Automatic text simplification of Russian texts using control tokens

Anna Dmitrieva
University of Helsinki
Yliopistonkatu 4, 00100 Helsinki
anna.dmitrieva@helsinki.fi

Abstract

This paper describes the research on the possibilities to control automatic text simplification with special tokens that allow modifying the length, paraphrasing degree, syntactic complexity, and the CEFR (Common European Framework of Reference) grade level of the output texts, i.e. the level of language proficiency a non-native speaker would need to understand them. The project is focused on Russian texts and aims to continue and broaden the existing research on controlled Russian text simplification. It is done by exploring available datasets for monolingual Russian machine translation (paraphrasing and simplification), experimenting with various model architectures, and adding control tokens that have not been used on Russian texts previously.

1 Introduction and related work

Easy and Plain Language are tailored languages (Leskelä et al., 2022) often aimed at a specific audience, such as people with learning disabilities, children, or second language learners. Easy Language is even considered to be a rule-based variety that reverts to purposeful language planning and shows similarities with controlled languages (Maaß, 2020). Despite the growing number of tools for automatic text simplification, most simplified texts are still produced by experts who understand and cater to the needs of a particular group of readers. Because of that, it seems reasonable to concentrate on the task of controllable text simplification so that in the future, simplification tools can be tailored to specific target audiences.

At present, text simplification is often viewed as a monolingual text-to-text generation task borrowing ideas from statistical machine translation (Zhang and Lapata, 2017), and simplification models are trained in a similar fashion to translation models. The training requires large parallel datasets where the target sentences are simplified versions of the source sentences. There are multiple ways to control the output of text simplification tools. For example, editing operations can be directly controlled. Dong et al. (2019) presented a simplification model that could learn explicit editing operations such as additions, deletions, and keeping. Alva-Manchego et al. (2017) proposed a sequence labeling model to predict which simplification operations should be performed as a first step for a complete simplification pipeline. The model is built on a corpus with automatically labeled simplification operations, and the approach is proven to produce more straightforward texts than end-to-end models.

Other research shows that, apart from controlling editing operations, it is also possible to control specific dimensions of the output texts. Martin et al. (2020) identify four attributes related to the text simplification process: the amount of compression, paraphrasing, lexical and syntactic complexity – and use control tokens that are put in front of the source sentences to modify these attributes in output texts. This approach was later used in Martin et al. (2022) and in Anastasyev (2021). The latter was the winning solution for the RuSimpleSentEval (Sakhovskiy et al., 2021) shared task on Russian text simplification. This methodology is used in the present study as well. Other studies have shown that control tokens can be used for all kinds of linguistic attributes, including politeness and monotonicity (the closeness of the word order in the target sentence to the word order in the source sentence) (Schioppa et al., 2021). Some studies also demonstrate the successful usage of control tokens to generate texts for a given school grade level (Scarton and Specia, 2018; Nishihara et al., 2019).

In this project, we use various datasets for monolingual Russian machine translation tasks, namely paraphrasing and simplification, to build models for controllable text simplification. The data is

Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023), pages 70–77

described in Section 2. Section 3 talks about the control tokens used in this study, the process of choosing the optimal model architecture, and the results of the experiments. The final parts of the paper present the conclusions and discuss the limitations of this research.

2 Data

For this project, four different data sources were used:

- ParaPhraser Plus: a large automatically developed corpus for Russian paraphrase generation (Gudkov et al., 2020). Contains news headlines crawled from publicly available websites;
- Opusparcus: a paraphrase corpus for six European languages comprising subtitles from movies and TV shows (Creutz, 2018). Only the Russian part of the corpus was used;
- RuAdapt: a parallel Russian-Simple Russian dataset which consists of texts adapted for learners of Russian as a foreign language (Dmitrieva and Tiedemann, 2021). RuAdapt has three subcorpora: literary texts, encyclopedic entries, and fairytales. Sentence pairs in RuAdapt were aligned automatically and have cosine similarity scores provided by the aligner. Only sentences with cosine similarity above 0.31 but below 0.98 were used;
- The RuSimpleSentEval[1] datasets: development and public test set (Sakhovskiy et al., 2021). The original training set is currently unavailable. The public test set was not included in the general dataset; it was only used separately.

