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

Moscow, June 15–18, 2022

Detecting Auto-generated Texts with Language Model and Attacking the Detector

Mikhail Orzhenovskii
Saint Petersburg, Russia
orzhan057@gmail.com

Abstract

We propose a simple approach to the detection of automatically generated texts. A pre-trained language model, fine-tuned on the shared task's dataset, achieved 3rd place on the binary task leaderboard with 82.6% accuracy. In the multi-task leaderboard, the language model achieved an F1 score of 64.5% after being fine-tuned with the same procedure. In order to investigate the weaknesses of this approach, we explore two possible attacks on the detector: selecting from language model outputs and directed beam search. These attacks reduce the likelihood of detecting the generated texts without significant loss in quality. Both attacks do not require retraining the generative model and are applied at inference time.

Keywords: generated text detection, text classification, pre-trained language models
DOI: 10.28995/2075-7182-2022-21-412-419

Обнаружение сгенерированного текста с помощью языковой модели и атаки на детектор

Орженовский М.В.
Санкт-Петербург, Россия
orzhan057@gmail.com

Аннотация

Мы предлагаем простой подход к обнаружению автоматически сгенерированных текстов. После дообучения на данных соревнования, предобученная языковая модель заняла 3 место в бинарной классификации с точностью 82.6%. В задаче множественной классификации, аналогичная модель достигла 64.5% по метрике F1 (также 3 место). Изучая слабые стороны такого подхода, мы рассматриваем два типа атак: выборка из сгенерированных языковой моделью текстов и направленный лучевой поиск. Эти атаки снижают вероятность обнаружения сгенерированных текстов без существенной потери их качества. Обе атаки не требуют переобучения генеративной модели, внедряются на этапе исполнения.

Ключевые слова: обнаружение сгенерированного текста, классификация текстов, предобученные языковые модели

1 Introduction

Large language models are capable of generating high-quality coherent texts. However, they can be used to generate fake news or product reviews. The shared task (Shamardina et al., 2022) is focused on detecting Russian texts that were created with large language models, distinguishing them from human-written ones. One of the tracks is binary classification: identifying whether a text was written by a human or a language model. Another track (multi classification) is to find which model was used to generate the text.

Humans cannot easily solve this problem. The longer the text is, the easier it is to identify the source. Most language models work with limited context, which causes them to lose coherence if a long text is produced. The dataset in the shared task includes short texts (one sentence long) which makes the task challenging.

Our approach is straightforward: fine-tuning a large pre-trained language model with a sentence classification head. We release the source code for training the models and the models' weights[1].

Additionally, we analyze the weaknesses of the proposed model. We explore two methods of generating texts that are less likely to be detected: selecting one of the language model's outputs with a discriminator model, and adversarial beam search driven by a discriminator model. While the unmodified language model's output is detected in 65% cases, these methods produce texts that are detected in 55% and 42% cases respectively. We are not publishing the code for the attacks.

2 Related work

(Gehrmann et al., 2019) use BERT and GPT models to identify top-k rank of each word in the text; generated texts consist of words with lower ranks, and human-generated texts include a high fraction of high-ranked words.

(Uchendu et al., 2020) apply neural models built on RNN and CNN as well as fine-tuned RoBERTa in the settings that are related to both binary and multi tracks of the shared task.

(Ippolito et al., 2020) show that humans and machines use different approaches to identify generated text. The authors show that common decoding strategies introduce statistical features that can be used by automatic systems.

(Pillutla et al., 2021) introduce a comparison measure to compare the distributions of human-written and machine-generated texts, using divergence frontiers.

(Scialom et al., 2020) use guided decoding with a discriminator to generate human-like texts in abstractive summarization task.

3 Datasets

3.1 Binary track

There are 129,065 samples in the training set, 21,511 samples in the validation set and 64,533 samples in the test set. In the training set, the length of the samples varies from 1 to 376 words, mean length 31 words, 75% of samples have no more than 22 words. 64,535 samples in the training set belong to class H (human-written) and 64,531 samples belong to class M (machine-written). The statistics for both classes are not very different. The dataset is balanced. Examples of the texts are displayed in Table 1.

