LLM-as-a-Coauthor: Can Mixed Human-Written and Machine-Generated Text Be Detected?

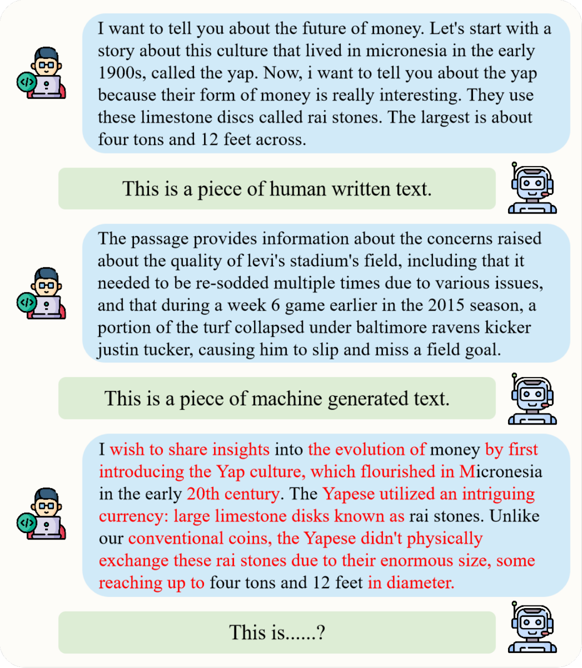
Qihui Zhang[1*†], Chujie Gao[1*†], Dongping Chen[2*], Yue Huang[3], Yixin Huang[4], Zhenyang Sun[1†], Shilin Zhang[1†], Weiye Li[1†], Zhengyan Fu[1†], Yao Wan[2], Lichao Sun[1‡]

[1]**Lehigh University**, [2]**Huazhong University of Science and Technology**, [3]**University of Notre Dame**, [4]**Institut Polytechnique de Paris**

{maskhui1003, gaochujie1107, dongpingchen0612, james.lichao.sun}@gmail.com

Abstract

With the rapid development and widespread application of Large Language Models (LLMs), the use of Machine-Generated Text (MGT) has become increasingly common, bringing with it potential risks, especially in terms of quality and integrity in fields like news, education, and science. Current research mainly focuses on purely MGT detection without adequately addressing mixed scenarios, including AI-revised Human-Written Text (HWT) or human-revised MGT. To tackle this challenge, we define *mixtext*, a form of mixed text involving both AI and human-generated content. Then, we introduce MIXSET, the first dataset dedicated to studying these mixtext scenarios. Leveraging MIXSET, we executed comprehensive experiments to assess the efficacy of prevalent MGT detectors in handling *mixtext* situations, evaluating their performance in terms of effectiveness, robustness, and generalization. Our findings reveal that existing detectors struggle to identify *mixtext*, particularly in dealing with subtle modifications and style adaptability. This research underscores the urgent need for more fine-grain detectors tailored for *mixtext*, offering valuable insights for future research. Code and Models are available at https://github.com/Dongping-Chen/MixSet.



I want to tell you about the future of money. Let's start with a story about this culture that lived in micronesia in the early 1900s, called the yap. Now, i want to tell you about the yap because their form of money is really interesting. They use these limestone discs called rai stones. The largest is about four tons and 12 feet across.

This is a piece of human written text.

The passage provides information about the concerns raised about the quality of levi's stadium's field, including that it needed to be re-sodded multiple times due to various issues, and that during a week 6 game earlier in the 2015 season, a portion of the turf collapsed under baltimore ravens kicker justin tucker, causing him to slip and miss a field goal.

This is a piece of machine generated text.

I wish to share insights into the evolution of money by first introducing the Yap culture, which flourished in Micronesia in the early 20th century. The Yapese utilized an intriguing currency: large limestone disks known as rai stones. Unlike our conventional coins, the Yapese didn't physically exchange these rai stones due to their enormous size, some reaching up to four tons and 12 feet in diameter.

This is......?

Figure 1: Three kinds of text: Machine Generative Text (MGT), Human Written Text (HWT), and mixtext. The text come from users 🧑‍💻 is classified by detectors 🤖. The text in red is the HWT polished by LLMs.

1 Introduction

The remarkable advancement of Large Language Models (LLM) has sparked global discussions on the effective utilization of AI assistants (OpenAI, 2022, 2023b). Given that LLMs can correctly follow human instructions and produce useful texts efficiently, more and more people prefer to integrate these powerful tools into their workflow by revising Machine Generated Text (MGT) or using LLMs to polish their Human Written Text (HWT), such as fact-checking revising in journalism (Guerra, 2023) and enhancing storytelling in the game industry [1].

Despite its various usages, The application of LLMs also causes the potential risk of MGT usage, raising public concerns on various misuse, as seen in the undermining of journalistic integrity and quality (Christian, 2023), reproducing and amplifying biases (Sison et al., 2023), plagiarism among students (Heavenarchive, 2023), and leading disruptions in trust towards scientific knowledge (Else, 2023). The misuse of machine-generated text has been a serious concern that is also raised by experts in different fields of work [2].

[*] Equal contribution.

[†] Visiting Students at LAIR Lab, Lehigh University.

[‡] Lichao Sun is the corresponding author.

[1] https://aicontentfy.com/en/blog/chatgpt-in-gaming-industry-enhancing-storytelling-and-interaction

[2] https://www.atlantanewsfirst.com/2023/01/24/experts-warn-about-possible-misuse-new-ai-tool-chatgpt/

Findings of the Association for Computational Linguistics: NAACL 2024, pages 409–436
June 16-21, 2024 ©2024 Association for Computational Linguistics

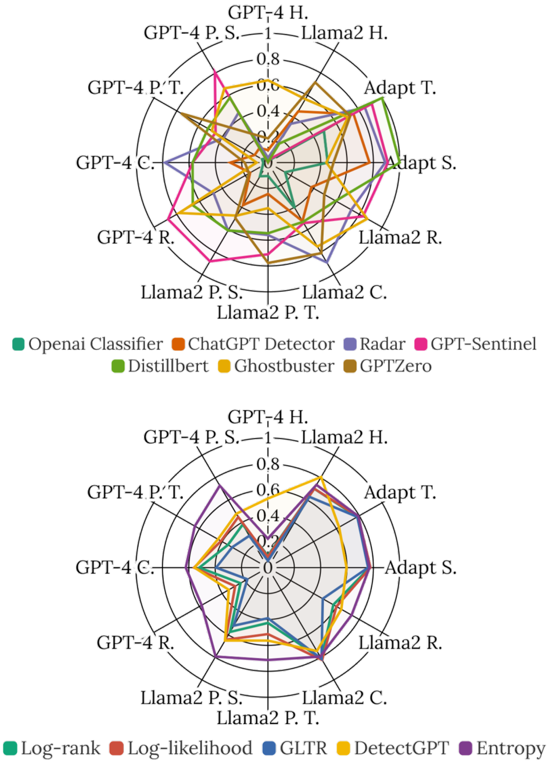


Figure 2: Accuracy of different dectors on MIXSET. (Above) Model-based methods; (Below) Metric-based methods. P.T. and P.S. signify token and sentence-level polish, respectively; C. for complete, R. for rewrite; Adapt T. and Adapt S. for token and sentence-level adapt. See 3 for details on revising operations.

Previous studies proposed many methods to detect MGT, including metric-based and model-based methods, where they have only tried to enhance the detection ability on binary classification, i.e., pure MGT or HWT. However, they did not pay much attention to revised texts (i.e., *mixtext*), but considered these cases as an attack on the detection system (Krishna et al., 2023) or complex cases for detection (Mitchell et al., 2023; Guo and Yu, 2023). However, the mixture of MGT and HWT is an essential scenario in our daily lives when using LLM assistants. For instance, thousands of non-native English speakers utilize LLMs to polish their drafts to avoid grammar problems. Moreover, LLMs can follow human instructions to produce new stories and interactive dialogue in game design [3]. Authors can also use LLMs to complete stories, providing them with new ideas and inspiration with LLM assistants like *Metaphoria* (Gero and Chilton, 2019) and *Sparks*, thereby generating metaphorical and science writing suggestions and supporting creative writing tasks (Gero et al., 2022).

Hence, there is a pressing demand to comprehensively analyze mixture cases and give a formal definition of them. Given that *mixtext* is a very common case in daily life and its amount continuously increases in NLP areas, it holds significant importance, especially in education. To end this, we propose a new dataset MIXSET, which is the first dataset that aims at the mixture of HWT and MGT, including both AI-revised HWT and human-revised MGT scenarios as illustrated in Figure 1, which addresses gaps in previous research. Further details of the dataset and definitions can be seen in Section 3. We also examine our dataset on mainstream detectors in binary and three-class settings to further analyze and raise concerns about these common but hard-to-detect cases.

To summarize, our work provides three main contributions:

- We defined *mixtext*, a form of mixed text involving both AI and human-generated content, providing a new perspective for further exploration in related fields.
- We proposed a new dataset MIXSET, which specifically addresses the mixture of MGT and HWT, encompassing a diverse range of operations within real-world scenarios, addressing gaps in previous research.
- Based on MIXSET, we conducted extensive experiments involving mainstream detectors and obtained numerous insightful findings, which provide a strong impetus for future research.

2 Related works

2.1 Machine Generated Text Detection

Current MGT detection methods can be broadly categorized into metric-based and model-based methods according to the previous study (He et al., 2023). Please refer to Appendix A for comprehensive related works.

Metric-based Methods. Building upon the observation that MGTs occupy regions with sharp negative log probability curvature, Mitchell et al. (2023) introduced a zero-shot whitebox detection method called DetectGPT, setting a trend in metric-based detection (Su et al., 2023; Mireshghallah et al., 2023; Bao et al., 2023). Recently, Yang et al. (2023a) also introduced a powerful detection method known as DNA-GPT, which leverages N-gram (Shannon, 1948) in a black-box setting.