The size of the dataset can be seen in Table 1. 3398 sentence pairs from the RuSimpleSentEval public test set were held out for further testing.

The data only includes sentences with five tokens or longer. Furthermore, to avoid hallucinations in the output (incoherent texts possibly including facts not justified by the training data), the larger parts of the dataset, Paraphraser Plus and Opusparcus, were cleaned from sentence pairs where named entities do not match. The Natasha toolkit[2] was used to exclude sentence pairs where the target sentence has named entities absent in the source.

Dataset	Train	Dev	Test
Paraphraser Plus	338865	37652	7638
Opusparcus	103186	11465	2405
RSSE	2570	285	59
RA literature	8530	948	169
RA encyclopedic	2041	227	50
RA fairytales	135	15	4
Total	455327	50592	10325

Table 1: General dataset partition counts in sentence pairs. RA stands for RuAdapt, RSSE for RuSimpleSentEval. Held out RSSE public test set not included.

3 Experiments

3.1 Control tokens

Following Martin et al. (2022) and Martin et al. (2020), we chose four control tokens to represent four attributes related to the process of simplification mentioned above in Section 1:

- **NbChars**: the ratio between the lengths of source and target sentences in characters; represents the amount of compression. Same as in Martin et al. (2020);
- **LevSim**: the Levenshtein ratio between source and target sentences; represents the amount of paraphrasing. Same as in Martin et al. (2020);
- **DepTreeDepth**: the ratio between the syntactic tree depths of target and source sentences; represents the syntactic complexity. Similar to Martin et al. (2020). The dependency parsing is performed with the deeppavlov's[3] ru_syntagrus_joint_parsing model;
- **CEFRgrade**: the CEFR grade level of the target sentence; represents multiple simplification-related attributes. It is the only token not represented by ratio because it is easier to control the output's grade level directly rather than control how simplified the output will be compared to the source. The grade levels were calculated using code from the Textometr (Laposhina et al., 2018) API. Textometr's grade levels go from elementary A1 up to what can be described as C2+ (too

[1] https://github.com/dialogue-evaluation/RuSimpleSentEval

[2] https://github.com/natasha/natasha

[3] https://github.com/deeppavlov/DeepPavlov

complicated even for a native speaker) and can be transformed to a 0.0 to 10.0 scale. Only sentence pairs where the source's grade level was higher than or equal to the target's (which means that some pairs had to be reversed) and the target's CEFR level was not higher than C2 were kept in the dataset.

Here is what a source sentence with control tokens looks like before encoding and preprocessing with sentencepiece and fairseq (this sentence is from the ParaPhraser.ru corpus):

<CEFRgrade_0> <LevSim_0.4> <NbChars_1.15> Погода на завтра: преимущественно без осадков.

Weather for tomorrow: mostly without precipitation.

Previous research has shown that the NbChars and LevSim tokens work well for both English and Russian; therefore, they were chosen for the initial experiments, including experiments with choosing the model architecture. To the best of our knowledge, the DepTreeDepth token was never tried on Russian but has shown a slight performance increase for English (Martin et al., 2020), so it was included in later experiments. The reason for choosing CEFR grade level as one of the tokens was twofold. The first goal was to find a way to simplify texts for a particular grade level. Secondly, since the WordRank token used in Martin et al. (2020) did not work well for Russian (Anastasyev, 2021), it was necessary to find something else to represent the change in lexical (and other) complexity between sentences. Moreover, studies such as Scarton and Specia (2018) have shown that annotating the source sentences with information about the target grade level can positively affect the model's simplification performance. All tokens except CEFRgrade levels have 40 unique values from 0.05 to 2.

