. Authors [24] show that BLEU assigns a higher score to the samples with the same complexity level and not penalizes them as SARI does. Along with FK-BLEU, these metrics achieve a much higher correlation with humans' evaluation of simplicity, keeping in mind grammaticality and meaning preservation.

The Python package EASSE [1], Easier Automatic Sentence Simplification Evaluation, is a helpful tool for automatic evaluation of simplification quality. It can evaluate simplification using BLEU and SARI metrics using references. In addition, it can calculate reference-independent quality metrics: FK grade level[7], Levenshtein similarity [8], Lexical Complexity score[11], and compression level. Lexical Complexity score is computing the third-quartile of log-ranks (inverse frequency order) of all words in a sentence. Compression level refers to the character length ratio between the original sentence and its simplified version. Levenshtein similarity [8], Lexical Complexity score (referred to as WordRank), and compression level was used as the tokens which control the simplification process in the ACCESS model [11], one of the state-of-the-art models.

3 Proposed Approach and Models

In this work, we conduct experiments on three different approaches: training the transformer-based NMT model, fine-tuning the MBart model, and applying back-translation to inference pre-trained Text Simplification model. Section 3 describes motivation and details of conducted experiments, along with data preparation, and details and results of these experiments described in Section 4.

3.1 Data Preprocessing

We have chosen automatically translated WikiLarge Dataset [25] provided by organizers as a dataset. It already split into train, test, and validation sets. However, this translated corpus has some problems, such as repetitive target sentences that do not save the sentence's meaning, so that dataset needs additional preprocessing.

Initially, we remove repetitive sentences from the dataset because such samples can distract the model during training. Then, since it is hard for any model to process long sequences and force padding for every sentence to maximum length, we keep only the sentence pairs in which a complex sentence length does not exceed 350 words, and the length of the simple sentence does not exceed 300 words.

As an additional dataset, we use ParaPhraserPlus [2] (a dataset of paraphrased headlines for the Russian Language) without any additional preprocessing for more training samples. The effect of extending the training dataset with ParaPhraserPlus is described in Section 4.

3.2 NMT Transformer as Sentence Simplification model

As we discussed in Section 2, the task of Sentence Simplification can be interpreted as sequence-to-sequence modelling. There is also evidence that sentence simplification is close to the NMT task [21]. Due to this, our next experiment is to train the Sentence Simplification model as the NMT model.

State-of-the-art NMT models use sequence-to-sequence architectures consist of two parts encoder and decoder. The encoder codes information of the input sequence, while the decoder tries to generate a new sequence based on the input sequence. The input sequence in the NMT task is in the source language, and the target sequence is in the target language. According to the NMT task and its application in sentence simplification, the source sequence will be a complex sentence, and the target sequence will be a simplified sentence.

Recently transformers [19] showed promising results in the translation task. So, as an NMT model, we chose transformer architecture described in [19] with three encoder layers and three decoder layers. We observed that sinusoidal positional encoding influences the model convergence (in our case model did not converge), so we decided to replace it with positional embeddings [22]. The number of heads in multi-head self-attention was set to 8.

As an activation function, we use GeLU, and the remaining model parameters were the same as in transformer [19]. For tokenization, a pre-trained ruBERT tokenizer was used. To train the model, standard cross-entropy loss was selected with Adam optimizer and reduce on plateau scheduler with a learning rate equal to $3 \cdot 10^{-4}$.

3.3 Fine-Tuning MBart Model

We fine-tune the MBart model on sentence simplification as our next experiment. MBart is the multilingual model for sequence-to-sequence generation that showed state-of-the-art results on various text generation tasks, including NMT [9]. We fine-tune two different MBart models, one that was trained to translate text between different languages and one that was trained to summarize Russian news [3]. Summarization is a task close to the Text Simplification problem, so we wanted to see if using a model trained for this task will improve results compared to the model trained on classic MBart.

We fine-tune both pre-trained models, for translation and summarization, by the same algorithm. We freeze the encoder and positional embeddings of the MBart, and train in a sequence-to-sequence manner using cross-entropy loss.