[3] https://aicontentfy.com/en/blog/chatgpt-in-gaming-industry-enhancing-storytelling-and-interaction

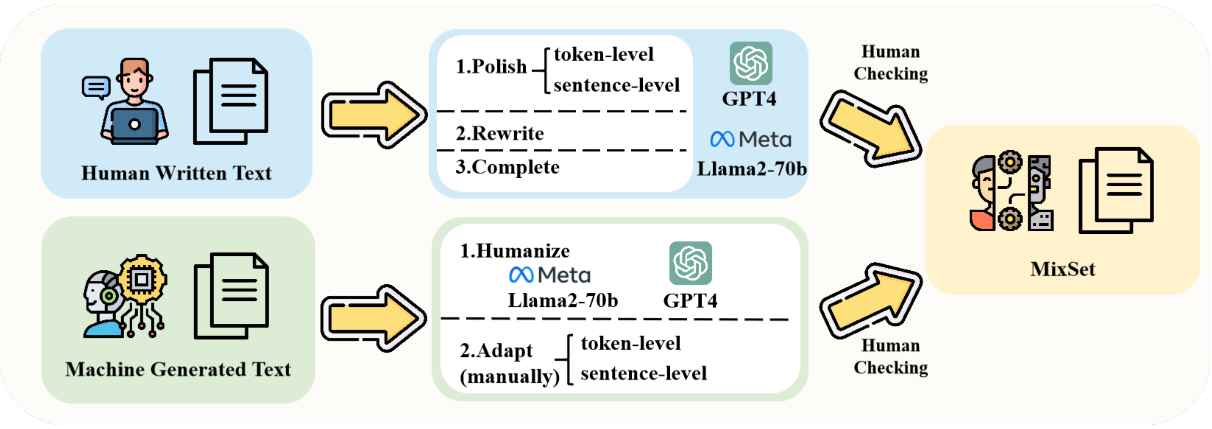


Figure 3: The process of MixSet generation. We perform distinct operations in HWT and MGT. In HWT, three operations—polish, rewrite, and complete—are employed. In MGT, we utilize LLama2 and GPT-4 to aid in humanization and conduct the adaptation operation manually.

Model-based Methods. In the era of Large Language Models (LLMs), Guo et al. (2023) developed the ChatGPT Detector based on a fine-tuned Roberta model. As for decoder-based detectors, GPT-sentinel (Chen et al., 2023) leverage the t5-small model (Muennighoff et al., 2022) and show convincing results when detecting MGT even in revised cases.

2.2 Previous study to mix of HWT and MGT

Prior studies have viewed the mixture of HWT and MGT in different settings. DNA-GPT (Yang et al., 2023a) and DetectGPT (Mitchell et al., 2023) notably utilized the T5 model (Raffel et al., 2020) to simulate scenarios where humans make limited, random modifications to MGT, creating complex test cases. Conversely, DIPPER (Krishna et al., 2023) and OUTFOX (Koike et al., 2023b) opted for a paraphrasing technique, using this method to craft adversarial attacks aimed at eluding the detection mechanisms of classifiers, thereby presenting a nuanced way to alter MGT while maintaining undetectability.

2.3 Datasets for MGT Detection

Previous studies have proposed many datasets of MGT, accompanied by their newly proposed detectors (Verma et al., 2023; Chen et al., 2023). Guo et al. (2023) leverages multiple previous Question-Answer (QA) datasets (Jin et al., 2019; Lin et al., 2021), allowing ChatGPT to generate corresponding answers without explicit prompts. This results in creating a comprehensive dataset comprising a large set of pairs of MGT and HWT. Following the QA pattern, many researchers (Mitchell et al., 2023; Su et al., 2023; Hu et al., 2023; He et al., 2023) propose datasets with the MGT from variant mainstream LLMs (OpenAI, 2022, 2023b).

However, these datasets typically consist of two distinct classes of texts, namely pure MGT or HWT, without accounting for the potential mixture cases. Furthermore, issues arise due to variations in prompts (Koike et al., 2023a), sampling methods, and the inherent differences in length, style, and quality among texts (He et al., 2023), posing variations challenges on the generalization ability of proposed detectors (Xu et al., 2023). In some instances, MGT included in datasets may not be thoroughly checked, with many noisy sentences not filtered well. For example, some sentences like *Let me know if you have any other questions* exist in the dataset, which will impact the effectiveness of the detectors (Guo et al., 2023).

3 MixSet Dataset

In this section, we present MixSet (**Mix**case Data**set**), the first dataset featuring a blend of HWT and MGT. Distinguished from earlier datasets exclusively composed of pure HWT and MGT, MixSet comprises a total of $3.6k$ mixtext instances, and the pipeline of its construction is shown in Figure 3. These operations are grounded in real-world application scenarios, each altered by a single LLM or through manual intervention, contributing 300 instances in our MixSet.

For our base data, we meticulously select pure HWT and MGT datasets. In the case of HWT, we gather datasets proposed before the widespread use of LLMs to mitigate potential contamination by MGT, as detailed in Table 1. For MGT, we choose samples from previous datasets (Rajpurkar et al., 2016a; Lin et al., 2022; Nishida et al., 2019), generated in a QA pattern by different LLMs, including the GPT family (OpenAI, 2022, 2023b), ChatGLM (Du et al., 2022), BloomZ (Muennighoff et al., 2022), Dolly [4], and StableLM (StabilityAI, 2023), all distinct from our MIXSET instances.

Table 1: The original resources of Human Written Texts in constructing our MIXSET.

Text Type	Original Resources
Email Content	Enron Email Dataset (CMU, 2015)
News Content	BBC News (Greene, 2006)
Game Review	Steam Reviews (Najzeko, 2021)
Paper Abstract	ArXiv-10 (Farhangi et al., 2022)
Speech Content	TED Talk (TheDataBeast, 2021)
Blog content	Blog (Schler et al., 2006)

3.1 Definition of Mixtext

Generally, *mixtext* is the mixed text involving both AI and human-generated content. To formulate it, we define a text sequence as $x \in X$, where X represents the set of all text sequences. The sequences in X can originate from either human-written text $\mathbb{X}_{\text{human}}$ or machine-generated text $\mathbb{X}_{\text{machine}}$. We denote the set of operations used to revise texts as $\mathbb{OP} = \{OP_1, OP_2, \ldots, OP_n\}$, categorized into two groups: $OP_{\text{human}}, OP_{\text{machine}}$. Here, OP_{human} refers to operations involving human revision on machine-generated text (MGT), while OP_{machine} refers to AI-driven operations on human-written text (HWT). In addition to $\mathbb{X}_{\text{human}}$ and $\mathbb{X}_{\text{machine}}$, we define $\mathbb{X}_{\text{mixtext}}$ as the union of all texts derived from $\mathbb{X}_{\text{human}}$ through OP_{machine} and all texts derived from $\mathbb{X}_{\text{machine}}$ through OP_{human}:

$$\begin{aligned}\mathbb{X}_{\text{mixtext}} = &\{OP_{\text{machine}}(x) \mid x \in \mathbb{X}_{\text{human}}\} \\ &\cup \{OP_{\text{human}}(x) \mid x \in \mathbb{X}_{\text{machine}}\}\end{aligned}$$

3.2 Dataset Construction

Combined with previous studies (Goyal et al., 2023; Wang et al., 2021) and real scenarios, we use five operations to generate mixtexts. They are divided into two operations shown in Table 2: 1) AI-revised: it contains three operations including 'polish', 'complete', and 'rewrite'. 2) Human-revised: it includes 'adapt' and 'humanize'.

Table 2: Different operations with their operation levels. ✔ demonstrate that MIXSET contains a subset operates at that level.

Operation	Token	Sentence	Paragraph
AI-Polish	✔	✔	✘
AI-Complete	✘	✘	✔
AI-Rewrite	✘	✘	✔
Human-Adapt	✔	✔	✘
Humanize	✔	✔	✔

- **Polish** (Chen, 2023): Polish operation contains token-level and sentence-level polishing. Token-level makes alterations at the individual word level, including changes such as adjusting words for precision or correcting spelling errors. On the other hand, sentence-level aims to enhance the overall coherence and clarity of the text by revising and restructuring the complete sentence.
- **Complete** (Zhuohan Xie, 2023): Complete operation involves taking 1/3 of every text and employing LLMs to generate the rest of the text.
- **Rewrite** (Shu et al., 2023): Rewrite operation requires LLMs to initially comprehend and extract key information from the given HWT and then rewrite them.
- **Humanize** (Bhudghar, 2023): Humanize operation typically refers to the modification of MGT to more closely mimic the natural noise for LLM (Wang et al., 2021) that human writing always brings. We employed LLMs to introduce various perturbations to the pure MGT, including *typo, grammatical mistakes, links*, and *tags*.
- **Adapt** (Gero et al., 2022): Adapt operation refers to modifying MGT to ensure its alignment to fluency and naturalness to human linguistic habits without introducing any error expression. The adapt operation is also divided into token-level and sentence-level adaptation. We accordingly performed manual annotations on the pure MGT dataset at both the token and sentence levels.

The detailed distribution of each category in MIXSET is shown in Table 3. All data generated from GPT-4 (300 items) and Llama2 (300 items) have undergone rigorous manual review and modification in the 'humanize' operation. For AI-revised *mixtext* generation, Llama2-70b and GPT-4 were used, both set to default parameters, including a temperature of 1. These models are chosen for their ability to produce high-quality, grammatically correct texts (Hugging Face, 2023). In human-revised

[4] https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

operation, we leverage two LLMs to assist with 'humanize' operation. For the adapt operation, we invite eight human experts with excellent English skills to revise MGT carefully to align it with human expression better. The details of human annotation guidelines and prompt template are shown in Appendix B.1 and D. After collecting all revised texts, we conducted a manual evaluation involving data filtering and cleaning to ensure MIXSET is high quality, such as removing conversational phrases like 'Sure! Here's a possible completion'.

Table 3: Detailed distribution of different operations in MIXSET.

	Operation	GPT-4	Llama2	Human
AI Revised	Polish Tok.	300	300	—
	Polish Sen.	300	300	—
	Complete	300	300	—
	Rewrite	300	300	—
Human Revised	Humanize	—	—	(300+300)
	Adapt Tok.	—	—	300
	Adapt Sen.	—	—	300

3.3 Dataset Analysis

Our comprehensive analysis of the MIXSET dataset covers length distribution, self-BLEU (Zhu et al., 2018), Levenshtein distance (Levenshtein, 1966), and cosine similarity. We only show analysis of length distribution and cosine similarity analysis here; for self-BLEU and POS distribution, refer to Appendix B.2.

- **Length distribution:** Given that detectors generally perform better with medium to long texts than with short texts (He et al., 2023), and to ensure that the text lengths in the MIXSET reflect real-world usage patterns, we have systematically selected data with a word count that falls within the range of 50 to 250 words. This range was chosen to ensure that the data were sufficiently detailed to provide meaningful insights while being concise enough to allow for effective analysis and comparison. As shown in Figure 4, the text lengths of both the MIXSET, as well as the HWT and MGT, follow a normal distribution.
- **Cosine Similarity:** Figure 5 illustrates that the texts processed with token-level polish operations exhibit the highest similarity to the original texts, followed by sentence-level polish, rewrite, and complete. Texts modified through the 'humanize' operation demonstrate lower similarity than those altered by adaptation.
- **Levenshtein Distance:** The Levenshtein distance (Levenshtein, 1966) is a metric for measuring the difference between two strings. We can observe in Figure 6 that in terms of the extent of modification, the rewrite operation results in the most significant alterations to the original texts, followed by complete and sentence-level polish. We also observe that manual annotations at both the token-level and sentence-level adaptation exhibit a high degree of differentiation.

