

# KInIT at SemEval-2024 Task 8: Fine-tuned LLMs for Multilingual Machine-Generated Text Detection

Michal Spiegel<sup>1,2</sup> and Dominik Macko<sup>1</sup>

<sup>1</sup> Kempelen Institute of Intelligent Technologies

<sup>2</sup> Faculty of Informatics, Masaryk University

michal.spiegel@intern.kinit.sk, dominik.macko@kinit.sk

## Abstract

SemEval-2024 Task 8 is focused on multigenerator, multidomain, and multilingual black-box machine-generated text detection. Such a detection is important for preventing a potential misuse of large language models (LLMs), the newest of which are very capable in generating multilingual human-like texts. We have coped with this task in multiple ways, utilizing language identification and parameter-efficient fine-tuning of smaller LLMs for text classification. We have further used the per-language classification-threshold calibration to uniquely combine fine-tuned models predictions with statistical detection metrics to improve generalization of the system detection performance. Our submitted method achieved competitive results, ranking at the fourth place, just under 1 percentage point behind the winner.

## 1 Introduction

Recent large language models (LLMs) are able to generate high-quality texts that are not easily detectable by human readers. A problem arises when such generated texts are misused for academic exams (Achiam et al., 2023), plagiarism (Wahle et al., 2022), disinformation spreading (Vykopal et al., 2023), etc. Therefore, it is crucial to develop automated means to detect machine-generated texts.

SemEval-2024 Task 8 (Wang et al., 2024) consists of three subtasks: A) binary human-written vs. machine-generated text classification, B) multi-way machine-generated text classification, and C) human-machine mixed text detection. In our work, we have focused on subtask A, especially its multilingual track. It covered 8 known languages for training (Arabic, Bulgarian, Chinese, English, German, Indonesian, Russian, Urdu), multiple domains (e.g., Wikipedia, news, abstracts), and multiple text generators (e.g., GPT-3, ChatGPT, BLOOMZ).

During our participation in the shared task, we have explored various alternatives. Our best sub-

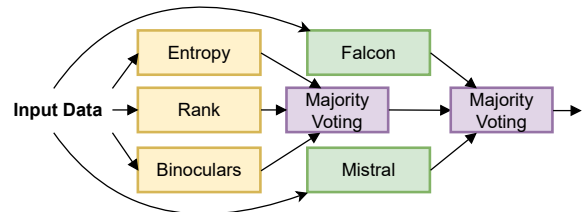


Figure 1: System components overview.

mitted solution (illustrated in Figure 1) combines two fine-tuned LLMs (green-colored) with statistical detection (orange-colored) using a two-step majority voting (purple-colored) based ensemble method. Such a system achieved fourth place in the final leaderboard, with a performance of 95% in accuracy, within 1 percentage point range behind the winning system. We have published the source code for easier replication purposes<sup>1</sup>. We have used the statistical detection methods implemented in the recently published IMGTB framework<sup>2</sup> (Spiegel and Macko, 2023), which will be extended to also support all the fine-tuning options that we have used in this work.

Our key contributions include:

- We have **proposed a unique way of combining the statistical and fine-tuned detection methods** using a two-way majority voting and a per-language threshold calibration.
- We have **proposed and compared three ensemble system alternatives** to cope with multilingual machine-generated text detection (additional two in the post-deadline study).
- We have **experienced a remarkably good performance of fine-tuned LLMs** of 7B parameters in this task.

<sup>1</sup><https://github.com/kinit-sk/semEval-2024-task-8-machine-text-detection>

<sup>2</sup><https://github.com/kinit-sk/IMGTB>

- We have proposed **the best-performing single-model system** called rMistral (Mistral-7B fine-tuned in a robust way – using both the train and dev sets and obfuscating 20% of the train data), achieving **0.97 AUC ROC** on the test data.