Таблица 1: Dataset examples

Sentence	Binary class	Multi class
Никто ни разу не навестил меня в больнице	M	OPUS-MT
Под монастырем, на самой верхушке скалы, обнаружил почти 200 древних археологических находок.	M	ruGPT3-Large
На лицо и руки садился тяжелый и липкий туман.	H	Human
Они чем-то кормились на земле и только в случае тревоги взлетали на деревья.	H	Human

3.2 Multi track

As we can see in Table 2, Human class is the largest. The classes of language models are not balanced, for example OPUS-MT has 7 times more samples than ruT5-Large. Average word count also differs between the models: average sample from ruGPT3-small has 71 word, and average M-BART50 sample has only 10 words. Examples of the texts are displayed in Table 1.

[1] https://github.com/orzhan/ruatd

Table 2: Value counts in the training part of Multi track dataset

Class	Count	Percentage	Average word count
Human	51150	39.63%	35.43
OPUS-MT	12087	9.36%	9.95
M-BART50	11913	9.23%	9.67
M2M-100	10817	8.38%	9.87
ruGPT3-Large	9870	7.64%	53.70
ruGPT3-Medium	7020	5.44%	68.42
ruGPT3-Small	6930	5.37%	71.06
mT5-Large	4860	3.77%	12.49
mT5-Small	2940	2.28%	10.52
ruT5-Base	2640	2.05%	32.05
M-BART	2510	1.94%	29.40
ruGPT2-Large	2370	1.84%	14.19
ruT5-Base-Multitask	2219	1.72%	11.59
ruT5-Large	1740	1.35%	16.21

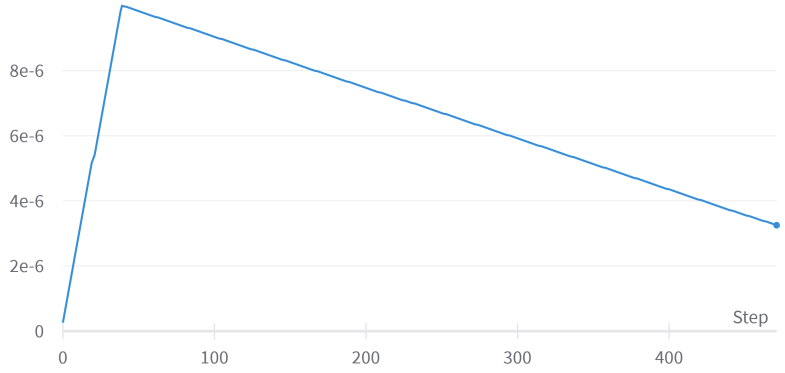

Figure 1: Learning rate of the binary model.

4 Classification model description

We used pre-trained language model sberbank-ai/ruRoberta-large[2] based on (Liu et al., 2019). This model scores high on the Russian SuperGLUE leaderboard (Shavrina et al., 2020), indicating its strong capabilities in various tasks. We directly fine-tune the model with a classification head on the training part of the dataset without any pre-processing. For the experiments, we set the learning rate to a relatively small value 1×10^{-5} and use weight decay 0.01 and label smoothing factor 0.1. For binary classification, we picked the model from epoch 2, which had the best evaluation accuracy; for multi classification, we also chose the model from epoch 2, with the best evaluation F1 score. For the experiments, we used an implementation by HuggingFace Transformers (Wolf et al., 2019).

The learning rate chart is shown on Figure 1. The accuracy chart is shown on Figure 2.

5 Attacks on the classification model

We explore an imaginary situation when the classification model is used to filter out malicious texts, and the attacker knows about it; however, the attacker does not have access to the model and the data that was used to train the model. The attacker collects another dataset of human-written and machine-generated

[2]https://huggingface.co/sberbank-ai/ruRoberta-large

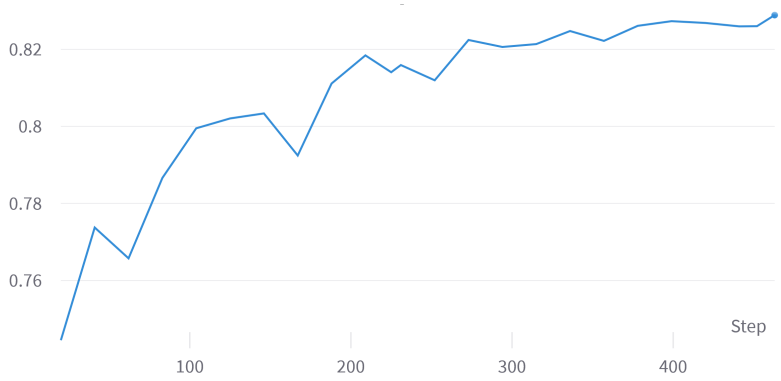

Figure 2: Accuracy of the binary model.

texts. The attacker already has some generative model and is going to use it to produce the texts. To reduce the percentage of rejected texts, the attacker trains some discriminator model and uses it to direct their generative model.