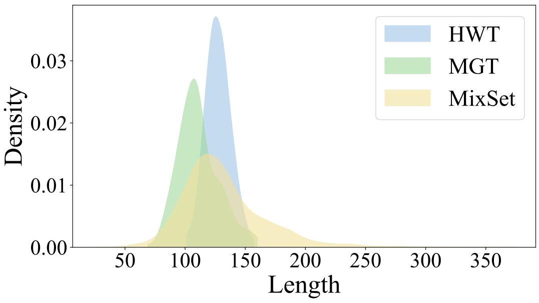


Figure 4: Length distribution of the HWT, MGT, and MixSet.

4 Experiments

4.1 Goals

We conduct experiments to understand better multiple facets of current detectors encountering our dataset MIXSET, including zero-shot and fine-tuning settings. We will figure out four questions:

- **Question 1.** How do current detectors perform in MIXSET dataset? Is there any classification preference in these detectors?
- **Question 2.** What is the performance of detectors retrained on our MIXSET? What about three-classed classification as we consider *mixtext* as a new class distinct from HWT and MGT?
- **Question 3.** What is the generalization ability of current detectors on our MIXSET?
- **Question 4.** Will the size of the training set impact the detection ability on *mixtext*?

4.2 Experiment Setup

Among our four experiments, We evaluate five metric-based and seven model-based detectors on three metrics in total, as shown in Tabel 4 and Table 5. We also outline the detailed training set construction in Table 6. Please refer to Appendix B.2 for a comprehensive introduction to detectors and metrics.

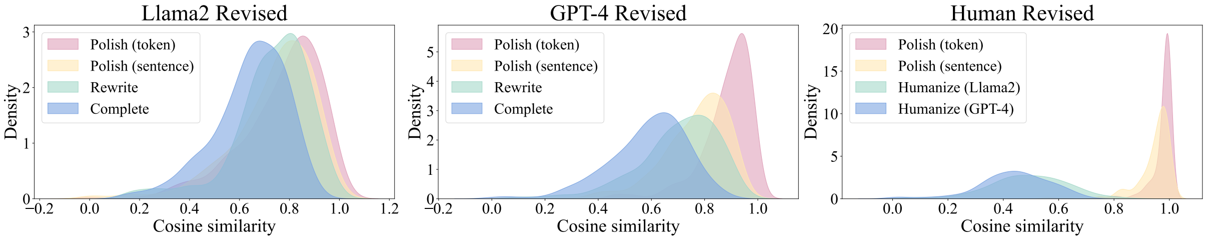


Figure 5: Cosine similarity of the MixSet

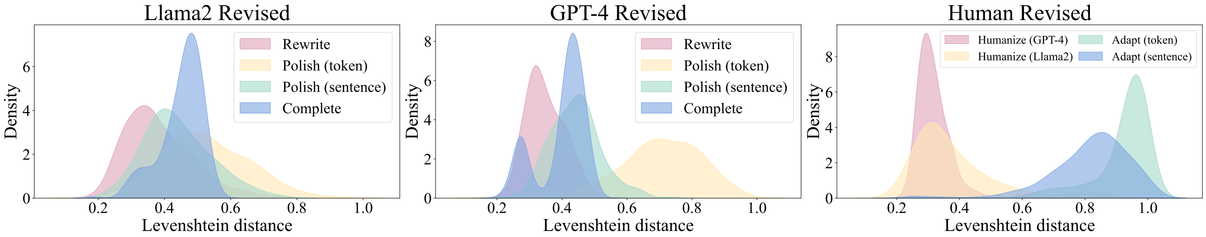


Figure 6: Levenshtein distance of the MixSet

Class Number. In real-world scenarios, people often aim to detect the presence of MGT in the text (e.g., spreading fake news or propaganda (Christian, 2023), reinforcing and intensifying prejudices (Sison et al., 2023)), and sometimes mixtext is also treated as MGT (e.g., student modified some words in MGT (i.e., mixtext) to generate homework, to avoid detection). Therefore, our experiments established two categorization systems: binary and three-class. In the binary classification, mixtext is categorized as MGT, while in the three-class classification, mixtext is treated as a separate class. The label setting is shown in Table 5.

Question 1. Based on MixSet, we evaluate current detectors to determine the classification preferences on mixtext, i.e., Does the detector tend to classify mixtext as MGT or HWT? We calculate the percentage of mixtext samples categorized to MGT in the experiment. For the DistilBERT detector and other metric-based detectors utilizing logistic regression models, we employ a training set comprising 10,000 pre-processed samples of both pure HWT and MGT. For other detectors, we use existing checkpoints [5] [6] or API [7] and evaluate them in a zero-shot setting.

Question 2(a). Following **Question 1**, our inquiry is whether the detector can accurately classify *mixtext* as MGT after training on MixSet. We fine-tune detectors on pure HWT and MGT data and a train split set of our MixSet labeled as MGT.

Table 4: Detectors used in different experiments.

	Detector	Q 1	Q 2	Q 3	Q 4
Metric-Based	Log-likelihood (Solaiman et al., 2019)	✔	✔	✘	✔
	Entropy (Gehrmann et al., 2019)	✔	✔	✘	✘
	GLTR (Gehrmann et al., 2019)	✔	✔	✘	✔
	Log-rank (Mitchell et al., 2023)	✔	✔	✘	✘
	DetectGPT (Mitchell et al., 2023)	✔	✔	✔	✔
Model-Based	Radar (Hu et al., 2023)	✔	✔	✘	✔
	ChatGPT Detector (Guo et al., 2023)	✔	✔	✔	✔
	DistillBert (Ippolito et al., 2019)	✔	✔	✔	✘
	GPT-sentinel (Chen et al., 2023)	✔	✔	✘	✔
	OpenAI Classifier (OpenAI, 2023a)	✔	✘	✘	✘
	Ghostbuster (Verma et al., 2023)	✔	✘	✘	✘
	GPTzero (Tian, 2023)	✔	✘	✘	✘

Question 2(b). On the other hand, assuming that *mixtext* lies outside the distribution of HWT and MGT, we conduct a three-class classification task, treating mixtext as a new label. In this scenario, we adopt multi-label training for these detectors while keeping all other settings consistent.

Question 3. As highlighted in prior research (Xu et al., 2023; He et al., 2023) that transfer ability

[5] `https://huggingface.co/TrustSafeAI/RADAR-Vicuna-7B`

[6] `https://github.com/haok1402/GPT-Sentinel-public`

[7] `https://gptzero.me/`

Table 5: The details of class number, metrics, and whether the detectors are retrained in our experiments. Except for Question 2(b), we implement binary classifications i.e., HWT and MGT. Per. stands for Percentage.

Setting	**Q 1**	**Q 2**		**Q 3**	**Q 4**
		(a)	**(b)**		
Class Num.	2-Class	2-Class	3-Class	2-Class	2-Class
Metric	MGT Per.	F1, AUC	F1	AUC	F1, AUC
Retrained?	✘	✔	✔	✔	✔

Table 6: An outline of detailed training set construction for each experiment. 'Ope.' denotes 'operation transfer' in Experiment 3, while 'LLM' refers to 'LLM transfer'.

Experiment	HWT/MGT	MIXSET
Q 1	$10k$	0
Q 2(a)	$10k$	$3k$
Q 2(b)	$10k$	$3k$
Q 3(Ope.)	$1k$	$0.5k$
Q 3(LLM)	$5k$	$1.5k$
Q 4	$1k/4k/7k/10k$	$0/1.5k/3k$

is crucial for detectors, our objective is to investigate the effectiveness of transferring across different subsets of MIXSET and LLMs. We establish two transfer experiments to assess whether the transferability of current detection methods is closely linked to the training dataset, referred to as operation-generalization and LLM-generalization:

- **Operation-generalization:** We initially train our detectors on one MIXSET subset operated by one of these operations, along with pure HWT and MGT datasets, and then proceed to transfer it to the subsets processed by other operations.
- **LLM-generalization:** In this experiment, we train detectors on GPT-generated texts and HWT, following which we evaluate the detectors on mixtext generated by GPT family (OpenAI, 2023b) and Llama2 (Touvron et al., 2023), respectively, to see whether there is a generalization gap between different LLMs.

Question 4. Empirically, incorporating more training data has been shown to enhance detection capabilities and robustness for generalization (Ying, 2019). To determine the relation between detectors' performance and the size of the training set, we follow **Question 2** and use varying sizes of training sets to retrain detectors, as illustrated in Table 6.

5 Empirical Findings

There is no obvious classification preference in current detectors on mixtext. In other words, the detectors do not exhibit a strong tendency to classify mixtext as either HWT or MGT. As we can observe from Figure 2 and Table 10, it is evident that the MGT percentage[8] of *mixtexts* is between MGT and HWT, indicating that the current detectors do not have a strong preference towards mixtext classification. This proves the success and effectiveness of our constructed MIXSET in presenting mixed features of HWT and MGT, demonstrating the limitations of existing detectors in recognizing mixtext.

When dealing with mixtext, the detectors treat it as an intermediate state between HWT and MGT. Most detectors exhibit inconsistent classification within a single subset, fluctuating between accuracies of 0.3 and 0.7, akin to random choice. In AI-revised scenarios, subsets, such as polished tokens or sentences, pose extreme detection challenges. Mainstream detectors generally perform poorly in these cases due to the subtle differences between *mixtext* and original text, highlighted in previous studies (Krishna et al., 2023). Furthermore, texts generated by Llama2-70b are easier to detect than those by GPT-4, possibly due to GPT-4's closer generative distribution to human writing.

Supervised binary classification yields profound results; however, three-classes classification encounters significant challenges when applied to mixtext scenarios except Radar. As indicated in Table 7, retrained model-based detectors outperform metric-based methods in both binary and three-class classification tasks. Notably, Radar ranks first in our results, achieving a significant lead over other detectors. We suppose that this superior performance can be attributed to its encoder-decoder architecture, which boasts 7 billion trainable parameters, substantially more than its counterparts. We also examined the impact of retraining on MixSet on MGT detection performance. As indicated in Table 8, there was a slight decrease in the F1 score, while the AUC metric remained largely unaffected. Notably, post-retraining, the detector acquired the capability to identify mixtext—an advancement deemed highly valuable. This ability to detect mixtext, despite a minor trade-off in F1 score for MGT detection, represents a significant step forward, suggesting a promising direction for

[8] MGT percentage means the percentage of identifying samples as MGT of different sets in Experiment 1.

