Automatic Detection of Machine Generated Texts: Need More Tokens

German Gritsay
Moscow Institute of Physics and Technology
Antiplagiat Company
Moscow, Russia
gritsai@ap-team.ru

Andrey Grabovoy
Moscow Institute of Physics and Technology
Antiplagiat Company
Moscow, Russia
grabovoy@ap-team.ru

Yury Chekhovich
FRC CSC of the RAS
Antiplagiat Company
Moscow, Russia
chehovich@ap-team.ru

***Abstract*—Current advances in text generation using neural approaches make it possible to create texts hardly distinguishable from human texts. A survey to improve the efficiency of automatic discriminators to detect machine-generated text could be useful in revealing features directly affecting the quality of detection. Recently, many works have appeared in the natural language processing (NLP) and machine learning (ML) communities to create accurate detectors for the English language. Despite the importance of this problem, all the works that exist for Russian rely only on short sequence length. In this work, we argue that context length matters. First, we present novel open dataset for Russian language with long texts for the task of machine-generated text detection. We describe the collection, generative models selection, and sampling process in detail and present exploratory analysis of the quality of various discriminators. Second, we conduct a set of learning experiments to build accurate machine-generated text detectors for both English and Russian languages. In addition, we conduct a comparative analysis of the quality of discriminators when training a multi-task model.**

***Index Terms*—machine-generated text, text classification, natural language generation, multi-task learning, datasets creation**

I. Introduction

Contemporary neural generative models are currently capable to generate texts very close to human ones in terms of logical structure, coherence and grammar [1], [2], [3]. Such models are useful in a wide variety of applications, but with their rising popularity, harmful users may use them for malicious purposes, such as fake news [4], spamming [5], fake reviews or plagiarism [6], [7]. Thus, it is important to build a proper detector for the task of identifying machine-generated texts in common case.

Speaking about the English language, in previous works a pipeline has been built for detecting the generated texts. The task is usually defined as a classification problem and approaches vary from using classical machine learning methods to fine-tuning pre-trained transformer-based models [8]. However, a major part of building an accurate classifier is training data. We decided to pay attention to the nature of the data fed to fine-tuning models. Texts generated by models with various numbers of parameters can, when fine-tuning at some point, improve the generalization ability of the classifiers. We have tested the hypothesis that increasing the length of the input sequence should improve the model's understanding of the context. It was found that this allows to reach a plateau in the metrics used.

For the Russian language, recent work in the field of machine-generated text detection research is based on examples from the RuATD (The Russian Artificial Text Detection) data [9]. However, in these datasets texts are represented by short sequences, we hypothesized that this length is not sufficient to build a correct discriminator for Russian in common case. For this reason, we present the construction of novel open dataset that contains long samples and texts generated by different models and sampling methods, which is an important aspect of further training, as was shown in [10]. We conduct experiments on presented data with negative samples from Wikipedia articles corpus. Experiments showed that our method significantly outperforms strong transformer-based baselines on RuATD datasets.

Our main contributions are three-fold:

1) We set up a new dataset for Russian containing long extracts generated by models with different number of parameters and sampling method.
2) We present methods for handling the data - mixing to improve generalizability, increasing the length of input sequence to provide a better understanding of the context.
3) We introduce a multi-task model to maximize the representational similarity texts created by the same machine.

II. Task Definition

We frame the detection problem as a binary classification task: given an fragment of text X, label it as human-written $\{0\}$ or machine-generated $\{1\}$.

$$F_{\mathbf{binary}} : \mathcal{X} \to \{0, 1\} \tag{1}$$

In particular, we are interested in how variables such as input sequence length, decoding strategy and mixing coefficient would affect performance on this classification task. We thus create several datasets by combining or cropping. Each is approximately balanced between positive examples of machine generated text and negative examples of human-written text.

For experiments with multi-task learning we introduce multiclass classification problem: given a fragment of text X, label it as human-written $\{0\}$ or various k classes that represent language models $\{1,2,\ldots,\text{k}\}$.

$$F_{\textbf{multiclass}} : \mathcal{X} \rightarrow \{0, 1, \ldots, k\} \qquad (2)$$

By training a separate classifier on each dataset, we are able to answer questions about sufficient context length for correct classification and decoding strategy resulting in text that is the easiest to automatically disambiguate from human-written text. Multiclass experiments allow us to close the embeddings of texts from one machine and separate the different ones.

