

# Automated Text Identification: Multilingual Transformer-based Models Approach

German Gritsay<sup>1,\*</sup>, Andrey Grabovoy<sup>1</sup>, Aleksandr Kildyakov<sup>1</sup> and Yury Chekhovich<sup>1</sup>

<sup>1</sup>*Advacheck, Tallinn, Estonia*

## Abstract

This paper describes our solution approach for the AuTextification (Automated Text Identification) competition held as part of the IberLEF 2023 conference. Generated text is an increasing problem nowadays. Due to the spread of large volumes of generated texts across the Internet, people are often confused by this kind of content. In this article, we present a model for machine generated text detection based on different BERT-like encoder models. To achieve better results, we applied a fine-tuning approach of large pre-trained language encoder models XLM-RoBERTa, mDeBERTa and MiniLM-V2. In order to improve the quality of the detectors, we performed extensive preprocessing and expansion of the training data, preserving the structural properties. The method described in the paper helped our team to achieve about 66% for the English binary dataset in the final competition result.

## Keywords

machine-generated text, text classification, transformer-based models, fine-tuning, data preprocessing

## 1. Introduction

The emergence of the Generative Pre-trained Transformer (GPT) language model [1] opens a new round of generated content development. A large zoo of PaLM [2], BLOOM [3], LLaMA [4] and ChatGPT [5] models are available today that handle the task of generating human-like text perfectly. Many of them can be used by anyone to produce text with any content that is difficult to distinguish from human at first glance. However, there is a downside: widespread access to these models often leads to the expansion of fake news [6], plagiarism [7] and misinformation. The malicious potential of generated text has just become a reality. Nevertheless, there are still many patterns in artificial excerpts that can be used to classify the author of a text. Thus, it is crucial to develop a quality detector of machine generated texts.

Today there are several attempts to build artificial text recognition systems. Ippolito in 2020 [8] managed to identify the dependence of the quality of GPT-2 [9] text detection on generative models sampling methods. The quality of detection was also revealed to depend on the length of the input sequence at the classification model input [10]. The most popular approaches [11] for machine generated text detection are those based on linguistic, grammatical and statistical feature generation and using classical machine learning methods (Logistic Regression, Random


---


*IberLEF 2023, September 2023, Jaén, Spain*

\*Corresponding author.

✉ [gritsai@advacheck.com](mailto:gritsai@advacheck.com) (G. Gritsay); [grabovoy@advacheck.com](mailto:grabovoy@advacheck.com) (A. Grabovoy); [kildyakov@advacheck.com](mailto:kildyakov@advacheck.com) (A. Kildyakov); [chekhovich@advacheck.com](mailto:chekhovich@advacheck.com) (Y. Chekhovich)

ORCID [0000-0002-4031-0025](https://orcid.org/0000-0002-4031-0025) (A. Grabovoy); [0000-0002-5204-5484](https://orcid.org/0000-0002-5204-5484) (Y. Chekhovich)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Forest, Gradient Boosting) or using encoders of pre-trained language models as a basis for fine-tuning on the selected domain. In different situations the strength of each approach varies, although classifiers with pre-trained models tend to have a higher generalisation ability and to classify robustly when the domain changes. Such pre-trained language encoders are also powerful in extracting fine-grained semantic information, which is not easily obtained using hand-crafted features and is at the same time often crucial in understanding natural language and further authorship attribution. The approach with fine-tuning pre-trained language encoder is experimented in this paper as part of the AuTextTification competition which aims to boost research on the detection of text generated automatically by text generation models.

In this paper we employ different BERT-based architectures (XLM-RoBERTa [12], mDeBERTa [13] and MiniLM-v2 [14]) to obtain embeddings for each text in the collection and classify it once. We also analysed provided training data and made some preprocessing and extension techniques with them.

## 2. Task

The AuTextTification competition consisted of 2 subtasks.

- Subtask 1 - participants need to determine whether the text has been automatically generated or not;
- Subtask 2 - participants are provided with a text and need to identify which model has generated it;

According to the organisers [15], the number of parameters in the generative models ranged from 2B to 175B, so participants' systems should be versatile enough to recognise a wide range of text generation models and writing styles. The subtasks described were given for two languages - English and Spanish.