Table 7: F1 score of experiment 2 (a) and (b). Tok. stands for token level and Sen. stands for sentence level. We underscore the best-performing detector and **bold** the score greater than 0.8, which we consider as a baseline threshold for detection.

Detection Method	Average	AI-Revised								Human-Revised			
		Complete		Rewrite		Polish-Tok.		Polish-Sen.		Humanize		Adapt-Sen.	Adapt-Tok.
		Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4		
Experiment 2 (a): Binary Classification													
log-rank	0.615	0.695	0.686	0.637	0.479	0.617	0.606	0.647	0.595	0.617	0.454	0.676	0.667
log likelihood	0.624	0.695	0.695	0.637	0.492	0.657	0.627	0.657	0.657	0.637	0.386	0.676	0.667
GLTR	0.588	0.686	0.647	0.606	0.441	0.574	0.585	0.637	0.540	0.617	0.400	0.657	0.667
DetectGPT	0.635	0.715	0.651	0.656	0.560	0.632	0.587	0.657	0.632	0.692	0.587	0.641	0.609
Entropy	0.648	0.690	0.671	0.681	0.613	0.681	0.671	0.681	0.671	0.623	0.430	0.681	0.681
Openai Classifier	0.209	0.171	0.359	0.031	0.197	0.145	0.270	0.247	0.439	0.247	0.316	0.000	0.090
ChatGPT Detector	0.660	0.705	0.696	0.676	0.583	0.676	0.647	0.647	0.594	0.667	0.615	0.705	0.705
Radar	**0.876**	**0.867**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**	**0.877**
GPT-sentinel	0.713	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.696	0.714	0.714	0.714
Distillbert	0.664	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.639	0.667	0.667	0.667
Experiment 2 (b): Three-class Classification													
DetectGPT	0.255	0.276	0.210	0.295	0.278	0.283	0.234	0.271	0.237	0.280	0.222	0.233	0.235
ChatGPT Detector	0.304	0.288	0.346	0.283	0.288	0.395	0.341	0.265	0.328	0.267	0.317	0.253	0.273
Radar	0.775	**0.804**	**0.842**	0.797	**0.837**	**0.831**	**0.820**	**0.815**	**0.837**	**0.884**	**0.889**	0.510	0.429
Distillbert	0.261	0.267	0.333	0.319	0.329	0.294	0.309	0.294	0.329	0.309	0.342	0.000	0.010

Table 8: The detection capabilities on pure HWT and MGT, comparing performances with (w.) and without (w.o.) MixSet labeling MGT during the training process, with the better one underscored.

Detector	F1		AUC	
	w.o.	w.	w.o.	w.
log-rank	0.830	0.821	0.922	0.922
log likelihood	0.845	0.834	0.931	0.931
GLTR	0.831	0.818	0.920	0.920
DetectGPT	0.746	0.725	0.820	0.820
Entropy	0.770	0.770	0.859	0.859
ChatGPT Det.	0.881	0.896	0.954	0.979
Radar	0.997	0.997	1.000	1.000
GPT-sentinel	0.988	0.982	1.000	0.999
Distillbert	0.996	0.984	1.000	1.000

Table 9: Result of LLM-transfer experiments. Although we retrain our detector on texts generated by GPT-4, it shows convincing generalization ability to Llama2.

Method	w.o *MixSet*		w. *MixSet*	
	Llama2	GPT-4	Llama2	GPT-4
GPT-sentinel	**0.813**	0.739	**0.972**	**0.971**
Radar	**0.834**	0.729	**0.997**	**1.000**
ChatGPT Det.	0.664	0.445	0.681	0.480
Distillbert	0.687	0.638	0.673	0.698

enhancing detector versatility and applicability in varied contexts.

In the three-class classification task, detectors based on LLMs, particularly the Radar detector, significantly outperformed those utilizing the BERT model. The BERT-based detectors' performance was markedly poor, akin to random guessing, with some models even underperforming a random baseline. This stark contrast underscores the efficacy of LLMs in capturing nuanced distinctions, as demonstrated in tasks like Mixtext. The superior performance of LLM-based Radar detectors lays a solid foundation for future explorations and applications in fine-grained classification tasks.

Current detectors struggle to generalize across different revised operation subsets of MIXSET and generative models. As shown in Figure 8 and Figure 13, significant variability is observed in the transfer capabilities of three different detectors. Additionally, training on texts generated by different revised operations results in different transfer abilities for these detectors. Overall, Radar exhibits the most robust transfer capability among the four model-based detectors, achieving an overall classification accuracy exceeding 0.9, followed by

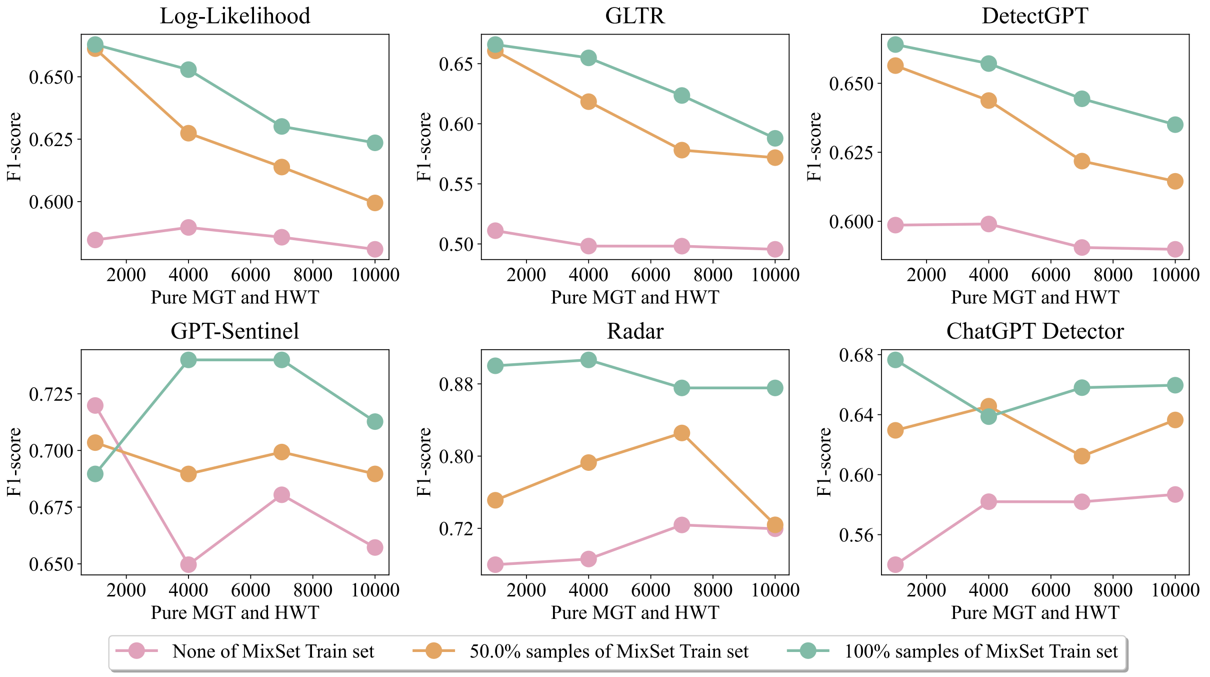


Figure 7: Analysis of the F1-score performance of various detectors across differing quantities of mixtext instances from MIXSET, as well as pure MGT and HWT.

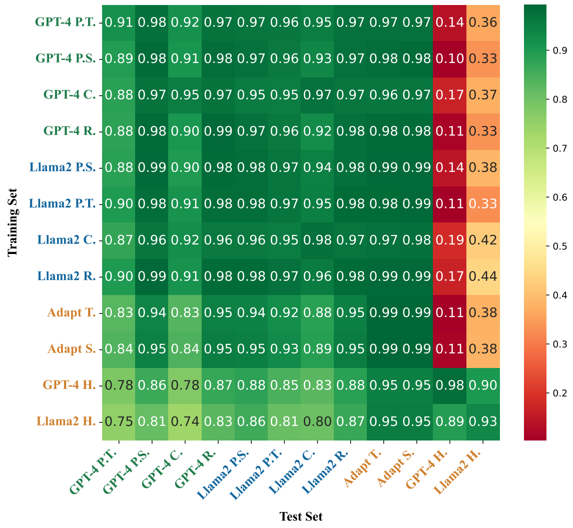


Figure 8: The AUC Heatmap of GPT-sentinel.

GPT-sentinel, DistillBert, and finally, the ChatGPT Detector. Among various operations, 'Humanize' exhibits the poorest transfer performance in almost all scenarios. Additionally, other operations also experience significant declines when dealing with 'Humanize' mixtexts. This suggests that 'Humanize' falls outside the current detectors' distribution of MGT, a gap that could be addressed by retraining on these specific cases. As shown in 9 It is also noteworthy that texts generated by Llama2-70b demonstrate stronger transfer abilities than those generated by GPT4.

Increasing the number of *mixtext* samples in the training set effectively enhances the success rate of *mixtext* detection. However, adding pure text samples does not yield significant improvements and may even have a negative impact on detector performance, especially for metric-based methods. This may be attributed to subtle distribution shifts between mixtext and pure text. The current detector still faces significant challenges in capturing these subtle shifts. For mixtext scenarios, a more powerful and fine-grained detection method is needed.

6 Conclusion

In this paper, we defined *mixtext*, the mixed text of human and LLM-generated content. Then, we proposed a dataset MIXSET to address the research gap in studying the mixed scenarios of machine-generated text (MGT) and human-written text (HWT). A thorough evaluation of the dataset is conducted, performing binary, three-class, and transfer experiments on mainstream detectors. The results underscore the complexities inherent in identifying mixtext, indicating the challenge of distinguishing the subtle differences in mixtext. As a result, there is a need for more robust and fine-grained detection methods.

7 Limitation

Bias Introduced by Human Participation. Although our study involved multiple human participants to modify the text, increasing the diversity and authenticity of the data, the text processing methods of different participants could vary due to their language habits and styles. This might affect the representativeness of the dataset and the generalization ability of the detection models.

Limitation in the Scale of the MixSet Dataset. As the MixSet dataset is the first to be proposed for studying mixed texts (mixtext), its overall scale is relatively small despite its comprehensive coverage in types. This could limit the comprehensiveness of model training and evaluation.

8 Ethics Statement

Opposition to Misuse of Mixed Text Scenarios. Our study highlights that the mixtext of HWT and MGT could significantly diminish the discerning abilities of detectors. However, we strongly oppose the misuse of mixtext to evade detection in specific scenarios, such as during examinations and homework assignments. We believe such misuse could severely harm the fairness of education and the integrity of academic practices.