An interesting observation regarding our rMistral system is that although our per-language threshold calibration method would not bring the best accuracy on the test set (0.93), the threshold fixed to 1.0 (only predictions with a probability of 1, i.e. 100% confident, are marked as machine-generated) would win the competition (accuracy of 0.97). Nevertheless, we have noticed such a threshold performance only after the deadline and we considered the model being over-fitted (we would not submitted the results) which turned-out to be false.

## 2 Background and Methodology

For the machine-generated text detection task, three main groups of methods are nowadays used (Uchendu et al., 2023). The first one is a *stylometric detection*, which uses linguistic features (e.g., n-grams) to differentiate between human and machine writing styles (Fröhling and Zubiaga, 2021; Kumarage et al., 2023). The second group is a *statistical detection*, which uses statistical distribution based on a pre-trained language model (e.g., GPT2) to calculate various metrics (e.g., entropy) that can be used even without training (i.e., better generalization) to differentiate machine and human written texts (Mitchell et al., 2023; Hans et al., 2024). The last group is a *fine-tuned detection*, which further trains an already pre-trained language model for the detection task (Uchendu et al., 2020; Macko et al., 2023).

In the SemEval2024 Task 8 (Wang et al., 2024), we have focused on the multilingual track of Subtask A, which aimed at a binary classification to differentiate between human-written and machine-generated texts. The provided dataset (not allowing additional training data) contained the predefined splits of train, dev, and test sets. The train and dev sets officially contained 8 languages (3 languages in the dev set only), while unknown number of languages is contained in the test set.

Due to a multilingual nature of the data and our previous experience in multilingual machine-generated text detection (Macko et al., 2023), we wanted to try-out something new in this shared task. Our initial idea was to experiment with a

per-language “mixture-of-experts”, which would consist of multiple models, fine-tuned in a monolingual way per each official language in the train and dev sets. Since it was expected that surprise languages will be present in the test set, we would have used an additional multilingually fine-tuned model for other languages. However, we have started the experiments only few weeks before the deadline, which gave us little time to cope with the problems such as over-fitting and hyper-parameter optimisation (shown as severe towards the deadline).

Therefore, while training these per-language models, we also started to fine-tune the Falcon-7B model (Almazrouei et al., 2023) for the machine-generated text detection task, inspired by the winning system (Gagiano and Tian, 2023) of the recent ALTA 2023 shared task (although English monolingual). Since Falcon-7B is pre-trained on two languages only (English and French), we did not want to use it in a standalone way due to uncertain cross-lingual capability. Therefore, we have similarly fine-tuned the Mistral-7B model (Jiang et al., 2023), which is similarly sized generative model outperforming even some 13B parameters models in common benchmarks. We have not previously experimented with such a “big hammer” for the task; therefore, it was an interesting new experience for us. We have further combined these LLMs with statistical detectors to ensure better generalization of the system, which is described in the following sections.

## 3 System Overview

Our best system (see Figure 1) combines the predictions of two fine-tuned LLMs (Falcon-7B and Mistral-7B) with the selected statistical metrics (Entropy, Rank, Binoculars) by using a **two-step majority voting**. Firstly, a single majority-voted prediction results out of the three statistical metrics. Then, the final majority-voted prediction is a combination of the previous one with the Falcon and Mistral predictions.

Each prediction uses a separate classification-decision threshold, which is applied on prediction probabilities and statistical metrics. These **thresholds are calibrated in a per-language way**, meaning that separate thresholds are used for each language officially present in the train and dev sets, plus an additional threshold for unknown languages (i.e., not officially present in the train and dev sets). The thresholds are calibrated based on the machine-

