Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2022”

Moscow, June 15–18, 2022

Russian Texts Detoxification with Levenshtein Editing

Ilya Gusev
Moscow Institute of Physics and Technology
Moscow, Russia
ilya.gusev@phystech.edu

Abstract

Text detoxification is a style transfer task of creating neutral versions of toxic t exts. In this paper, we use the concept of text editing to build a two-step tagging-based detoxification model using a parallel corpus of Russian texts. With this model, we achieved the best style transfer accuracy among all models in the RUSSE Detox shared task, surpassing larger sequence-to-sequence models.

Keywords: detoxification, style transfer, BERT, T5, tagging, text editing
DOI: 10.28995/2075-7182-2022-21-264-272

Преобразование оскорбительных текстов на русском языке с помощью предсказания редакционных предписаний

Илья Гусев
Московский физико-технический институт
Москва, Россия
ilya.gusev@phystech.edu

Аннотация

Детоксикация текста — это задача создания нейтральных версий оскорбительных текстов. В этой статье мы используем концепцию преобразования текста с помощью предсказания редакционных предписаний для построения двухэтапной модели детоксикации русских текстов при наличии параллельного корпуса. С помощью этой модели мы добились наилучшей точности передачи стиля среди всех участаников дорожки RUSSE Detox, превзойдя более крупные sequence-to-sequence модели.

Ключевые слова: оскорбительные тексты, детоксификация, перенос стиля, BERT, T5, тегирование, редакционные предписания

1 Introduction

There is a vast amount of user-generated content on the Internet containing hate speech, profanity, toxicity, and aggression. It may not be appropriate for some platforms to show toxic texts. Some countries can even consider illegal writing or showing such content.

There are several ways to combat this problem. The obvious solution is to censor all toxic messages. Such texts can be deleted completely, covered with a warning, or placed at the very bottom of the page. However, it is ethically questionable apparent censorship.

Another way is to prevent writing such messages by suggesting alternative neutral options to a user. We will refer to the task of making such neutral variants of toxic texts as detoxification. It is a style transfer task where the source style is toxic, and the target style is neutral. The goal of this work was to build a system to solve this task.

Why is this task difficult?

1. Indistinct boundaries of what to consider toxic
2. Obfuscations that hide the meaning of words

3. Occasions of some rare insults
4. Sarcasm and other issues that require external world knowledge

Toxicity is a broad term that includes hate speech, obscene or condescending language, aggression, or grave insults. An instruction for annotators should define the particular rules for it. From the perspective of the shared task organizers, a text should contain «insults or obscene and rude words» to be considered toxic.

From a scientific perspective, it is a curious sequence-to-sequence task, where a target text is almost the same as a source one but with a different style. It allows specific methods that rely on the similarity of source and target texts.

This work is a part of the RUSSE Detox shared task (Dementieva et al., 2022), organized by a group of researchers as a part of the Dialogue-2022 conference. The goal of the shared task was to build a detoxification model with provided parallel corpus. Organizers also provided several baselines.

Our contributions:

1. We adopt a concept from the LEWIS paper (Reid and Zhong, 2021) to build a two-step tagging-based detoxification model using a parallel corpus of Russian texts.
2. We compare this tagging-based model with sequence-to-sequence baselines trained on the same corpus.
3. We propose a better model for toxicity classification.
4. We achieve the best style transfer accuracy among all models in the shared task.

Our code[1] and models[2][3][4] are available online.

2 Related work

2.1 Toxicity classification

Nobata et al. (2016) made one of the first attempts to formulate the task of toxicity classification and collect a unified test dataset for it. They used comments posted on Yahoo Finance and News and rated by their in-house workers. The model was Vowpal Wabbit's regression over different manual NLP features.

Gordeev (2016) selected anonymous imageboards (4chan.org, 2ch.hk) as the material for their corpus for the task of analysis of aggression. Authors utilized convolutional neural networks to detect the state of aggression in English and Russian texts.

Andrusyak et al. (2018) collected a dataset from Russian YouTube comments in an unsupervised way using a seed dictionary of abusive words and an iterative process updating this dictionary.