Purpose for Scientific Research. This study aims purely for scientific exploration and understanding of the behavior and impact of mixtext in natural language processing. Our goal is to enhance understanding of mixed text processing and to advance the technological development in this area, not to encourage or support applications that may violate ethical standards.

Compliance with Licensing and Distribution Regulations. We affirm that all open-source resources utilized in our study, including detectors, language models, and datasets, have been employed in strict accordance with their respective licenses and distribution terms. This adherence extends to ensuring that any modifications, redistributions, or applications of these resources in our research comply with their original licensing agreements. Our commitment to these principles upholds the integrity of our research and contributes to a responsible and ethical academic environment.

Use of Publicly Available Data and Consideration for Privacy. The datasets used in our research are exclusively sourced from publicly available, open-source collections. While these datasets are publicly accessible and generally considered devoid of sensitive personal information, we acknowledge the potential for inadvertent inclusion of personal identifiers in datasets. We emphasize that our use of these datasets is aligned with their intended purpose and distribution terms. We also recognize the importance of respecting privacy and are committed to ongoing vigilance in this regard.

We reiterate that this research adheres to strict scientific and ethical standards, aiming to contribute to the field of natural language processing while ensuring that the results are not used for improper purposes. We also encourage our peers to consider these ethical factors when utilizing our research findings, ensuring their applications do not adversely affect society and individuals.

9 Acknowledgements

Lichao Sun is supported by the National Science Foundation Grants CRII-2246067.

References

Youdao translate. http://fanyi.youdao.com/.

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. *GitHub*.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Gandhi Gram Bhudghar. 2023. Ai text converter. https://aitextconverter.com/.

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with python: Analyzing text with the natural language toolkit. http://nltk.org/.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Xuhang Chen. 2023. Gpt academic prompt. https://github.com/xuhangc/ChatGPT-Academic-Prompt.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Ramakrishnan. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jon Christian. 2023. Cnet secretly used ai on articles that didn't disclose that fact, staff say. https://futurism.com/cnet-ai-articles-label.

CMU. 2015. Enron email dataset. https://www.cs.cmu.edu/~enron/.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Holly Else. 2023. Abstracts written by chatgpt fool scientists. *Nature*, 613(7944):423–423.

Ashkan Farhangi, Ning Sui, Nan Hua, Haiyan Bai, Arthur Huang, and Zhishan Guo. 2022. Protoformer: Embedding prototypes for transformers. In *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part I*, pages 447–458.

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

Katy Ilonka Gero and Lydia B Chilton. 2019. Metaphoria: An algorithmic companion for metaphor creation. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12.

Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for science writing using language models. pages 1002–1019.

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Singh Bedi. 2023. Towards possibilities & impossibilities of ai-generated text detection: A survey. *arXiv preprint arXiv:2310.15264*.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Computing Surveys*, 55(14s):1–39.

et al. Greene, Derek. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. 2023. On the learnability of watermarks for language models.

Marci Guerra. 2023. Chat gpt for journalism: Revolutionizing the future of reporting. https://brandalytics.co/chat-gpt-for-journalism/.

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

Zhen Guo and Shangdi Yu. 2023. Authentigpt: Detecting machine-generated text via black-box language models denoising. *arXiv preprint arXiv:2311.07700*.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Will Douglas Heavenarchive. 2023. Chatgpt is going to change education, not destroy it. https://www.technologyreview.com/2023/04/06/1071059/chatgpt-change-not-destroy-education-openai/.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *arXiv preprint arXiv:2307.03838*.

Hugging Face. 2023. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

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

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023a. How you prompt matters! even task-oriented constraints in instructions affect llm-generated text detection. *arXiv preprint arXiv:2311.08369*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023b. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *arXiv preprint arXiv:2307.11729*.

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*.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Y. Zou. 2024. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better blackbox machine-generated text detectors. *arXiv preprint arXiv:2305.09859*.

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*.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. Covid-qa: A question answering dataset for covid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Najzeko. 2021. Steam reviews dataset 2021.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style generative reading comprehension. pages 2273–2284, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2022. Openai models - gpt3.5. `https://platform.openai.com/docs/models/gpt-3-5`.

OpenAI. 2023a. Ai text classifier. `https://beta.openai.com/ai-text-classifier`.

OpenAI. 2023b. Gpt-4 technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. SQuAD: 100,000+ questions for machine comprehension of text. pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Canoee Liu, Simon Tong, Jindong Chen, and Lei Meng. 2023. Rewritelm: An instruction-tuned large language model for text rewriting. *arXiv preprint arXiv:2305.15685*.

Alejo Jose G Sison, Marco Tulio Daza, Roberto Gozalo-Brizuela, and Eduardo C Garrido-Merchán. 2023. Chatgpt: More than a weapon of mass deception, ethical challenges and responses from the human-centered artificial intelligence (hcai) perspective. *arXiv preprint arXiv:2304.11215*.

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*.

StabilityAI. 2023. Stablelm.

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*.

TheDataBeast. 2021. Ted talk transcripts (2006 - 2021).

Edward Tian. 2023. Gptzero: An ai text detector. https://gptzero.me/.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multitask benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. Llmdet: A third party large language models generated text detection tool. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Han Xu, Jie Ren, Pengfei He, Shenglai Zeng, Yingqian Cui, Amy Liu, Hui Liu, and Jiliang Tang. 2023. On the generalization of training-based chatgpt detection methods. *arXiv preprint arXiv:2310.01307*.

Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023a. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Xianjun Yang, Kexun Zhang, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023b. Zero-shot detection of machine-generated codes. *arXiv preprint arXiv:2310.05103*.

Xue Ying. 2019. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing.

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Weiming Zhang, and Nenghai Yu. 2023. Gpt paternity test: Gpt generated text detection with gpt genetic inheritance. *arXiv preprint arXiv:2305.12519*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, page 1097–1100.

Jey Han Lau Zhuohan Xie, Trevor Cohn. 2023. The next chapter: A study of large language models in storytelling. https://aclanthology.org/2023.inlg-main.23/.

A Full Related Works

A.1 Detecting Machine Generated Text

Current MGT detection methods can be broadly categorized into metric-based and model-based methods according to previous study (He et al., 2023). Moreover, other detection methods such as watermark, retrieval-based methods, and in-context learning leveraging LLMs also lead to promising detection methods.

Metric-based Methods. Metric-based methods leverage the LLM backbone directly to extract its distinguishing features between HWT and MGT, operating within a white-box setting that requires access to the model. Former methods like Log-Likelihood (Solaiman et al., 2019), Entropy, Rank (Gehrmann et al., 2019), and Log-Rank (Mitchell et al., 2023) employ statistical analysis to measure information beyond the token level. GLTR (Gehrmann et al., 2019) utilizes a suite of metric-based methods to aid in human identification. However, with the advent of LLMs, the progressively increasing similarity between the distributions of HWT and MGT has weakened its detection accuracy (Ghosal et al., 2023).

Building upon the observation that MGTs occupy regions with sharp negative log probability curvature, Mitchell et al. (2023) introduced a zero-shot whitebox detection method called DetectGPT, setting a trend in metric-based detection (Su et al., 2023; Mireshghallah et al., 2023; Bao et al., 2023). Yang et al. (2023a) also introduced a powerful detection method known as DNA-GPT, which leverages N-gram in a black-box setting by analyzing the differences between truncated original text and regenerated text. Recently, they even extended the detection method to MGT code in a zero-shot setting, which is proven to achieve promising results (Yang et al., 2023b).

Model-based Methods. In the Large Language Models (LLMs) era, Guo et al. (2023) developed the ChatGPT Detector based on a fine-tuned Roberta model. As for decoder-based or encoder-decoder detectors, GPT-sentinel (Chen et al., 2023) and RADAR (Hu et al., 2023), utilizing T5-small (Raffel et al., 2020) and Vicuna-7B (Chiang et al., 2023) respectively, show convincing results when detecting MGT even in revised cases. Moreover, Verma et al. (2023) proposes a novel detection framework called Ghostbuster, which employs passing documents through a series of weaker language models. Using a small amount of training data, Guo and Yu (2023) leverages a black-box LLM to denoise input text with artificially added noise and then semantically compares the denoised text with the original to determine if the content is machine-generated, leading a new method for MGT detection.

However, it's important to note that some researchers have raised concerns about fine-tuning models for MGT detection. Bakhtin et al. (2019) and Uchendu et al. (2020) have argued that fine-tuning models can lead to overfitting and a loss of generalization, particularly when dealing with text generated by the latest LLMs. They highlight the challenge posed by out-of-distribution editing texts, which can undermine the effectiveness of pre-trained detectors, as demonstrated by research on paraphrasing.

Other detection methods. Watermarking imprints specific patterns of the LLM output text that can be detected by an algorithm while being imperceptible to humans. Kirchenbauer et al. (2023) developed watermarks for language modeling by adding a green list of tokens during sampling. Currently, Gu et al. (2023) introduces a learnable watermark by distilling LLM and watermark technology into one student model, finding that models can learn to generate watermarked text with high detectability.

In retrieval-based methods, Krishna et al. (2023) introduce a method to retrieve semantically similar generations and search a database of sequences previously generated by specific Large Language Models (LLMs), looking for sequences that match the candidate text within a certain threshold. Delving deeper, Wu et al. (2023) proposes a model-specific detection tool called LLMDet, which can detect source text from specific LLMs by constructing a text collection dictionary for each LLM.

In the in-context learning setting, Yu et al. (2023) introduced a straightforward method that analyzes the similarity between re-answering a question by generating a corresponding question in the context of the given answer. Moreover, Koike et al. (2023b) employed a pure in-context learning approach for detection and found that LLMs can distinguish between human and machine styles.

A.2 Previous study to mix of HWT and MGT

Prior studies have viewed the mixture of HWT and MGT in different settings. DNA-GPT (Yang et al., 2023a) and DetectGPT (Mitchell et al., 2023) notably utilized the T5 model (Raffel et al., 2020) to simulate scenarios where humans make limited,

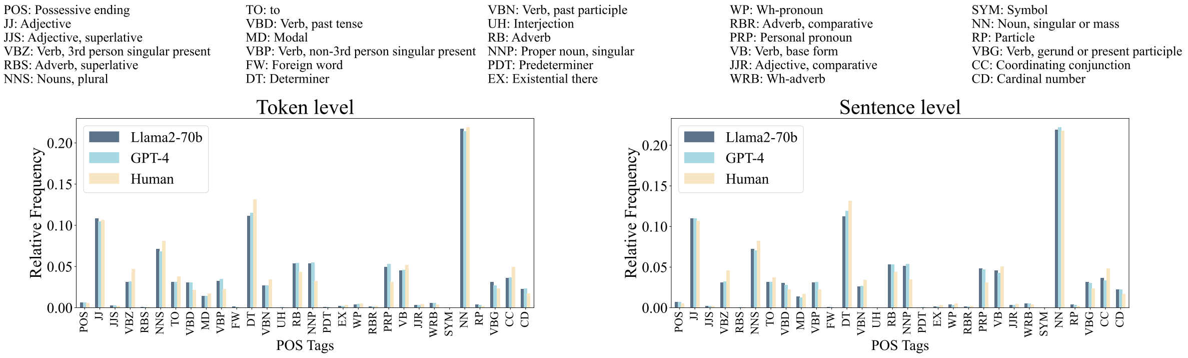


Figure 9: POS distribution of the MIXSET by NLTK (Bird et al., 2009).