For the purpose of evaluating the attack methods, we used RuSimpleSentEval task (Sakhovskiy et al., 2021) from Dialogue Evaluation 2021. This text-to-text task requires to write a simplified version of a Russian sentence. The target metric is SARI (Xu et al., 2016) (generated text is compared to several human-written reference sentences). We calculate attack success rate as percentage of the generated sentences that were falsely identified as human-written by the main classification model. First, we generate simplifications with fine-tuned ruGPT3 by (Orzhenovskii, 2021) and calculate SARI metric on the public test part of RuSimpleSentEval dataset.

For both attacks we use the same discriminator model, which is based on rubert-tiny2[3]. To classify unfinished sentences, we augment the dataset with partial sentences, making 160,000 samples out of 20,000 from the validation set. This small discriminator model is further used to score the generative language model's outputs.

In the first attack (scoring), we produce several outputs from the generative model using nucleus sampling (Holtzman et al., 2019) and calculate the class probabilities with the discriminator model. The output with the highest human-class probability is selected.

In the second attack (beam search), we use beam search with one modification. Following (Scialom et al., 2020), on each step of the beam search algorithm, we add discriminator model's log-probability $S_{dis}(\hat{y})$ to the generator's log-probability $S_{gen}(\hat{y})$ of the partial sequence $\hat{y}$, so that beams that look machine-generated are less likely to be selected.

$$S_{DAS}(\hat{y}) = S_{dis}(\hat{y}) + \alpha \times S_{gen}(\hat{y})$$

where $\alpha > 0$ is a weighting factor.

The process is repeated until end-of-sentence token is generated. This method slow, because we have to run inference of the discriminator model N times per token. In our experiments, we used $\alpha = 1$ and $N = 10$ beams.

6 Results and analysis

6.1 Binary classification model

The binary model scored 0.82629 on the private leaderboard (3rd) which is close to the validation score of 0.83054. These numbers are far behind the results from (Uchendu et al., 2020) where RoBERTa-

[3]https://huggingface.co/cointegrated/rubert-tiny2

tuned scored 0.9702 in the same binary setting; the difference can be attributed to significantly longer texts (average word count was 432 against 31 in RuATD shared task).

Analysis of the validation results indicates positive correlation 0.177 between word count and results. Longer texts are easier to classify, as expected. See Table 3.

Table 3: Binary model: accuracy and text length

Number of words	Number of samples	Accuracy
1 to 9	6208	0.759
10 to 13	4238	0.785
14 to 22	5738	0.811
23 and more	5324	0.952

6.2 Multi classification model

The multi model achieved 0.64573 on the private leaderboard (3rd). This value is close to the validation score of 0.64427.

F1 score is highest for Human (the largest class), ruT5-Base, and ruGPT3-Small (Table 4). In the dataset, the samples from these classes were longer (30+ words), which could help in their identification. Another possibility is that the base model ruROBERTa-large has been trained a dataset that overlaps with the training data of ruT5 and ruGPT3 models, which improves the score.

The confusion matrix (Figure 3) indicates that certain models are more difficult to distinguish. Interestingly, different sizes of the same model (like ruGPT3-small and ruGPT3-medium, or ruT5-base and ruT5-large) that have been trained on the same data, can be separated by the classifier.

Models OPUS-MT, ruGPT3-small, ruGPT3-medium, ruGPT3-large, ruT5-base, M-BART are the easiest to distinguish from human. Models M2M-100 and ruGPT2-large which had the shortest samples in the dataset, were the most difficult to identify.