3.4 Sentence simplification through back-translation

One of the techniques to get pseudo parallel corpora for context-aware NMT models is data augmentation using back-translation [17]. So, taking this approach, we assume that sentence simplification can be partially solved with the back-translation technique without fine-tuning to a downstream task or training a new model. This approach does not require additional computing power, which necessary to train modern models on large datasets.

The idea of the method is to first translate the source sentence from Russian to English and translate the sentence back to Russian. As our machine translation model, we chose MarianMT model [4] from hugging-face trained on different language pairs, including Russian-English and English-Russian. We leverage a machine translation system to perform a two-step approach: (1) translating forward ($RU \rightarrow EN$) followed by (2) back translation ($EN \rightarrow RU$). The assumption behind the approach is that the machine translation system will probably have a limited vocabulary and, therefore, will produce simplification as a part of translation; performing the back-translation can potentially further simplify the sentence.

The proposed two-step approach complicates the whole process of sentence simplification. More advanced techniques for simplifications at the sentence level exist, such as MUSS [10]. They were already tested for English, French and Spanish languages but required much computational power for training. Besides the bigger carbon footprint, one may find it challenging to participate in deep learning research due to the high cost of such computations. However, machine translation for English is already good enough and can be used 'out of the box'. This consideration justifies using the back-translation method for the sake of rational use of computing resources. In our experiments, we try to answer the question, is it worth complicating the process of sentence simplification using the back-translation in terms of the quality of the result?

4 Experiments and Results

We conduct experiments with training Transformer, fine-tuning the two MBarts (a model pre-trained for NMT and a model pre-trained for Russian Text Summarization). Finally, we test the Back-translation method. We should also mention that using the ParaPhraserPlus dataset, described in Section 2.2, for Transformer training does not give any improvements but increases training time drastically. Due to this fact, we do not use ParaPhraserPlus for MBart fine-tuning.

Represented in SARI score results of all methods are shown in Table 1. The Back-translation method shows the best result among the approaches that we applied. Thus, further we provide a more profound analysis of its result.

Method	SARI
MBart fine-tuned for translation	26.38
MBart pretrained on news summarization	32.32
Transformer	32.50
Back-translation (MarianMT-based model)	**37.08**

Table 1: Results on the sentence simplification task on public test set. Back-translation significantly outperforms other models with no training or fine-tuning on downstream task. Although, we should mention that the final score of the system calculated on the private test set was 36.94.

We use the EASSE package [1], which provides useful metrics to evaluate the result of the Back-translation method. Figure 1 shows plots of Levenshtein Similarity and Compression ratio between system output and reference sentences. Levenshtein similarity quantifies the extent to which the source sentence has been changed (through paraphrasing, adding, and deleting content). Compression ratio is the proportion of the number of characters between the source and target sentences. Both can be interpreted as indirect indicators of sentence simplification quality. We provide the results in Table 2, where identity baseline takes input sentence as system output; truncate baseline takes the first 80% of words of input as system output, and the reference takes randomly one of the references as system output. One can mention that both compression ratio and Levenshtein values are small for reference data. Indeed, the simpler the sentence the shorter it has to be, at the same time preserving the original content. However, one can see that in the first row of the Table 2 compression ratio is greater than 1.0 for the system output. This can be explained by the fact that a simpler sentence should not be always shorter than the original.