It should be noted that the studies that this paper is based on, namely Martin et al. (2022) and Anastasyev (2021), have different approaches to appending the control tokens to the model. In Martin et al. (2022), the tokens are appended to the beginning of the sentence. Then the sentence is encoded with sentencepiece, preprocessed with fairseq, and fed to the model. Therefore, no special embeddings just for the control tokens are added to the pretrained model, and the vectorization of control tokens happens as is. Anastasyev (2021) uses a different approach, in which he utilizes tokens from the mBART's dictionary that were not used in the training data to denote control tokens. To our understanding, all possible values of the control tokens receive their own embeddings from the pool of tokens known to the model but not utilized in the training data. During inference, if a control token with a certain value is not present in the training data, the closest possible value is found, and the model uses the embedding assigned to that value. Our study follows the Martin et al. (2022)'s approach for this project. It would be interesting to try and append new embeddings to the pretrained models for control tokens. For instance, in Schioppa et al. (2021), the authors introduce attribute control during fine-tuning by affecting a smaller subset of the original model parameters. However, not all frameworks currently have instruments for that.

3.2 Choosing the model architecture

The following versions of two transformer architectures, mBART (Liu et al., 2020) and T5 (Raffel et al., 2020), both proven very capable at monolingual translation tasks such as paraphrasing, were used in this project:

- mBART cc25, a model with 12 encoder and decoder layers trained on 25 languages' monolingual corpus[4]. The preprocessing, training, and inference process was identical to that of the RuSimpleSentEval competition baseline[5].
- a version of Google's multilingual T5 (Xue et al., 2021) with only Russian and some English embeddings left[6]. The training process was similar to the one used by David Dale for fine-tuning a T5 model for multiple tasks, including paraphrasing Russian texts (Dale, 2021). During inference, we used the number of beams of 3 and a no-repeat ngram size of 5.

The models' performance was evaluated with the SARI score (Xu et al., 2016) from the EASSE (Alva-Manchego et al., 2019) library. SARI compares **s**ystem output **a**gainst **r**eferences and against the **i**nput sentence, and correlates with

[4] https://github.com/facebookresearch/fairseq/blob/main/examples/mbart/README.md

[5] https://github.com/dialogue-evaluation/RuSimpleSentEval

[6] https://huggingface.co/cointegrated/rut5-base

Test set	mBART	T5
General	**44.3776**	40.781
RSSE	33.3876	**35.2519**

Table 2: Highest SARI scores for models with no control tokens.

Test set	mBART	T5
General test, true tokens	**53.9269**	38.9376
General test, $\text{NbChars}_{0.95}$, $\text{LevSim}_{0.4}$	**43.1563**	40.0487
RSSE, $\text{NbChars}_{0.95}$, $\text{LevSim}_{0.4}$	**38.9894**	34.6402
RSSE, $\text{NbChars}_{1.0}$, $\text{LevSim}_{1.0}$	15.944	**35.1672**

Table 3: Highest SARI scores for models with NbChars and LevSim control tokens. "True tokens" means tokens that represent the actual attribute values between source and target sentences.

human judgments of simplicity (Xu et al., 2016). It uses an arithmetic average of n-gram precisions and recalls of editing operations: addition, keeping, and deletions between the source, output, and references (ibid.). The models were evaluated on two test sets: a general test set from Table 1 and the public test set from RuSimpleSentEval. Before evaluation, sanity tests were conducted on the RSSE public test set: if the source file is used as the output file, the SARI score is 14.7, and if the target is used as output, the score is 100. During RuSimpleSentEval, the best system had a SARI score of 40.23 on the public test set.

As seen in Table 2, when trained without any control tokens, mBART has a much higher score on the general test set, but on the RSSE public test set, the scores are much lower, with T5 performing slightly better. However, adding two control tokens, NbChars and LevSim, improved the performance of mBART significantly on both test sets (see Table 3). T5, however, did not show a considerable performance gain. Moreover, when both tokens were set to 1.0, only mBART showed a SARI score similar to the SARI that can be obtained if the source sentences are passed as output (which means that the sentences were left unchanged as it is supposed to happen when these tokens are set to 1.0). It should be noted, however, that, despite high SARI scores, the output of mBART contained some incoherent sentences, similar to what Anastasyev (2021) reports (the models with highly rated performance still hallucinating in some cases).