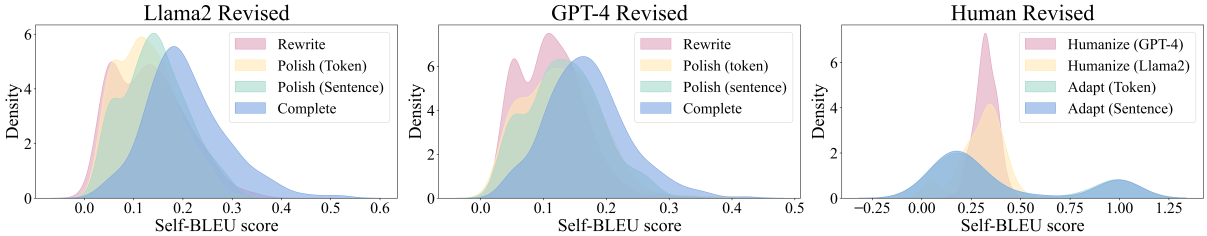


Figure 10: Self-BLEU score of the HWT, MGT, and MixSet.

random modifications to MGT, creating complex test cases. Conversely, DIPPER (Krishna et al., 2023) and OUTFOX (Koike et al., 2023b) opted for a paraphrasing technique, using this method to craft adversarial attacks aimed at eluding the detection mechanisms of classifiers, thereby presenting a nuanced way to alter MGT while maintaining undetectability. Recent research efforts have started to explore real-world applications of human-AI mixtext. Liang et al. (2024) explores the impact of AI, such as ChatGPT, on modifying content in academic peer reviews, aligning with our focus on the detection of mixtext.

A.3 Datasets for MGT Detection

Previous studies have proposed many datasets of MGT, often accompanied by their newly proposed detectors (Verma et al., 2023; Chen et al., 2023). Guo et al. (2023) leverages multiple previous Question-Answer (QA) datasets (Rajpurkar et al., 2016b; Kočiský et al., 2018; Jin et al., 2019; Lin et al., 2021), allowing ChatGPT to generate corresponding answers without explicit prompts. This approach results in creating a comprehensive dataset comprising a large set of pairs of MGT and HWT. Following the QA pattern, many researchers (Mitchell et al., 2023; Su et al., 2023; Hu et al., 2023; He et al., 2023) propose datasets with the MGT from variant mainstream LLMs (Du et al., 2022; Black et al., 2022; Anand et al., 2023; OpenAI, 2022, 2023b) [9]. Yu et al. (2023) only utilizes the answer section within the QA dataset (Hamborg et al., 2017; Möller et al., 2020) and employs ChatGPT to generate corresponding questions and re-answers.

However, these datasets typically consist of two distinct classes of texts, namely pure MGT or HWT, without accounting for the potential mixtext. Furthermore, issues arise due to variations in prompts (Koike et al., 2023a), sampling methods, and the inherent differences in length, style, and quality among texts in some datasets (He et al., 2023). These variations challenge the generalization of proposed detectors (Xu et al., 2023) and lie a vast diversity in distribution between the original and revised text (Ghosal et al., 2023). In some instances, the MGT included in datasets may not undergo thorough and careful evaluation. Many noisy sentences are not filtered well in the datasets. For example, some sentences like *Let me know if you have any other questions* exist in the dataset, which will impact the effectiveness of the detectors (Guo et al., 2023).

[9] https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

B Dataset Details

B.1 Construction Details

Eight Human revised the MGT to mixtext. The MGT is revised by eight human experts with professional English proficiency and costs them a total of 280 hours to complete this part. The guidelines for human revision are shown in Figure 22. And the labeling screenshot is shown in Figure 37.

B.2 Other Metrics in Evaluating MixSet

- **Self-BLEU Score:** Self-BLEU is a metric used to assess the diversity of generated text. Generally, a lower Self-BLEU score indicates higher textual diversity. These results are shown in Figure 10. Overall, the HWT shows greater diversity than MGT, and the Rewrite category has the highest textual diversity in the MixSet. The self-BLEU score of HWT, WGT, and mixtext is shown in Figure 11 and 10.
- **POS distribution:** POS distribution refers to the frequency and pattern of Part-of-Speech tags in a text, categorizing words into grammatical classes like nouns, verbs, and adjectives. This analysis is key for understanding the text's syntactic structure and linguistic characteristics, which is important in NLP research fields.

Seven Model-Based detectors. We implement seven Machine Generative Text (MGT) detectors, encompassing both supervised and zero-shot settings. Firstly, we consider a robust closed-source online detector baseline: GPTZero (Tian, 2023). Secondly, we implement three open-source encoder-based detectors: OpenAI's classifier (OpenAI, 2023a), Roberta-based classifier (Guo et al., 2023). We also implement GPT-sentinel (Chen et al., 2023), RADAR (Hu et al., 2023), and Ghostwriter (Verma et al., 2023) as strong baselines. We also finetune a pre-trained language model built by Sanh et al. (2019) with an extra classification layer on top.

Three Evaluation Metrics Previous studies (Sadasivan et al., 2023; Mitchell et al., 2023) have proven the feasibility of using the Area Under The ROC Curve (AUROC) score for evaluating detection algorithm effectiveness. Given that most detectors can only give a predictive probability, we build a logistic regression model to provide concrete predictions, i.e., MGT or HWT, converting probability to accuracy and f1-score as the two other metrics for our detection evaluation.

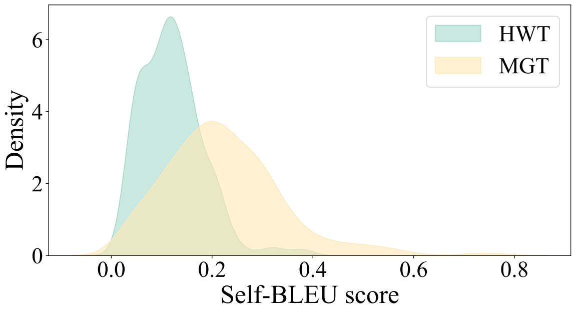


Figure 11: Self-BLEU score of HWT and MGT.

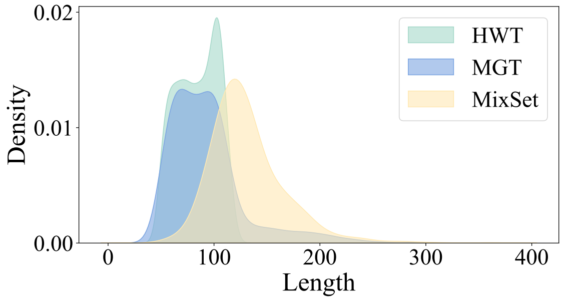


Figure 12: Length distribution of the training datasets and the MixSet.

Training set construction. We respectively select pure HWT and MGT for the train set from different datasets as illustrated in 1 and MGTBench (He et al., 2023), which is also the original dataset of our MixSet. Since all datasets are specific, this selection strategy ensures only a small difference in data distribution. Firstly, we do data deduplication and pre-process it to erase the Unicode or other special tokens like `\n\n`. Then, we select pieces of sentences with a similar length distribution in our MixSet, as illustrated in Figure 12. As we use accuracy as our evaluation metric, we restrict the amount of HWT and MGT to be the same in our dataset, as illustrated in Tabel 6.

Training details. We employ the standard binary-classification loss function and the AdamW optimizer (Loshchilov and Hutter, 2019), with an empirically determined learning rate. Specifically, for the Hello-Ai/Roberta-based model and the DistilBERT model, the learning rate is set to 5×10^{-7}. In contrast, for Radar and GPT-sentinel, the learning rates are 5×10^{-6} and 5×10^{-5}, respectively. Each supervised model undergoes training for three epochs on a dual-4090 server.

C Detailed Experiment Results

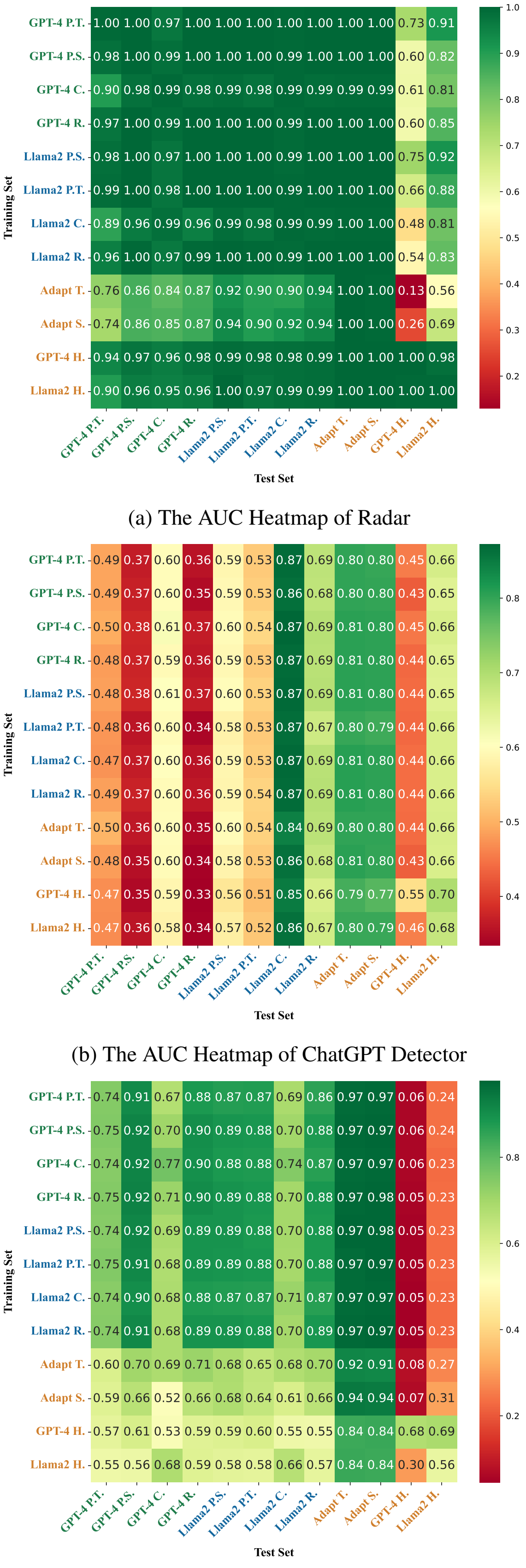
As for experiment 1, we put the detailed accuracy for different detectors in Table 10. In experiment 2, we also evaluate detectors with AUC metric, as shown in Table 11. We also post other detectors undergo our experiment 3 illustrated in Figure 13. As for experiment 4, we evaluate detectors with accuracy, precision, and recall metrics, as illustrated in Figure 14, 15, and 16.