Table 4: Classification report for Multi model

	precision	recall	f1-score	support
Human	0.81	0.90	0.85	8524
M-BART	0.69	0.53	0.60	418
M-BART50	0.56	0.39	0.46	1986
M2M-100	0.36	0.23	0.28	1804
OPUS-MT	0.36	0.64	0.46	2014
mT5-Large	0.37	0.22	0.27	810
mT5-Small	0.61	0.43	0.50	490
ruGPT2-Large	0.67	0.63	0.65	395
ruGPT3-Large	0.63	0.59	0.61	1645
ruGPT3-Medium	0.72	0.56	0.63	1170
ruGPT3-Small	0.66	0.72	0.69	1155
ruT5-Base	0.66	0.79	0.72	440
ruT5-Base-Multitask	0.48	0.22	0.30	370
ruT5-Large	0.37	0.09	0.14	290

6.3 Attacks

We evaluate our approaches based on two metrics: SARI (the target metric of the text generation task, higher is better) and attack success rate (higher is better).

The unmodified model achieved SARI 39.06, and only 35% of samples were not detected by the binary classification model. Scoring attack using discriminator model with the best parameters achieved SARI

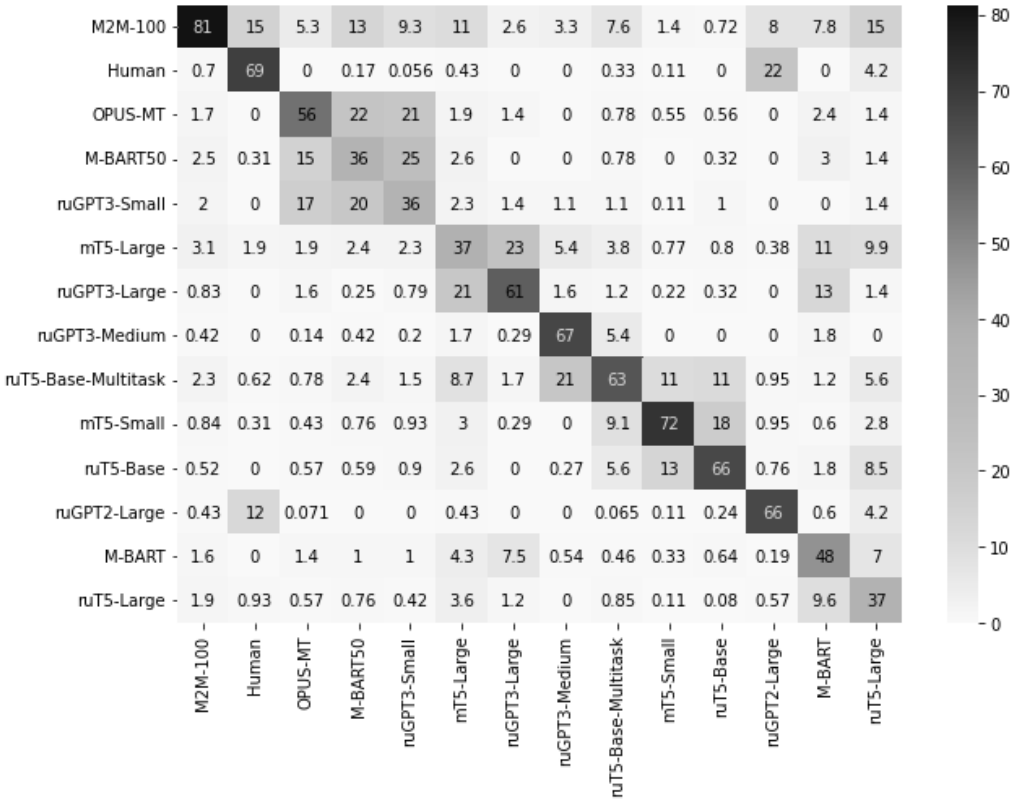

Figure 3: Confusion matrix for the multi model.

38.27, and 46% samples successfully escaped detection. By modifying the number of generated samples before scoring, we can trade-off between increasing SARI and detection rate.

Beam search attack was able to achieve SARI 38.61, and 56% samples were not identified as machine-generated. See Figure 4.

Examples of original and adversarial samples are shown in Table 5. Binary class is predicted by the Binary model. Adversarial beam search sometimes produces longer sentences compared to the original model. This may be the cause of reduced SARI score, however some of the sentences are more fluent and are not detected as machine-generated.