System	Description
*LLM2S3	The system described in this paper. It is an ensemble using two-step majority voting for predictions, consisting of 2 LLMs (Falcon-7B and Mistral-7B) fine-tuned using the train set only, 3 zero-shot statistical methods (Entropy, Rank, Binoculars) using Falcon-7B and Falcon-7B-Instruct for calculation of the metrics, utilizing language identification and per-language threshold calibration.
PLMoE	Our initial idea representing a per-language mixture of experts. It uses Electra-Large-Discriminator for English and XML-RoBERTa-Large for each of other languages officially present in the train and dev sets. Models for languages present in the dev set only are trained using the dev set. For unknown languages the Mistral-7B fine-tuned using the whole train set is used.
rLLM2S3	The same ensemble system as LLM2S3; however, the LLMs are fine-tuned using both the train and dev sets. Also, to increase the robustness of the system, we have obfuscated 20% of the train samples during fine-tuning, by using HomoglyphAttack and inserting zero-width-joiner character, inspired by our recent work (Macko et al., 2024).
rLLM2B-ES	The post-deadline ensemble system similar to rLLM2S3; however, the Llama-2-7B is used instead of Falcon-7B and Binoculars is used solely in the statistical part (instead of a combination of 3 methods). Moreover, the fine-tuning process used the early stopping mechanism to alleviate the over-fitting.
LLM2B1	The post-deadline ensemble system using the original LLM2S3 fine-tuned Falcon-7B and Mistral-7B models; however, classification thresholds are not calibrated, but only predictions with a probability of 1 (i.e., 100% confident) are marked as machine-generated. Such predictions are combined with Binoculars zero-shot prediction using the per-language threshold calibration.

Table 1: Description of system alternatives. The main system described in this paper is denoted by \*. The last two alternatives were evaluated post-deadline.

class prediction probabilities and statistical metrics for samples in the train and dev sets combined. The calibration maximized the difference between true positive rate (TPR) and false positive rate (FPR) based on the ROC (receiver operating characteristic) curve. The texts with probabilities (or statistical metrics) outreaching the thresholds are considered machine-generated, otherwise they are considered human-written. The thresholds are saved and used for prediction of test samples.

Due to unknown languages in the test set and using the per-language threshold calibration, we have utilized the FastText<sup>3</sup> **language identification**. Since it is not fully accurate, we have used such language information only if the prediction probability was greater than 0.5, otherwise the language was handled as unknown.

As mentioned, the system includes **two fine-tuned LLMs**, namely Falcon-7B and Mistral-7B. For the fine-tuning process, we have used a parameter efficient fine-tuning (PEFT) technique called **QLoRA** (Dettmers et al., 2023) to minimize the computational costs of our system training.

To enhance the system performance generalization, we have integrated a **statistical part** of the system, which is based on the three statistical metrics, namely Entropy (Lavergne et al., 2008), Rank (Gehrmann et al., 2019), and recently proposed

Binoculars (Hans et al., 2024). The statistical metrics are calculated using the Falcon-7B as a base model. Since Binoculars requires two models, it also uses Falcon-7B-Instruct (as a performer model).

Besides the described best submitted system, we have tried multiple system alternatives, which are briefly described in Table 1. In addition to those ensembles, we have evaluated single detectors, namely Falcon, Mistral, S5 (a combination of 5 statistical metrics – likelihood, entropy, rank, log-rank, and llm-deviation), and Binoculars. After the deadline, we have also finished fine-tuning of Llama-2-7B and retrained the detectors using the early stopping (patience of 5) to prevent over-fitting. Also, when knowing the gold labels of the test set, we have evaluated various combinations of the trained detectors to see whether we have done the right decision for the submission.

## 4 Experimental Setup

For the experimental purpose, we have used the defaults splits of the provided dataset, namely the train and dev sets in the pre-deadline experiments, and the gold labels of the test set for the post-deadline evaluation of the pre-deadline system alternatives. The main system described in this paper uses only the train set in the training process; however, uses both the train and dev sets for the classification threshold calibration. Some of the

<sup>3</sup><https://pypi.org/project/fasttext-langdetect/>

system alternatives used both the train and dev sets in the training process, as described in Table 1.

As the key evaluation metric in the shared task is **accuracy**, we have also used this metric for the preliminary system evaluation and selection of the alternative for submission. Since classification task is sensitive to the used classification threshold, we have also used **AUC ROC** (area under curve of the receiver operating characteristic) as a threshold independent metric, providing better information about the classification capability.