D Prompt Template

We show the prompt template of LLM's operation, including complete, polish (token-level and sentence-level), rewrite, and humanize in Figure 17, Figure 18, Figure 19, Figure 20 and Figure 21.

E Case study in MixSet

We selected two cases to show the comparison between the revised mixtext and the original text, where the highlighted content represents the modified content. The HWT original text can be found in figure 23, the AI revised text are shown in Figure 24, 25, 26, 27, 28, 29, 30, and 31. The MGT original text can be found in Figure 32, and the Human revised text can be found in Figure 33, 34, 35, and 36.



(a) The AUC Heatmap of Radar

(b) The AUC Heatmap of ChatGPT Detector

(c) The AUC Heatmap of distilbert-based

Figure 13: The AUC Heatmap of the other three detectors.

Table 10: Percentage of identifying samples as MGT of different sets in Experiment 1. For example, the Log-Rank detector categorizes 57.30% of samples in the Llama2-revised set as MGT. We underscore the best-performing detector and **bold** the score greater than 0.8, which we consider as a baseline threshold for detection. (Tok. stands for token level, and Sen. stands for sentence level)

			AI-Revised								**Human-Revised**			
Detection Method	**HWT**	**MGT**	Rewrite		Complete		Polish-Tok.		Polish-Sen.		Humanize		Adapt-Tok.	Adapt-Sen.
			Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4		
Metric-based Detector														
Log-rank	0.213	**0.847**	0.573	0.240	**0.810**	0.520	0.573	0.383	0.427	0.350	0.703	0.093	0.783	0.770
Log-likelihood	0.223	**0.867**	0.600	0.287	**0.823**	0.560	0.643	0.450	0.513	0.410	0.703	0.083	0.790	0.777
GLTR	0.207	**0.840**	0.480	0.180	**0.813**	0.393	0.517	0.283	0.390	0.313	0.630	0.053	0.783	0.760
DetectGPT	0.350	**0.823**	0.643	0.343	0.743	0.557	0.650	0.480	0.563	0.437	**0.807**	0.533	0.623	0.597
Entropy	0.353	**0.840**	0.733	0.580	0.793	0.623	0.793	0.730	0.713	0.640	0.737	0.223	0.793	0.770
Model-based Detector														
Openai Classifier	0.060	0.607	0.150	0.047	0.407	0.037	0.123	0.037	0.103	0.053	0.023	0.007	0.490	0.453
ChatGPT Detector	0.040	0.757	0.380	0.157	0.523	0.287	0.380	0.130	0.243	0.117	0.457	0.097	0.750	0.770
Radar	0.307	**0.857**	0.730	0.477	**0.893**	0.783	0.607	0.447	0.560	0.387	0.347	0.037	**0.850**	**0.890**
GPT-Sentinel	0.133	**0.887**	**0.833**	**0.877**	0.540	0.573	**0.883**	**0.807**	0.710	0.460	0.033	0.000	**0.910**	**0.910**
Distillbert	0.483	**0.993**	0.593	0.660	0.530	0.573	0.607	0.580	0.547	0.527	0.170	0.003	**1.000**	**1.000**
Ghostbuster	0.103	0.610	**0.870**	0.780	0.750	0.087	0.353	0.493	0.473	0.663	0.567	0.637	0.700	0.443
GPTZero	0.017	0.730	0.493	0.167	**0.810**	0.177	0.497	0.260	0.777	0.763	0.717	0.187	0.720	0.497

Table 11: AUC of Experiment 2 (a). We underscore the best-performing detector and **bold** the score greater than 0.8, which we consider as a baseline threshold for detection. (Tok. stands for token level and Sen. stands for sentence level)

	AI-Revised								**Human-Revised**			
Detection Method	Rewrite		Complete		Polish-Tok.		Polish-Sen.		Humanize		Adapt-Tok.	Adapt-Sen.
	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4		
Metric-based Detector												
log-rank	**0.921**	0.629	0.632	0.318	0.569	0.531	0.662	0.462	0.641	0.245	0.778	0.778
log likelihood	**0.933**	0.650	0.672	0.352	0.610	0.569	0.709	0.508	0.652	0.206	0.782	0.786
GLTR	**0.870**	0.504	0.546	0.268	0.511	0.466	0.602	0.345	0.595	0.208	0.764	0.768
DetectGPT	**0.852**	0.644	0.669	0.352	0.612	0.466	0.664	0.482	0.677	0.461	0.548	0.557
Entropy	**0.814**	0.581	0.662	0.463	0.656	0.636	0.686	0.596	0.580	0.185	0.733	0.730
Model-based Detector												
Openai Classifier	0.294	0.601	0.126	0.360	0.433	0.492	0.383	0.590	0.321	0.517	0.182	0.187
ChatGPT Detector	0.706	0.399	**0.874**	0.640	0.567	0.508	0.617	0.410	0.679	0.483	**0.818**	**0.813**
Radar	**0.992**	**0.994**	**0.997**	**0.999**	**0.998**	**0.986**	**0.998**	**1.000**	**0.984**	**0.984**	**0.999**	**0.999**
GPT-sentinel	**0.994**	**0.992**	**0.987**	**0.993**	**0.995**	**0.964**	**0.992**	**0.996**	**0.915**	**0.953**	**0.958**	**0.986**
Distillbert	0.756	**0.856**	0.746	**0.859**	0.790	0.730	0.791	**0.856**	0.416	0.330	**0.837**	**0.861**

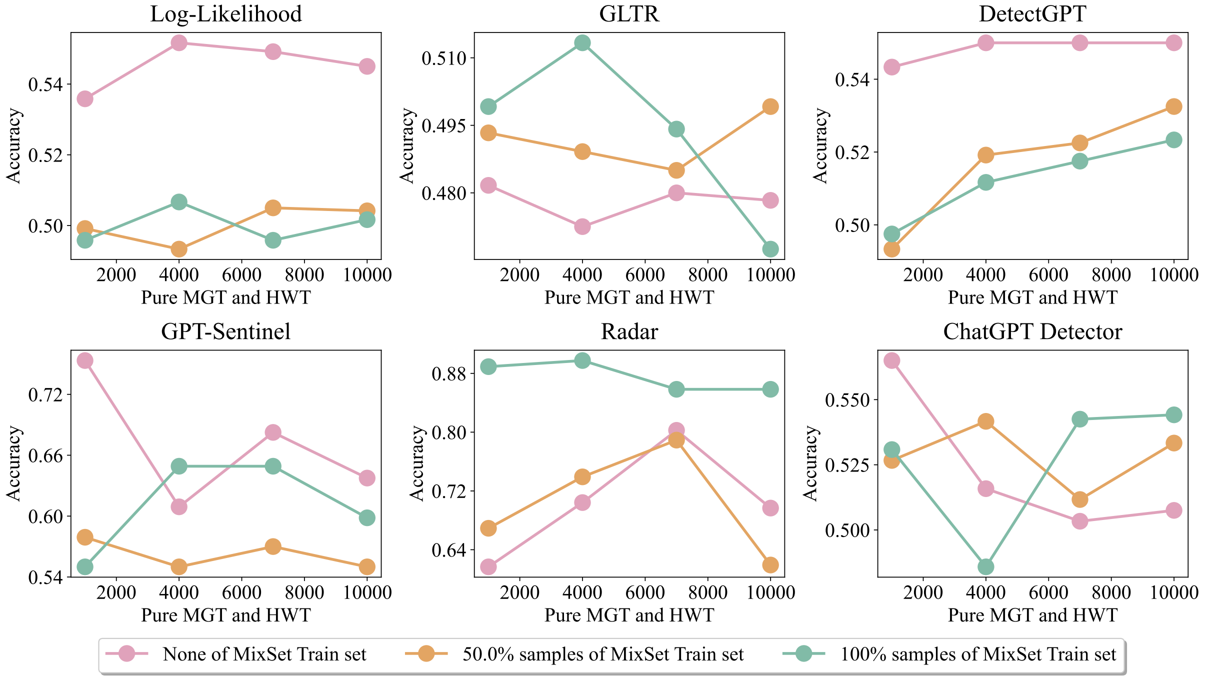


Figure 14: Analysis of the accuracy of various detectors across differing quantities of *mixtext* instances from MIXSET, as well as pure MGT and HWT.

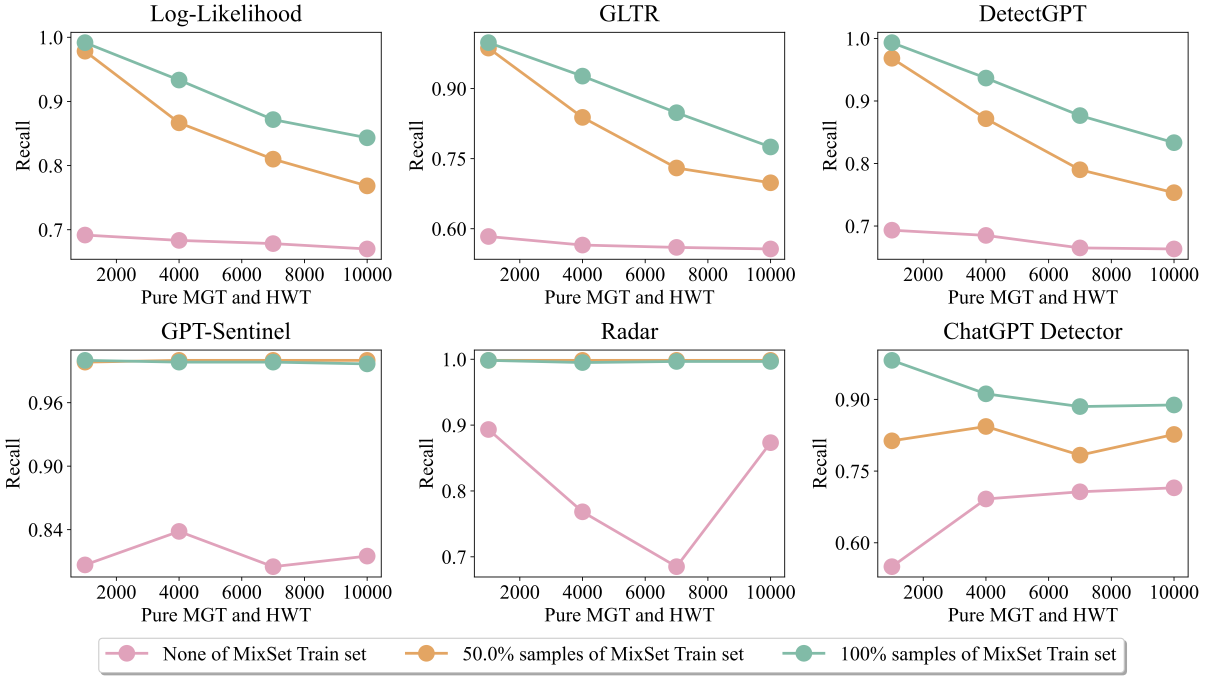


Figure 15: Analysis of the recall rate of various detectors across differing quantities of *mixtext* instances from MIXSET, as well as pure MGT and HWT.