III. Datasets

The model architecture underlying all the state-of-the-art language models is the Transformer [11]. Text generation from such models as GPT [12], GROVER [13], FAIR [14] which are based on transformer architecture tends to be grammatically correct, coherent and uses world knowledge. It was proved by a study of linguistic features [15]. These models are generally trained using the language modeling objective on large amounts of raw text from a diverse set of sources (like Wikipedia, Reddit, and news sources).

A. English language

GPT-2 An autoregressive language model that defines the probability distribution over the next token given the previous tokens in a sequence. Next token selection is the task of the sampling method. By choosing a decoding method, it is possible to obtain texts of variable coherence. In our experiments, we used data from `Gpt-2 output dataset`[1], which allows us to compare data from models with different numbers of parameters and different sampling methods. The authors provided data obtained by three decoding strategies: *top-k* (restricts distribution to all but the k most likely tokens), *top-p* (truncates the distribution to most-likely next tokens such that the cumulative likelihood of these tokens is no greater than a constant) and *pure* (sample from the untruncated distribution). Also, there is a choice of texts depending on the number of parameters of the model with which it was generated:

- small - 117M
- medium - 345M
- large - 742M
- xl - 1542M

This is what we will use to blend data from multiple models. In addition, we form “negative” examples of human-written text by taking excerpts of web text that come from the same distribution as GPT-2’s training data [10].

TuringBench A benchmark environment that contains benchmark tasks (Turing Test and Authorship Attribution) and datasets (Binary and Multiclass settings) [16]. This corpus contains human-written news articles, collectively categorized by a single human author, and machine-generated texts from 19 different neural language generators. The models generate the texts based on the titles of the human-written articles. This controls for topic differences between samples by different authors. There are a total of 200,000 texts from 20 authors.

B. Russian language

RuATD The Russian Artificial Text Detection (RuATD) shared task explores the problem of artificial text detection in Russian. The first of its kind a diverse automatic text detection corpus in Russian. The generated texts were obtained in different ways (machine and back translations, paraphrase generation, simplification, etc.). Unfortunately, the dataset has a huge disadvantage: the short length of the samples. Data from this dataset is limited to only one sentence. Experiments have shown that the quality of the discriminator directly depends on the length of the input sequence, due to the limited sample RuATD, it is difficult to build an accurate detector that takes into account this feature.

Our Dataset The problem described above led us to the task of building not only a correct detector for Russian, but also a dataset with long excerpts to help the model capture the long context. It was important for us to get samples of different generative models with a variety of decoding methods. During the selection of models, we encountered the problem of a small number of pre-trained generative models for Russian. After analysis, we have chosen 3 models for generation: two from Sberbank-AI (`rugpt3small-based-on-gpt2`[2], `rugpt3large-based-on-gpt2`[3]) and another one from Facebook (`XGLM-1.7B`[4]).

The major issue when generating text with a language model is whether to provide a priming sequence and what the length should be so that the language model will be able to continue it. To generate our data, we rejected the idea of no priming texts, given the limited number of generative models for the Russian language, because the comparison includes multilingual models that may produce unexpected behavior without priming sequence.

We decided to use two priming approaches: an input word and an input sentence. Once we have the machine-generated text, we need to connect it to the real scripts. We took them from the Russian Wikipedia, and exactly from these texts the first words and the first sentences came to the input of the generative model. We fixed the length of the generated sequence from 500 to 1200 tokens, which is comparable to a paragraph of real text.

In addition, we settled on three sampling strategies to generate the data: *top-k* = 40, *top-p* = 0.96 and *pure* (top-p = 1.0), like in [10]. Thus, we obtained 18 different corpora 25k samples each with generated texts featuring different numbers of parameters, sampling methods, and priming sequence. Combining all our generated data and balancing it with human texts, we have created and published a dataset for the detection of machine-generated texts for the Russian language [17].

[1] `https://github.com/openai/gpt-2-output-dataset`

[2] `https://huggingface.co/sberbank-ai/rugpt3small_based_on_gpt2`

[3] `https://huggingface.co/sberbank-ai/rugpt3large_based_on_gpt2`

[4] `https://huggingface.co/facebook/xglm-1.7B`

IV. Experiments

Based on previous achievements in the field of machine-generated text detection, the transformer-based pretrained model was chosen as the basic discriminator. Comparing its results with classical methods of machine learning or other neural networks it was shown that the quality of classification fell. Experimenting with these neural models, we split the dataset into the training, validation and testing parts in 8:1:1 ratio.