Finally, we provide some of the worst and best examples of the simplification results (Tables 3 and

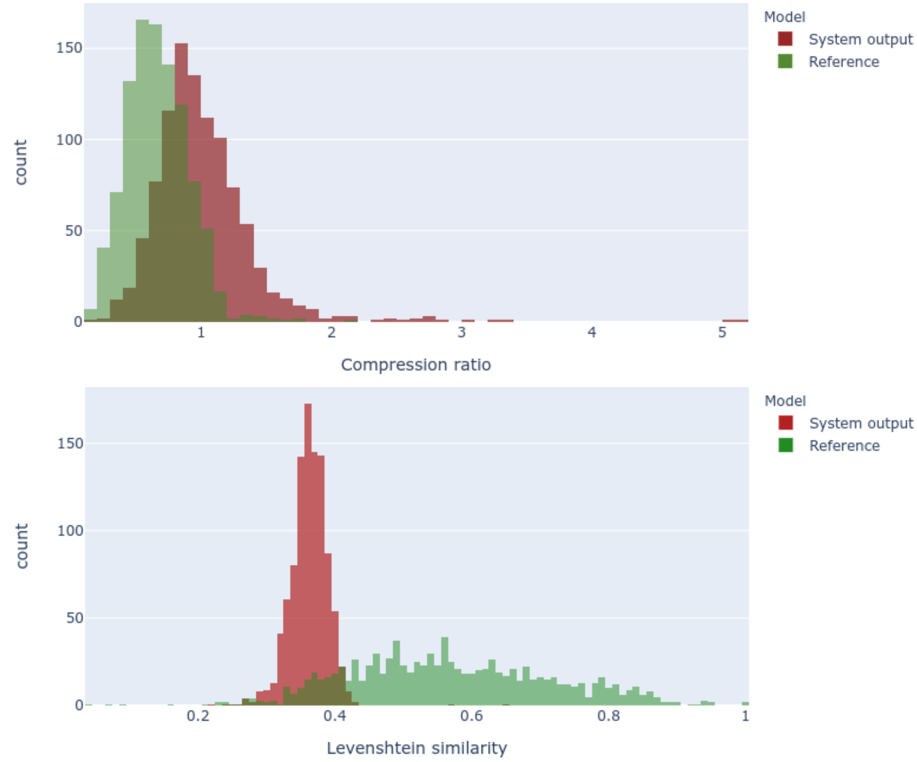

Figure 1: Compression ratio and Levenshtein similarity between system output and reference sentences on the dev part of public dataset (images were generated by the EASSE tool).

	SARI	Compression ratio	Levenshtein similarity
System output	32.47	1.01	0.37
Identity baseline	11.04	1.0	1.0
Truncate baseline	22.84	0.78	0.88
Reference	40.84	0.67	0.58

Table 2: Comparing back-translation method metrics to simple baselines (results were generated by the EASSE tool using the dev set).

4 respectively). One can see that the back-translation method sometimes can copy the source sentence, which we attribute to the performance of the underlying NMT model. In fact, we observed that such 'errors' appear when the sentence is not 'complex enough'; the simplification does not become a part of the translation process. The examples with high SARI scores show that the non-trivial transformation derives the simplified version of the sentence.

5 Conclusion

The proposed in this paper method can be applied to the Russian sentences simplification task. We show that the simple back-translation technique for sentence simplification can provide competitive results without fine-tuning or training. Such a result might be significant in green AI because the required computations for deep learning research have been doubling every few months [15], leading to significant carbon footprints. Besides the problem of air pollution, researchers, students, especially those from developing economies, may find it challenging to participate in deep learning research either due to the high computations cost or due to the absence of the datasets.

There is a limitation for applying our method outside the "cottonwool" conditions of the Dialogue

	Sentence	SARI
Original	Дания является одним из мировых лидеров в использовании возобновляемых источников энергии, в частности энергии ветра.	0.0
Simplified	Дания является одним из мировых лидеров в использовании возобновляемых источников энергии, в частности энергии ветра.	
Original	Разделение равнинных и горных районов между двумя государствами лишило бы многочисленных азербайджанских кочевников летних пастбищ.	9.8
Simplified	Разделение равнинных и горных районов между двумя государствами лишило бы многих азербайджанских кочевников летних пастбищ.	

Table 3: Examples of the worst simplifications according to SARI score.