Smetanin (2020) used the Russian Language Toxic Comments Dataset (RTC dataset) from Kaggle[5]. It is the collection of annotated comments from 2ch[6] and Pikabu[7] websites. Fine-tuned RuBERT (Kuratov and Arkhipov, 2019) was the best model from this paper.

Zueva et al. (2020) introduced a novel corpus of 100000 comments posted on a major Russian social network (VK). As their primary model, they used a self-attentive encoder to get interpretable weights for each input token. They also used several tweaks, such as identity dropout and multi-task learning.

Saitov and Derczynski (2021) utilized Russian subtitles from the «South Park» TV show (RSP dataset) and the RTC dataset. Crowdsourcers annotated these subtitles for toxicity. Again, RuBERT held the best result.

Pronoza et al. (2021) focused on ethnicity-targeted hate speech detection in Russian texts. The authors composed a dataset of 5600 texts with 12000 mentioned instances of different ethnic groups. They named it RuEthnoHate. One more time, the modified RuBERT with additional linguistic features was the best model.

[1] https://github.com/IlyaGusev/rudetox
[2] https://huggingface.co/IlyaGusev/rubertconv_toxic_clf
[3] https://huggingface.co/IlyaGusev/rubertconv_toxic_editor
[4] https://huggingface.co/IlyaGusev/sber_rut5_filler
[5] https://www.kaggle.com/blackmoon/russian-language-toxic-comments
[6] https://2ch.hk/
[7] https://pikabu.ru/

We used several of the mentioned datasets for toxicity classification to fine-tune a conversational RuBERT model.

2.2 Style transfer

Li et al. (2018) started the whole field of research, proposing a set of simple baselines for unsupervised text style transfer. The baselines were based on detecting style tokens with n-gram statistics and replacing them with altered retrieved similar sentences with the target style.

Wu et al. (2019) introduced a way to augment texts without breaking the label compatibility. They trained a **conditional BERT** (Devlin et al., 2019) using a conditional MLM task on a labeled dataset. Aside from data augmentation, they proposed to use this method as a part of a style transfer system, using an attention-based method to find style words and conditional BERT to replace them.

Krishna et al. (2020) suggested **STRAP**, **S**tyle **Tra**nsfer via **P**araphrasing. First, they generated a pseudo-parallel corpus. They started with styled texts and applied paraphrasers to normalize these texts in terms of style. The diversity of paraphrasing was promoted by filtering outputs heavily. Then, they fine-tuned style-changing inverse paraphrasers on this pseudo-parallel corpus. GPT2 (Radford et al., 2018) language model was used to implement both the paraphrasers and inverse paraphrasers. This scheme can also be used to augment an existing parallel corpus.

They also criticized existing style transfer evaluation methods and proposed an evaluation scheme based on transfer accuracy, semantic similarity, and fluency that we use in this work.

Malmi et al. (2020) introduced **Masker**, a system that used two language models to detect style tokens and padded masked language models to replace them. They tested it on sentence fusion and sentiment transfer. As for supervised tasks, they created **LaserTagger** (Malmi et al., 2019), a sequence tagging approach that casts text generation as a text editing task.

Krause et al. (2021) used **GeDi** (**Ge**nerative **Di**scriminator) to control generation towards the desired style. They use three language models: a base one, one for the desired style, and one for the undesired anti-style. The Bayes rule is applied during generation to compute style modifiers for every token from a vocabulary. Then these modifiers are applied to predictions of the base language model. This method allows computationally effective style-guided generation, but there is no source sequence, unlike the style transfer task. Dale et al. (2021) introduced the **ParaGeDi** method that applies GeDi for style transfer using a paraphrasing model instead of the base language model.

Dementieva et al. (2021) introduced the first study of automatic detoxification of Russian texts. They proposed two methods, the unsupervised one based on condBERT and the supervised one based on fine-tuning pretrained language GPT-2 model on a small manually created parallel corpus.

