Authorship Obfuscation in Multilingual Machine-Generated Text Detection

Dominik Macko[1], Robert Moro[1], Adaku Uchendu[2], Ivan Srba[1], Jason Samuel Lucas[3],
Michiharu Yamashita[3], Nafis Irtiza Tripto[3], Dongwon Lee[3],
Jakub Simko[1], Maria Bielikova[1]

[1] Kempelen Institute of Intelligent Technologies, Slovakia
{name.surname}@kinit.sk
[2] MIT Lincoln Laboratory, USA
adaku.uchendu@ll.mit.edu
[3] The Pennsylvania State University, PA, USA
{jsl5710, michiharu, nit5154, dongwon}@psu.edu

Abstract

High-quality text generation capability of recent Large Language Models (LLMs) causes concerns about their misuse (e.g., in massive generation/spread of disinformation). Machine-generated text (MGT) detection is important to cope with such threats. However, it is susceptible to authorship obfuscation (AO) methods, such as paraphrasing, which can cause MGTs to evade detection. So far, this was evaluated only in monolingual settings. Thus, the susceptibility of recently proposed multilingual detectors is still unknown. We fill this gap by comprehensively benchmarking the performance of 10 well-known AO methods, attacking 37 MGT detection methods against MGTs in 11 languages (i.e., $10 \times 37 \times 11 = 4,070$ combinations). We also evaluate the effect of data augmentation on adversarial robustness using obfuscated texts. The results indicate that all tested AO methods can cause evasion of automated detection in all tested languages, where homoglyph attacks are especially successful. However, some of the AO methods severely damaged the text, making it no longer readable or easily recognizable by humans (e.g., changed language, weird characters).

1 Introduction

Recent advances in Language Modeling have birthed Large Language Models (LLMs) which exhibit significant improvements, including the ability to generate texts easily misconstrued as human-written (Zellers et al., 2019). In addition, LLMs have been found to exhibit emergent abilities (Wei et al., 2022), such as performing several NLP tasks at or over the human level, e.g., an LLM passing academic exams designed for humans in top 10% of test takers (Achiam et al., 2023). While this is a celebratory feat for the NLP community, LLMs currently have several shortcomings like the generation of hate and biased speech (Deshpande al.,
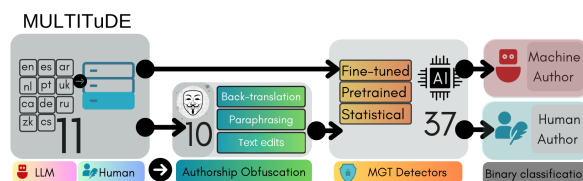

Figure 1: Benchmarking authorship obfuscation techniques for machine-generated text detection.

2023; Venkit et al., 2023), the generation of memorized and plagiarized content (Nasr et al., 2023; Carlini et al., 2021; Lee et al., 2023), and the generation of mis/disinformation (Lucas et al., 2023; Chen and Shu, 2023a; Vykopal et al., 2024; Zellers et al., 2019). Furthermore, we find that some of these LLMs like ChatGPT, Flan-T5 (Chung et al., 2022), Falcon (Almazrouei et al., 2023), LLaMA (Touvron et al., 2023), etc. can have their alignment tuning bypassed by jailbreaking prompting techniques (Lucas et al., 2023; Chen and Shu, 2023b). Therefore, due to the great potential to be misused, it is imperative that at the minimum, we can accurately distinguish **Machine-Generated Texts** (**MGT**) from human-written ones.

The MGT detection task has been studied very well for the binary case (*Turing Test*), showing that accurate detection of machine-generated texts is possible (Uchendu et al., 2023; Crothers et al., 2023; Wu et al., 2023). On the other hand, in this work, we investigate the less comprehensively studied problem for MGT detection–**Authorship Obfuscation** (**AO**), which refers to the process of masking an author's writing style/signature, while preserving semantics. In the MGT detection task, AO represents an attack vector for MGT evading detection, being successful when true positives (i.e., detected MGT) are changed to false negatives (i.e., undetected MGT).

We evaluate the robustness of MGT detectors to adversarial perturbations by implementing AO techniques (see Figure 1 for an illustration). Since

the task has been solely studied for monolingual settings, mostly English (Zellers et al., 2019; Mitchell et al., 2023), but also Chinese or Russian (Orzhenovskii, 2022; Pu et al., 2022; Guo et al., 2023a), we focus on the *multilingual* perspective. Our key contributions are:

(1) We provide the first comprehensive **multilingual benchmark of AO methods to evade the MGT detection** by implementing 10 AO methods under 3 categories: *Backtranslation*, *Paraphrasing*, and *Text edits*. We confirmed that eight of the ten used AO methods are usable in multilingual settings. Homoglyph-based attacks reached 70% attack success in some languages.

(2) We provide **a unique dataset of multilingual obfuscated texts (human and machine-generated)**, which resulted into approximately 740k samples (although some of them contains the same text as original, i.e., the obfuscation failed). The dataset is available at Zenodo upon request for research purposes only[1].

(3) We provide the first **evaluation of robustness of multilingual MGT detection methods** against authorship obfuscation. We evaluate 37 multilingual MGT detectors using the MULTI-TuDE (Macko et al., 2023a) multilingual benchmark dataset, containing 11 languages and 8 LLMs. Out of multiple novel findings, we found out, for example, that while basic *ChatGPT paraphrasing* is not an effective AO technique, the *homoglyph attacks* are very effective if not considered during detector training or text preprocessing.

(4) We evaluate the effect of **data augmentation using obfuscated texts on adversarial robustness** of multilingual detectors. We show that in most cases even simple data augmentation can improve the robustness of detectors to AO techniques.

2 Related Work

With the advancements of LLMs, the MGT detection methods also adapt to multilingual settings (Wang et al., 2023; Macko et al., 2023a). However, it is known that MGT detection is susceptible to AO methods (e.g. paraphrasing), which can drop the detection performance even by 90% (Crothers et al., 2022; Krishna et al., 2023; Shi et al., 2023). As previously mentioned, AO methods have been evaluated in monolingual settings only. For some other NLP tasks, adversarial attacks (as a subset of AO) have been evaluated in multilingual set-

tings (Rosenthal et al., 2021; Wang et al., 2021). However, multilingual evaluation of AO in MGT detection task is still missing, thus the severity of a potential threat of detection evasion is unknown.

2.1 MGT Detection

The current most popular MGT detectors are represented by two main groups: fine-tuned and statistical models. *Fine-tuned models* are transformer-based models fine-tuned specifically for the MGT detection task (Uchendu et al., 2021; Munir et al., 2021; Ai et al., 2022; Liyanage et al., 2022). There are also available several models already *pre-trained* for the MGT detection task, such as RoBERTa-base-OpenAI-Detector (Solaiman et al., 2019), Longformer detector (Li et al., 2023), ChatGPT-Detector-RoBERTa-Chinese (Guo et al., 2023b), GROVER detector (Zellers et al., 2019), etc. These can be directly used in a zero-shot manner for detection of MGT. However, they are mostly monolingual. Multilingual models, such as XLM-RoBERTa (Conneau et al., 2020) or mBERT (Devlin et al., 2019), can be fine-tuned for the task on a custom dataset to be used as multilingual detectors (Wang et al., 2023; Macko et al., 2023a).

Statistical models use the statistical distribution of human-written and machine-generated texts to calculate the differences between the two. These detectors are usually based on a pre-trained LLM without fine-tuning, such as GPT-2 (Radford et al., 2019), mGPT (Shliazhko et al., 2022), Falcon (Almazrouei et al., 2023), etc. to calculate a single or multiple metrics. Usually, a separate classifier (e.g., Logistic Regression or Random Forest) is trained to make a prediction based on such metrics. Popular statistical models include LLMDeviation (Wu and Xiang, 2023), Rank (Gehrmann et al., 2019), GLTR Test 2 (Gehrmann et al., 2019), MFD (Wu and Xiang, 2023), DetectGPT (Mitchell et al., 2023), or DetectLLM (Su et al., 2023). There are new promising statistical methods available, such as Fast-DetectGPT (Bao et al., 2024), GPT-who (Venkatraman et al., 2024), DNA-GPT (Yang et al., 2024), or Binoculars (Hans et al., 2024), which are however not covered in this study.

2.2 MGT Obfuscation

To evaluate the adversarial robustness of MGT detectors, researchers have implemented several AO methods, which can be grouped into: *Backtranslation*, *Paraphrasing*, and *Text edits*. *Backtranslation* is the process of changing a text from one

---

[1] https://doi.org/10.5281/zenodo.13846588

language to another and then back to the original (e.g., English → Spanish → English) (Almishari et al., 2014; Keswani et al., 2016). The idea is that the final output, the backtranslated version, will be subtly different from the original and thus, evade accurate detection. Backtranslation has been well studied in the authorship attribution niche field, making it suitable to apply to this task (Altakrori et al., 2022). Next, *Paraphrasing* is similar to the backtranslation method, however, the goal is to paraphrase/re-write the text and keep it in the same language as the original. It is currently the most popular AO technique in this detection task (Sadasivan et al., 2023; Krishna et al., 2023; Lu et al., 2023; Shi et al., 2023; Koike et al., 2024; Tripto et al., 2024). In *Text edits*, the goal is to obfuscate authorship through one or more of the linguistic categories: Lexical, Syntactic, Morphological, or Othographic. Various text-edit attacks are available, such as lexical-based attacks implemented in GPTZzzs[2] or DFTFooler (Pu et al., 2023), a syntactic-based attack implemented in ALISON (Xing et al., 2024), or an orthographic attack implemented in GPTZeroBypasser[3].

3 Methodology

The methodology is illustrated in Figure 1, summarizing key components. We benchmark 10 well-known AO methods attacking 37 MGT detection methods against texts in 11 languages (i.e., $10 \times 37 \times 11 = 4,070$ combinations). All source codes, data, and full results are publicly available to enable full replication of our study[4].

3.1 Authorship Obfuscation Methods

To make the scope of this work feasible, we have used fully automated AO methods (leaving out human-in-the-loop methods) without any heavy modification (out-of-the-box, just changing the base model and/or tokenizer to a multilingual version if available). A single run of these methods have been used without any combinations (i.e., no iterative modification of the texts). We have used 10 existing AO methods in this work (see Table 1, links to and parameters of the AO methods are available in Appendix B), grouped into three main categories: backtranslation, paraphrasing, and text edits. We aimed to use at least two representatives of each group, while avoiding paid services and

[2] https://github.com/Declipsonator/GPTZzzs
[3] https://github.com/o2161405/GPTZero-Bypasser
[4] https://github.com/kinit-sk/mAO

	AO Method	Description
Backtranslation	m2m100 (Fan et al., 2021)	A dedicated multi-language machine translation model.
	nllb-200 (Costa-jussà et al., 2022)	A research-purpose machine translation model.
Paraphrasing	Pegasus-paraphrase	A fine-tuned PEGASUS model for paraphrasing task, working on a sentence level.
	DIPPER (Krishna et al., 2023)	A contextual paragraph-level paraphraser with a controllable diversity, based on English-only T5-xxl.
	ChatGPT	A popular OpenAI chat model. We have used the basic prompt of "Paraphrase the following text in <language> language: <text>".
Text edits	GPTZzzs	A tool using an English dictionary of synonyms to replace a number of words.
	GPTZeroBypasser	A homoglyph attack to replace 9 specific Latin letters for Cyrillic letters and inserting a zero-width joiner pseudorandomly.
	HomoglyphAttack	Our generic version of a homoglyph attack using the whole confusables table to psudorandomly replace letters for their homoglyphs.
	ALISON (Xing et al., 2024)	An adversarial perturbation attack that requires no queries to the target LLM, instead it targets its own classifier trained on most frequent ngrams from the train corpus.
	DFTFooler (Pu et al., 2023)	An adversarial perturbation attack that attacks only machine-labeled samples, also requires no queries to the target model.

Table 1: Authorship obfuscation methods overview.

including at least one representative with naturally multilingual capability.

Backtranslation uses English (as a high resource language) as an intermediary language for non-English texts and Spanish (as a representative from a different language family branch) for English texts. Since we have not found any dedicated multilingual **paraphrasing** method available, we have used ChatGPT instead (as used for this purpose in literature, e.g., Tripto et al., 2024; Cegin et al., 2023). **Text edits** include synonym, homoglyph, as well as specialized adversarial attacks.

3.2 Dataset

As a source of original unobfuscated texts, we use the MULTITuDE benchmark dataset (Macko et al., 2023b) containing about 8k human-written news texts along with 66k texts generated by 8 LLMs (i.e., alpaca-lora, gpt-3.5-turbo, gpt-4, llama, opt, opt-iml-max, text-davinci-003, vicuna) in 11 languages (i.e., Arabic, Catalan, Chinese, Czech, Dutch, English, German, Portuguese, Russian, Spanish, Ukrainian). For each AO method, we duplicate the MULTITuDE data and replace the texts by their obfuscated version (we retry this if some obfuscated texts are the same as the unobfuscated ones; we limit the number of trials to 10). We evaluate similarity between the obfuscated and the original texts using a range of similarity metrics along with a human check of data pseudo-random subset to gauge the quality of the obfuscation and detect potential pitfalls (see Appendix C).

3.3 MGT Detection Methods

The MGT detection methods used in this work are grouped into three categories (based on their nature): fine-tuned, pre-trained, and statistical.

We include **fine-tuned methods** which are based on multilingual language models (monolingual ones were omitted due to having low performance in the original MULTITuDE benchmark). They are trained on various portions of the train split of the MULTITuDE dataset (for monolingual and multilingual fine-tuning using all LLMs data) using the published scripts of the MULTITuDE benchmark[5]. These include the following models: mDeBERTa-v3-base (He et al., 2022), XLM-RoBERTa-large (Conneau et al., 2020), BERT-base-multilingual-cased (Devlin et al., 2019), mGPT (Shliazhko et al., 2022), and OPT-IML-Max-1.3B (Iyer et al., 2022).

As **pre-trained methods**, we include publicly available detection models that were fine-tuned for MGT detection tasks using different datasets (i.e., out-of-distribution data) and use them in a zero-shot manner in this work. These include RoBERTa-Base and RoBERTa-Large OpenAI Detectors (Solaiman et al., 2019), ChatGPT-detector-RoBERTa and ChatGPT-detector-RoBERTa-Chinese (Guo et al., 2023a), Longformer Detector (Li et al., 2023), RoBERTa-base-autextification-Detection[6], and ruRoBERTa-ruatd-binary[7].