7 Conclusion

With our simple approach, we are able to achieve high accuracy on the leaderboard. However, it does not indicate practical applicability. For shorter texts, the model's accuracy is as low as 75%. The future development of large language models will make this task even more challenging. Additionally, adversarial generation can fool some models of automatic classification. Ensemble-based models or statistical methods may be more resistant to such attacks.

References

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration.

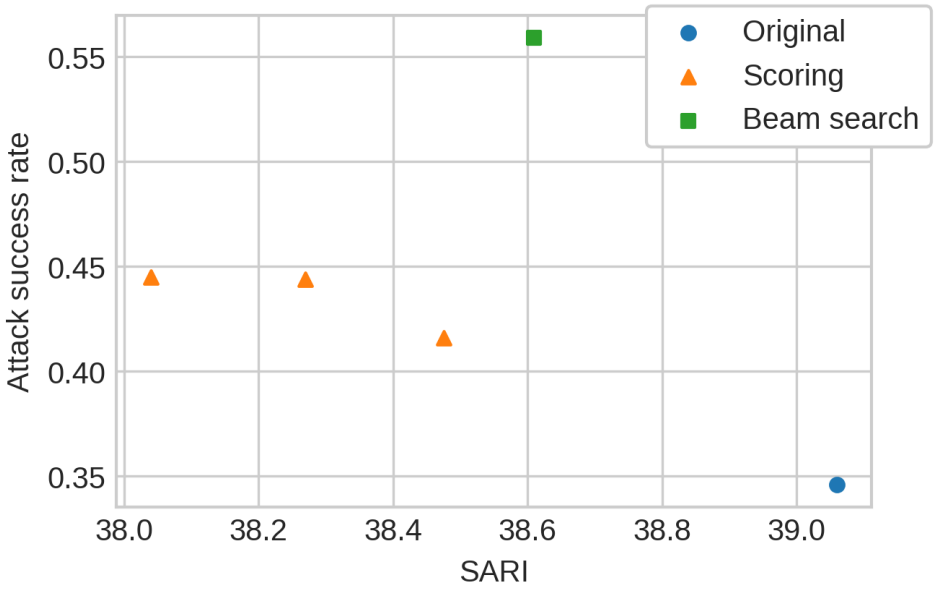

Figure 4: Attack success rate and SARI score for different attack methods.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. P 1808–1822, 01.

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. *CoRR*, abs/1907.11692.

Mikhail Orzhenovskii. 2021. Rusimscore: unsupervised scoring function for russian sentence simplification quality. // *Proceedings of the international conference Dialogue 2021.*

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers.

Andrey Sakhovskiy, Alexandra Izhevskaya, Alena Pestova, Elena Tutubalina, Valentin Malykh, Ivan Smurov, and Ekaterina Artemova. 2021. Rusimplesenteval-2021 shared task: Evaluating sentence simplification. // *Proceedings of the international conference Dialogue 2021.*

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Discriminative adversarial search for abstractive summarization.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the The RuATD Shared Task 2022 on Artificial Text Detection in Russian. // *Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference Dialogue*, volume 21, P xxx–xxx.

Tatiana Shavrina, Alena Fenogenova, Anton Emelyanov, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. Russiansuperglue: A russian language understanding evaluation benchmark. *arXiv preprint arXiv:2010.15925.*

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. // *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, P 8384–8395, Online, November. Association for Computational Linguistics.

Таблица 5: Generated text examples

Sentence	Model	Binary class
Сенат США постановил, что законы штатов запрещают компаниям платить за пользование интернетом.	Original	M
Сенат США принял закон, запрещающий властям штатов и местным администрациям взимать налоги с пользователей Интернета.	Beam search	H
Английский язык был языком в раннем средневековье.	Original	M
Английский язык - это язык, на котором говорили в раннем средневековье в Британии.	Beam search	M
Додд сказал, что британскую речь можно назвать лицемериями.	Original	M
Додд сказал, что британская речь - пример лицемерия. Он вспомнил события в Ирландии, а также в Индии.	Beam search	M
Анализ грунта, который доставил Аполлон, дал понять, что лунная почва по составу отличается от земной.	Original	H
Анализ лунного грунта показал, что он по составу сильно отличается от земной почвы.	Beam search	H

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

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.