To further investigate how the control tokens affect the model, we measured the actual values of the character length ratio and the Levenshtein similarity ratio between the model's output and the source sentences. Intuitively, suppose a model was asked to simplify sentences with NbChars set to 0.95. In that case, the average character length ratio between the system output and source sentences should be close to 0.95. As seen in Table 4, both models seem to learn the meaning of the tokens with further training, even though it does not necessarily mean SARI score improvement. Evidently, the mBART architecture was better at understanding the meaning of both control tokens, which is why it was chosen for further experiments. It should also be noted that the training process for mBART with fairseq was faster than training T5 with transformers, which influenced our choice of model.

3.3 Syntactic complexity

Training an mBART model with the same configuration as before on texts with just the DepTreeDepth token resulted in a considerable decrease in performance. After 5 initial epochs and additional 7 epochs after early stopping, the best SARI score on the general test set was 28.77 on epoch 7. Despite generally standard loss scores (not much different from previous experiments with and without control tokens), the models hallucinated quite a bit. The hallucinations made calculating the actual syntactic tree depth of the outputs impossible because there were too many word repetitions to create adequate syntactic trees. In conclusion, the tree depth ratio may not be an adequate enough metric to control syntactic complexity in Russian sentences. It should be noted that, as reported in Martin et al. (2020), the identical DepTreeDepth token also did not seem to control its attribute as well as the NbChars and LevSim tokens did in English texts, although it had the desired effect on the output.

3.4 CEFR grade levels

Firstly, we conducted multiple experiments to determine how many unique values should be allocated to this token. The starting range was from 0.7 to 8.5 with a step of 0.1 (the way the values come from Textometr). After a decrease in performance compared to models with no tokens (the highest SARI score obtained on the general test set was 35.84 on

Token	mBART			T5				
	4 epochs	3 epochs	2 epochs	1 epoch	800k	700k	600k	500k
$NbChars_{0.95}$	0,9119	0,9004	**0,9140**	0,8496	0,8976	0,8792	0,8684	0,7327
$LevSim_{0.4}$	**0,4812**	0,4814	0,5074	0,4980	0,5336	0,5648	0,6909	0,6666
$NbChars_{1.0}$	**0,9999**	0,9997	1,0002	0,9993	0,9914	0,9989	0,9315	0,8590
$LevSim_{1.0}$	0,9990	0,9989	**0,9993**	0,9987	0,8762	0,8442	0,7573	0,7085

Table 4: Mean attribute values calculated between the output and the source files (RSSE public test set). k (in 800k, 700k, etc.) = thousands of steps.

Control token CEFR level	SARI
0 (A1)	**46.4875**
1 (A2)	44.8701
2 (B1)	42.2034
3 (B2)	38.0583
Actual target CEFR level (best model)	38.9731

Table 5: SARI scores on the general test set for the model with a CEFR grade level control token: manually set values and actual values (CEFR grade levels of target sentences in the test set).

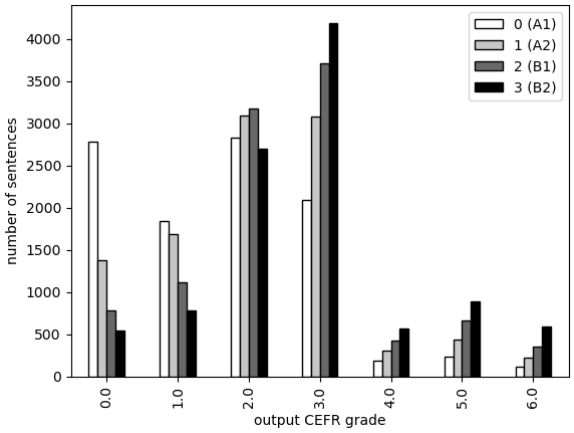

Figure 1: Influence of the CEFR grade level control token on the output. General test set. The numbers in the legend denote the control token values given to the model.

epoch 8/12), the number of unique values was lowered to 8, from 1 to 8. After that, the SARI scores increased up to 41 (epoch 4/7), but the model still hallucinated quite a lot. After that, the number of unique values was reduced to 6, corresponding to levels A1 (0) to C2 (5). This decreased the SARI scores slightly (highest SARI 38.97, epoch 8/10); however, the outputs became more coherent.