We adopt `RoBERTa-base`[5] [18] as the direct baseline for our experiments because RoBERTa achieves state-of-the-art performance on several benchmark NLP tasks. For datasets in Russian we will use a multilingual model `XLM-RoBERTa-base`[6] [19]. In addition, we compare the results of our hypotheses with `DeBERTa-base`[7]. This transformer-based model improves the BERT and RoBERTa using two novel techniques. In the works presented earlier, DeBERTa was not used to estimate the quality of detection, although the results presented in [20] make it clear that this transformer-based model is capable of outperforming RoBERTa in multiple challenges. Since regular transformer-based models are limited to an input sequence length of 512 tokens, for experiments with longer lengths we chose `Longformer-base`[8] - transformer model for long documents which started from the RoBERTa checkpoint and pretrained for MLM on long documents [21].

A. Experiments with data

Original data We decided to measure the quality of state-of-the-art detectors for English language on two samples from `Gpt-2 output dataset`. The description of the selected parts is presented in Appendix A.

We fine-tuned models on the described data, training the classifier first, with frozen encoder weights, and then the full model. This method of training allows not to shift the distribution of encoder weights totally, but only to bias it towards the presented data. Testing was performed on the respective datasets.

This experiment showed DeBERTa claims to be able to adapt more coherent and human-like texts better, because as the parameters of the generative model increase, the naturalness of the text presented also rises.

Mixed data Since we noticed that increasing the parameters of the generative model causes the detector to make errors, we decided to measure the quality of the transformer-based discriminators after training on mixed data.

In this approach, we blended data corpus on which the fine-tuning of the models is performed. In equal portions, it contains samples from *small*, *medium*, *large* and *xl* data. In a such way, the model sees both ”simple” samples from models with a small number of parameters and more ”complicated” ones during fine-tuning. The test sample, on the other hand, remained unchanged, with samples from just one generative model. We also fine-tuned these models by training the classifier first, with frozen encoder weights, and then the full model. The description of the selected architecture for the classifier head is presented in Appendix B.

This training approach improved the quality of the RoBERTa detector, while it had no positive effect on the DeBERTa results.

Sequential data mixing In this experiment, we decided to improve the generalization ability of the discriminators by sequentially mixing texts from models with a high number of parameters. This way of fine-tuning requires more phases of learning, however, detector faces more coherent texts sequentially and starting with ”simpler” texts adjust more accurately.

The training consisted of four successive stages. The first of these was to train the classifier with encoder frozen weights and then the entire model on the data from the ”small-117M” generative model only. Then, in the second stage, we mixed to ”small” samples, examples from ”medium-345M” data in equal proportions. On the third, we added ”large-762M” and at the end – ”xl-1542M”. Testing was done on untouched single data. By mixing examples from generative models, we also added real texts, so the classification problem was always solved on a balanced set. The description of the experimental setup is presented in Appendix C.

TABLE I
FINE-TUNING TRANSFORMER-BASED DETECTORS FOR ENGLISH

Models	**Datasets**					
	Original		Mixed		Sequential Mix	
	Small	***Xl***	***Small***	***Xl***	***Small***	***Xl***
RoBERTa	**0.98**	0.85	0.98	0.89	0.98	**0.94**
DeBERTa	0.98	**0.93**	**0.99**	**0.92**	**0.99**	0.90

In Tab. I. it can be seen that any mixing of data improved the quality of classification for RoBERTa and worsened it for DeBERTa. The increase in accuracy metrics on ”Xl” data for Roberta amounted to 10%, while Deberta’s quality dropped by 2%. DeBERTa is a good discriminator for data from models with fewer parameters. Whereas RoBERTa improves its generalizability by looking at data from generative models with different number of parameters.