Reid and Zhong (2021) proposed **LEWIS** (**L**evenshtein **E**diting **WI**th unsupervised **S**ynthesis), the editing and synthesis framework for text style transfer. They had no parallel data, so the first task was to create a pseudo-parallel corpus. They used an attention-based detector of style words and two style-specific BART (Lewis et al., 2020) masked language models to replace these style words. Then they filtered resulting pairs with a style classifier, keeping only examples where the language models and the classifier agree.

After obtaining the pseudo-parallel corpus, they trained a RoBERTa-tagger (Liu et al., 2019) on it, predicting coarse edit types: «insert», «keep», «replace» and «delete» (Levenshtein, 1966). Then they trained a fine-grain edit generator to produce the target text, filling in phrases for coarse-grain edit types «insert» and «replace». We use this scheme almost without any modifications, but with a different language, with different base models, and already existing parallel corpus.

3 Evaluation

We built our style classifier by fine-tuning conversational RuBERT (Kuratov and Arkhipov, 2019) instead of the model[8] proposed by organizers of the shared task. In addition to ok.ru[9] and 2ch/Pikabu[10] datasets,

[8] https://huggingface.co/SkolkovoInstitute/russian_toxicity_classifier

[9] https://www.kaggle.com/datasets/alexandersemiletov/toxic-russian-comments

[10] https://www.kaggle.com/datasets/blackmoon/russian-language-toxic-comments

Test type	Test description	Skolkovo clf., ER, %	Our clf., ER, %
INV	Replace yo	0.6	0.0
INV	Remove exclamations	0.9	0.4
INV	Add exclamations	0.9	0.3
INV	Captioned sentences to lowercase	73.9	34.8
INV	Remove question marks	4.0	0.2
INV	Add typos	3.6	1.9
INV	Masking of characters in toxic words	5.2	0.5
INV	Add typos to toxic words only	24.2	2.8
MFT	Concatenate non-toxic and toxic texts	15.5	3.1
MFT	Concatenate two non-toxic texts	2.1	0.6
MFT	Add toxic words from a vocabulary	16.3	0.1

Table 1: Error rates on different tests for two toxic classification models

Model	AUC, %	Accuracy, %	F1, %
Skolkovo classifier	66.2	86.4	37.2
Our classifier	73.5	90.3	51.3

Table 2: Metrics of toxicity classifiers on unseen crowdsourced test set: 3642 unique texts, 355 of them are toxic

we used Russian Persona Chat dataset[11] as a reliable source of non-toxic sentences.

We also tested models using a «checklist» (Ribeiro et al., 2020) methodology and augmented the resulting dataset with all the transformations. Test results are in Table 1. There are invariance (INV) and minimum functionality tests (MFT). Invariance tests ensure that a label will not change after a transformation, and MF tests have a fixed label to be predicted. It is clear from the table that our model has much lower error rates. From the user's perspective, it is harder to pick up an adversarial example for our model than for the default one.

Two models have different dataset splits, so comparing them on their native test sets is wrong. However, we used crowdsourcing to evaluate the style transfer model, so we can use these annotations as an independent test set, keeping in mind that these samples are adversarial. Results for this new set are in Table 2. Our classifier shows better results in this setting.

We used models provided by organizers of the shared task for measuring semantic similarity[12] and fluency[13]. They have similar problems, but we did not come up with better options.

However, automatic metrics are not reliable, especially when being used with near-adversarial examples. Table 2 gives a glance at how unreliable they can be. To overcome this, we arranged our in-house annotation process with crowdsourcing through the Toloka[14] platform in addition to the final evaluation provided by organizers of the shared task. We measured only style accuracy and semantic similarity, as fluency was much harder to define. Annotation instructions were close as possible to ones provided by the organizers and are available in the repository. Five workers annotated every sample. Samples were aggregated by majority vote. The average agreement was 90% for the style accuracy project, with Krippendorff's alpha of 46%. For the similarity project, the average agreement was 88%, with Krippendorff's alpha of 49%.