In order to test the influence of different token values on the output, during the inference, the token was set to lower grade levels, from A1 (0) to B2 (3). The testing has shown that the SARI score decreases when the CEFR grade level goes up (see Table 5). As expected, the lowest CEFR grade gives the highest SARI score. When studying this token's influence further, it became clear that, even though setting the token to a particular grade level leads to more sentences of that level in the output, the model still produces a lot of B1 and B2 (2 and 3) level sentences, as shown on Figure 1. The reason is likely because many sentences with these grade levels are in the training data.

Despite the model being able to learn the NbChars and LevSim control tokens together and the CEFRgrade separately, combining them in one model did not increase performance. On the contrary, there was no noticeable SARI increase across 18 epochs, and many outputs were incoherent with a lot of word repetitions. The reason for such behavior is unclear since in previous studies (see, for example, Martin et al., 2022, and Schioppa et al., 2021), different control tokens were successfully combined.

4 Conclusions

This paper continues and expands previous research on controlled text simplification. We studied the influence of control tokens on Russian texts using open-source datasets. Also, another transformer architecture was tested not previously used for these kinds of experiments. In the end, the choice fell on mBART, but the experiments have shown that T5 can also learn the meaning of control tokens. Two tokens were tested that have not been applied to Russian data before. The findings show that the DepTreeDepth token does not perform as well on Russian data as it did on English, according to previous research. The CEFRgrade token can influence the model's output in a desirable way, but according to the experiments' results, it cannot be combined with other tokens. Finally, it was confirmed that the other two tokens, NbChars

and LevSim, work well on Russian data. Some examples of the models' outputs can be found in Appendix A. The best models' checkpoints and other supplementary materials can be found on GitHub: https://github.com/annadmitrieva/controlled_simplification_ru.

The findings show that some tokens are "harder" to learn for the models than others. Possible topics for future research include more in-depth studies of "difficult" tokens and finding methods for representing their attributes in more understandable ways to the models. Another possible topic is studying how to combine tokens more effectively and why some combinations do not work well.

Limitations

Data: the bigger portion of the dataset used in this study consists of paraphrases and not professionally done simplifications. There was an attempt to compensate for it by assigning CEFR grade levels to each sentence and reversing the pairs where the source was originally "easier" than the target. This is also partially why the distribution of target CEFR levels is so skewed towards B1 and B2: lower grade levels require more effort made by the author specifically towards simplification. A more balanced dataset would likely improve the models' performance and their ability to simplify sentences for any given grade level.

CEFR grade levels: it should be noted that Textometr, the software used for assigning the grade levels, is used primarily for texts, not single sentences, since CEFR grade levels are generally assigned to a text, and estimating an exact level of a single sentence can be difficult even for an expert. For some sentences, it is also challenging to lower the level below B: for example, when it contains mentions of phenomena that, in order to be understood by someone on level A, would need a detailed explanation, such as "Покров Пресвятой Богородицы" (*Intercession of the Theotokos*) or "Дом профсоюзов" (*Trade Unions Building*). On the other hand, some source sentences in the dataset are already quite simple, and modifying them to become more complex is out of the scope of the simplification task. The observations also show that in many cases, the model could not simplify a sentence to all possible grade levels: for example, sometimes, the model could only simplify a given sentence to levels 0 to 2 but not to 3. The model's behavior and limitations when it comes to controlling the grade level are in itself a separate topic for discussion.

Models: for the sake of time, the models' parameters were not changed during training or inference, and no search for more optimal parameters has been performed. It is likely that finding proper parameters could have improved the results of the experiments. However, the goal was not to increase the performance but to compare how models behave in different settings (with different tokens).

Acknowledgements

This work was partially supported by the Cultura foundation (Application 2021143).

References

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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 (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Daniil Anastasyev. 2021. RuSimpleSentEval. [Online; released 11-April-2021].

Mathias Creutz. 2018. Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

David Dale. 2021. Перефразирование русских текстов: корпуса, модели, метрики. [Online; posted 28-June-2021].