As for **statistical methods**, we include *single-metric methods* (Entropy by Lavergne et al., 2008, LogLikelihood by Mitchell et al., 2023, Rank by Gehrmann et al., 2019, LogRank by Mitchell et al., 2023, LLMDeviation by Wu and Xiang, 2023, DetectLLM-LRR by Su et al., 2023), *multi-metric methods* (GLTR Test 2 Features by Gehrmann et al., 2019, MFD by Wu and Xiang, 2023), and *perturbation-based methods* (DetectGPT by Mitchell et al., 2023, DetectLLM-NPR by Su et al., 2023). Due to the long run-time of perturbation-based methods, these have been used only on a portion of the test set. All of the mentioned statistical methods are implemented in the publicly available IMGTB framework[8] (Spiegel and Macko, 2024), which was used in its default configuration (i.e., Logistic Regression classifier for predictions without hyper-parameters optimization). Analogously to the fine-tuned methods, the classifiers of the statistical methods have been trained using the original

---
[5] https://github.com/kinit-sk/mgt-detection-benchmark
[6] www.huggingface.co/arincon/roberta-base-autextification
[7] https://huggingface.co/orzhan/ruroberta-ruatd-binary
[8] https://github.com/kinit-sk/IMGTB

MULTITuDE data only (i.e., unobfuscated).

4 Experiments and Results

Firstly, our experiments focus on finding whether the existing AO methods (so far evaluated primarily for English) are even usable for non-English languages. Secondly, we focus on finding how robust the existing MGT detection methods are in regard to susceptibility to AO. Lastly, we evaluate the effect of using obfuscated texts for data augmentation to increase their adversarial robustness.

To compare MGT detection performance, we use standard metrics, namely **AUC ROC** (area under the curve of receiver operating characteristic) as a classification-threshold independent metric (not affected by a threshold calibration on a domain data) and **Macro avg. F1-score** as a metric balancing between a precision and a recall of the classification commonly used for the MGT detection task using imbalanced datasets. We further provide Macro avg. F1-score values theoretically achievable on the used data with the classification thresholds calibrated based on the ROC curve for optimal (i.e., maximal difference between TPR and FPR, where TPR is true positive rate and FPR is false positive rate), 1% FPR and 5% FPR conditions. We are aware that such a performance is not actually achievable due to calibration on the evaluation data (i.e., data leakage). Therefore, we are not using it to compare performance of the MGT detection methods, but to evaluate effect of AO on the detection (i.e., compare AO methods), while not being biased due to sub-optimal predictions.

We use two primary metrics to evaluate the effect of AO methods. **ASR** (attack success rate) measures the effectiveness of changing true positives to false negatives, considering all AO methods as attacks on detection. **AUC ROC drop** (detection performance decrease) is used for measuring the effect of obfuscated texts on detection performance.

Before diving deeper into the experiments, let us briefly examine the general detection performance on our data (i.e., difficulty of the data for this task). Table 2 shows a comparison of detection performance of all MGT detection methods using the whole test set (i.e., the original MULTITuDE human and machine texts from the test split and the obfuscated machine texts). Obfuscated human texts are not used in the experiments due to uncertain labeling after obfuscation (i.e., whether the texts modified by machine should still be consid-

Rank	MGT Detection Method	Category	AUC ROC (sorted)	Macro avg. F1-score	Macro avg. F1-score (optimal)	Macro avg. F1-score (1% FPR)	Macro avg. F1-score (5% FPR)
1	XLM-RoBERTa-large (all)	F	0.9247	0.5745	0.5119	0.2296	0.4538
2	XLM-RoBERTa-large (ru)	F	0.9231	0.5983	0.4709	0.3298	0.4290
3	mDeBERTa-v3-base (all)	F	0.9076	0.5388	0.4917	0.2826	0.4057
4	mDeBERTa-v3-base (ru)	F	0.8895	0.6434	0.5160	0.1497	0.3405
5	mDeBERTa-v3-base (es)	F	0.8616	0.5185	0.4743	0.1633	0.3573
6	BERT-base-multilingual-cased (all)	F	0.8515	0.5215	0.4479	0.2176	0.3594
7	mGPT (all)	F	0.8511	0.5347	0.4640	0.2525	0.3110
8	BERT-base-multilingual-cased (es)	F	0.8505	0.5306	0.4764	0.0875	0.2648
9	mGPT (ru)	F	0.8427	0.5901	0.4924	0.0110	0.0110
10	XLM-RoBERTa-large (es)	F	0.8346	0.5035	0.4996	0.0110	0.2747
11	mGPT (es)	F	0.8312	0.5074	0.4363	0.0110	0.3217
12	OPT-IML-Max-1.3B (all)	F	0.8261	0.5265	0.4406	0.0110	0.1958
13	OPT-IML-Max-1.3B (es)	F	0.7697	0.5024	0.4905	0.0110	0.0110
14	BERT-base-multilingual-cased (ru)	F	0.7315	0.5072	0.3823	0.0602	0.0602
15	BERT-base-multilingual-cased (en)	F	0.7198	0.4999	0.4361	0.1231	0.2303
16	OPT-IML-Max-1.3B (ru)	F	0.7101	0.5346	0.4063	0.0776	0.1961
17	XLM-RoBERTa-large (en)	F	0.6815	0.5285	0.3734	0.1274	0.2073
18	MFD	S	0.6713	0.4799	0.3564	0.1069	0.2526
19	RoBERTa-large-OpenAI-Detector	P	0.6618	0.2266	0.4972	0.4672	0.4997
20	Entropy	S	0.6191	0.4972	0.2433	0.1562	0.2229
21	mGPT (en)	F	0.6178	0.4936	0.3181	0.0110	0.0110
22	Longformer Detector	P	0.6135	0.4972	0.2531	0.0582	0.1362
23	mDeBERTa-v3-base (en)	F	0.6112	0.4660	0.4166	0.0152	0.0775
24	RoBERTa-base-OpenAI-Detector	P	0.5955	0.1924	0.4972	0.4328	0.4897
25	OPT-IML-Max-1.3B (en)	F	0.5824	0.5452	0.1489	0.0968	0.1489
26	DetectLLM-NPR	S	0.5764	0.4926	0.2844	0.0636	0.1469
27	ChatGPT-Detector-RoBERTa-Chinese	P	0.5585	0.3463	0.4107	0.0110	0.0110
28	GLTR Test 2	S	0.5385	0.4922	0.3001	0.0454	0.1188
29	DetectGPT	S	0.5382	0.4926	0.2700	0.0581	0.1231
30	ChatGPT-Detector-RoBERTa	P	0.5311	0.1036	0.4972	0.0110	0.4551
31	DetectLLM-LRR	S	0.5250	0.4966	0.2587	0.0573	0.1494
32	RoBERTa-base-autextification-Detection	P	0.4946	0.4883	0.0727	0.0727	0.0727
33	LogRank	S	0.4669	0.4965	0.2504	0.0344	0.0635
34	LLMDeviation	S	0.4589	0.4967	0.2349	0.0292	0.0664
35	LogLikelihood	S	0.4508	0.4966	0.2521	0.0323	0.0595
36	ruRoBERTa-ruatd-binary	P	0.4406	0.4772	0.0110	0.0110	0.0110
37	Rank	S	0.3859	0.4972	0.0110	0.0110	0.0110

Table 2: Detection performance comparison of all MGT detection methods using all MULTITuDE test data (human and machine samples) and machine samples obfuscated by each AO method. *Macro avg. F1-score (optimal)* is just a theoretical performance and refers to this metric after calibration of the classification threshold based on ROC curve maximizing difference between TPR (true positive rate) and FPR (false positive rate). Similarly, *Macro avg. F1-score (1% FPR) and (5% FPR)* refer to this metric for the classification threshold closest to (but below) 1% and 5% of FPR. For the exact calculations of such thresholds, see the published source code. F refers to fine-tuned, P to pre-trained, and S to statistical MGT detection methods category.

ered as human) (Tripto et al., 2024). Based on the results of achieved AUC ROC values, existing detection methods can be also used for distinguishing between human-written and obfuscated machine-generated texts. However, based on Macro avg. F1-score maximum values, the data are quite challenging for the task. A high difference between AUC ROC and Macro avg. F1-score values are due to highly imbalanced data with a ratio between human and machine classes around 1:80.

4.1 Multilingual Capability of AO Methods

In this experiment, we aim to answer the following research question: **RQ1:** *Are the available out-of-the-box automated AO methods usable in multilingual settings?* The objective is to find out whether an adversary can actually easily use the available methods to evade detection of MGTs in non-English languages. Is there a difference in the effect between languages (and their relationships)? Are there differences between AO method categories in transferability to non-English languages?

To answer these questions, we measure ASR of individual AO methods for each language by using optimal predictions of all MGT detection methods (i.e., the classification thresholds optimized to maximize the difference between TPR and FPR). Table 3 shows the results, where a darker color represents a higher ASR value. For each test language, we test whether the differences between AO methods are statistically significant. To do this, we conduct repeated measures ANOVA tests for each test language: we use ASR for a given test language as a dependent variable, the MGT detection methods as "subjects" and the AO method as an independent within-subjects variable. For all 11 test languages, the observed differences are statistically significant ($p < 0.05$). We further conduct post hoc pairwise tests between pairs of AO methods per each test language for a more in depth analysis.

To avoid biased results due to low-quality machine-generated texts, the results are reported for the data generated by the three best LLM generators (gpt-4, gpt-3.5-turbo, vicuna) based on the

AO Method (Category)	Test Language [mean (±confidence interval)]											→ Average
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
m2m100-1.2B (B)	0.1060 (±0.19)	0.2759 (±0.13)	0.1429 (±0.16)	0.2451 (±0.13)	0.3063 (±0.11)	0.2251 (±0.12)	0.2691 (±0.11)	0.2484 (±0.11)	0.2014 (±0.14)	0.1449 (±0.13)	0.1841 (±0.05)	0.2136
nllb-200-distilled-1.3B (B)	0.0983 (±0.01)	0.3262 (±0.02)	0.1980 (±0.01)	0.1987 (±0.01)	0.2133 (±0.06)	0.1846 (±0.01)	0.2307 (±0.01)	0.1990 (±0.01)	0.1868 (±0.01)	0.1542 (±0.01)	0.2725 (±0.00)	0.2057
Pegasus-paraphrase (P)	0.1706 (±0.03)	0.5279 (±0.05)	0.4143 (±0.04)	0.4564 (±0.08)	0.1912 (±0.08)	0.4730 (±0.06)	0.4877 (±0.05)	0.5883 (±0.05)	0.1395 (±0.05)	0.0489 (±0.04)	0.2754 (±0.07)	0.3430
DIPPER (P)	0.2425 (±0.03)	0.1830 (±0.06)	0.2042 (±0.05)	0.2191 (±0.05)	0.1556 (±0.05)	0.2478 (±0.04)	0.2120 (±0.04)	0.2645 (±0.04)	0.1199 (±0.04)	0.0719 (±0.05)	0.2991 (±0.12)	0.2018
ChatGPT (P)	0.0803 (±0.00)	0.1076 (±0.10)	0.0958 (±0.06)	0.1020 (±0.11)	0.1332 (±0.13)	0.0798 (±0.11)	0.0920 (±0.11)	0.0766 (±0.12)	0.1026 (±0.08)	0.0831 (±0.03)	0.1052 (±0.00)	0.0962
GPTZzzs (T)	0.0053 (±0.14)	0.0741 (±0.07)	0.0173 (±0.11)	0.0308 (±0.12)	0.4140 (±0.06)	0.0849 (±0.13)	0.0797 (±0.13)	0.0928 (±0.09)	0.0031 (±0.06)	0.0010 (±0.05)	0.0095 (±0.11)	0.0739
GPTZeroBypasser (T)	0.3764 (±0.20)	0.4698 (±0.17)	0.2328 (±0.13)	0.4492 (±0.14)	0.5752 (±0.17)	0.5378 (±0.15)	0.5634 (±0.15)	0.5857 (±0.16)	0.4713 (±0.18)	0.3633 (±0.15)	0.1923 (±0.10)	0.4379
HomoglyphAttack (T)	0.3767 (±0.00)	0.7154 (±0.05)	0.4593 (±0.01)	0.6440 (±0.02)	0.7684 (±0.14)	0.7033 (±0.05)	0.7447 (±0.04)	0.7495 (±0.05)	0.5371 (±0.00)	0.4147 (±0.00)	0.1131 (±0.00)	0.5660
ALISON (T)	0.0216 (±0.02)	0.0451 (±0.04)	0.0180 (±0.03)	0.0275 (±0.03)	0.1092 (±0.06)	0.0312 (±0.02)	0.0298 (±0.02)	0.0340 (±0.02)	0.0227 (±0.03)	0.0261 (±0.03)	0.0094 (±0.04)	0.0340
DFTFooler (T)	0.0036 (±0.16)	0.2149 (±0.13)	0.1105 (±0.16)	0.3061 (±0.17)	0.3011 (±0.07)	0.2206 (±0.14)	0.3147 (±0.14)	0.2678 (±0.15)	0.0960 (±0.10)	0.0372 (±0.05)	0.0081 (±0.14)	0.1710
↓ Average	0.1481	0.2940	0.1893	0.2679	0.3167	0.2788	0.3024	0.3107	0.1880	0.1345	0.1469	

Table 3: Attack success rate of AO methods based on classification thresholds of individual MGT detection methods optimized for each language separately. Only human data and machine data generated by the selected best three LLM generators (gpt-4, gpt-3.5-turbo, vicuna) are used. Only MGT detection methods achieving above 0.8 AUC ROC per each language on original test data are included (see Table 14). Per-language mean values of all detection methods are reported along with 95% confidence interval error bounds. B refers to backtranslation, P to paraphrasing, and T to text edits AO method category.

text similarity to human-written counterparts (see Table 8). Similarly, to avoid low-performing detectors affecting the mean values, we include only detection methods achieving above 0.8 AUC ROC per each test language on original test data (see Table 14). It results in averaging different sets of MGT detection methods for each language; therefore, we provide an ablation study including all detection methods and all LLM-generated data in Appendix D. The results might be also influenced by a potential overfit on the test data (due to calibration of the classification thresholds). Therefore, we also provide an ablation study with thresholds calibrated by using only 10% of the original (unobfuscated) test data (it confirmed the conclusions).

All (even English-only) AO methods are usable in multilingual settings. The results indicate that each AO method is able to successfully cause evading detection at least for some samples. However, only in about 12% of cases, the ASR reached 0.5 on average (representing about 50% chance of the obfuscation being successful). Thus, a practical usage of most of AO methods is questionable.