[11] https://toloka.ai/ru/datasets

[12] https://huggingface.co/cointegrated/LaBSE-en-ru

[13] https://huggingface.co/SkolkovoInstitute/rubert-base-corruption-detector

[14] https://toloka.ai

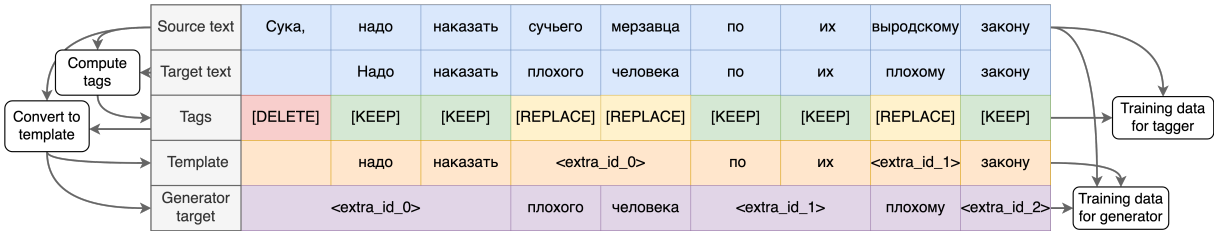

Figure 1: Data generation for training a tagger and a generator.

4 Model

We see text detoxification as a two-step process. In the first step, a model should determine what words should be deleted or replaced. We can explicitly do it through tagging. In the second step, a generator replaces words or adds new ones. From this perspective, any classical sequence-to-sequence model has a trivial first step, as all words can be replaced.

4.1 Tagger — first step

4.1.1 Based on interpretation of a classifier

One way to find style tokens is to interpret a classification model. As for attention-based models, one can find such tokens using attention distribution. Tokens with high attention scores correlate with tokens that manifest style. Many researchers used this method (Xu et al., 2018; Wu et al., 2019; Hoover et al., 2020; Reid and Zhong, 2021).

It is also possible to use models that allow interpretation by design. Dementieva et al. (2021) utilized logistic regression and its weights for each word from the vocabulary for this task, and Li et al. (2018) used a Naive Bayes classifier.

4.1.2 Based on language models

Another way is to use two language models, one trained on texts of one style and another trained on texts of a different style. We can calculate the proportion of their predictions for every token if we have such models. If a prediction of the first model is much higher than that of the second model, then a corresponding token can be style-loaded. For instance, Masker (Malmi et al., 2020) used a similar approach.

4.1.3 Based on tags from parallel corpus

Finally, if we have a parallel corpus, we can directly compute edits required to transform source texts into target texts, convert these edits to tags, and then predict these tags with a token classification model.

4.2 Generator — second step

4.2.1 Based on MLM models

One way of filling the gaps is to use models pretrained for masked language modeling tasks (MLM) such as BERT (Devlin et al., 2019) or T5 (Raffel et al., 2020). It is their original task, but one can fine-tune them on texts of the required style (Wu et al., 2019). However, these models have no access to original words, so they can rely solely on context remained after masking.

4.2.2 Based on pairs from parallel corpus

The most direct way is to fine-tune a sequence-to-sequence model on a parallel corpus. Inputs are templates from the tagger, and outputs are masked words from the target sentence. The whole process of data generation for training is in Figure 1. We also concatenate a source sentence with the generated template, as in Reid and Zhong (2021), to provide access to the original masked words.

Architecture	Generator	STA, %	SIM, %
Seq2seq	T5 baseline	83.5	**87.0**
Seq2seq	cointegrated/ruT5-base	73.0	83.5
Seq2seq	sber/ruT5-base	81.0	85.0
LEWIP	cointegrated/ruT5-base	93.5	76.0
LEWIP	sber/ruT5-base	**94.5**	81.5

Table 3: In-house manual annotation with crowdsourcing, 200 random samples from the test set

4.3 Final model

Our final model uses tagger and generator, both based on a parallel corpus, so we will call it LEWIP (**L**evenshtein **e**diting **wi**th **p**arallel corpus), following the LEWIS (Reid and Zhong, 2021) scheme, as there is no «unsupervised synthesis» step.