In this paper we considered the approach for solving the subtask 1 on samples with binary classification in English language. There is a given dataset  $\mathcal{D} = (x_i, y_i)$ :

$$x_i = \{x_i^1, \dots, x_i^m\}, \quad x_i^j \in \mathcal{W}, \quad j \in \{1, \dots, m\}, \quad y_i \in \{0, 1\},$$

where  $\mathcal{W}$  corresponds to all possible strings in the given language. The label  $y_i = 1$  corresponds to text that is likely machine-generated,  $y_i = 0$  corresponds to human excerpt.

Formally, the task is to find the binary classifier that minimizes an empirical risk on the dataset  $\mathcal{D}$ :

$$f = \operatorname{argmin}_{f \in \mathfrak{F}} \sum_{x_i, y_i \in \mathcal{D}} [f(x_i) \neq y_i],$$

where  $\mathfrak{F}$  is a set of all considered classification models.

## 3. Dataset

The dataset proposed by the organisers for the training stage consisted of 33,845 examples with the labels 'human' and 'generated'. According to the authors, the texts are based on five

Sample text	Label
@CathrineSchack hahaha noo i was waiting for this to come out to see what you would do :D Kerilynn	<i>generated</i>
all these random people are at my house drinking out of my kegg of beer sorry beau cant bring it, its nre	<i>human</i>

Table 1: Example of raw rows from training data.

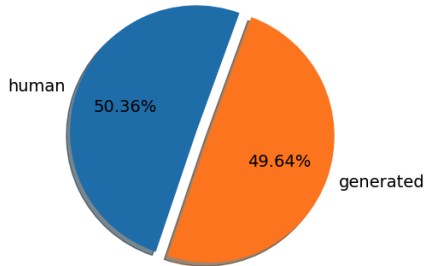


Figure 1: Distribution of classes in train data.

Data Part	Mean Length	Median Length
human	297.13	345.0
generated	313.49	377.0
all	305.25	361.0

Table 2: Length statistics in train data.

different domains, including legal documents, practical articles and social media. In this way, it will be possible to identify the robustness of the developed algorithm to the style of writing: from more structured and formal to less structured and informal. Examples of generated and human texts are provided in the Table 1.

Note that we split the provided train data into two parts (30,000 and 3,845 samples) in order to use the second part as test data for our approaches. The second part was class balanced and all studies and experiments in the paper were performed on the first part.

Before starting to build the classification algorithm, we decided to analyse the data provided. The texts were balanced in terms of their class, as illustrated in Figure 1. In terms of length statistics, the samples turned out to be relatively short. Often the length of the text for detection makes a difference and affects the quality of the detection [10]. The length values by class are shown in the Table 2.

### 3.1. Data Cleaning

The authors of the competition imposed a restriction on the use of the data: only submitted samples could be used, and no external sources were allowed. It was decided to clean up the texts and increase their number.

We did not want to change the style of the texts too much, as there are studies that show that most of the generated texts have common features that are unique to them, for example in the frequency of using certain parts of speech [16], the difference between inverted commas and white space [17]. Therefore, in our cleanup phase we have included the removal of user mentions via the '@' symbol, as this part of the text is incapable of carrying useful information.

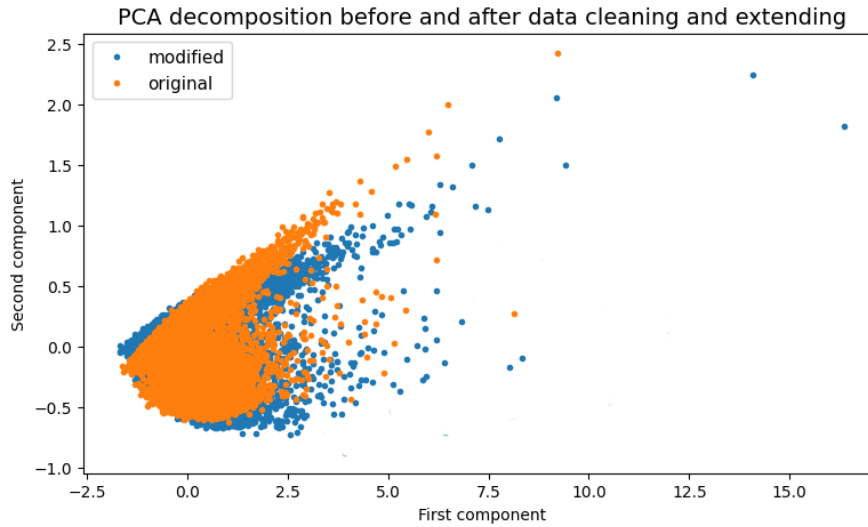


Figure 2: Two component PCA decomposition presented with XLM-RoBERTa embeddings on train samples before and after cleaning and preprocessing.