Effect of the English-only AO methods is inconsistent. GPTZzzs affects mostly English, as expected since it uses an English dictionary of synonyms. Other English-only AO methods seem to have lower (the lowest in some cases) effect on English than on the other languages. Pegasus-paraphrase seems to achieve unusually high ASR

on non-English languages, but not affecting non-Latin scripts as much. Similar behavior (regarding scripts) can be observed for DFTFooler, although in a lesser intensity. On the other hand, DIPPER seems to affect non-Latin scripts in a higher amount. A high ASR of Pegasus-parahrase cannot be solely due to changed language of the texts after obfuscation, since DIPPER changed the language for almost twice as many texts but achieves only about a half ASR.

Homoglyph-based attacks are most successful across languages. For most languages, these attacks offer above 50% chance of successfully avoiding MGT detection. This effect can be expected since these attacks change the characters of the words, thus influencing inner representations (embeddings) in the detection models. However, such attacks can be easily detected/alleviated by preprocessing (e.g., by detecting multi-script texts). Out of the test languages, Chinese seems to be the most resistant to this kind of AO.

Backtranslation and other paraphrasers are more successful than ChatGPT in obfusation. However, we have used only a basic prompt for paraphrasing by ChatGPT. A more sophisticated prompt directing the writing style could help to increase the ASR. Nevertheless, the ChatGPT paraphrasing does not suffer by language change in multilingual settings as do the English-only dedicated paraphrasers (Pegasus-parphrase and DIPPER).

		Test Language [mean (±confidence interval)]											→ Average
		ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
Train Language	en	-12.73% (±8.59%)	-22.54% (±8.37%)	-9.28% (±6.10%)	-12.07% (±5.41%)	-3.93% (±3.00%)	-19.20% (±7.59%)	-23.63% (±8.75%)	-19.79% (±7.79%)	-16.62% (±7.83%)	-14.51% (±7.44%)	-7.26% (±2.97%)	-14.69%
	es	-5.28% (±4.99%)	-8.21% (±5.30%)	-3.64% (±4.02%)	-7.67% (±4.01%)	-21.54% (±10.07%)	-5.15% (±3.60%)	-9.31% (±5.50%)	-8.28% (±5.39%)	-11.24% (±5.95%)	-8.02% (±5.00%)	-2.86% (±1.75%)	-8.29%
	ru	-3.90% (±3.69%)	-12.58% (±6.61%)	-3.22% (±2.99%)	-4.70% (±4.67%)	-17.11% (±9.37%)	-6.68% (±4.94%)	-11.50% (±5.51%)	-8.03% (±5.99%)	-2.80% (±1.50%)	-3.30% (±1.94%)	-3.87% (±1.87%)	-7.06%
	all	-4.77% (±4.05%)	-8.72% (±4.27%)	-4.01% (±2.60%)	-8.00% (±3.75%)	-2.14% (±1.79%)	-4.65% (±2.68%)	-11.11% (±4.73%)	-6.71% (±3.59%)	-4.35% (±2.23%)	-4.56% (±2.59%)	-4.27% (±2.09%)	-5.75%
Category	F	-6.67% (±2.80%)	-13.01% (±3.19%)	-5.04% (±2.05%)	-8.11% (±2.22%)	-11.18% (±3.67%)	-8.92% (±2.62%)	-13.89% (±3.18%)	-10.70% (±2.96%)	-8.75% (±2.61%)	-7.60% (±2.40%)	-4.57% (±1.11%)	-8.95%
	P	-9.58% (±7.23%)	-3.07% (±6.73%)	-5.94% (±6.22%)	-6.76% (±6.72%)	-20.46% (±7.89%)	-7.00% (±6.71%)	-3.63% (±6.04%)	-8.30% (±6.26%)	-3.91% (±7.86%)	-5.73% (±6.53%)	-0.59% (±4.98%)	-6.82%
	S	-6.22% (±7.36%)	-9.28% (±10.31%)	-7.10% (±9.28%)	-13.66% (±8.20%)	-17.79% (±9.98%)	-14.32% (±8.14%)	-10.04% (±8.69%)	-12.76% (±8.20%)	-19.42% (±7.95%)	-14.54% (±8.54%)	-6.22% (±3.50%)	-11.94%

Table 4: AUC ROC drop of differently fine-tuned MGT detection methods (top) and of different categories (F – fine-tuned, P – pre-trained, S – statistical) of MGT detection methods (bottom). Per-language mean value is reported for each group (train language or category) along with 95% confidence interval error bounds. Only test data of AO methods passing the quality check are used.

4.1.1 Human Validation of Readability

Such an experiment properly evaluates the impact of AO methods on the automated MGT detection methods. However, the AO methods could have severely damaged the texts, which would be either unreadable by humans or clearly suspicious. Therefore, we have done a manual human check (see Appendix C) of the obfuscated data sample by three human evaluators (average Pearson correlation between annotations of 0.7, total annotation agreement accuracy of 0.65 with majority 2 of 3 accuracy of 0.99). Based on such a quality check along with text-similarity analysis between unobfuscated and obfuscated texts, Pegasus-paraphrase and DIPPER AO methods are unusable in the multilingual settings. Therefore, these two methods are disqualified from the further experiments (although included in the Appendices for a reference).

4.2 Adversarial Robustness of Multilingual MGT Detection Methods

This experiment aims to answer the following research question: **RQ2:** *How robust are multilingual MGT detection methods against out-of-the-box automated AO methods?* Namely, is there a difference in robustness between languages or MGT detection method categories? Also, is there a difference between monolingually and multilingually fine-tuned MGT detection methods?

To answer these questions, we measure general detection performance (AUC ROC) drop caused by texts being obfuscated by individual AO methods for each language separately. A smaller AUC ROC drop (a higher value) means a detection method is more robust against obfuscation. To see differences, we provide aggregated results based on train

language used for fine-tuning and based on MGT detection method category in Table 4. Analogously to the previous experiment, we have tested statistical significance for each test language. AUC ROC and AUC ROC drop values per each AO method and each test language is reported in Appendix E.

There are clear differences among languages and among MGT detection method categories. All of them are most sensitive to homoglyph attacks, both of the attacks having similar effect on fine-tuned and pre-trained methods. However, statistical methods are more resistant to GPTZeroBypasser and more vulnerable to HomoglyphAttack. Fine-tuned and statistical methods are also affected by backtranslation, unlike pre-trained methods.

English is most vulnerable to obfuscation in pre-trained methods. However, this might be influenced by being mostly represented by English-only detection models. In fine-tuned and statistical methods, Arabic, Czech and Chinese are more resistant to obfuscation than the others. However, these observations are not statistically significant due to high variation between AO methods.

English-only fine-tuned MGT detection methods are least robust against obfuscation. These methods are affected more by obfuscation of non-English texts. On the other hand, detection methods fine-tuned monolingually by Spanish or Russian are mostly sensitive to English texts obfuscation. Multilingually fine-tuned detection methods are the most robust, although difference between them and the Russian or Spanish monolingually fine-tuned methods are not statistically significant.

4.3 Multilingual Obfuscation for Data Augmentation

Finally, this experiment aims to answer the following research question: **RQ3:** *Does the training of multilingual MGT detection methods also on obfuscated data increase their adversarial robustness for individual languages?* Is the effect the same when using texts obfuscated by individual AO methods separately and by all of them combined?

To answer these questions, we measure AUC ROC and focus on a difference before and after adversarial retraining by data augmentation. We refer to these detection methods as originally trained and adversarially trained. For this experiment, we use only multilingually fine-tuned detection methods to limit the scope. As in the previous experiments, only obfuscated machine-class samples (of AO methods passing the quality check) were used for augmenting the train data for adversarial retraining. This caused highly imbalanced (human vs. machine classes) train set; therefore, we have pseudorandomly upsampled the minority class using a simple duplication method. For the results not to be biased due to this difference in the fine-tuning process, we have also retrained original detection methods in this manner. For individual AO methods data adversarial retraining, we have selected one from backtranslation, one from paraphrasing, one from homoglyph attacks, and one from adversarial attacks, based on the highest absolute number of successfully obfuscated texts. For each adversarial retraining process, we used the same amounts of original and obfuscated samples.

The performance results of originally and adversarially trained MGT detection methods in Figure 2 show that in general **adversarial retraining**
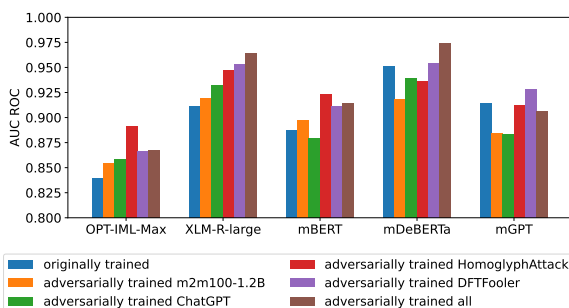

Figure 2: Detection performance (AUC ROC) of originally (leftmost bar) and adversarially trained MGT detection methods on the test data including all unobfuscated texts and obfuscated texts by AO methods passing the quality check. Note that the y-axis starts at 0.8.

by data augmentation using obfuscated texts increases the overall performance on the used test set. However, in some cases of adversarially trained models, we observe a decrease in the overall performance. To provide more insights, Table 5 in Appendix E shows the results per AO method (i.e., a portion of the test set containing obfuscated machine-class texts by a particular AO method). It provides absolute AUC ROC values as well as relative values reflecting the effect of adversarial retraining (i.e., whether it helped or not).

Table 5 shows the results per AO method (i.e., a portion of the test set containing obfuscated machine-class texts by a particular AO method). It provides absolute AUC ROC values (mean across detection methods) to illustrate performance on different portions of the test set, as well as relative values reflecting the effect of adversarial retraining. Using texts obfuscated by *all* AO methods for adversarial retraining slightly decreases the performance on the original test data, but it significantly increases the performance for homoglyph attacks, having negligible impact on other portions of the test data. Using *m2m100-1.2B* and *ChatGPT* increases performance for backtranslation and paraphrasing, but slightly drops the performance in other cases, especially *HomoglyphAttack*. Using *HomoglyphAttack* significantly increases performance for homoglyph attacks, but slightly decreases performance in other cases. And finally, using *DFTFooler* increases performance for most of the text-edits attacks, but slightly drops performance in other cases.

There are differences in the effect of adversarial retraining based on the used AO method data. Using texts obfuscated by all AO methods is the best option on average, while using backtranslation only is the worst option (although comparable with ChatGPT basic paraphrasing). Adversarial retraining using single AO method data does not only increase the performance on that particular AO method obfuscated test data, but transfers also to some other obfuscated texts. For example, using adversarial attack also increased performance for homoglyph attacks. For additional per-language results, see Table 21 in Appendix E.

5 Discussion

Existing authorship-obfuscation methods present a potential threat in non-English languages. All of them are able to confuse all

AO Method (Category)	Absolute AUC ROC ↑						Relative AUC ROC Difference ↓				
	originally trained	all	m2m100-1.2B	ChatGPT	Homoglyph Attack	DFTFooler	all	m2m100-1.2B	ChatGPT	Homoglyph Attack	DFTFooler
original	0.9372	0.9139	0.9275	0.9312	0.9270	0.9317	-2.54%	-1.07%	-0.64%	-1.09%	-0.58%
m2m100-1.2B (B)	0.9069	0.8985	0.9392	0.9019	0.8911	0.8951	-1.02%	3.56%	-0.57%	-1.80%	-1.32%
nllb-200-distilled-1.3B (B)	0.9060	0.8989	0.9234	0.9214	0.8900	0.8957	-0.87%	1.89%	1.70%	-1.81%	-1.14%
ChatGPT (P)	0.9254	0.9169	0.9258	0.9587	0.9132	0.9116	-0.97%	0.01%	3.67%	-1.35%	-1.52%
GPTZzzs (T)	0.9311	0.9170	0.9216	0.9258	0.9221	0.9349	-1.58%	-1.05%	-0.57%	-0.97%	0.41%
GPTZeroBypasser (T)	0.8443	0.9783	0.8316	0.8548	0.9554	0.9197	16.27%	-1.71%	1.37%	13.43%	9.07%
HomoglyphAttack (T)	0.8580	0.9760	0.8219	0.8183	0.9903	0.9453	14.15%	-4.34%	-4.63%	15.91%	10.40%
ALISON (T)	0.9328	0.9346	0.9244	0.9252	0.9215	0.9234	0.16%	-0.93%	-0.80%	-1.22%	-0.99%
DFTFooler (T)	0.9172	0.9306	0.9031	0.9074	0.9183	0.9626	1.44%	-1.58%	-1.05%	0.14%	5.04%
↓ Average	0.9065	0.9294	0.9021	0.9050	0.9254	0.9244	2.78%	-0.58%	-0.17%	2.36%	2.15%

Table 5: Comparison of detection performance between originally and adversarially trained MGT detection methods (in columns) on the original (unobfuscated) and per AO method (obfuscated) data (shown in rows). Mean across detection models absolute AUC ROC values are shown in the left part of the table. Relative difference in AUC ROC caused by adversarially trained methods in comparison to originally trained are shown in the right part of the table.

detection methods at least for some texts. There is a sole exception of GPTZeroBypasser not being able to fool RoBERTa-base-autextification-Detection in any test language (due to its rather random detection ability). Still, in majority of cases (combinations of AO method, MGT detection method, and test language), the attack success rate is below 50%. However, we have not considered combinations of multiple AO methods or sophisticated adjustments to multilingualism.

Pre-processing to avoid homoglyphs can protect against this type of attacks. Homoglyphs are usually considered as being able to fool humans due to visual similarity of characters but not the computers due to unrelated binary representations. If the classification models are not trained considering such cases, homoglyph-including samples can easily be misclassified. However, pre-processing steps specifically targeting homoglyphs (e.g., characters of different scripts contained in a single word) can presumably alleviate this kind of attacks. Data augmentation using such obfuscated samples also helps, as shown in our experiments.

A simple data augmentation using obfuscated texts can increase adversarial robustness of MGT detection methods. Usually, a complex adversarial training including generation of adversarial examples targeting a specific model is used to increase adversarial robustness of a model. We have shown that inclusion of generic obfuscated samples (not necessarily being adversarial examples) also helps to increase the adversarial robustness. We also observe a transferability between AO methods, meaning that using samples from one AO method for adversarial training increases resistance to another AO method during testing.

6 Conclusion

In this work, we have done the first comprehensive benchmark of authorship obfuscation methods in multilingual settings. We have also done the first evaluation of adversarial robustness (using authorship obfuscation methods) of multilingual machine-generated text detection methods. We have crafted a new public dataset of 740k obfuscated texts, which has wide usage possibilities, even beyond the purpose outlined in this work (e.g., to study differences in obfuscated human-written and obfuscated machine-generated texts, or to evaluate robustness of multi-class authorship-attribution systems).

