# DIALOG-22 RuATD Generated Text Detection

**Narek Maloyan**
Lomonosov MSU
maloyan.narek@gmail.com

**Bulat Nutfullin**
Lomonosov MSU
bulat15g@gmail.com

**Eugene Ilyshin**
Lomonosov MSU
john.ilyushin@gmail.com

**Abstract**

Text Generation Models (TGMs) succeed in creating text that matches human language style reasonably well. Detectors that can distinguish between TGM-generated text and human-written ones play an important role in preventing abuse of TGM. In this paper, we describe our pipeline for the two DIALOG-22 RuATD tasks: detecting generated text (binary task) and classification of which model was used to generate text (multiclass task) (Shamardina et al., 2022). We achieved 1st place on the binary classification task with an accuracy score of 0.82995 on the private test set and 4th place on the multiclass classification task with an accuracy score of 0.62856 on the private test set. We proposed an ensemble method of different pre-trained models based on the attention mechanism[1].

**Keywords:** Generated text, text classification, ensemble methods, multi-class classification task
**DOI:** 10.28995/2075-7182-2022-21-394-401

**Аннотация**

Модели генерации текста успешно синтезируют текст, который сложно отличим от написанного человеком текста. Детекторы, способные отличить текст, созданный автоматически, от написанного человеком текста позволяют предотвратить злоупотребление сгенерированными текстами. В этой статье мы описываем наше решение для задач DIALOG-22 RuATD по обнаружению сгенерированного текста и классификации с помощью какой модели был сгенерирован текст. Мы заняли 1-е место в задаче бинарной классификации с оценкой точности 0,829 в частном наборе тестов и 4-е место в задаче мультиклассовой классификации с оценкой точности 0,628 в тестовом наборе. Наше решение является ансамблем дообученных моделей, основанных на механизме внимания.

**Ключевые слова:** Сгенерированные тексты, классификация текстов, ансамблевые методы, задача многоклассовой классификации

## 1 Introduction

As the language neural nets got better at generating texts, it's getting harder and harder to distinguish the human-written text from generated one. So manual detection of these texts got impossible. For that reason, it's desirable to build a system that can automatically detect generated text.

The proposed system will use various features extracted from the text such as length, punctuation, word choice etc. To determine whether the text is human-generated or not. The accuracy of this system can be improved by using machine learning algorithms which will learn how humans generate texts and then use those features for detection purpose.

There are many ways to build such a system, but probably the most reliable one is based on machine learning algorithms. These algorithms can be trained on a large number of examples - both human-generated and computer-generated texts. After being trained, they should be able to identify which texts are computer-generated with high accuracy.

This approach already works in other areas, such as spam detection. Some early experiments have shown promising results and indicate that this approach works well for the detection of the generated text.

The DIALOG-22 shared task of RuATD 2022 had 2 tracks for binary classification and multiclass classification. In this report, we will describe the data, how we handled it, the models we used, and the ensembling technique.

---

[1] github.com/maloyan/ruatd

## 2    Task Definition

For the binary classification task $F_{binary}$, we frame the generated text detection problem as follows: given a piece of text $X$, label it as either human-written or machine-generated $y_{binary} = \{H, M\}$.

$$F_{binary} : X \to y_{binary}$$

For the multiclass classification task $F_{multiclass}$, we set up the problem as follows: given a piece of text $X$, label it as one of the 14 classes that represent deep neural models

$y_{multiclass} = \{M2M{-}100, Human, OPUS{-}MT, M{-}BART50, ruGPT3{-}Medium,$
$ruGPT3{-}Small, mT5{-}Large, ruGPT3{-}Large, ruT5{-}Base{-}Multitask,$
$mT5{-}Small, ruT5{-}Base, ruGPT2{-}Large, M{-}BART, ruT5{-}Large\}$

$$F_{multiclass} : X \to y_{multiclass}$$

## 3    Datasets

Provided datasets offer the train and test splits. The part of the set was annotated automatically by different generative models. Various language models were fine-tuned on different tasks: machine translation, paraphrasing, summarization, simplification, and unconditional text generation - are used to generate texts. The texts written by a human were collected from open sources from different domains. (rua, a), (rua, b).

| Text | Class |
|---|---|
| Обустройство тротуаров, мостовых (в том числе тротуарной плиткой). | H |
| Минстрой обозначил способы снижения энергоемкости российской экономики. | M |
| В конце 1873 года военный суд вынес решение по делу Франциска Ахиллы Базейн. | M |
| увеличение правовой грамотности и развитие правосознания граждан. | H |

Table 1: Example of the binary classification data.

| Text | Class |
|---|---|
| Прочла автобиографию Каутского, Одесса, 1905. | Human |
| Вы не можете быть в печи и в myulin. | M-BART50 |
| Вторая попытка привела к тому же результату. | OPUS-MT |
| Сколько учеников в вашем классе? | M2M-100 |

Table 2: Example of the multiclass classification data.