URLs and HTML tags have also been cleaned up.

### 3.2. Data Instance Preprocessing

As for increasing the number of texts, our idea is based on statistics about the average length of the data provided. When we saw that the classification model would be more likely to see short samples, we decided to split some of the long texts into medium-length excerpts. We selected texts with a minimum length of 450 characters and divided them into sentences with the condition that the new sequence should be at least 50 characters long. This kind of preprocessing extended the data to 42,484 samples with real data 30,000 count.

We looked at the Principal Component Analysis (PCA) decomposition of the two main components of the texts embeddings received by XLM-RoBERTa encoder as one of the most popular baseline encoding methods. The decomposition is shown in Figure 2, it can be seen that the distribution has retained its structural properties after data expansion and cleaning, allowing machine-generated patterns, if they exist, to be retained for a future model.

## 4. Experiments

### 4.1. Methods Description

Based on a review conducted on the task, we were able to determine the most relevant models for solving the problem in English. In recent years, transformer models have been the most frequently used in natural language processing tasks. Their efficiency has been proved by various researches, so in this paper the experiments were carried out with this group of models. Transfer learning is commonly utilized in the implementation of such models. This is an

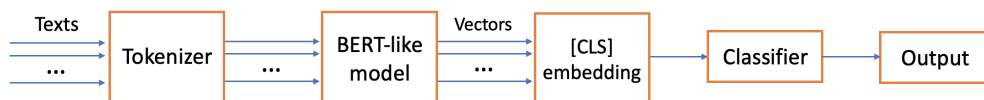


Figure 3: Full text classification pipeline. The input is raw text, the output is a class label.

approach in deep learning, where network knowledge from one task is transferred to solve another, related task, thus making it narrowly focused. Initially, such networks are trained on large data collections, after which they are fine-tuned to the specific task, making these models quite flexible. The fragment embeddings that can be obtained by those models are generally able to have an excellent contextual understanding and may not only be multidomain, but may also be multilingual. The following models have been considered to solve the task set by the organisers:

1. RoBERTa (Robustly Optimized BERT Pretraining Approach) - has the same architecture as BERT [18], but uses a byte-level BPE as a tokenizer (same as GPT-2) and uses a different pretraining scheme and optimization features. For this task we used XLM-RoBERTa which is multilingual version of RoBERTa and was pre-trained on 2.5TB of filtered Common Crawl data containing 100 languages. In our own earlier research, we found that the performance of the multilingual version was superior to that of the monolingual version on most tasks. This can be explained by the fact that a multilingual task setting for training a large model helps to improve the quality of the embeddings, thus helping them to achieve greater generalizability.
2. DeBERTa (Decoding-enhanced BERT with Disentangled Attention) - improves the BERT and RoBERTa models using two techniques: a disentangled attention mechanism, where each word is represented by two vectors encoding its content and position respectively, and an enhanced mask decoder, which replaces the output softmax layer to predict the masked tokens for model pretraining. For this task we also used its multilingual version mDeBERTa and it was trained using the 2.5T Common Crawl 100 data too.
3. MiniLM-L12-v2 (Multi-Head Self-Attention Relation Distillation for Compressing Pre-trained Transformers) - generalizes deep self-attention distillation in MiniLM [19] by employing multi-head self-attention relations to train the student. In general, it is distilled model from large-size teachers (BERT, RoBERTa, XLM-RoBERTa-large) that uses relational knowledge. The authors showed that transferring the self-attention knowledge of an upper middle layer achieves better performance for large-size teachers. This model is initially multilingual, so for this task we used its checkpoint from the Sentence Transformers hub - miniLM-L12-v2.

All of the above models have been used as encoder for samples. For classification, we have redefined the head that will handle with the [CLS] embeddings at the encoder output. It consisted of 3 fully-connected layers, a GELU [20] activation function and a dropout technique. The complete pipeline is demonstrated in Figure 3.

## 4.2. Comparison

For each of the models described above, it was decided to run an experiment with default settings and on train data without preprocessing (labelled "processed"). As default settings, we chose the loss function - cross-entropy, the batch size was set 16, the AdamW [21] optimiser and the linear LR scheduler were also selected. Fine-tuning technique was performed for 5 epochs: 1 epoch only the classifier with frozen encoder weights is trained, 3 epochs the full model is trained and 1 epoch again only the classifier with frozen encoder weights. This learning stages helps to shift the distribution of the encoder weights in the right direction.