Our results (namely nonzero attack success rate values) indicate that the existing authorship obfuscation methods, although focused primarily on English, are usable in all 11 tested languages, thus posing a potential threat for automated multilingual detection. However, human quality check of the obfuscated texts disqualified two of them as unusable in non-English and/or non-Latin script languages. The adversarial robustness results show that there are differences among machine-generated text detection method categories as well as among tested languages. The results confirmed that English monolingual fine-tuning of multilingual detectors is not suitable, since it is the least resistant against obfuscation. As we have shown, the adversarial retraining using a simple data augmentation by obfuscated texts can significantly increase the adversarial robustness of detectors, especially against homoglyph and paraphrasing attacks.

Future work should be focused on iterative and hybrid authorship obfuscation of multilingual texts to evaluate severity of such threat on machine-generated text detection.

Limitations

Selection of AO Methods. We have limited the scope of this work by constraining which **authorship obfuscation methods** (e.g., fully automated) have been used under which conditions, as described in Section 3.1. We have limited the number of trials of obfuscation methods to 10, leaving some texts not modified (i.e., they might have remained the same as the original). Due to computational and time costs, we have avoided paid services (e.g., Google Translation[9]) and slow adversarial attacks (e.g., TextFooler[10]). As already noted, in real settings, actors trying to evade the detection of machine-generated texts might use a combination of AO methods or use humans in the loop.

Settings of AO Methods. We have used only the default settings of AO methods where available. By tuning the obfuscation settings, one could achieve slightly different results, as shown by our ablation study for HomoglyphAttack in Appendix D (Table 13). Therefore, a comparison between individual AO methods (e.g., the best and worst AO methods) should be interpreted with this in mind. However, general observations about multilingual usage of existing AO methods, per-language differences, or adversarial robustness evaluations are not significantly affected by these and still hold.

Dataset Selection. We have used a **single dataset** (MULTITuDE) for experiments, limiting the number of **languages** for training (3) and for evaluation (11), limiting the number of **LLMs** used to generate the machine texts (8), and limiting the **domain** of the texts to news articles. Extension in all the mentioned aspects as well as in the **number of samples** (especially better balancing between human and machine classes) could help the generalizations; however, not influencing the conclusions we could already make.

Evaluation of Obfuscation Quality. We evaluated the quality (usability) of the selected AO methods using a range of heuristics (i.e., similarity metrics, change in text length, language change, etc., see Appendix C). Additionally, we did a manual quality check on a small random subset of the obfuscated data to evaluate their quality from the users' perspective, i.e., whether the obfuscation does not make the texts easily detectable by humans in the process. Nevertheless, we did not evaluate the quality on the whole dataset, which was unfeasible due to the scale of our dataset (740k obfuscated texts).

Ethics Statement

Our work identifies the most effective authorship obfuscation methods for each tested language, thus identifying potential vulnerabilities of machine-generated text detection methods. Although there is a risk that this can be misused by adversaries, the benefits of our work outweigh such a risk. First, we use existing freely available AO methods. Secondly, although we identify weaknesses of already existing detection methods, we also provide simple means to eliminate them through data augmentation during adversarial retraining.

We use an existing data resource in our work (MULTITuDE) in accordance with its intended use and license. As a part of our work, we extend this resource by generating obfuscated human and machine-generated texts. We will publish this data (for research purposes only) together with our code to ensure reproducibility of our work.

Acknowledgements

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*.

Bo Ai, Yuchen Wang, Yugin Tan, and Samson Tan. 2022. Whodunit? learning to contrast for authorship attribution. In *AACL*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Mishari Almishari, Ekin Oguz, and Gene Tsudik. 2014. Fighting authorship linkability with crowdsourcing. In *Proceedings of the Second ACM Conference on Online Social Networks*, COSN '14, page 69–82, New York, NY, USA. Association for Computing Machinery.

Malik Altakrori, Thomas Scialom, Benjamin C. M. Fung, and Jackie Chi Kit Cheung. 2022. A multifaceted framework to evaluate evasion, content preservation, and misattribution in authorship obfuscation techniques. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2391–2406, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Florian Beil, Martin Ester, and Xiaowei Xu. 2002. Frequent term-based text clustering. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 436–442, New York, NY, USA. Association for Computing Machinery.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Jan Cegin, Jakub Simko, and Peter Brusilovsky. 2023. ChatGPT to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Canyu Chen and Kai Shu. 2023a. Can llm-generated misinformation be detected? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Canyu Chen and Kai Shu. 2023b. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Evan Crothers, Nathalie Japkowicz, Herna Viktor, and Paula Branco. 2022. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22:1–48.

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.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023a. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. *arXiv preprint arxiv:2301.07597*.

Jiaxin Guo, Daimeng Wei, Zhanglin Wu, Zongyao Li, Zhiqiang Rao, Minghan Wang, Hengchao Shang, Xiaoyu Chen, Zhengzhe Yu, Shaojun Li, Yuhao Xie, Lizhi Lei, and Hao Yang. 2023b. The HW-TSC's simultaneous speech-to-text translation system for IWSLT 2023 evaluation. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 376–382, Toronto, Canada (in-person and online). 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. *arXiv preprint arXiv:2401.12070*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. OPT-IML: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Yashwant Keswani, Harsh Trivedi, Parth Mehta, and P Majumder. 2016. Author masking through translation—notebook for pan at clef 2016. In *Conference and Labs of the Evaluation Forum, CLEF*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. Vancouver, Canada.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*.

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.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In

Proceedings of the ACM Web Conference 2023, pages 3637–3647.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Deepfake text detection in the wild. *arXiv preprint arxiv:2305.13242*.

Vijini Liyanage, D. Buscaldi, and Adeline Nazarenko. 2022. A benchmark corpus for the detection of automatically generated text in academic publications. In *International Conference on Language Resources and Evaluation*.

Ning Lu, Shengcai Liu, Rui He, and Ke Tang. 2023. Large language models can be guided to evade ai-generated text detection. *arXiv preprint arXiv:2305.10847*.

Jason Lucas, Adaku Uchendu, Michiharu Yamashita, Jooyoung Lee, Shaurya Rohatgi, and Dongwon Lee. 2023. Fighting fire with fire: The dual role of llms in crafting and detecting elusive disinformation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14279–14305.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023a. 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, Jason Samuel Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023b. MULTITuDE. Zenodo dataset.

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*.

Shaoor Munir, Brishna Batool, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2021. Through the looking glass: Learning to attribute synthetic text generated by language models. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Mikhail Orzhenovskii. 2022. Detecting auto-generated texts with language model and attacking the detector. *Computational Linguistics and Intellectual Technologies*.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1613–1630.

Jiashu Pu, Zi Feng Huang, Yadong Xi, Guandan Chen, Weijie Chen, and Rongsheng Zhang. 2022. Unraveling the mystery of artifacts in machine generated text. In *International Conference on Language Resources and Evaluation*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sara Rosenthal, Mihaela Bornea, and Avirup Sil. 2021. Are multilingual BERT models robust? A case study on adversarial attacks for multilingual question answering. *arXiv preprint arXiv:2104.07646*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2023. Red teaming language model detectors with language models. *arXiv preprint arXiv:2305.19713*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

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*.

Michal Spiegel and Dominik Macko. 2024. IMGTB: A framework for machine-generated text detection benchmarking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 172–179, Bangkok, Thailand. Association for Computational Linguistics.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Nafis Irtiza Tripto, Saranya Venkatraman, Dominik Macko, Robert Moro, Ivan Srba, Adaku Uchendu, Thai Le, and Dongwon Lee. 2024. A ship of theseus: Curious cases of paraphrasing in LLM-generated texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6608–6625, Bangkok, Thailand. Association for Computational Linguistics.

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, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. In *Conference on Empirical Methods in Natural Language Processing*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2024. Gpt-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

Pranav Narayanan Venkit, Sanjana Gautam, Ruchi Panchanadikar, Ting-Hao Huang, and Shomir Wilson. 2023. Nationality bias in text generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–122.

Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2024. Disinformation capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14830–14847, Bangkok, Thailand. Association for Computational Linguistics.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

6360

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research.*

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724.*

Zhendong Wu and Hui Xiang. 2023. MFD: Multi-feature detection of LLM-generated text. *PREPRINT (Version 1) available at Research Square.*

Eric Xing, Saranya Venkatraman, Thai Le, and Dongwon Lee. 2024. ALISON: Fast and effective stylometric authorship obfuscation. In *38th AAAI Conf. on Artificial Intelligence (AAAI).*

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent N-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations.*

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations.*

A Computational Resources

In the experiments, we have utilized a computational infrastructure consisting of $1\times$ A100 40GB GPUs for application of the authorship obfuscation methods, text-similarity metrics calculations, and machine-generated text detectors requiring LLM inference and fine-tuning of pre-trained models. Cumulatively, these processes required approximately 1,875 GPU hours. For other processes, such as data analysis, simpler metric calculations (not requiring GPU), using some obfuscation methods (ChatGPT inference via OpenAI API, GPTZzzs, GPTZeroBypasser, and HomoglyphAttack), and

results analysis, we used Google Colab[11] without GPU acceleration.

B Authorship Obfuscation Methods

The used authorship obfuscation methods, grouped into three categories, are summarized in Table 6 along with the used default parameters or settings for obfuscation.

	AO Method	Parameters
Backtranslation	m2m100-1.2B[12] (Fan et al., 2021)	We have used English as an intermediary language for non-English texts and Spanish for English texts.
	nllb-200-distilled-1.3B[13] (Costa-jussà et al., 2022)	We have used English as an intermediary language for non-English texts and Spanish for English texts, with max_length set to 512.
Paraphrasing	Pegasus-paraphrase[14]	We have used the model for paraphrasing each sentence separately, with max_length of 60, num_beams of 10, and $temperature$ of 1.5, as provided in exemplar usage on HuggingFace.
	DIPPER[15] (Krishna et al., 2023)	We have used both the $lex\_diversity$ and $order\_diversity$ set to 40 (as the most intensive settings in the DIPPER paper), the nucleus sampling with top_p of 0.75 and max_length of 512.
	ChatGPT[16]	We have used a basic paraphrasing prompt of "Paraphrase the following text in <language> language: <text>". We have limited the number of output tokens to 512, and used the nucleus sampling with top_p of 0.95.
Text edits	GPTZzzs[17]	We have used a random seed of 42. We have set 30% of words to be replaced by synonyms using Zaibacu Thesaurus and 50% of adjectives to changed emphasis on, without ignoring quotations and without using only common words.
	GPTZeroBypasser[18]	We have used a random seed of 42, inserting a zero-width joiner pseudorandomly with a probability of 0.2.
	HomoglyphAttack	We have used a random seed of 42. We have used the whole confusables table[19] to psudorandomly replace letters for their homoglyphs with a probability (of a character being replaced) set to 0.1.
	ALISON[20] (Xing et al., 2024)	We have use bert-base-multilingual-cased as a base model and min_length of 2, other parameters used the default values.
	DFTFooler[21] (Pu et al., 2023)	We have used bert-base-multilingual-cased as the backend model and the number of samples to attack set to 100,000 to obfuscate all texts. Other parameters used the default values.

Table 6: Authorship obfuscation methods parameters.

C Post-Obfuscation Similarity Analysis

To evaluate the similarity between the obfuscated and the original texts (and thus the quality of the obfuscation), we have used the following metrics: *METEOR* (Banerjee and Lavie, 2005) as a standard metric used in machine translation based on

---

[11] https://colab.research.google.com/

[12] https://huggingface.co/facebook/m2m100_1.2B

[13] https://huggingface.co/facebook/nllb-200-distilled-1.3B

[14] https://huggingface.co/tuner007/pegasus_paraphrase

[15] https://github.com/martiansideofthemoon/ai-detection-paraphrases

[16] https://openai.com/blog/chatgpt

[17] https://github.com/Declipsonator/GPTZzzs

[18] https://github.com/o2161405/GPTZero-Bypasser

[19] https://www.unicode.org/Public/security/8.0.0/confusables.txt

[20] https://github.com/EricX003/ALISON

[21] https://github.com/jmpu/DeepfakeTextDetection/tree/main/DFTFooler

AO Method	METEOR ↑	BERTScore ↑	USE ↑	ngram ↑	TF ↑	LD ↓	CharLenDiff → 1	LangCheck ↓
m2m100-1.2B	0.452 (±0.22)	0.853 (±0.07)	0.842 (±0.13)	0.485 (±0.18)	0.810 (±0.16)	0.467 (±0.21)	0.678 (±0.24)	0.55%
nllb-200-distilled-1.3B	0.398 (±0.23)	0.833 (±0.08)	0.797 (±0.17)	0.431 (±0.20)	0.775 (±0.18)	0.542 (±0.33)	0.638 (±0.39)	**0.30%**
Pegasus-paraphrase	0.331 (±0.24)	0.708 (±0.15)	0.575 (±0.34)	0.324 (±0.23)	0.646 (±0.28)	0.698 (±0.40)	0.556 (±0.49)	28.17%
DIPPER	0.276 (±0.23)	0.760 (±0.10)	0.683 (±0.26)	0.282 (±0.23)	0.528 (±0.34)	0.704 (±0.28)	0.756 (±0.32)	51.79%
ChatGPT	0.566 (±0.22)	0.867 (±0.07)	0.884 (±0.11)	0.546 (±0.18)	0.819 (±0.16)	0.418 (±0.22)	0.920 (±0.24)	1.38%
GPTZzzs	0.968 (±0.06)	0.974 (±0.03)	0.988 (±0.02)	0.918 (±0.09)	**0.986 (±0.02)**	0.046 (±0.05)	1.017 (±0.02)	2.78%
GPTZeroBypasser	0.131 (±0.10)	0.651 (±0.21)	0.375 (±0.18)	0.168 (±0.14)	0.130 (±0.17)	0.495 (±0.17)	1.238 (±0.03)	37.33%
HomoglyphAttack	0.568 (±0.10)	0.778 (±0.05)	0.762 (±0.11)	0.596 (±0.06)	0.179 (±0.16)	0.094 (±0.02)	**1.003 (±0.00)**	2.74%
ALISON	**0.987 (±0.06)**	**0.991 (±0.02)**	**0.993 (±0.01)**	**0.971 (±0.04)**	0.968 (±0.07)	**0.009 (±0.01)**	1.005 (±0.01)	2.77%
DFTFooler	0.948 (±0.07)	0.977 (±0.02)	0.990 (±0.02)	0.920 (±0.08)	0.963 (±0.06)	0.033 (±0.04)	1.004 (±0.01)	2.78%

Table 7: Similarity analysis between obfuscated and original texts [mean (± std)]. *LD* is normalized Levenshtein distance, *LangCheck* is percentage of texts with changed languages based on FastText predictions, arrows refer to values representing more similar texts, boldfaced values represent the most similar texts for each metric.