We use conversational RuBERT as a base model for tagger and two versions of the ruT5-base model for generator[15][16], with the final submission based on the Sber model. We did not use ruT5-large in a shared task submission, as it did not fit into our GPU memory.

Organizers of the shared task provided a parallel corpus of 11090 pairs for training, a development set of 800 samples, and a test set of 875 samples. We used only that data for the style transfer model.

5 Results

5.1 In-house annotation and automatic metrics

Some results of in-house annotation are in Table 3. In addition to the T5 baseline provided by the organizers, we fine-tuned two ruT5-base models on our own. LEWIP with the Sber generator shows the best style transfer accuracy among all tested models but a lower similarity score than the baseline. Sber ruT5-base is better than the original ruT5-base for this task in both setups.

Examples where LEWIP failed to generate non-toxic texts according to in-house annotation:

1. **Original**: сколько же е**нутых в россии в месте с тобой
   Translation: how many f**ked up people in russia are there with you
   Generated: сколько же неадекватных в россии в месте с тобой
   Translation: how many inadequate people in russia are there with you
2. **Original**: какие же эти люди сволочи!!!
   Translation: what kind of bastards are these people!!!
   Generated: какие же эти люди плохие !
   Translation: how bad are these people!

Examples where LEWIP failed to generate similar texts according to in-house annotation:

1. **Original**: у курки е**ные не оболщатесь это миф.
   Translation: f**king s toners, don't be fooled, it's a myth.
   Generated: у вас не оболденьтесь это миф
   Translation: you don't go crazy, it's a myth.
2. **Original**: Только хотел спросить, что за завалы. Е**ть хреновые в Рашке плотники
   Translation: Just wanted to ask what are these obstructions. The carpenters in Russia are f**king bad
   Generated: Только хотел спросить, что за завалы. в Рашке плотники
   Translation: Just wanted to ask what are these obstructions. The carpenters in Russia

It seems that the tagger works well in most cases, and problems are mostly in the generator.

[15] https://huggingface.co/cointegrated/rut5-base
[16] https://huggingface.co/sberbank-ai/ruT5-base

Architecture	Model	Our STA, %	SIM, %	FL, %	J, %
Seq2seq	T5 baseline	86.3	82.7	83.7	59.3
Seq2seq	cointegrated/ruT5-base	78.8	**85.0**	83.9	55.2
Seq2seq	sber/ruT5-base	83.8	83.6	83.4	57.8
LEWIP	cointegrated/ruT5-base	**93.6**	79.7	88.4	**66.1**
LEWIP	sber/ruT5-base	93.1	79.8	**88.5**	65.8

Table 4: Automatic metrics on the test set

Team	STA, %	SIM, %	FL, %	J, %
Human References	88.8	82.4	89.4	65.3
T5 baseline	79.1	82.2	**92.5**	60.6
SomethingAwful	79.4	**87.2**	90.3	**63.3**
FRC CSC RAS	73.4	86.5	91.8	59.8
Our system	**82.4**	79.1	84.6	58.2

Table 5: Final results of the shared task, human evaluation, 3 top teams out of 10

Automatic metrics for the same set of models are in Table 4. Joint and STA scores for both LEWIP models are higher than the baseline.

5.2 Final human evaluation

The final results are in Table 5. Our model's style transfer accuracy is much worse than our in-house annotation. We explain it with different instructions and annotation protocols. Still, our model has the best style transfer accuracy among all other models but with lower semantic similarity than the baseline. Pang and Gimpel (2019) showed that these metrics are complementary and challenging to optimize simultaneously.

We attempted to rank several beam search hypotheses from generators with automatic metrics to find different trade-offs, and we were successful in the sense of these automatic metrics. Nevertheless, it did not yield better human assessments. Generators were coming up with adversarial examples that were wrong for humans but good for automatic metrics.