After several stages of testing different strategies, we came up with other settings for the fine-tuning and tested the chosen models on them. The batch size remained the same, the label smoothing regularization technique with 0.1 value was added to the loss function, chose another scheduler - MultiStepLR with milestones = [30,90,130] and gamma = 0.3, and also extended and cleaned train collection (labelled "cleaned") was selected for experiment.

Model	Datasets	
	<i>Original Data</i>	<i>Processed Data</i>
XLM-RoBERTa	86.86	88.75
mDeBERTa V3	90.42	<b>93.07</b>
MiniLM-L12-v2	89.63	90.49

Table 3: Fine-tuning on two different datasets with various settings transformer-based detectors

The results obtained in the experiment on our test data are presented in Table 3. The metric chosen was f1-score, the same as in the competition. The data expansion improved the ability of the models to learn the data representation better and increased the generalizability. The multilingual version of the DeBERTa model performed best on cleaned and expanded data with selected hyperparameters. The model with these settings was submitted by our team as a solution to the AuTextification competition, which placed us in the top-25 at the end of the contest.

## 5. Conclusion

The paper describes an approach to the problem of machine generated text detection. We propose a model to detect artificial texts using mDeBERTa encoder to obtain embeddings of single excerpts and for the further classification. We also provide an analysis of different vectorization models based on the BERT architecture. We preprocessed the original training dataset with cleaning and extension to improve the quality of the recognition. The PCA decomposition of the two datasets showed that the distributions retained structural features. The resulting model shows an f1-score in AuTextification competition final results of about 66% for the English binary dataset.

## References

- [1] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).
- [2] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, arXiv preprint arXiv:2204.02311 (2022).
- [3] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al., Bloom: A 176b-parameter open-access multilingual language model, arXiv preprint arXiv:2211.05100 (2022).
- [4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.
- [5] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.
- [6] O. Bakhteev, A. Ogaltsov, 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, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
- [7] O. Bakhteev, Y. Chekhovich, G. Gorbachev, T. Gorlenko, A. Grabovoy, K. Grashchenkov, A. Kildyakov, A. Khazov, V. Komarnitsky, A. e. a. Nikitov, Cross-language plagiarism detection: a case study of european universities academic works, in: Academic Integrity: Broadening Practices, Technologies, and the Role of Students, Springer, 2022, pp. 143–161.
- [8] D. Ippolito, D. Duckworth, C. Callison-Burch, D. Eck, Automatic detection of generated text is easiest when humans are fooled, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 1808–1822. URL: <https://doi.org/10.18653/v1/2020.acl-main.164>. doi:10.18653/v1/2020.acl-main.164.
- [9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019.
- [10] G. Gritsay, A. Grabovoy, Y. Chekhovich, Automatic detection of machine generated texts: Need more tokens, in: 2022 Ivannikov Memorial Workshop (IVMEM), 2022, pp. 20–26. doi:10.1109/IVMEM57067.2022.9983964.
- [11] G. Jawahar, M. Abdul-Mageed, L. Lakshmanan, V.S., Automatic detection of machine generated text: A critical survey, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 2296–2309. URL: <https://aclanthology.org/2020.coling-main.208>. doi:10.18653/v1/2020.coling-main.208.
- [12] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020. arXiv:1911.02116.
- [13] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023. arXiv:2111.09543.
- [14] W. Wang, H. Bao, S. Huang, L. Dong, F. Wei, Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers, 2021. arXiv:2012.15828.

- [15] A. M. Sarvazyan, J. Á. González, M. Franco Salvador, F. Rangel, B. Chulvi, P. Rosso, Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, in: *Procesamiento del Lenguaje Natural*, Jaén, Spain, 2023.
- [16] S. Mitrović, D. Andreoletti, O. Ayoub, Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text, 2023. [arXiv:2301.13852](#).
- [17] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, B. Raj, Gpt-sentinel: Distinguishing human and chatgpt generated content, 2023. [arXiv:2305.07969](#).
- [18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. [arXiv:1810.04805](#).
- [19] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. [arXiv:2002.10957](#).
- [20] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), 2020. [arXiv:1606.08415](#).
- [21] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. [arXiv:1711.05101](#).