LLM Generator	METEOR ↑	BERTScore ↑	USE ↑	ngram ↑	TF ↑	LangCheck ↓
alpaca-lora-30b	0.110 (±0.07)	0.668 (±0.04)	0.516 (±0.20)	0.170 (±0.08)	0.619 (±0.20)	1.01%
gpt-3.5-turbo	0.139 (±0.07)	0.678 (±0.04)	0.584 (±0.20)	0.215 (±0.09)	0.650 (±0.20)	0.02%
gpt-4	**0.163 (±0.08)**	**0.688 (±0.04)**	**0.629 (±0.21)**	**0.253 (±0.10)**	**0.667 (±0.21)**	0.00%
llama-65b	0.099 (±0.06)	0.619 (±0.06)	0.448 (±0.22)	0.138 (±0.10)	0.513 (±0.23)	14.29%
opt-66b	0.116 (±0.08)	0.655 (±0.05)	0.464 (±0.25)	0.175 (±0.10)	0.595 (±0.20)	3.53%
opt-iml-max-1.3b	0.106 (±0.08)	0.635 (±0.06)	0.402 (±0.26)	0.159 (±0.10)	0.548 (±0.23)	4.80%
text-davinci-003	0.123 (±0.07)	0.674 (±0.04)	0.542 (±0.21)	0.196 (±0.09)	0.620 (±0.21)	0.00%
vicuna-13b	0.131 (±0.07)	0.667 (±0.04)	0.548 (±0.21)	0.199 (±0.09)	0.630 (±0.21)	2.89%

Table 8: Similarity analysis between machine-generated and human-written texts in original MULTITuDE data [mean (± std)]. *LangCheck* is percentage of texts with intended languages mismatching FastText predictions, arrows refer to values representing more similar texts, boldfaced values represent the most similar texts for each metric.

unigrams, *BERTScore* (Zhang et al., 2019) with mBERT model as a contextual embeddings based similarity metric that is more robust to adversarial texts, and cosine similarity of *USE* (multilingual version of universal sentence encoder) (Yang et al., 2020) sentence-level embeddings as a semantic similarity metric. For a better explainability, we have also used character-level *ngram*[22] (3-grams in our case) as a language-independent string similarity metric in the form of a ratio of the shared ngrams between two strings and cosine similarity of *TF* (term frequency) (Beil et al., 2002) as a word-level frequency similarity metric by using polyglot[23] multilingual tokenizer. All of the mentioned metrics represent a score between 0 and 1, while closer to 1 means higher similarity. In addition, we have measured amount (severity) of text modification by using the *Levenshtein distance* as a character-level edit distance[24], which was normalized to the text length (a number closer to 0 means lesser modification). Since some LLMs occasionally fail in the task by generating duplicated characters, we have calculated a ratio of character-level lengths between obfuscated and unobfuscated texts after removal of subsequent duplicate characters (e.g., 'ssssssss' is changed to a single 's' for the cal-

culation purpose). We call this ratio *CharLenDiff*. Such a ratio value above 1 represents that the obfuscation lengthened the text and between 0 to 1 means that obfuscation shortened the text (while not taking duplicated characters into account). Also, since multiple of the used AO methods are English-only by their nature, we have checked possible changed language of the obfuscated texts by *FastText*[25] language detection.

A summary of the post-obfuscation analysis is provided in Table 7. For a reference, a summary of per-generator similarity analysis between the corresponding human-written and machine-generated texts in the original MULTITuDE dataset is provided in Table 8. Based on the results, we can see that the texts obfuscated by adversarial attacks are the most similar to the original versions. On the other hand, English-only paraphrasers (Pegasus-paraphrase and DIPPER) are clearly the outliers (by most of the metrics). As *LangCheck* indicates, these methods probably changed the language of the texts to English (or incorporated several English parts), which must be considered during the experiments. Based on *CharLenDiff*, we might assume that Pegasus-paraphrase AO method shortened the obfuscated texts by a high amount on average due to failing of the model to generate a meaningful text (shorter strings of duplicated char-

AO Method	Test Language [mean (± std)]										
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh
m2m100-1.2B	0.9 (±0.31)	0.8 (±0.41)	0.9 (±0.25)	0.8 (±0.41)	0.7 (±0.47)	0.9 (±0.31)	0.8 (±0.43)	0.9 (±0.25)	0.8 (±0.53)	1.0 (±0.00)	0.7 (±0.52)
nllb-200-distilled-1.3B	1.0 (±0.00)	0.6 (±0.72)	0.6 (±0.68)	0.6 (±0.76)	0.8 (±0.43)	0.5 (±0.73)	0.5 (±0.78)	0.8 (±0.57)	0.8 (±0.50)	0.9 (±0.31)	0.9 (±0.35)
Pegasus-paraphrase	-1.0 (±0.00)	0.2 (±0.68)	0.5 (±0.51)	0.4 (±0.63)	0.8 (±0.46)	0.2 (±0.95)	0.4 (±0.61)	0.4 (±0.77)	-1.0 (±0.00)	-1.0 (±0.00)	-1.0 (±0.00)
DIPPER	-0.9 (±0.51)	-0.7 (±0.65)	-0.9 (±0.43)	-1.0 (±0.18)	0.5 (±0.78)	-0.5 (±0.82)	-0.9 (±0.35)	-0.9 (±0.51)	-1.0 (±0.00)	-1.0 (±0.00)	-1.0 (±0.00)
ChatGPT	1.0 (±0.00)	0.9 (±0.40)	0.9 (±0.51)	0.6 (±0.82)	0.9 (±0.37)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	0.9 (±0.37)	1.0 (±0.00)	0.8 (±0.63)
GPTZzzs	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)
GPTZeroBypasser	1.0 (±0.00)	0.7 (±0.48)	0.7 (±0.48)	0.7 (±0.48)	0.7 (±0.48)	0.7 (±0.48)	0.6 (±0.50)	0.6 (±0.49)	0.9 (±0.25)	0.9 (±0.35)	0.6 (±0.49)
HomoglyphAttack	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.3 (±0.48)	0.4 (±0.49)	0.6 (±0.50)
ALISON	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)
DFTFooler	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)

Table 9: Human quality check of the obfuscated data subset. Mean value for each AO method and each test language is reported, where -1.0 is the lowest quality (a red-like background color gradient) and 1.0 is the highest quality (a green-like background color gradient).

acters mentioned above). Also, there is indicated a severe language change for GPTZeroBypasser AO method. In this case, however, we assume that FastText language detection was confused due to a change of portion of Latin characters to Cyrillic, while in HomoglyphAttack, the distribution of homoglyphs using different scripts is not as uniform (thus not causing the same effect). Homoglyph attacks achieve lower values for most of the similarity metrics, which is expected since tokens as well as ngrams are affected by changed characters (in contrast to humans being possibly fooled visually by such texts). Backtranslation and paraphrasing AO methods behave similarly in this analysis, both categories tending to shorten the texts (ChatGPT being exception, where shortening is not as severe).

C.1 Human Quality Check

The aim of the human quality check is to evaluate the obfuscated data quality, i.e., whether the obfuscated data is/is not clearly damaged from a human view. Massive human study is infeasible due to the scope and complexity of the work (it would require proficient annotators in 11 languages); therefore, a manual quality check by three human evaluators (selected from the paper authors) has been executed to identify unusable AO methods.

To reduce load and noise by LLM generators, we use just the obfuscated human data for AO evaluation (except for DFTFooler, which obfuscates only machine-class samples, where GPT-4 generated data are used instead). We have pseudo-randomly selected 10 samples per each test language for each AO method, which resulted in N = 1100 samples. The samples were annotated on the 3 step scale:

- *-1 – unusable text* – no meaningful text at all (e.g., just some numbers, or repeated/random characters), text completely in a different language than intended, massive repetition (more than one duplicated word)

- *0 – obviously damaged but readable text* – mixed languages (partly in the intended language), missing letters (e.g., letters with accents), mixed script (changed characters)

- *1 – no obvious damage in the text* – no obvious modification unusual for a human text

The annotations by the three annotators achieved averaged pairwise Pearson correlation of 0.7. The total annotation agreement accuracy of 0.65 (i.e., full agreement – 65% of the samples were annotated by the same values by all three reviewers). When considering majority agreement (i.e., 2 of 3 annotations matched for a given sample), accuracy achieved 0.99.

The mean quality annotation values are provided in Table 9 per each AO method and for each test language. Pegasus-paraphrase and DIPPER AO methods are clearly disqualified form being usable for some non-English languages. Also, homoglyph attacks are identifiable by humans; however, if not used intensively, they can deceive a human reader.

An example of obfuscated texts by each AO method with a corresponding annotation of the quality and explanation in a comment is provided in Table 10.

D Ablation Study

For ablation study regarding attack success rate (ASR) experiments, in which we included only detection methods achieving above 0.8 AUC ROC per each language (based on original MULTITuDE test data subset) and data generated by only best three LLM generators, we provide full coverage (all detection methods, all LLM generated data) in Table 11. Similarly, the results calculated when the classification thresholds are calibrated by using only 10% of the test data are provided in Table 12. Although the values are slightly different from Table 3, the general conclusions hold.

AO Method	Obfuscated Text	Quality	Comment
original	Los dos soldados franceses que permanecían ingresados en el Hospital Universitario de Albacete tras ...	N/A	
m2m100-1.2B	Los muertos dos soldados franceses que quedaron en el Hospital Universitario de Albacete tras el acc...	1	
nllb-200-distilled-1.3B	Los soldados franceses que fueron ingresados en el Hospital Universitario de Albacete después de...	1	
Pegasus-paraphrase	The hospital Universitario de Albacete tras el accidente el pasado lunes de un F-16, ha sido traslad...	1	
DIPPER	Most importantly, the Italian minister of defense visited the wounded in the hospital in Los Llanos...	-1	language change
ChatGPT	Los soldados franceses que estaban hospitalizados en el Hospital Universitario de Albacete despu...	1	
GPTZzzs	Los soldados franceses que permanecían ingresados en el Hospital Universitario de Albacete tras ...	1	
GPTZeroBypasser	Los dos sol dados franceses que permanecían ingresados en el Hospital Universitario...	0	multiple scripts
HomoglyphAttack	Los dos soldados franceses Que permaecín ingesados en el Hӧspiťaʹl UniＶersitario de Abaceţe trãs ...	0	weird characters
ALISON	Los dos soldados franceses que permanecían ingresados en el Hospital Universitario de Albacete tras ...	1	
DFTFooler	Los dos soldados franceses que permanecían ingresados en el Hospital Universitario de Albacete tras ...	1	

Table 10: An example of original and obfuscated sample by each AO method along with a human annotation.

AO Method	Test Language [mean (±confidence interval)]											→ Average
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
m2m100-1.2B	0.3177	0.3101	0.2008	0.2671	0.2727	0.2389	0.2764	0.2495	0.2729	0.2510	0.2911	0.2680
	(±0.12)	(±0.11)	(±0.12)	(±0.10)	(±0.11)	(±0.11)	(±0.10)	(±0.11)	(±0.11)	(±0.11)	(±0.09)	
nllb-200-distilled-1.3B	0.2890	0.3475	0.2492	0.2369	0.1985	0.2195	0.2564	0.2297	0.2721	0.3025	0.3350	0.2669
	(±0.04)	(±0.03)	(±0.04)	(±0.04)	(±0.05)	(±0.04)	(±0.03)	(±0.04)	(±0.04)	(±0.06)	(±0.02)	
Pegasus-paraphrase	0.3259	0.4803	0.4151	0.4367	0.1708	0.4469	0.4542	0.5319	0.2898	0.2228	0.2371	0.3647
	(±0.09)	(±0.05)	(±0.05)	(±0.06)	(±0.06)	(±0.05)	(±0.05)	(±0.04)	(±0.08)	(±0.06)	(±0.08)	
DIPPER	0.3614	0.1842	0.2096	0.2151	0.1382	0.2148	0.1853	0.2262	0.2314	0.2418	0.3444	0.2320
	(±0.08)	(±0.05)	(±0.06)	(±0.05)	(±0.04)	(±0.05)	(±0.04)	(±0.04)	(±0.07)	(±0.07)	(±0.09)	
ChatGPT	0.2355	0.1878	0.1865	0.1801	0.1211	0.1426	0.1735	0.1528	0.2154	0.2044	0.2095	0.1826
	(±0.04)	(±0.07)	(±0.06)	(±0.06)	(±0.10)	(±0.09)	(±0.09)	(±0.09)	(±0.07)	(±0.07)	(±0.02)	
GPTZzzs	0.0481	0.0837	0.0362	0.0629	0.3269	0.1094	0.0860	0.1181	0.0296	0.0300	0.0387	0.0882
	(±0.10)	(±0.05)	(±0.08)	(±0.09)	(±0.04)	(±0.09)	(±0.06)	(±0.09)	(±0.08)	(±0.11)	(±0.09)	
GPTZeroBypasser	0.5392	0.4947	0.3705	0.5409	0.6147	0.5374	0.5696	0.5947	0.5170	0.4818	0.3095	0.5064
	(±0.13)	(±0.14)	(±0.13)	(±0.11)	(±0.14)	(±0.13)	(±0.13)	(±0.13)	(±0.13)	(±0.13)	(±0.10)	
HomoglyphAttack	0.5723	0.6496	0.4868	0.6305	0.6839	0.6324	0.7009	0.6687	0.5520	0.5449	0.2532	0.5796
	(±0.04)	(±0.04)	(±0.02)	(±0.04)	(±0.11)	(±0.05)	(±0.03)	(±0.05)	(±0.03)	(±0.03)	(±0.02)	
ALISON	0.0740	0.0679	0.0625	0.0700	0.0945	0.0705	0.0582	0.0751	0.0867	0.1208	0.0337	0.0740
	(±0.07)	(±0.06)	(±0.06)	(±0.05)	(±0.04)	(±0.05)	(±0.06)	(±0.05)	(±0.08)	(±0.07)	(±0.07)	
DFTFooler	0.0459	0.2070	0.1376	0.3230	0.2289	0.2299	0.2841	0.2810	0.1616	0.1474	0.0397	0.1896
	(±0.10)	(±0.11)	(±0.12)	(±0.12)	(±0.05)	(±0.12)	(±0.12)	(±0.11)	(±0.11)	(±0.10)	(±0.08)	
↓ Average	0.2809	0.3013	0.2355	0.2963	0.2850	0.2842	0.3045	0.3128	0.2628	0.2547	0.2092	

Table 11: Ablation of attack success rate of AO methods based on classification thresholds of individual MGT detection methods optimized for each language separately. Per-language mean values of all detection methods are reported along with 95% confidence interval error bounds.