	Sentence	SARI
Original	В вечерне-ночное время могут возникать ощущения нехватки воздуха, сердцебиение, потливость, озноб или приливы жара.	32.39
Simplified	В вечернее время может возникнуть чувство отсутствия воздуха, сердцебиения, потности, холода или жары.	
Original	1960 году была выпущена модель 172A. Изменения: хвостовое оперение и руль направления с обратной стреловидностью и крепления для поплавкового шасси.	35.74
Simplified	Модель 172A была выпущена в 1960 году.	

Table 4: Examples of the best simplification according to SARI score.

competition. The scalability of the applied technology to the problem of simplification depends on many factors. Most of them, such as dependency on the neural machine translator, the complexity of a sentence, memory requirements etc., are out of the scope of this study. However, some of the factors are easy to assess and overcome (for instance, switch to another sequence modelling toolkit can improve execution time). Although the scalability of the method is questionable, we claim that the approach, in general, can be investigated further by exploring other languages as well as other ‘backbone’ neural machine translators.

Acknowledgements

We thank Innopolis University for generously funding this research and anonymous reviewers for valuable comments.

References

[1] Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 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 (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

[2] Vadim Gudkov, Olga Mitrofanova, and Elizaveta Filippskikh. Automatically ranked Russian paraphrase corpus for text generation. In *Proceedings of the Fourth Workshop on Neural Generation*

and Translation, pages 54–59, Online, July 2020. Association for Computational Linguistics.

[3] Ilya Gusev. Dataset for automatic summarization of russian news. In Andrey Filchenkov, Janne Kauttonen, and Lidia Pivovarova, editors, *Artificial Intelligence and Natural Language*, pages 122–134, Cham, 2020. Springer International Publishing.

[4] 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. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[5] Tomoyuki Kajiwara and Mamoru Komachi. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[6] David Kauchak. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[7] J.P. Kincaid. *Derivation of New Readability Formulas: (automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. Research Branch report. Chief of Naval Technical Training, Naval Air Station Memphis, 1975.

[8] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[9] Yinhan Liu, Jiatao Gu, Naman Goyal, X. Li, Sergey Edunov, Marjan Ghazvininejad, M. Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

[10] Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. Multilingual Unsupervised Sentence Simplification. working paper or preprint, January 2021.

[11] Louis Martin, Benoît Sagot, Éric de la Clergerie, and Antoine Bordes. Controllable sentence simplification. In *LREC*, 2020.

[12] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[13] Ellie Pavlick and Chris Callison-Burch. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany, August 2016. Association for Computational Linguistics.

[14] Andrey Sakhovskiy, Izhevskaya, Alexandra, Alena Pestova, Elena Tutubalina, Valentin Malykh, Ivana Smurov, and Ekaterina Artemova. RuSimpleSentEval-2021 Shared Task: Evaluating Sentence Simplification for Russian. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference "Dialogue"*, volume XX, pages xx–xx, 2021.

[15] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *CoRR*, abs/1907.10597, 2019.

[16] Punardeep Sikka, Manmeet Singh, Allen Pink, and Vijay Mago. A survey on text simplification. *arXiv preprint arXiv:2008.08612*, 2020.

[17] Amane Sugiyama and Naoki Yoshinaga. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China, November 2019. Association for Computational Linguistics.

[18] E. Sulem, O. Abend, and A. Rappoport. Bleu is not suitable for the evaluation of text simplification. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[20] Tong Wang, Ping Chen, Kevin Amaral, and Jipeng Qiang. An experimental study of lstm encoder-decoder model for text simplification. *arXiv preprint arXiv:1609.03663*, 2016.

[21] Tong Wang, Ping Chen, John Rochford, and Jipeng Qiang. Text simplification using neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[22] Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding, 2020.

[23] Wei Xu, Chris Callison-Burch, and Courtney Napoles. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297, 2015.

[24] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016.

[25] Xingxing Zhang and Mirella Lapata. Sentence simplification with deep reinforcement learning. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 584–594, 2017.

[26] S. Zhao, R. Meng, D. He, S. Andi, and P. Bambang. Integrating transformer and paraphrase rules for sentence simplification. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020.

[27] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China, August 2010. Coling 2010 Organizing Committee.