For the fine-tuning process, we have used the official baseline script<sup>4</sup>, modified to export machine-class prediction probabilities in addition to the predictions. Since, it was not clear which version of the XLM-RoBERTa model was marked as a baseline in the multilingual track (with the known accuracy of 0.72), we have trained both the base (*XLM-R-B*) and large (*XLM-R-L*) versions. In addition, we have also included mDeBERTa-v3-base (*mDeBERTa*) model in our baselines, since it performed the best in our previous work (Macko et al., 2023).

To perform per-language models fine-tuning, we have used the source field of the train and dev data to select data only for the specific language. Other parameters of the fine-tuning process remained unchanged. The FastText language identification is used for a prediction, which uses the machine-class probability of the corresponding language-specific model.

The used QLoRA PEFT fine-tuning process used the binary cross entropy with logits for loss calculations and 4-bit quantization using BitsAndBytes<sup>5</sup>. The LoRA configuration<sup>6</sup> used an *alpha* of 16, a *dropout* of 0.1, *r* of 64, and the *task type* of sequence classification. Unlike the baseline fine-tuning, this version used half-precision training, gradient accumulation of 4 steps, and evaluation each 1,000 steps. Other parameters were the same.

Due to time constraints, we have not done any hyper-parameter optimization; thus, further improvements of the system are very likely possible.

## 5 Results

The experimental results are provided in Table 2. It must be noted that the results in the bottom part of

<sup>4</sup>[https://github.com/mbzuai-nlp/SemEval2024-task8/blob/main/subtaskA/baseline/transformer\\_baseline.py](https://github.com/mbzuai-nlp/SemEval2024-task8/blob/main/subtaskA/baseline/transformer_baseline.py)

<sup>5</sup><https://pypi.org/project/bitsandbytes>

<sup>6</sup><https://pypi.org/project/peft>

the table are not part of the competition, since those experiments were performed after the submission deadline of the shared task. Also, the performance results using the test set were not known before the deadline; gold labels have been released only afterwards. Therefore, the design decisions could be made purely using the dev set.

Due to high accuracy and high AUC ROC metrics using the dev set, we considered *rFalcon* and *rMistral* over-fitted; therefore, we decided not to submit our *rLLM2S3* system. This turned-out to be a mistake, since it performed slightly better than the submitted *LLM2S3* on the test set. On the other hand, our suspicion of over-fitting *PLMoE* (due to the similar observations) turned-out to be valid, since it performed much worse using the test set. Therefore, it seems that per-language monolingually fine-tuned (i.e., lower amount of samples than multilingually fine-tuned) models require optimization of hyper-parameters to prevent over-fitting and to better generalize to unseen texts.

As an ablation study, we also provide the results for individual components of our system alternatives. As the results show, the ensembling into more complex systems of *LLM2S3* and *rLLM2S3* helped generalization of the classification performance. Individual methods would not outperform the submitted system.

### 5.1 Post-Deadline Study

In the post-deadline experiments (already knowing the gold labels of the test set for evaluation), we have finished Llama-2-7B model fine-tuning and retraining all three robust-version LLMs using the early stopping (to minimize the over-fitting). The results revealed that the *rLlama-2* model does not suffer by over-fitting as much. Based on the test set evaluation and by examining various combinations, the retrained *rLlama-2-ES* and *rMistral-ES* seemed like good candidates to combine with Binoculars (*rLLM2B-ES*), outperforming the winning system in the competition.

Early stopping helped a lot in boosting performance generalization (i.e., reducing over-fitting) of our per-language mixture-of-experts ensemble system (*PLMoE-ES*), achieving one of the highest AUC ROC using the test set. Nevertheless, in the accuracy as an official metric, it would not outperform the other system alternatives.