AO Method	Test Language [mean]											→ Average
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
m2m100-1.2B	0.0980	0.2455	0.1281	0.2295	0.2545	0.1738	0.2474	0.2062	0.1718	0.1419	0.1155	0.1829
nllb-200-distilled-1.3B	0.0954	0.2792	0.1589	0.1756	0.1670	0.1338	0.2109	0.1450	0.1513	0.1400	0.1542	0.1647
Pegasus-paraphrase	0.1572	0.4963	0.3787	0.4410	0.1570	0.4026	0.3532	0.4889	0.0597	0.0734	0.1358	0.2858
DIPPER	0.2405	0.1499	0.1585	0.1143	0.1153	0.0919	0.0910	0.1051	0.0885	0.0810	0.2267	0.1330
ChatGPT	0.0847	0.0768	0.0865	0.1137	0.1071	0.0577	0.0919	0.0627	0.0804	0.0791	0.0826	0.0839
GPTZzzs	0.0036	0.0540	0.0145	0.0189	0.3712	0.0369	0.0290	0.0364	0.0013	0.0005	0.0079	0.0522
GPTZeroBypasser	0.3639	0.4910	0.2516	0.5986	0.5430	0.6894	0.6778	0.7087	0.4551	0.3692	0.2067	0.4868
HomoglyphAttack	0.3446	0.7220	0.4143	0.6106	0.7376	0.6785	0.7383	0.7286	0.4699	0.3848	0.0858	0.5377
ALISON	0.0201	0.0283	0.0171	0.0261	0.0739	0.0216	0.0240	0.0221	0.0154	0.0245	0.0071	0.0255
DFTFooler	0.0023	0.1961	0.1001	0.2533	0.2625	0.1165	0.1603	0.1349	0.0361	0.0228	0.0056	0.1173
↓ Average	0.1410	0.2739	0.1708	0.2582	0.2789	0.2403	0.2624	0.2639	0.1530	0.1317	0.1028	

Table 12: Ablation of attack success rate of AO methods based on classification thresholds of individual MGT detection methods optimized for each language separately by using 10% of original (unobfuscated) test data. Only MGT detection methods achieving above 0.8 AUC ROC per each language on original test data are included.

	Detector	Test Language [mean]										
		ar	ca	cs	de	en	es	nl	pt	ru	uk	zh
a)	MFD	0.9427	0.9293	0.7589	0.8467	0.7289	0.8888	0.8417	0.7203	0.9071	0.9699	0.3180
	RoBERTa-base-OpenAI-Detector	0.9855	0.6718	0.9501	0.8781	0.9941	0.2658	0.8682	0.3069	1.0000	0.8889	0.7921
	XLM-RoBERTa-large (en)	0.6619	0.9105	0.9172	0.7779	0.3806	0.9669	0.9025	0.9828	0.9530	0.9719	0.1622
b)	MFD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	RoBERTa-base-OpenAI-Detector	0.6662	0.2029	0.3918	0.5938	0.9776	0.1907	0.5462	0.2855	0.5481	0.2253	0.6237
	XLM-RoBERTa-large (en)	0.0826	0.1299	0.1376	0.1152	0.0454	0.1884	0.1456	0.1837	0.1905	0.2283	0.0468

Table 13: Ablation of attack success rate of the best-performing AO method *HomoglyphAttack* for the selected MGT detection methods. a) original settings used in the experiments, b) less intensive settings.

6364

For ablation of attack success rate of AO methods depending on the obfuscation settings, we performed *HomoglyphAttack* (as the most successful attack in our study) using a less intensive settings (changing characters to their homoglyphs with a probability of 0.01 instead of 0.1 in the original experiments). The classification threshold calibration was the same as in original experiments, i.e., the thresholds calibrated per-language using the original (unobfuscated) test data. The results for the three selected MGT detection methods (one of each category with the highest attack success rates in the original experiment) are provided in Table 13. As shown, the ASR is much lower in the less intensive settings. Using these settings, the *HomoglyphAttack* was not able to fool the *MFD* statistical MGT detection method for any sample in any language. The results indicate that the achieved attack success rates are highly sensitive to the used settings for obfuscation. Even low-performing AO methods, such as ALISON, can achieve higher attack success rates with proper settings adjustments. We have not experimented with settings optimization.

For ablation study regarding detection performance, such as reported in Figure 2, we provide the results in Figure 3 for filtered test samples, in which the obfuscated texts meet the chosen similarity metrics (reported in Table 7) thresholds in addition to not using the data of AO methods not passing the quality check in general. These thresholds include: *CharLenDiff* is in the interval of $(0.5, 2)$, *LD* is above 1, and *METEOR, BERTScore, USE, ngram, and TF* are above 0.5. The results show that the detection methods performance is slightly higher when using such high-quality texts than when using the whole test set including noise. The differences between originally trained and adversarially trained

detection methods are lower, but shows the same trend. Nevertheless, the adversarial retraining used not only high-quality obfuscated texts, but all of them, which could affect the results.

E Results Data

Table 14 shows a per-language comparison of AUC ROC detection performance of all MGT detection methods using subset of MULTITuDE test data, namely all human samples along with the machine samples generated by the selected three best LLM generators (gpt-4, gpt-3.5-turbo, vicuna), generating the texts most similar to human counterparts based on automated similarity metrics. Tables 15–17 include resulted mean values (across detection methods) of AUC ROC detection performance per each AO method data and per each test language. To see differences between MGT detection method categories, we provide the results separately for them. Furthermore, we provide AUC ROC drop in Tables 18–20, reflecting the robustness of MGT detection method categories in regard to each AO method.

Table 21 provides the results of per-language comparison of AUC ROC drop caused by obfuscated data before and after adversarial retraining, reflecting whether the adversarial robustness is increased or decreased. As the results indicate, it is increased in most cases (only in four cases it was slightly decreased, even those are not statistically significant). For example, HomoglyphAttack caused mean AUC ROC drop of -17.68% for the German texts in originally trained MGT detection methods and only of -3.21% in adversarially trained methods; thus, resulting in 14.47% difference.
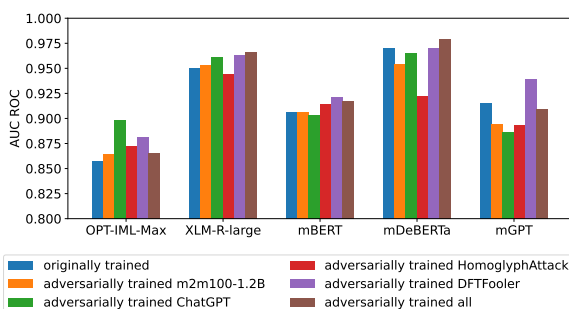

Figure 3: Ablation of detection performance (AUC ROC) of originally (leftmost bar) and adversarially trained MGT detection methods on the high-quality test data. Note that the y-axis starts at 0.8.

Rank	MGT Detection Method	Category	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	all
			Test Language [AUC ROC]											
1	XLM-RoBERTa-large (all)	F	0.981	0.990	0.990	0.982	0.994	0.995	0.982	0.977	0.988	0.988	0.958	0.983
2	mDeBERTa-v3-base (all)	F	0.938	0.987	0.937	0.925	0.995	0.993	0.984	0.987	0.991	0.980	0.935	0.966
3	XLM-RoBERTa-large (ru)	F	0.964	0.967	0.995	0.981	0.959	0.983	0.971	0.966	0.993	0.981	0.934	0.954
4	mDeBERTa-v3-base (es)	F	0.936	0.985	0.979	0.971	0.929	0.993	0.985	0.980	0.964	0.973	0.925	0.952
5	BERT-base-multilingual-cased (all)	F	0.906	0.986	0.932	0.898	0.995	0.991	0.945	0.975	0.982	0.950	0.871	0.951
6	XLM-RoBERTa-large (es)	F	0.965	0.971	0.984	0.974	0.839	0.986	0.979	0.968	0.971	0.972	0.957	0.946
7	mDeBERTa-v3-base (ru)	F	0.963	0.923	0.993	0.958	0.882	0.941	0.953	0.886	0.988	0.988	0.893	0.933
8	BERT-base-multilingual-cased (es)	F	0.877	0.980	0.938	0.940	0.698	0.987	0.984	0.970	0.925	0.921	0.947	0.917
9	mGPT (all)	F	0.960	0.944	0.916	0.977	0.996	0.986	0.962	0.974	0.988	0.964	0.710	0.911
10	mGPT (ru)	F	0.978	0.944	0.944	0.972	0.757	0.960	0.939	0.922	0.987	0.975	0.741	0.906
11	mGPT (es)	F	0.930	0.970	0.901	0.976	0.939	0.991	0.989	0.980	0.964	0.957	0.904	0.906
12	BERT-base-multilingual-cased (ru)	F	0.933	0.915	0.943	0.893	0.714	0.890	0.926	0.902	0.978	0.953	0.838	0.890
13	XLM-RoBERTa-large (en)	F	0.836	0.976	0.934	0.948	0.998	0.909	0.943	0.890	0.926	0.898	0.909	0.874
14	OPT-IML-Max-1.3B (all)	F	0.486	0.957	0.933	0.903	0.996	0.981	0.929	0.982	0.913	0.807	0.393	0.858
15	BERT-base-multilingual-cased (en)	F	0.725	0.967	0.891	0.840	0.996	0.898	0.900	0.891	0.823	0.885	0.740	0.856
16	MFD	S	0.727	0.766	0.703	0.899	0.955	0.945	0.901	0.938	0.837	0.858	0.761	0.833
17	OPT-IML-Max-1.3B (es)	F	0.685	0.948	0.868	0.906	0.896	0.982	0.958	0.981	0.616	0.641	0.450	0.812
18	mDeBERTa-v3-base (en)	F	0.602	0.907	0.869	0.765	0.998	0.938	0.856	0.780	0.703	0.835	0.547	0.802
19	DetectLLM-LRR	S	0.659	0.931	0.886	0.881	0.939	0.911	0.938	0.887	0.734	0.764	0.663	0.791
20	mGPT (en)	F	0.511	0.884	0.863	0.860	0.997	0.936	0.884	0.899	0.795	0.796	0.596	0.782
21	LLMDeviation	S	0.615	0.961	0.905	0.876	0.965	0.911	0.958	0.896	0.684	0.775	0.683	0.765
22	GLTR Test 2	S	0.599	0.893	0.874	0.850	0.943	0.912	0.939	0.914	0.672	0.791	0.702	0.759
23	LogRank	S	0.596	0.965	0.917	0.873	0.972	0.916	0.960	0.904	0.669	0.762	0.689	0.758
24	LogLikelihood	S	0.581	0.963	0.912	0.853	0.971	0.907	0.960	0.900	0.634	0.738	0.685	0.743
25	OPT-IML-Max-1.3B (ru)	F	0.518	0.562	0.865	0.725	0.714	0.742	0.703	0.704	0.918	0.815	0.603	0.696
26	Rank	S	0.558	0.869	0.729	0.767	0.829	0.769	0.875	0.736	0.598	0.511	0.606	0.683
27	OPT-IML-Max-1.3B (en)	F	0.353	0.832	0.770	0.788	0.997	0.758	0.663	0.785	0.480	0.497	0.373	0.674
28	ChatGPT-Detector-RoBERTa-Chinese	P	0.734	0.819	0.547	0.709	0.564	0.536	0.854	0.618	0.615	0.509	0.918	0.638
29	DetectLLM-NPR	S	0.527	0.758	0.685	0.628	0.690	0.687	0.711	0.689	0.668	0.717	0.506	0.636
30	DetectGPT	S	0.452	0.765	0.792	0.584	0.685	0.630	0.640	0.652	0.616	0.635	0.503	0.593
31	Longformer Detector	P	0.532	0.677	0.513	0.621	0.974	0.782	0.726	0.672	0.476	0.472	0.551	0.588
32	ChatGPT-Detector-RoBERTa	P	0.513	0.502	0.427	0.736	0.855	0.709	0.681	0.640	0.489	0.461	0.628	0.570
33	RoBERTa-large-OpenAI-Detector	P	0.688	0.479	0.471	0.448	0.927	0.564	0.412	0.730	0.614	0.585	0.475	0.544
34	RoBERTa-base-OpenAI-Detector	P	0.637	0.510	0.574	0.480	0.952	0.518	0.440	0.504	0.707	0.455	0.430	0.542
35	ruRoBERTa-ruatd-binary	P	0.577	0.505	0.501	0.484	0.611	0.496	0.507	0.495	0.619	0.591	0.505	0.535
36	RoBERTa-base-autextification-Detection	P	0.500	0.478	0.501	0.509	0.500	0.533	0.351	0.512	0.500	0.500	0.500	0.489
37	Entropy	S	0.563	0.117	0.164	0.417	0.303	0.373	0.151	0.346	0.584	0.425	0.539	0.419

Table 14: Per-language detection performance (AUC ROC) comparison of all MGT detection methods using original MULTITuDE test data (human and machine samples of the 3 selected best LLM generators – gpt-4, gpt-3.5-turbo, vicuna). F refers to fine-tuned, P to pre-trained, and S to statistical MGT detection methods category. MGT detection methods achieving above threshold (0.8) performance are highlighted.