## 4    Related Works

In this section, we will discuss various methods for detecting machine-generated texts.

Over the past years, many approaches appeared for generated text detection. The latest works are usually based on using transformer-based models. Either fine-tuning the proposed task or using probabilities distribution to make decision-based on them (Ippolito et al., 2019). Here we list some examples of different methods:

- First, we calculate the mean likelihood over all machine-generated sequences, then the mean of human-written ones. If the likelihood according to some language model is closer to the machine-generated mean likelihood, then we consider it as generated text (Solaiman et al., 2019);
- In GLTR (Gehrmann et al., 2019) described a method using a language model to compute the probability distribution of the next word given the text sequence. For each sequence position, we get the likelihood of the ground-truth next word within this list. Then these ranks are displayed on a histogram. Based on the distribution of bins, we can decide if this text is generated or not.

- Bert fine-tuning on the classification task. Having a label of text if it's machine-generated or not, we can fine-tune the language model to predict (Solaiman et al., 2019)
- Also, there is possible to use human-machine collaboration. Real or Fake Tool provides a game-like interface for humans to decide at what point the text begins to be written by a computer (Dugan et al., 2020).

More approaches are described in the survey (Jawahar et al., 2020).

## 5 Experiment Setup

In this section, we present the experiment configurations we use to solve binary and multiclass tasks.

### 5.1 Data preproccessing

We decided not to perform any preprocessing of the text itself. Regarding the data split:
- **Binary classification**. We concatenated the train set with the validation set. This new concatenated dataset gave us an option to perform a 5-fold cross-validation;
- **Multiclass classification**. We took the data as is without changing anything.

### 5.2 Models

All models we used for fine-tuning we got from transformers library (Wolf et al., 2020). Here we will describe these models:
- **sberbank-ai/sbert_large_nlu_ru**(hab, 2020), **sberbank-ai/ruBert-large**, **DeepPavlov/rubert-base-cased** are BERT models (Devlin et al., 2018). Sber-ai models were fine-tuned on a closed dataset collected by their research group (sbe, ). DeepPavlov's RuBERT was fine-tuned on the Russian part of Wikipedia and news data.
- **IlyaGusev/mbart_ru_sum_gazeta** is a mBART model fine-tuned on the dataset for summarization of Russian news(Gusev, 2020). The original MBart model was pretrained on large-scale monolingual corpora in many languages using the BART objective. mBART is one of the first methods for pre-training a complete sequence-to-sequence model by denoising full texts in multiple languages, while previous approaches have focused only on the encoder, decoder, or reconstructing parts of the text (Liu et al., 2020).
- **MoritzLaurer/mDeBERTa-v3-base-mnli-xnli** (He et al., 2021) This multilingual model is suitable for multilingual zero-shot classification. The original model was pre-trained by Microsoft on the CC100 multilingual dataset (Wenzek et al., 2020).
  This model was fine-tuned on the XNLI development dataset and the MNLI train dataset. The XNLI development set consists of translated texts for each of the 15 languages (Conneau et al., 2018).
- **DeepPavlov/xlm-roberta-large-en-ru-mnli** is an XLM-RoBERTa model (Conneau et al., 2019) which was fine-tuned on the english-russian part of the MNLI (Williams et al., 2017) dataset.

### 5.3 Binary classification (with ensembling technique)

Five chosen models were used in the experiement: **sberbank-ai/sbert_large_nlu_ru**, **sberbank-ai/ruBert-large**, **IlyaGusev/mbart_ru_sum_gazeta** , **MoritzLaurer/mDeBERTa-v3-base-mnli-xnli**, **DeepPavlov/xlm-roberta-large-en-ru-mnli**

Let's describe steps for training each of these models:
1. We split our training dataset into non-overlapping 5-folds and performed cross-validation (Figure 1);
2. for each validation fold we predicted the target and as a result, we got out-of-fold predictions for training data;
3. for the test set, we predicted the target from every 5 models from cross-validation and averaged the result which is marked with red;