TABLE II
FINE-TUNING TRANSFORMER-BASED DETECTORS FOR RUSSIAN

Models	**Datasets**			
	Original		Sequential Mix	
	Small	***Large***	***Small***	***Large***
XLM-RoBERTa	0.92	0.89	0.95	0.95

We decided to test the hypothesis of quality improvement by data mixing on our generated dataset for the Russian language. After experiments for English, we settled on a stable RoBERTa but on its multilingual version and chose approach with sequential data mixing with identical techniques. To

[5] `https://huggingface.co/roberta-base`

[6] `https://huggingface.co/xlm-roberta-base`

[7] `https://huggingface.co/microsoft/deberta-base`

[8] `https://huggingface.co/allenai/longformer-base-4096`

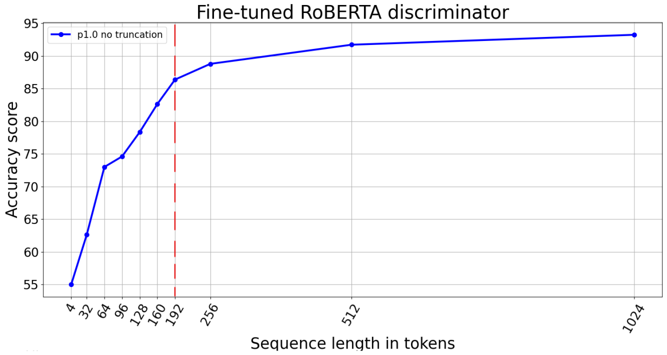

Fig. 1. Accuracy increases as the length of the sequences used to train the discriminator is increased. After **n=512** metric reaches a platea. Red line shows the boundary of the previous experiments with increasing length.

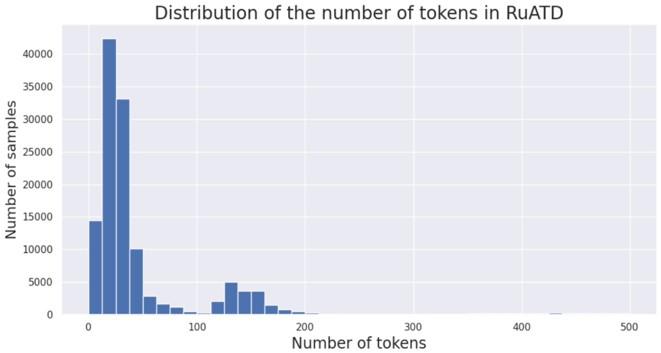

Fig. 2. The majority samples of RuATD data is up to 200 tokens long, with given amount of tokens, it will be difficult for the model to understand the long context.

begin with, we conducted a basic experiment with original data. Tab. II showed that the mixing hypothesis improves the generalization ability of the model for Russian as well.

B. Experiments with data length

The length of the input sequence can directly affect the classifier's ability to understand the context. D. Ippolito in [10] showed that for different sampling methods, increasing the input sequence length improves the quality of transformer-based discriminators. However, the experimenters settled on a length of *n=192*. For samples based on *top-k*, this is enough; nevertheless, for *top-p* and *pure*, Fig. 1. showed that the detector for English language reaches a plateau in the given metrics only when considering from *n=512* to *n=1024* tokens of the input text. Solaiman et al. in [22] set up an experiment with training up to *n=128* tokens and testing at long lengths up to *n=512*. According to this task formulation, it means that the training and test samples are from different general data distributions. In our experiment it is assumed that the samples are aligned in length, and therefore from the same general data distribution.

This increase of input data significantly grows the detector's training time, however, length is what assists the model in capturing context in a longer range, which at several points is the key feature for detecting machine-generated texts.

We plotted in Fig. 2. the distribution of the number of tokens in RuATD data and noticed that the majority are samples up to 200 tokens long. The accuracy of the detector based on these samples will show poor results in common case.

We decided to test the hypothesis of quality improvement by increasing the number of tokens on our generated dataset for the Russian language. Moreover, we repeated the tests for all sampling methods mentioned above. Due to the limited choice of pretrained models, the experiment was conducted only up to *n=512* tokens, although it was enough for the declared metric to reach the plateau. Fig. 3 showed that sequence length *n=256* to *n=512* token range is key for the proposed detector to understand context, which is why we argued for the importance of using longer texts in the training data. The accuracy score achieved with *n=64* (average value of token length in samples from the RuATD data) correlates with the results of DIALOG-22 RuATD [23].