AO Method	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	→ Average
	Test Language [mean (±confidence interval)]											
original	0.7696	0.8832	0.8997	0.8843	0.8788	0.9098	0.8797	0.8766	0.8653	0.8661	0.7470	0.8600
	(±0.13)	(±0.10)	(±0.09)	(±0.08)	(±0.16)	(±0.11)	(±0.09)	(±0.10)	(±0.12)	(±0.11)	(±0.09)	
m2m100-1.2B	0.7875	0.7986	0.8721	0.8476	0.8232	0.8635	0.7970	0.8271	0.8264	0.8351	0.7114	0.8172
	(±0.09)	(±0.04)	(±0.03)	(±0.04)	(±0.06)	(±0.04)	(±0.05)	(±0.04)	(±0.07)	(±0.05)	(±0.08)	
nllb-200-distilled-1.3B	0.7702	0.7958	0.8657	0.8584	0.8431	0.8731	0.8163	0.8376	0.8290	0.8269	0.6972	0.8194
	(±0.09)	(±0.06)	(±0.04)	(±0.05)	(±0.08)	(±0.06)	(±0.07)	(±0.05)	(±0.07)	(±0.07)	(±0.09)	
Pegasus-paraphrase	0.7904	0.6742	0.7820	0.7486	0.8876	0.7562	0.7205	0.6580	0.8620	0.9105	0.7576	0.7771
	(±0.10)	(±0.07)	(±0.05)	(±0.06)	(±0.07)	(±0.06)	(±0.06)	(±0.06)	(±0.08)	(±0.08)	(±0.09)	
DIPPER	0.7729	0.8976	0.9157	0.9148	0.8925	0.9163	0.8908	0.8894	0.8833	0.9083	0.7346	0.8742
	(±0.10)	(±0.05)	(±0.04)	(±0.06)	(±0.06)	(±0.05)	(±0.07)	(±0.06)	(±0.09)	(±0.07)	(±0.08)	
ChatGPT	0.7932	0.8582	0.8793	0.8779	0.9192	0.9108	0.8588	0.8871	0.8571	0.8588	0.7348	0.8577
	(±0.09)	(±0.04)	(±0.04)	(±0.04)	(±0.05)	(±0.04)	(±0.04)	(±0.04)	(±0.07)	(±0.05)	(±0.09)	
GPTZzzs	0.7664	0.8720	0.8966	0.8796	0.8328	0.8989	0.8665	0.8611	0.8643	0.8657	0.7417	0.8496
	(±0.13)	(±0.13)	(±0.11)	(±0.09)	(±0.17)	(±0.12)	(±0.12)	(±0.12)	(±0.13)	(±0.12)	(±0.09)	
GPTZeroBypasser	0.5593	0.5623	0.7840	0.6962	0.5560	0.6619	0.5471	0.5880	0.6524	0.6683	0.6609	0.6306
	(±0.10)	(±0.05)	(±0.04)	(±0.04)	(±0.07)	(±0.04)	(±0.05)	(±0.04)	(±0.07)	(±0.06)	(±0.08)	
HomoglyphAttack	0.5955	0.5562	0.7756	0.6680	0.5985	0.6646	0.5359	0.5839	0.6458	0.6715	0.7003	0.6360
	(±0.09)	(±0.05)	(±0.03)	(±0.04)	(±0.06)	(±0.04)	(±0.04)	(±0.04)	(±0.07)	(±0.06)	(±0.08)	
ALISON	0.7645	0.8793	0.8967	0.8811	0.8769	0.9064	0.8747	0.8746	0.8616	0.8602	0.7476	0.8567
	(±0.09)	(±0.04)	(±0.04)	(±0.04)	(±0.05)	(±0.04)	(±0.06)	(±0.04)	(±0.08)	(±0.07)	(±0.10)	
DFTFooler	0.7673	0.8406	0.8766	0.8013	0.8666	0.8773	0.7984	0.8213	0.8358	0.8570	0.7427	0.8259
	(±0.12)	(±0.12)	(±0.09)	(±0.11)	(±0.06)	(±0.12)	(±0.12)	(±0.12)	(±0.10)	(±0.08)	(±0.10)	
↓ Average	0.7397	0.7834	0.8585	0.8234	0.8159	0.8399	0.7805	0.7913	0.8166	0.8298	0.7251	

Table 15: AUC ROC per-language mean value along with 95% confidence interval error bounds of finetuned MGT detection methods. A higher value indicates better detection performance.

Table 16

AO Method	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	→ Average
	Test Language [mean (±confidence interval)]											
original	0.6058 (±0.21)	0.5872 (±0.19)	0.5494 (±0.17)	0.5849 (±0.18)	0.7259 (±0.20)	0.5791 (±0.24)	0.6011 (±0.11)	0.5851 (±0.22)	0.5403 (±0.26)	0.5061 (±0.19)	0.5642 (±0.05)	0.5845
m2m100-1.2B	0.6190 (±0.07)	0.5606 (±0.07)	0.5490 (±0.10)	0.5881 (±0.06)	0.6593 (±0.23)	0.5733 (±0.12)	0.6032 (±0.11)	0.5766 (±0.10)	0.5317 (±0.06)	0.5019 (±0.05)	0.6191 (±0.05)	0.5802
nllb-200-distilled-1.3B	0.6175 (±0.08)	0.5645 (±0.10)	0.5644 (±0.09)	0.5895 (±0.08)	0.6810 (±0.23)	0.5806 (±0.10)	0.5968 (±0.15)	0.5776 (±0.08)	0.5451 (±0.10)	0.4955 (±0.04)	0.6282 (±0.13)	0.5855
Pegasus-paraphrase	0.5027 (±0.09)	0.6986 (±0.12)	0.6258 (±0.16)	0.7059 (±0.10)	0.6843 (±0.23)	0.6490 (±0.12)	0.7079 (±0.15)	0.6343 (±0.08)	0.5664 (±0.15)	0.6266 (±0.12)	0.6427 (±0.14)	0.6404
DIPPER	0.6373 (±0.07)	0.7657 (±0.09)	0.7213 (±0.11)	0.7271 (±0.13)	0.7037 (±0.05)	0.7405 (±0.07)	0.7696 (±0.13)	0.7600 (±0.09)	0.7081 (±0.08)	0.7265 (±0.03)	0.6126 (±0.05)	0.7157
ChatGPT	0.6150 (±0.27)	0.5463 (±0.23)	0.5189 (±0.27)	0.5700 (±0.24)	0.7226 (±0.20)	0.5447 (±0.24)	0.5544 (±0.26)	0.5366 (±0.25)	0.5540 (±0.16)	0.5026 (±0.26)	0.5651 (±0.26)	0.5664
GPTZzzs	0.6001 (±0.30)	0.5764 (±0.36)	0.5466 (±0.30)	0.5818 (±0.31)	0.5400 (±0.35)	0.5608 (±0.34)	0.5942 (±0.27)	0.5630 (±0.31)	0.5386 (±0.24)	0.5049 (±0.25)	0.5544 (±0.14)	0.5601
GPTZeroBypasser	0.4257 (±0.08)	0.5705 (±0.08)	0.4310 (±0.08)	0.4552 (±0.06)	0.4316 (±0.21)	0.4919 (±0.07)	0.5543 (±0.12)	0.4533 (±0.06)	0.5006 (±0.08)	0.4517 (±0.03)	0.5017 (±0.05)	0.4789
HomoglyphAttack	0.3287 (±0.07)	0.5825 (±0.09)	0.4844 (±0.08)	0.4737 (±0.07)	0.3595 (±0.18)	0.4998 (±0.09)	0.5343 (±0.13)	0.5089 (±0.08)	0.3958 (±0.08)	0.3980 (±0.03)	0.5113 (±0.05)	0.4615
ALISON	0.5728 (±0.09)	0.5702 (±0.11)	0.5241 (±0.10)	0.5506 (±0.09)	0.6221 (±0.17)	0.5334 (±0.08)	0.5756 (±0.13)	0.5441 (±0.07)	0.5020 (±0.12)	0.4708 (±0.03)	0.5490 (±0.11)	0.5468
DFTFooler	0.5995 (±0.27)	0.5773 (±0.21)	0.5334 (±0.28)	0.5505 (±0.21)	0.5700 (±0.23)	0.5522 (±0.21)	0.5786 (±0.23)	0.5402 (±0.24)	0.5296 (±0.22)	0.4970 (±0.28)	0.5509 (±0.19)	0.5527
↓ Average	0.5567	0.6000	0.5499	0.5797	0.6091	0.5732	0.6064	0.5709	0.5375	0.5165	0.5727	

Table 16: AUC ROC per-language mean value along with 95% confidence interval error bounds of pretrained MGT detection methods. A higher value indicates better detection performance.

Table 17

AO Method	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	→ Average
	Test Language [mean (±confidence interval)]											
original	0.4016 (±0.23)	0.7323 (±0.27)	0.7015 (±0.27)	0.6212 (±0.25)	0.7704 (±0.27)	0.6837 (±0.26)	0.6698 (±0.26)	0.6629 (±0.26)	0.4793 (±0.24)	0.4904 (±0.25)	0.5320 (±0.15)	0.6132
m2m100-1.2B	0.4290 (±0.11)	0.6965 (±0.11)	0.7139 (±0.11)	0.5449 (±0.05)	0.6403 (±0.10)	0.6246 (±0.07)	0.6353 (±0.08)	0.6272 (±0.06)	0.4453 (±0.10)	0.4916 (±0.06)	0.5102 (±0.09)	0.5781
nllb-200-distilled-1.3B	0.4209 (±0.11)	0.6625 (±0.09)	0.6721 (±0.12)	0.5834 (±0.06)	0.7240 (±0.06)	0.6493 (±0.05)	0.6620 (±0.05)	0.6344 (±0.05)	0.4581 (±0.11)	0.4809 (±0.05)	0.4987 (±0.11)	0.5860
Pegasus-paraphrase	0.6235 (±0.11)	0.4384 (±0.06)	0.3884 (±0.08)	0.3613 (±0.06)	0.7208 (±0.06)	0.3753 (±0.05)	0.3781 (±0.07)	0.3645 (±0.05)	0.6362 (±0.11)	0.5748 (±0.06)	0.6806 (±0.11)	0.5038
DIPPER	0.4482 (±0.11)	0.6695 (±0.08)	0.5411 (±0.12)	0.5152 (±0.19)	0.7532 (±0.14)	0.5182 (±0.16)	0.6084 (±0.13)	0.5215 (±0.20)	0.3924 (±0.16)	0.3471 (±0.09)	0.5180 (±0.10)	0.5303
ChatGPT	0.4714 (±0.19)	0.7422 (±0.05)	0.7317 (±0.16)	0.6521 (±0.14)	0.7339 (±0.07)	0.7000 (±0.16)	0.7253 (±0.10)	0.6930 (±0.16)	0.4978 (±0.23)	0.5459 (±0.27)	0.5576 (±0.17)	0.6410
GPTZzzs	0.3945 (±0.16)	0.6899 (±0.28)	0.6873 (±0.26)	0.5959 (±0.23)	0.4414 (±0.28)	0.6114 (±0.26)	0.6242 (±0.26)	0.5982 (±0.24)	0.4760 (±0.21)	0.4896 (±0.25)	0.5148 (±0.11)	0.5567
GPTZeroBypasser	0.3384 (±0.11)	0.7223 (±0.11)	0.7172 (±0.11)	0.5908 (±0.06)	0.6557 (±0.21)	0.6245 (±0.10)	0.6683 (±0.07)	0.6475 (±0.08)	0.2956 (±0.09)	0.2752 (±0.05)	0.4710 (±0.10)	0.5460
HomoglyphAttack	0.2007 (±0.10)	0.2646 (±0.12)	0.2709 (±0.11)	0.2533 (±0.05)	0.2846 (±0.12)	0.2596 (±0.07)	0.2666 (±0.08)	0.2740 (±0.06)	0.2156 (±0.09)	0.2097 (±0.05)	0.4248 (±0.09)	0.2658
ALISON	0.3977 (±0.08)	0.7212 (±0.14)	0.6980 (±0.14)	0.6113 (±0.06)	0.7473 (±0.10)	0.6710 (±0.07)	0.6623 (±0.08)	0.6502 (±0.08)	0.4670 (±0.08)	0.4739 (±0.04)	0.5304 (±0.06)	0.6028
DFTFooler	0.3953 (±0.13)	0.5848 (±0.22)	0.6132 (±0.28)	0.4304 (±0.25)	0.5682 (±0.06)	0.4908 (±0.23)	0.4791 (±0.23)	0.4721 (±0.24)	0.3245 (±0.15)	0.4331 (±0.16)	0.5146 (±0.14)	0.4824
↓ Average	0.4110	0.6295	0.6123	0.5236	0.6400	0.5644	0.5799	0.5587	0.4262	0.4375	0.5230	

Table 17: AUC ROC per-language mean value along with 95% confidence interval error bounds of statistical MGT detection methods. A higher value indicates better detection performance.

Table 18

AO Method	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	→ Average
	Test Language [mean (±confidence interval)]											
m2m100-1.2B	2.65% (±2.23%)	-9.39% (±5.99%)	-3.09% (±1.96%)	-4.27% (±2.58%)	-7.22% (±4.07%)	-5.42% (±3.01%)	-9.88% (±4.47%)	-5.84% (±3.20%)	-4.63% (±2.17%)	-3.69% (±2.25%)	-5.10% (±3.51%)	-5.08%
nllb-200-distilled-1.3B	-0.49% (±2.56%)	-10.40% (±4.24%)	-3.96% (±3.80%)	-3.13% (±1.76%)	-4.60% (±2.43%)	-4.49% (±2.57%)	-7.66% (±3.13%)	-4.82% (±2.20%)	-4.86% (±2.80%)	-5.39% (±4.06%)	-7.25% (±3.31%)	-5.18%
Pegasus-paraphrase	1.51% (±10.10%)	-24.87% (±11.71%)	-13.82% (±8.30%)	-16.21% (±10.41%)	1.13% (±2.49%)	-18.22% (±11.15%)	-19.67% (±11.27%)	-26.19% (±12.22%)	-1.49% (±7.64%)	4.96% (±7.37%)	0.75% (±9.47%)	-10.19%
DIPPER	2.38% (±8.83%)	2.38% (±5.51%)	1.88% (±3.31%)	3.76% (±4.19%)	1.76% (±2.44%)	0.91% (±3.64%)	1.55% (±4.79%)	1.91% (±2.81%)	2.40% (±3.41%)	5.46% (±7.65%)	-1.53%	2.08%
ChatGPT	3.74% (±4.13%)	-3.26% (±3.06%)	-2.50% (±2.92%)	-0.89% (±1.68%)	4.98% (±2.12%)	-0.10% (±1.70%)	-2.87% (±3.72%)	1.12% (±1.60%)	-1.33% (±2.32%)	-1.26% (±2.49%)	-2.67% (±3.47%)	-0.46%
GPTZzzs	-0.57% (±0.36%)	-1.37% (±0.78%)	-0.36% (±0.21%)	-0.56% (±0.30%)	-5.56% (±3.33%)	-1.28% (±0.92%)	-1.64% (±0.86%)	-1.90% (±1.16%)	-0.14% (±0.09%)	-0.05% (±0.03%)	-0.87% (±0.46%)	-1.30%
GPTZeroBypasser	-31.12% (±12.80%)	-37.22% (±13.99%)	-13.15% (±12.45%)	-21.36% (±10.26%)	-40.25% (±16.55%)	-28.32% (±12.00%)	-38.86% (±13.23%)	-33.63% (±13.22%)	-26.50% (±12.92%)	-24.23% (±12.02%)	-12.64% (±4.29%)	-27.93%
HomoglyphAttack	-26.33% (±9.48%)	-37.14% (±10.84%)	-14.12% (±8.98%)	-24.64% (±8.75%)	-35.18% (±15.47%)	-27.71% (±10.46%)	-39.82% (±9.51%)	-33.71% (±10.66%)	-27.75% (±10.01%)	-24.05% (±9.26%)	-7.28% (±2.61%)	-27.07%
ALISON	-0.79% (±0.69%)	-0.37% (±0.95%)	-0.38% (±0.62%)	-0.34% (±0.70%)	-0.17% (±0.85%)	-0.39% (±0.72%)	-0.64% (±0.65%)	-0.21% (±0.93%)	-0.52% (±0.57%)	-0.83% (±0.95%)	0.02% (±0.51%)	-0.42%
DFTFooler	-0.43% (±0.34%)	-4.97% (±2.79%)	-2.73% (±1.95%)	-9.69% (±4.74%)	-1.43% (±1.42%)	-3.65% (±3.42%)	-9.74% (±5.67%)	-6.60% (±4.00%)	-4.30% (±3.94%)	-1.28% (±1.40%)	-0.75% (±0.49%)	-4.14%
↓ Average	-4.94%	-12.66%	-5.22%	-7.73%	-8.65%	-8.87%	-12.92%	-10.99%	-6.91%	-5.04%	-3.73%	

Table 18: AUC ROC drop of finetuned MGT detection methods. Per-language mean value is reported along with 95% confidence interval error bounds. A higher drop (i.e., a lower value – a darker color) indicates that the detection ability is affected by the corresponding AO method in a greater scale.