5.3 Computational effectiveness

In some cases, we do not need the second step of the system. For 238 examples of 875 (27%) in the test set, the generator model was not run because there were no «replace» or «insert» tags. Sequence-to-sequence models are much more computationally expensive than encoder-only taggers. Moreover, a generator requires fewer steps than a raw sequence-to-sequence model, as it only fills the gaps. Overall, our system is more computationally effective than a T5 baseline.

6 Conclusions

- Thorough testing of a classification model helps in building data augmentations and, eventually, a much more stable model.
- Current automatic metrics are not reliable for evaluating systems trained on parallel corpora. They can work in a range of low values, e.g., for unsupervised style transfer, but there are too unstable to work with accurate models.
- Text editing models can perform at least as well as pure sequence-to-sequence models. They have the inductive bias based on the assumption that input and output are very close. They are also more environmental-friendly than pure sequence-to-sequence models.

References

Bohdan Andrusyak, Mykhailo Rimel, and Roman Kern. 2018. Detection of abusive speech for mixed sociolects of russian and ukrainian languages. // Ales Horák, Pavel Rychlý, and Adam Rambousek, *The 12th Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2018, Karlova Studanka, Czech Republic, December 7-9, 2018*, P 77–84. Tribun EU.

David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text detoxification using large pre-trained neural models. // *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, P 7979–7996, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Daryna Dementieva, Daniil Moskovskiy, Varvara Logacheva, David Dale, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Methods for detoxification of texts for the russian language. *Multimodal Technologies and Interaction*, 5(9).

Daryna Dementieva, Irina Nikishina, Varvara Logacheva, Alena Fenogenova, David Dale, Irina Krotova, Nikita Semenov, Tatiana Shavrina, and Alexander Panchenko. 2022. Russe-2022: Findings of the first russian detoxification task based on parallel corpora.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. // *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, P 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Denis Gordeev. 2016. Detecting state of aggression in sentences using cnn. // *International conference on speech and computer*, P 240–245. Springer.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. // *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, P 187–196, Online, July. Association for Computational Linguistics.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. // *Findings of the Association for Computational Linguistics: EMNLP 2021*, P 4929–4952, Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. // *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, P 737–762, Online, November. Association for Computational Linguistics.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, feb. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. // *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, P 7871–7880, Online, July. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. // *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, P 1865–1874, New Orleans, Louisiana, June. Association for Computational Linguistics.

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*, abs/1907.11692.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. // *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, P 5054–5065, Hong Kong, China, November. Association for Computational Linguistics.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. // *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, P 8671–8680, Online, November. Association for Computational Linguistics.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. // *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, P 145–153, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Richard Yuanzhe Pang and Kevin Gimpel. 2019. Unsupervised evaluation metrics and learning criteria for non-parallel textual transfer. // *Proceedings of the 3rd Workshop on Neural Generation and Translation*, P 138–147, Hong Kong, November. Association for Computational Linguistics.

Ekaterina Pronoza, Polina Panicheva, Olessia Koltsova, and Paolo Rosso. 2021. Detecting ethnicity-targeted hate speech in russian social media texts. *Information Processing and Management*, 58(6):102674.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Colin Raffel, Noam M. 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. *ArXiv*, abs/1910.10683.

Machel Reid and Victor Zhong. 2021. Lewis: Levenshtein editing for unsupervised text style transfer. // *FINDINGS*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. // *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, P 4902–4912, Online, July. Association for Computational Linguistics.

Kamil Saitov and Leon Derczynski. 2021. Abusive language recognition in Russian. // *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, P 20–25, Kiyv, Ukraine, April. Association for Computational Linguistics.

Sergey Smetanin. 2020. Toxic comments detection in russian. // *Computational Linguistics and Intellectual Technologies, Papers from the Annual International Conference "Dialogue"*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. // *ICCS*.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. // *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, P 979–988, Melbourne, Australia, July. Association for Computational Linguistics.

Nadezhda Zueva, Madina Kabirova, and Pavel Kalaidin. 2020. Reducing unintended identity bias in Russian hate speech detection. // *Proceedings of the Fourth Workshop on Online Abuse and Harms*, P 65–69, Online, November. Association for Computational Linguistics.