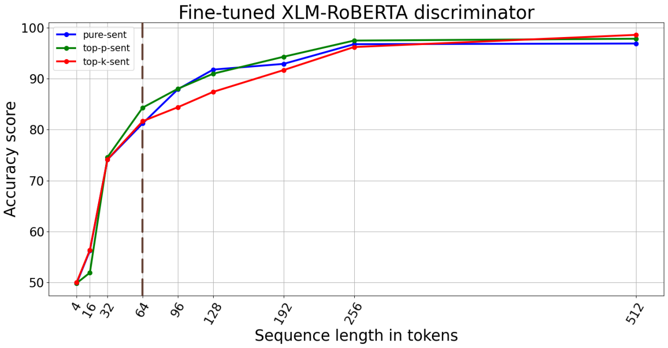

Fig. 3. Accuracy increases as the length of the sequences used to train the discriminator is increased for all presented methods of sampling. After **n=256** metric reaches a platea. Brown line shows the average length of tokens in the RuATD dataset.

Also, it was noticed that in contrast to the English dataset, where texts with *top-k* reache a plateau on 16 tokens [10] for the Russian language this assumption does not work, here all 3 stated sampling methods showed comparable results.

C. Experiments with multi-task learning

Multi-task learning – the ability to make the model learn multiple tasks simultaneously. As stated in [24] this type of learning can offer advantages like improved data efficiency, reduced overfitting through shared representations, and fast learning by leveraging auxiliary information. We chose an approach for multi-task learning that is based on adding additional prediction heads to the BERT-like model and a common encoder.

We hypothesized that this method of training detectors could improve text representations, in particular, to close embeddings of texts by the same author and separate different ones, since one of the heads would solve the problem of Authorship Attribution [15]. This way, with ***k*** datasets from each language generative model, we create ***k + 1*** heads for the multi-task model. Of these, ***k*** will solve the binary classification problem on their samples, and the last one will solve the multiclass

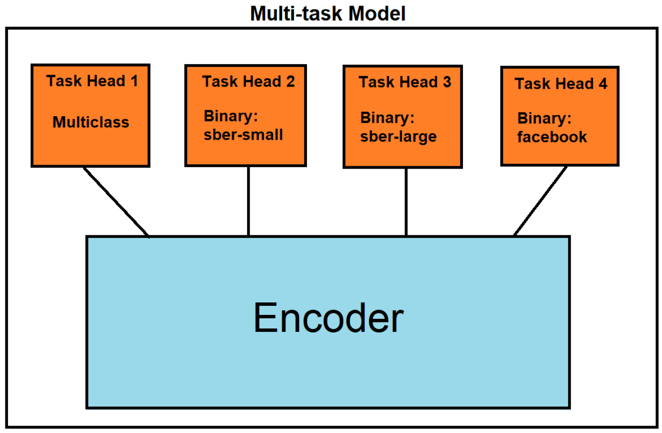

Fig. 4. Multi-task model to solve the task of detection machine-generated texts. The encoder is the same for all heads. First head fine-tuned for the problem of multiclass classification into 4 classes on merged data (human, sber-small, sber-large, facebook) and another for binary on single data respectively.

classification problem with merged generated data from all binaries.

We decided to test the multi-task model for English on the Turing Bench. However, when analyzing the samples we concluded that it would not be possible to obtain a correctly trained classifier. Samples presented in datasets overlap with each other in train and test parts, there is leakage among all datasets, and some sentences are repeated several times in the same corpus. The model would be overfitted and would not learn at all the representations necessary for proper classification. We could clean up the data and remove all dependencies, but then corpus would become so small that it would not be enough to train a correct detector.

So we settled on an experiment for the Russian language on our generated datasets. The model we trained is presented at Fig. 4. Training consisted of three stages: first we trained on 4 epochs only classifier heads with frozen encoder weights, second, we trained on 4 epochs full model and the third stage again froze the weights and fine-tune classifier heads on 1 epoch. This learning stages helps to shift the distribution of the encoder weights in the right direction.

Tab. III. showed that multi-task learning improves the quality of the discriminator on binary datasets. The amount of data for the multiclass head was greater than the binary parts, so we can see a better score for this classification task. This is exactly what we expected, in general, the problem of multi-task can be imagined as data mixing only supervised approach. Since in the previously described approach of mixing we just concatenate samples without giving model the correct labels. However, here the model receives this information. Multiclass head helps the embeddings of texts by the same author to converge in the representation space, which positively affects the quality of the classification within the same author in other words binary.