In addition, we have noticed that optimal thresholds for fine-tuned LLMs are often set to 1.0 by using purely the dev set samples machine-class

	System	Accuracy		AUC ROC	
		Dev	Test	Dev	Test
<b>Baselines</b>	XLM-R-B	0.7158	0.7935	0.8262	0.9040
	XLM-R-L	0.7275	0.8841	0.8187	0.9063
	mDeBERTa	0.6968	0.8943	0.7952	0.9832
<b>System Alternatives</b>	*LLM2S3●	0.9035	0.9501	N/A	N/A
	PLMoE●	0.9878	0.5819	0.9943	0.6268
	rLLM2S3●	0.9965	0.9560	N/A	N/A
<b>Ablation Study</b>	Falcon	0.8043	0.9102	0.8775	0.9492
	Mistral	0.8560	0.9027	0.9138	0.9579
	rFalcon	0.9905	0.8843	0.9991	0.9395
	rMistral	<b>0.9980</b>	0.9268	<b>0.9997</b>	0.9713
	S3●	0.7248	0.8328	N/A	N/A
	S5●	0.5880	0.4763	N/A	N/A
	Binoculars	0.5430	0.7979	0.6304	0.8777
	Binoculars●	0.6240	0.8434	0.6304	0.8777
<b>Post-Deadline Study</b>	PLMoE-ES●	0.9885	0.8417	0.9947	0.9635
	Llama-2	0.7335	0.7587	0.9342	0.7400
	rLlama-2	0.8903	0.8907	0.8416	0.9400
	rLlama-2-ES	0.9838	0.8805	0.9960	0.9108
	rFalcon-ES	0.9410	0.8672	0.9872	0.9503
	rMistral-ES	0.9863	0.9412	0.9984	<b>0.9834</b>
	rLLM2B-ES●	0.9915	0.9700	N/A	N/A
	LLM2B1	0.8668	<b>0.9708</b>	N/A	N/A
rMistral1	0.9975	0.9675	<b>0.9997</b>	0.9713	

Table 2: Detection performance evaluated using the dev (pre-deadline) and test (post-deadline) splits separately. The main system described in this paper is denoted by \*, the systems using the per-language threshold calibration are denoted by ●, systems using fixed threshold of 1.0 are denoted by “1”. “-ES” denotes using of early-stopping mechanism to prevent over-fitting. “N/A” denotes not available values due to prediction-based majority voting (i.e., no probabilities to calculate AUC ROC). The gray color denotes unrepresentative performance values due to training on the dev set.

probabilities. Therefore, we have fixed the thresholds to 1.0 for the *LLM2B1* system (containing only models we have trained before the deadline), meaning that the machine-class predictions of the LLMs are used only when having 100% confidence (otherwise considered human-written). Such predictions, when combined with Binoculars, achieved even higher performance using the test set data (0.9708). Thus, we had such a system trained before the deadline; however, we have not noticed such a threshold bringing the best performance in time. Moreover, when looking at the accuracy for the dev set, we do not see why we would select such a system for the submission. It can be just a coincidence that it performs so well using the test set data. Further experiments are required to examine this phenomenon using independent out-of-distribution data.

Also, even when our *rLLM2B-ES* system alternative or the *rMistral1* single-model system would win the competition, we are now not sure that we would be confident enough (about not being over-fitted) to submit it as the final system without eval-

uation on the external dataset. Thus, we have submitted best what we could at the time.

## 5.2 Per-language Analysis

For a deeper insight of the proposed system (*LLM2S3*) performance, we have performed an analysis per each language identified in the test set. The results are provided in Figure 2. Interesting is that it achieved the highest accuracy for the Italian surprise language (*it*). Lower accuracy is evident for German and Arabic languages, although present in the dev set. It must be noted that this version of the system was not trained using the dev set, only the classification threshold calibration used such data. Therefore, the robust versions of system alternatives are expected to provide higher performance especially in those languages.