Anna Dmitrieva and Jörg Tiedemann. 2021. Creating an aligned Russian text simplification dataset from language learner data. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 73–79, Kiyv, Ukraine. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Vadim Gudkov, Olga Mitrofanova, and Elizaveta Filippskikh. 2020. Automatically ranked Russian paraphrase corpus for text generation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 54–59, Online. Association for Computational Linguistics.

Antonina Laposhina, Tatyana Veselovskaya, Maria Lebedeva, and Olga Krivenko. 2018. Automated Text Readability Assessment For Russian Second Language Learners. In *Proceedings of the International Conference on Computational Linguistics and Intellectual Technologies "DIALOGUE"*, pages 396–406.

Leealaura Leskelä, Arto Mustajoki, and Aino Piehl. 2022. Easy and plain languages as special cases of linguistic tailoring and standard language varieties. *Nordic Journal of Linguistics*, 45(2):194–213.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation.

Christiane Maaß. 2020. *Easy Language – Plain Language – Easy Language Plus*, 1 edition, volume 3 of *Easy – Plain – Accessible*. Frank & Timme.

Louis Martin, Angela Fan, Éric Villemonte De La Clergerie, Antoine Bordes, and Benoît Sagot. 2022. MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664.

Louis Martin, Éric Villemonte de La Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable Sentence Simplification. In *LREC 2020 - 12th Language Resources and Evaluation Conference*.

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, Florence, Italy. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Andrey Sakhovskiy, Alexandra Izhevskaya, Alena Pestova, Elena Tutubalina, Valentin Malykh, Ivan Smurov, and Ekaterina Artemova. 2021. RuSimpleSentEval-2021 shared task: evaluating sentence simplification for Russian. In *Proceedings of the International Conference on Computational Linguistics and Intellectual Technologies "DIALOGUE"*, pages 607–617.

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 (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Andrea Schioppa, David Vilar, Artem Sokolov, and Katja Filippova. 2021. Controlling machine translation for multiple attributes with additive interventions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6676–6696, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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(0):401–415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

A Example Appendix

Some examples of simplifications performed by the models can be found in Tables 6 and 7.

Partition	Text	Actual grade level
Source	Семья Березовского не дает согласия на закрытие уголовных дел против него	3
	Berezovsky's family does not consent to the closure of criminal cases against him	
Target	Родственники Березовского не будут давать согласие на прекращение уголовных дел в отношении него	3
	Berezovsky's relatives will not consent to the termination of criminal cases against [in relation to] him	
CEFRgrade_0	Семья Березовского не хочет закрывать дела	0
	Berezovsky's family does not want to close cases	
CEFRgrade_1	Семья Березовского не хочет закрывать дела против него	1
	Berezovsky's family does not want to close cases against him	
CEFRgrade_2	Семья Березовского не дает согласия на закрытие уголовных дел	2
	Berezovsky's family does not consent to the closure of criminal cases	
CEFRgrade_3	Семья Березовского не согласна на закрытие уголовных дел против него	3
	Berezovsky's family does not agree to the closure of criminal cases against him	

Table 6: Examples of simplifications with arbitrary CEFR grade levels. Original dataset: ParaPhraser.ru.

Partition	Text
Source	Андропов,военный атташе и водитель уцелели и пешком добрались до посольства.
	Andropov, the military attache and the driver survived and reached the embassy on foot.
Target	Андропов вместе с военным атташе и водителем уцелели, но пешком два часа по ночному городу пробирались в посольство.
	Andropov, along with the military attache and the driver, survived, but they made their way to the embassy on foot for two hours through the night city.
$\text{NbChars}_{1.0}$, $\text{LevSim}_{1.0}$	Андропов,военный атташе и водитель уцелели и пешком добрались до посольства.
	Andropov, the military attache and the driver survived and reached the embassy on foot.
$\text{NbChars}_{0.95}$, $\text{LevSim}_{0.4}$	До посольства добрались Андропов, атташе и водитель.
	Andropov, the attache and the driver reached the embassy.

Table 7: Examples of simplifications with arbitrary NbChars and LevSim parameters. Original dataset: RuSimpleSentEval public test.