TABLE III
FINE-TUNING TRANSFORMER-BASED DETECTORS WITH MULTI-TASK LEARNING

Models	Datasets			
	Merged Data	*Sber-small*	*Sber-large*	*Facebook*
XLM-RoBERTa	**0.97**	0.92	0.89	0.90
XLM-RoBERTa multi-task	0.96	**0.95**	**0.95**	**0.96**

V. CONCLUSION

Successful machine-generated texts detection necessitates correctly provided data. In this work, we made the first attempt to create an effective discriminator for Russian language for long texts classification. For this reason, we presented our own open dataset which is collected from different generative models and various sampling methods. We managed to assemble a corpus of 450k machine-generated samples with lengths from 500 to 1200 tokens and real texts in the same amount.

Experiments were also conducted with data mixing approaches, in both cases, the quality of the stated metric increased for English and Russian languages, which indicates the growing generalization ability of the fine-tuned detector.

Furthermore, we have studied the effect of the input sequence length on the quality of the discriminator. We extended earlier experiments and showed that the model reached a plateau with more tokens, which suggests that the upper bound for this type of classifier has been reached. For Russian, the *top-k* sampling method proved to be similar to the *top-p* and *pure* in terms of detection quality as the number of tokens increased, which was not observed for English. In addition, reaching the plateau occurs earlier than for the English samples.

Finally, we applied multi-task learning to our generated data in order to confirm the hypothesis about the increasing the similarity of the embeddings of texts by the same author. This type of training results in an improvement in the quality of binary tasks heads.

APPENDIX A
DATASETS DESCRIPTION

Table IV shows the number of samples used for training and evaluating each method and models we have described. When we talk about the *original* data, we mean that half of the texts are machine-generated and half are real. The *mixed* section assumes that machine-generated texts contain samples from multiple machines with different number of parameters in equal proportions. In the case of the *sequential mix*, the table shows a figure for the amount of data in the last stage of training, when the corpus contains generated texts from all claimed models. It is worth noting that at all stages above the balance between real and generated samples is still present. *Merged* data is dedicated to multiclass classification and contain samples of each class in equal proportions, there are 4 of them (real, sber-small, sber-large, facebook). *Small* and *xl (large)* from the tables in the article meaning corpora

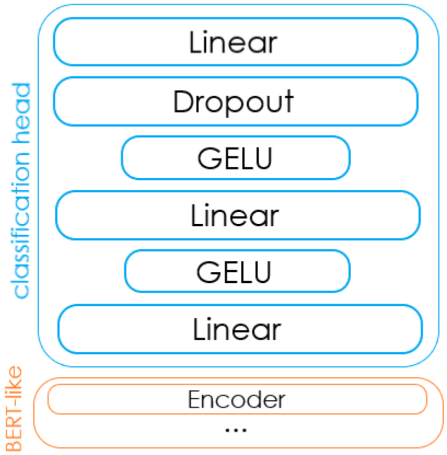

Fig. 5. Classification head architecture that we use for fine-tuning models.

with generated samples from *Gpt-2 output dataset* (small-117M and Xl-1542M) for English language and samples from *Sberbank-AI* (rugpt3small, rugpt3large) for Russian. For `Gpt-2 output dataset` we set pure sampling (p=1.0) the sampling strategy of generator, which leads to better-generated text quality [10].

TABLE IV
DATA AND ITS AMOUNT FOR EXPERIMENTS

Datasets	**Parts**		
	Train	***Validation***	***Test***
GPT2 - original	200k	10k	10k
GPT2 - mixed	200k	10k	10k
GPT2 - sequential mix	200k	10k	10k
OurData - original	76k	12k	12k
OurData - sequential mix	76k	12k	12k
Multi-Task - original	38k	6k	6k
Multi-Task - merged	76k	12k	12k

APPENDIX B
CLASSIFICATION HEAD

In all of the experiments described in the article, the architecture shown in Fig. 5 was used as the classification head. We decided to establish 3 linear layers, which were separated by activation functions and dropout.