As a meta-model, we choose Logistic Regression, which was trained on out-of-fold predictions. Then we predict the final results for the test set.
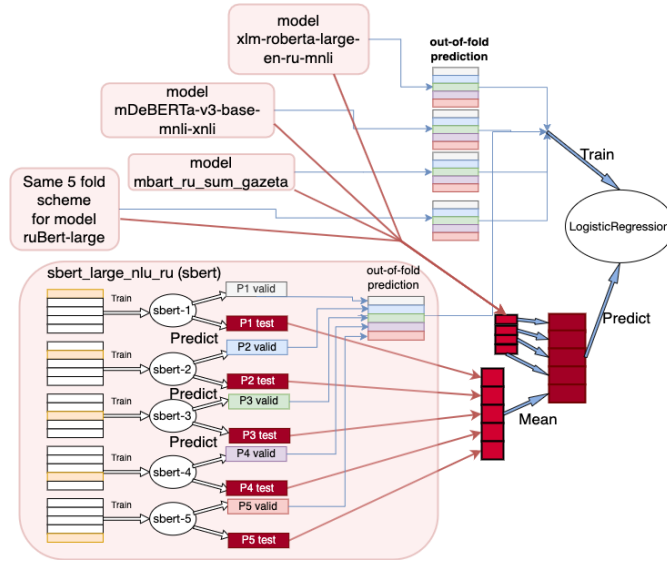
Figure 1: Ensembling scheme

## 5.4 Multiclass classification

For the multiclass classification we chose these models: **DeepPavlov/rubert-base-cased**, **DeepPavlov/xlm-roberta-large-en-ru**, **IlyaGusev/mbart_ru_sum_gazeta**. We fine-tune these models without cross-validation only on provided train set.

## 6 Results

Table 3 shows each model accuracy score on binary classification task. The multiclass classification accuracy is shown in Table 4. On both tasks, the best performing single model was **DeepPavlov/xlm-roberta-large-en-ru-mnli**. We managed to ensemble models in the binary task, so the ensemble of models showed the best accuracy.

| Model name | Accuracy |
|---|---|
| sberbank-ai/sbert_large_nlu_ru | 0.79986 ±0.003 |
| sberbank-ai/ruBert-large | 0.80154 ±0.002 |
| IlyaGusev/mbart_ru_sum_gazeta | 0.80566 ±0.001 |
| MoritzLaurer/mDeBERTa-v3-base-mnli-xnli | 0.80710 ±0.001 |
| **DeepPavlov/xlm-roberta-large-en-ru-mnli** | **0.81708 ± 0.002** |
| **Ensemble** | **0.82995**[2] |

Table 3: Results for binary classification for different models

| Model name | Accuracy(validation) | Accuracy(kaggle public/private) |
|---|---|---|
| IlyaGusev/mbart_ru_sum_gazeta | 0.6142 | 0.61459/0.61092 |
| DeepPavlov/rubert-base-cased. | 0.6045 | 0.60433/0.60472 |
| **DeepPavlov/xlm-roberta-large-en-ru-mnli**. | **0.6242** | **0.62856/0.62644** |

Table 4: Results for multiclass classification for different models

# 7 Conclusion

In this paper, we described our pipeline for the DIALOG-22 RuATD challenge. Our solution achieved 1st place in binary classification using ensembling techniques and 4th place for the multiclass classification task using only a single model. However, the proposed solution requires a lot of computational power, so it cannot be used in real-time systems to detect generated texts. But it gives us an understanding that we still need to upgrade methods to distinguish generated texts from human-written ones.

# References

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. Roft: A tool for evaluating human detection of machine-generated text. *arXiv preprint arXiv:2010.03070*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Ilya Gusev. 2020. Dataset for automatic summarization of russian news. // *Conference on Artificial Intelligence and Natural Language*, P 122–134. Springer.

2020. Natulral language model learning with bert and tensorflow link:https://habr.com/ru/company/sberdevices/blog/527576/, Nov.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

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. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ruatd 2022 (binary task). `https://www.kaggle.com/competitions/ruatd-2022-bi`. Accessed: 2022-03-28.

Ruatd 2022 (multi task). `https://www.kaggle.com/competitions/ruatd-2022-multi-task`. Accessed: 2022-03-28.

Sberbank-ai. `https://huggingface.co/sberbank-ai`.

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.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. // *Proceedings of the 12th Language Resources and Evaluation Conference*, P 4003–4012, Marseille, France, May. European Language Resources Association.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. // *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, P 38–45.