## 6 Conclusion

To cope with the problem of multilingual, multidomain, and multigenerator machine-generated text detection, we have proposed an ensemble sys-

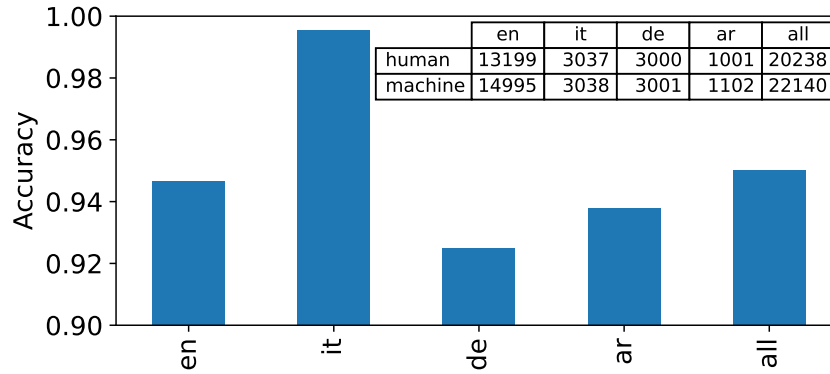


Figure 2: Per-language test-set performance (*it* is a surprise language, *de* and *ar* are in the dev set only). Axis scale for Accuracy is shown from 0.9 to 1.0. The per-class samples counts are provided in the top-right table.

tem using 2 LLMs (Falcon-7B and Mistral-7B) fine-tuned for the binary sequence classification task. We have further combined the predictions with the statistical metrics of Entropy, Rank, and Binoculars using a two-stage majority voting. The classification thresholds in our system have been calibrated in a per-language manner, for which we have utilized the FastText language identification. A combination of fine-tuned LLMs and statistical detection seems to be the right way to cope with generalization of the detection performance. Out of the evaluated single-model systems, Mistral-7B is the best candidate for fine-tuning, which by itself can bring a remarkable classification performance. Further improvements of the system could be easily achievable by hyper-parameters optimization, which we have not done in the submitted system due to lack of time.

## Acknowledgements

This work was partially supported by the projects funded by the European Union under the Horizon Europe: *AI-CODE*, a project funded by the European Union under the Horizon Europe, GA No. 101135437, and *VIGILANT*, GA No. 101073921. Part of the research results was obtained using the computational resources procured in the national project *National competence centre for high performance computing* (project code: 311070AKF2) funded by European Regional Development Fund, EU Structural Funds Informatization of Society, Operational Program Integrated Infrastructure.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,

Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Hestlow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: An open large language model with state-of-the-art performance.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. *QLoRA: Efficient finetuning of quantized llms*.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Computer Science*, 7:e443.

Rinaldo Gagiano and Lin Tian. 2023. A prompt in the right direction: Prompt based classification of machine-generated text detection. In *Proceedings of ALTA*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. *Spotting LLMs with binoculars: Zero-shot detection of machine-generated text*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. *Mistral 7b*.

- Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.
- Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377, PAN'08*, page 27–31, Aachen, DEU. CEUR-WS.org.
- Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. [MULTITuDE: Large-scale multilingual machine-generated text detection benchmark](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.
- Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Samuel Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. [Authorship obfuscation in multilingual machine-generated text detection](#). *arXiv preprint arXiv:2401.07867*.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. [DetectGPT: Zero-shot machine-generated text detection using probability curvature](#). *arXiv preprint arXiv:2301.11305*.
- Michal Spiegel and Dominik Macko. 2023. [IMGTB: A framework for machine-generated text detection benchmarking](#). *arXiv preprint arXiv:2311.12574*.
- Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Attribution and obfuscation of neural text authorship: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 25(1):1–18.
- Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. [Authorship attribution for neural text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.
- Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2023. [Disinformation capabilities of large language models](#). *arXiv preprint arXiv:2311.08838*.
- Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. [How large language models are transforming machine-paraphrase plagiarism](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 Task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval 2024*, Mexico, Mexico.

## A Computational Resources

For experiments regarding model fine-tuning and inference processes, we have used  $1 \times$  NVIDIA GeForce RTX 3090 24GB GPU and  $1 \times$  A100 40GB GPU, cumulatively consuming around 10,000 GPU-core hours. For combining the results and analysis, we have used Jupyter Lab running on 4 CPU cores, without the GPU acceleration.