APPENDIX C
EXPERIMENTAL SETUP

In this topic, we describe training details of each part of Experiments section. We employed cross-entropy loss as the loss function and applied AdamW as the optimizer for all experiments. Additionally, we used the linear learning rate scheduler. We set the learning rate as 1e-5 and used 2 NVIDIA GeForce RTX 3090 and NVIDIA GeForce RTX 3070 Ti for training, evaluating and generating stages. For sections IV-A and IV-B we established 2 epochs for all methods. At the first epoch we updated only classifier weights and at the second all model weights were engaged in training operations. By the way, in the sequential mix training, there were 2 epochs per dataset.

REFERENCES

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” 2018. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[2] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, “Defending against neural fake news,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 2019, pp. 9051–9062. [Online]. Available: http://papers.nips.cc/paper/9106-defending-against-neural-fake-news

[3] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen, “Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection,” in *Advanced Information Networking and Applications*, L. Barolli, F. Amato, F. Moscato, T. Enokido, and M. Takizawa, Eds. Cham: Springer International Publishing, 2020, pp. 1341–1354. [Online]. Available: https://doi.org/10.48550/arXiv.1907.09177

[4] O. Bakhteev, A. Ogaltsov, and P. Ostroukhov, “Fake News Spreader Detection Using Neural Tweet Aggregation—Notebook for PAN at CLEF 2020,” in *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org, Sep. 2020. [Online]. Available: http://ceur-ws.org/Vol-2696/

[5] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, Z. Kenton, S. Brown, W. Hawkins, T. Stepleton, C. Biles, A. Birhane, J. Haas, L. Rimell, L. Hendricks, and I. Gabriel, “Ethical and social risks of harm from language models,” 12 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2112.04359

[6] A. S. Khritankov, P. V. Botov, N. S. Surovenko, S. V. Tsarkov, D. V. Viuchnov, and Y. V. Chekhovich, “Discovering text reuse in large collections of documents: A study of theses in history sciences,” in *2015 Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, 2015, pp. 26–32.

[7] K. Zhuravlev, K. Rudakov, A. Inyakin, A. Kirsanov, A. Lisitsa, G. Nikitov, N. Peskov, R. Yaminov, and Y. Chekhovich, “The system of recognition of intellectual text reuse “antiplagiat”,” in *Mathematical methods of pattern recognition: 12th All-Russian conference: Collection of reports. MAKS Press*, 2005, pp. 329–332.

[8] G. Jawahar, M. Abdul-Mageed, and L. Lakshmanan, “Automatic detection of machine generated text: A critical survey,” 11 2020. [Online]. Available: https://aclanthology.org/2020.coling-main.208.pdf

[9] T. Shamardina, V. Mikhailov, D. Cherniavskii, A. Fenogenova, M. Saidov, A. Valeeva, T. Shavrina, I. Smurov, E. Tutubalina, and E. Artemova, “Findings of the the ruatd shared task 2022 on artificial text detection in russian,” *CoRR*, vol. abs/2206.01583, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2206.01583

[10] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, “Automatic detection of generated text is easiest when humans are fooled,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 1808–1822. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.164

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” 2019. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[13] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, “Defending against neural fake news,” *Advances in neural information processing systems*, vol. 32, 2019.

[14] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, and S. Edunov, “Facebook FAIR’s WMT19 news translation task submission,” in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 314–319. [Online]. Available: https://aclanthology.org/W19-5333

[15] A. Uchendu, T. Le, K. Shu, and D. Lee, “Authorship attribution for neural text generation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 8384–8395. [Online]. Available: https://aclanthology.org/2020.emnlp-main.673

[16] A. Uchendu, Z. Ma, T. Le, R. Zhang, and D. Lee, “TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2001–2016. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.172

[17] G. Gritsay, A. Grabovoy, and Y. Chehovich, “Open access dataset for machine-generated text detection in russian.” [Online]. Available: https://doi.org/10.17632/4ynxfp3w53.1

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *ArXiv*, vol. abs/1907.11692, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1907.11692

[19] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747

[20] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention.” *CoRR*, vol. abs/2006.03654, 2020. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr2006.html#abs-2006-03654

[21] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[22] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, and J. Wang, “Release strategies and the social impacts of language models,” 08 2019. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1908/1908.09203.pdf

[23] N. Maloyan, B. Nutfullin, and E. Ilyushin, “Dialog-22 ruatd generated text detection,” 06 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2206.08029

[24] X. Liu, P. He, W. Chen, and J. Gao, “Multi-task deep neural networks for natural language understanding,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4487–4496. [Online]. Available: https://aclanthology.org/P19-1441