Table 19

AO Method	Test Language [mean ±confidence interval]											→ Average
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
m2m100-1.2B	2.51% (±9.04%)	-4.92% (±6.36%)	-0.26% (±3.72%)	0.28% (±4.54%)	-10.91% (±13.18%)	-1.12% (±3.06%)	-0.60% (±6.69%)	-1.45% (±3.62%)	-2.04% (±4.10%)	-0.99% (±4.10%)	10.28% (±24.37%)	-0.84%
nllb-200-distilled-1.3B	1.86% (±6.45%)	-4.42% (±13.28%)	0.77% (±14.72%)	0.21% (±9.27%)	-7.97% (±12.19%)	-0.26% (±8.27%)	-1.38% (±9.46%)	-1.26% (±6.30%)	-1.36% (±18.85%)	-2.93% (±21.19%)	12.11% (±26.97%)	-0.42%
Pegasus-paraphrase	-18.71% (±40.93%)	17.78% (±28.60%)	9.65% (±37.18%)	19.99% (±32.35%)	-7.19% (±11.87%)	13.94% (±38.86%)	18.13% (±33.19%)	7.83% (±38.48%)	9.06% (±47.50%)	21.98% (±49.37%)	15.64% (±36.74%)	9.83%
DIPPER	3.62% (±39.63%)	28.49% (±33.96%)	30.00% (±45.69%)	21.63% (±35.07%)	-3.58% (±6.58%)	27.45% (±41.35%)	27.91% (±40.94%)	27.88% (±38.16%)	31.05% (±25.13%)	41.80% (±48.43%)	9.56% (±47.02%)	22.35%
ChatGPT	1.44% (±7.19%)	-7.51% (±7.09%)	-6.02% (±8.10%)	-2.94% (±8.24%)	-0.04% (±4.53%)	-5.45% (±8.31%)	-8.05% (±4.82%)	-7.92% (±7.87%)	1.67% (±9.62%)	-0.58% (±5.23%)	-0.50% (±11.96%)	-3.26%
GPTZzzs	-0.98% (±1.80%)	-1.71% (±2.33%)	-0.54% (±1.70%)	-0.44% (±1.97%)	-22.46% (±26.49%)	-2.59% (±3.98%)	-0.93% (±3.85%)	-3.31% (±4.90%)	-0.32% (±0.90%)	-0.21% (±0.54%)	-1.71% (±3.15%)	-3.20%
GPTZeroBypasser	-31.85% (±44.71%)	-2.43% (±60.72%)	-22.35% (±49.26%)	-23.93% (±49.49%)	-44.88% (±40.66%)	-20.22% (±51.41%)	-5.61% (±50.66%)	-25.98% (±45.02%)	-1.96% (±47.99%)	-11.32% (±48.09%)	-10.52% (±26.95%)	-18.28%
HomoglyphAttack	-43.28% (±36.40%)	0.65% (±29.90%)	-10.42% (±31.17%)	-17.05% (±33.53%)	-46.49% (±35.83%)	-14.80% (±37.36%)	-5.96% (±30.54%)	-12.81% (±35.56%)	-18.51% (±60.36%)	-21.03% (±37.78%)	-9.38% (±5.52%)	-18.10%
ALISON	-5.26% (±5.12%)	-2.61% (±3.34%)	-5.16% (±7.76%)	-5.48% (±6.69%)	-13.49% (±23.78%)	-8.36% (±13.04%)	-3.57% (±5.85%)	-7.03% (±10.85%)	-6.62% (±5.69%)	-6.98% (±7.58%)	-2.61% (±3.30%)	-6.11%
DFTFooler	-1.05% (±1.49%)	-1.62% (±7.17%)	-3.52% (±8.53%)	-4.73% (±24.28%)	-17.45% (±16.19%)	-3.24% (±13.86%)	-2.90% (±15.28%)	-6.65% (±16.68%)	-2.15% (±2.24%)	-1.79% (±1.39%)	-2.38% (±2.50%)	-4.32%
↓ Average	-9.17%	2.17%	-0.78%	-1.25%	-17.45%	-1.46%	1.70%	-3.07%	0.88%	1.80%	2.05%	

Table 19: AUC ROC drop of pretrained MGT detection methods. Per-language mean value is reported along with 95% confidence interval error bounds. A higher drop (i.e., a lower value – a darker color) indicates that the detection ability is affected by the corresponding AO method in a greater scale.

Table 20

AO Method	Test Language [mean ±confidence interval]											→ Average
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
m2m100-1.2B	6.97% (±6.23%)	-2.02% (±10.05%)	1.60% (±4.50%)	-11.59% (±11.19%)	-11.25% (±23.56%)	-7.55% (±9.28%)	-3.66% (±7.77%)	-4.81% (±5.48%)	-8.58% (±6.52%)	0.39% (±2.99%)	-5.04% (±7.16%)	-4.14%
nllb-200-distilled-1.3B	4.91% (±5.85%)	-3.88% (±19.31%)	-1.71% (±10.60%)	-5.68% (±7.97%)	-2.94% (±13.17%)	-4.09% (±8.27%)	-0.17% (±5.95%)	-3.78% (±5.40%)	-5.78% (±5.69%)	-2.13% (±3.92%)	-6.72% (±10.18%)	-2.91%
Pegasus-paraphrase	67.02% (±42.20%)	-21.23% (±64.83%)	-26.60% (±70.02%)	-40.46% (±42.26%)	-2.57% (±16.73%)	-41.69% (±39.83%)	-34.14% (±49.40%)	-42.13% (±41.54%)	37.60% (±34.57%)	14.42% (±26.13%)	32.96% (±29.22%)	-5.17%
DIPPER	5.66% (±23.34%)	-0.02% (±29.41%)	-10.70% (±47.52%)	-16.25% (±23.44%)	0.93% (±14.13%)	-21.83% (±28.24%)	-3.64% (±30.60%)	-19.30% (±28.77%)	-27.03% (±31.74%)	-35.43% (±45.69%)	-6.06% (±16.21%)	-12.15%
ChatGPT	21.70% (±13.69%)	0.08% (±4.58%)	2.58% (±7.10%)	4.85% (±3.32%)	-4.13% (±2.72%)	2.30% (±2.58%)	6.36% (±9.85%)	4.24% (±5.17%)	4.38% (±4.00%)	12.26% (±8.29%)	6.28% (±6.74%)	5.54%
GPTZzzs	-2.31% (±2.26%)	-4.13% (±8.55%)	-1.62% (±2.59%)	-4.10% (±5.27%)	-33.10% (±42.59%)	-10.37% (±12.56%)	-5.74% (±7.34%)	-9.63% (±10.60%)	-0.81% (±0.81%)	-0.20% (±0.81%)	-3.91% (±3.16%)	-6.90%
GPTZeroBypasser	-16.43% (±29.57%)	-9.32% (±34.10%)	-5.42% (±32.17%)	-5.34% (±35.17%)	-19.48% (±32.81%)	-10.18% (±35.62%)	-4.40% (±35.16%)	-4.62% (±33.66%)	-42.54% (±34.88%)	-48.16% (±42.89%)	-12.05% (±15.55%)	-16.18%
HomoglyphAttack	-60.98% (±36.79%)	-40.91% (±80.86%)	-42.57% (±73.17%)	-55.91% (±45.63%)	-48.46% (±59.29%)	-56.31% (±47.36%)	-48.74% (±56.66%)	-54.03% (±47.55%)	-62.52% (±38.86%)	-61.80% (±42.87%)	-23.73% (±18.44%)	-50.54%
ALISON	-1.48% (±2.17%)	-0.84% (±2.63%)	-0.33% (±1.07%)	-1.56% (±1.85%)	-2.25% (±3.34%)	-1.78% (±1.68%)	-0.87% (±1.44%)	-1.92% (±2.10%)	-3.28% (±3.22%)	-3.81% (±3.65%)	-0.52% (±0.80%)	-1.69%
DFTFooler	-2.10% (±2.41%)	-13.20% (±25.23%)	-9.35% (±18.82%)	-29.95% (±31.24%)	-20.72% (±28.41%)	-26.54% (±27.10%)	-23.08% (±30.09%)	-27.51% (±32.54%)	-36.28% (±23.44%)	-12.89% (±10.20%)	-4.03% (±3.48%)	-18.69%
↓ Average	2.30%	-9.55%	-9.41%	-16.60%	-14.40%	-17.80%	-11.81%	-16.35%	-14.48%	-13.74%	-2.28%	

Table 20: AUC ROC drop of statistical MGT detection methods. Per-language mean value is reported along with 95% confidence interval error bounds. A higher drop (i.e., a lower value – a darker color) indicates that the detection ability is affected by the corresponding AO method in a greater scale.

Table 21

AO Method	Test Language [mean ±confidence interval]											→ Average
	ar	ca	cs	de	en	es	nl	pt	ru	uk	zh	
m2m100-1.2B	-1.36% (n.s.)	1.60% (n.s.)	2.11% (±1.50%)	1.60% (n.s.)	1.10% (±0.50%)	0.50% (±0.50%)	2.18% (±1.50%)	1.43% (±0.50%)	0.72% (n.s.)	0.97% (n.s.)	-1.20% (n.s.)	0.88%
nllb-200-distilled-1.3B	1.34% (n.s.)	1.86% (±2.00%)	3.31% (±1.50%)	0.72% (n.s.)	0.90% (±0.50%)	0.48% (±0.50%)	0.90% (n.s.)	0.94% (±1.00%)	1.38% (±0.50%)	1.85% (±1.00%)	-1.77% (n.s.)	1.08%
Pegasus-paraphrase	-0.31% (n.s.)	7.43% (±5.50%)	7.21% (±4.50%)	8.50% (±4.50%)	1.16% (±0.50%)	3.55% (±2.50%)	6.61% (±3.50%)	8.11% (±4.50%)	1.12% (±0.50%)	1.82% (±1.00%)	3.57% (n.s.)	4.43%
DIPPER	1.21% (n.s.)	2.26% (±1.50%)	3.23% (±2.00%)	2.57% (±1.50%)	0.78% (±0.50%)	1.02% (±0.00%)	3.26% (±1.00%)	2.07% (±1.00%)	1.70% (±0.50%)	2.33% (±1.50%)	0.13% (n.s.)	1.87%
ChatGPT	2.09% (n.s.)	1.05% (n.s.)	2.18% (±1.50%)	1.27% (n.s.)	0.52% (±0.50%)	0.40% (±0.50%)	1.41% (±1.50%)	0.88% (±1.00%)	1.20% (±0.50%)	1.63% (±1.50%)	0.39% (n.s.)	1.18%
GPTZzzs	0.19% (±0.00%)	0.81% (±0.50%)	0.24% (±0.00%)	0.34% (±0.50%)	1.04% (±1.00%)	0.54% (±0.50%)	0.38% (±0.50%)	0.98% (±0.50%)	0.01% (n.s.)	0.00% (n.s.)	0.33% (±0.50%)	0.44%
GPTZeroBypasser	11.80% (±7.00%)	8.07% (±6.00%)	8.66% (±3.50%)	12.57% (±5.50%)	4.78% (±3.00%)	5.76% (±3.00%)	9.57% (±5.50%)	8.44% (±4.50%)	6.40% (±3.00%)	8.50% (±3.50%)	5.78% (n.s.)	8.21%
HomoglyphAttack	11.77% (±9.00%)	6.86% (±6.00%)	6.73% (±3.50%)	14.47% (±7.50%)	2.64% (±1.50%)	3.74% (±3.00%)	8.33% (±6.50%)	6.82% (±5.00%)	5.34% (±3.50%)	7.37% (±4.50%)	5.03% (±4.50%)	7.19%
ALISON	0.69% (n.s.)	0.68% (n.s.)	0.17% (n.s.)	0.61% (n.s.)	0.17% (n.s.)	0.10% (n.s.)	0.49% (n.s.)	0.30% (n.s.)	0.40% (±0.50%)	0.70% (n.s.)	0.73% (±0.50%)	0.46%
DFTFooler	0.18% (±0.00%)	3.96% (±3.00%)	2.05% (±1.00%)	5.38% (±3.50%)	0.72% (±0.50%)	1.17% (±1.00%)	4.14% (±2.50%)	2.54% (±1.50%)	0.50% (±0.50%)	0.83% (±0.50%)	0.37% (±0.50%)	1.99%
↓ Average	2.76%	3.46%	3.59%	4.80%	1.38%	1.73%	3.73%	3.25%	1.88%	2.60%	1.33%	

Table 21: Per-language AUC ROC drop difference between originally and adversarially trained MGT detection methods. Per-language mean difference value is reported for each AO method along with 95% confidence interval error bounds. The results that are not statistically significant are marked by (n.s.). A higher number represents more robust MGT detection after adversarial retraining.