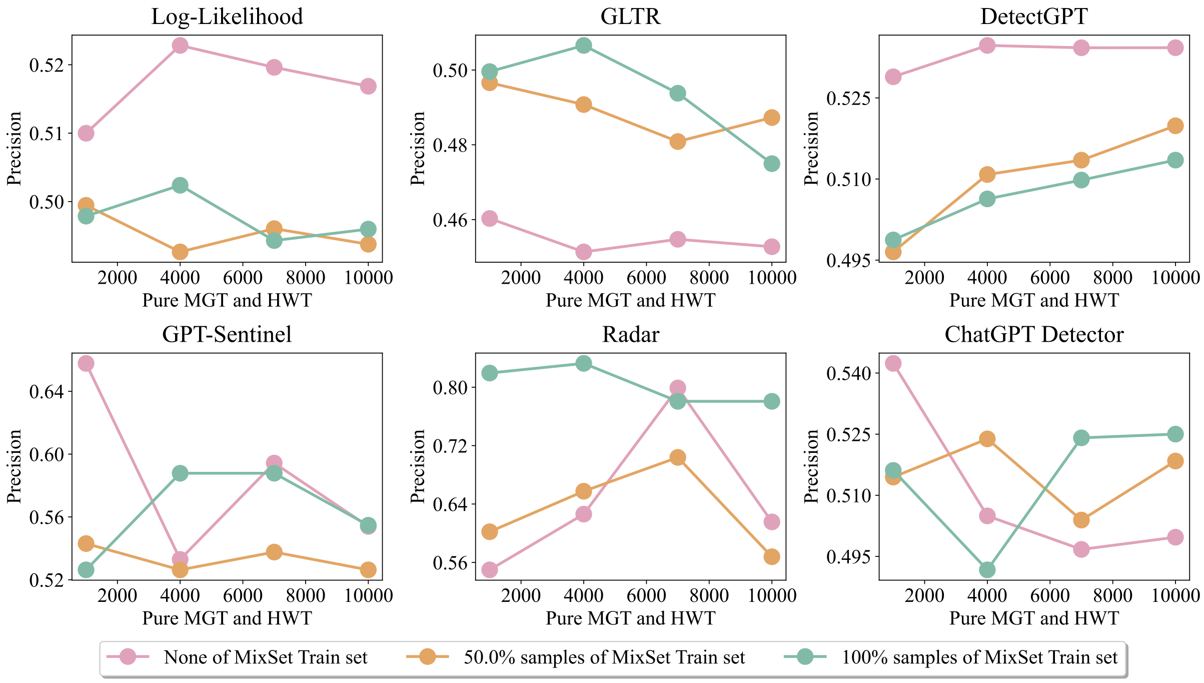


Figure 16: Analysis of the precision rate of various detectors across differing quantities of *mixtext* instances from MIXSET, as well as pure MGT and HWT.

Figure 17: Prompt(①)-LLM complete the HWT

```
I have an incomplete text and need it to be completed. Please expand this into
a complete text where the total word count, including the original text I have
provided, does not exceed 180 words. The original text must remain exactly as
is, with its format (such as capitalization and punctuation) intact. Please do
not modify any part of the original text. Here’s the text: {HWT}
```

Figure 18: Prompt(②)-LLM polish HWT in token level

```
Please carefully examine the following paragraph solely for spelling and
grammatical errors, and replace any words that are repetitive, inaccurate,
or poorly chosen. It is crucial to avoid any changes to the sentence order or
structure. The focus should be strictly on the choice and usage of individual
words to improve the clarity and appropriateness of the text without altering
the original sentence construction: {HWT}
```

Figure 19: Prompt(③)-LLM polish HWT in sentence level

```
Please optimize the sentences in the following paragraph to enhance fluency and
clarity. Do not alter the overall content or structure of the paragraph. Focus
on the construction and expression of the sentences, ensuring that the text is
coherent and the information is accurate: {HWT}
```

Figure 20: Prompt(④)-LLM rewrite HWT

Please extract the core ideas and keywords from the following English text and then rewrite a passage based on this information. The new text should maintain the essence of the original, with the word count varying by no more than 10% from the original. There's no need to list the core ideas and keywords. Here is the text that needs to be processed: {HWT}

Figure 21: Prompt(④)-LLM humanize MGT

I need to modify a machine-generated text to make it appear more like it was written by a human. The objective is to introduce elements commonly found in human-written texts. Here are some optional modifications you can choose to apply:
1. Introduce spelling errors or typos(optional)
2. Create grammatical errors, such as randomly adding or deleting words (optional).
3. Include relevant but internet links, like blog posts or image links pertaining to the topic, you don't have to use the real links, so you can freely write one (optional).
4. Add relevant hashtags, for instance, #TopicKeyword #Location #Activity (optional).
5. Use internet slang and abbreviations, e.g., 'OMG', 'How r u', 'LOL', (optional).
Please select any combination of these modifications to enhance the text's human-like quality. The aim is to simulate the imperfections and stylistic choices typical in casual human writing.
The word count of the new text should not exceed 1.1 times that of the original text.
You should just give me the revised version without any other words.
Emojis are strictly prohibitive, so please ensure that it contains no emojis.
Here is the machine-generated text:{HWT}

Figure 22: Guidelines for Human Revision

The content under this document is generated by a large language model, such as ChatGPT. You are required to revise it to make it closer to the style of human-written text. You are responsible for the text under the IDs xx-xx, and you need to make the following three types of modifications to the content, generating two different sentences for each ID (each ID corresponds to 3 sentences):

The document is in JSON format. You can choose to use code editors like Visual Studio Code or text editors like Notepad for reading and writing.

1. Adapt Token: You need to modify any words or phrases in this passage that you think are too rigid, mechanical, obscure, or unusual into vocabulary typical of human texts. Be careful not to alter the sentence order or structure; only modify 'words' or 'phrases.'

2. Adapt Sentence: You need to revise any sentence structures that you find too mechanical or rigid to make them more in line with how humans typically write texts. This involves changes at the sentence level, which may include altering sentence order and structure.

Note:

1. After writing, please ensure to check that there are no 'grammatical errors' or 'spelling mistakes' in the text paragraphs.

2. Do not use ChatGPT or other large language models for data annotation, as it will severely degrade the data quality.

3. You may use translation platforms like Youdao (you) or Google [a]; or use Grammarly [b] to check for grammatical errors.

Below are some examples you can refer to:

[Example 1]

[Example 2]

[Example 3]

[a] https://translate.google.com/

[b] https://www.grammarly.com/

Figure 23: Case Study of revision on HWT: Original HWT

[Original HWT] We present a new mixed variable symplectic (mvs) integrator for planetary systems, that fully resolve close encounters. The method is based on a time regularisation that allows keeping the stability properties of the symplectic integrators, while also reducing the effective step size whenever two planets encounter. We use a high order mvs scheme such that it is possible to integrate with large time steps far away from close encounters. We show that this algorithm is able to resolve almost exact collisions (i.e. with a mutual separation of a fraction of the physical radius) while using the same time-step as in weakly perturbed problems such as the solar system. We demonstrate the long-term behavior on systems of six super-earths experiencing strong scattering for 50 kyr. We compare our algorithm to hybrid methods such as mercury and show that for an equivalent cost, we obtain much better energy conservation.

Figure 24: Case Study of revision on HWT: GPT-4 Polish (Token)

[GPT-4 Polish (Token)] We present a new mixed-variable symplectic (MVS) integrator for planetary systems that fully resolves close encounters. The method is grounded in a temporal regularization that maintains the stability properties of symplectic integrators while also diminishing the effective step size whenever two planets draw near each other. We employ a high-order MVS scheme, enabling integration with extended time steps when distant from close encounters. We demonstrate that this algorithm capably resolves nearly exact collisions (i.e., with a mutual separation of a small fraction of the physical radius) using a consistent time-step as in weakly perturbed problems such as the solar system. We exhibit the long-term dynamics on systems of six super-Earths undergoing intense scattering over 50 kyr. We contrast our algorithm with hybrid methods such as Mercury, revealing that, for a comparable cost, we achieve much improved energy conservation.

Figure 25: Case Study of revision on HWT: GPT-4 Polish (Sentence)

[GPT-4 Polish (Sentence)] We introduce a novel mixed variable symplectic (MVS) integrator specifically designed for planetary systems that efficiently manages close encounters. This method utilizes time regularization to maintain the stability characteristics inherent to symplectic integrators, while dynamically adjusting the effective step size during planetary encounters. Our high-order MVS approach enables integration at large time steps when planets are distantly separated, thereby optimizing computational efficiency. The algorithm proves capable of resolving near-exact collisions—with mutual separations as minute as a fraction of the planets' physical radii—without altering the time step used for more stable configurations, such as within the solar system. We examine the algorithm's long-term stability in simulations of six super-Earths undergoing significant scattering over 50,000 years. Furthermore, we benchmark our algorithm against hybrid methods like Mercury, demonstrating superior energy conservation at a comparable computational cost.

Figure 26: Case Study of revision on HWT: GPT-4 Complete

[GPT-4 Complete] We present a new mixed variable symplectic (mvs) integrator for planetary systems, that fully resolve close encounters. The method is based on a time regularisation that allows keeping the stability properties of the symplectic integrators, while also reducing the effective step size whenever two planets encounter. We use a modified leapfrog scheme to address the issues of traditional integrators when handling near-collision scenarios. The key improvement lies in an adaptive algorithm that dynamically adjusts the step size according to the proximity of the planetary bodies. As a result, our integrator ensures high accuracy and energy conservation even during complex dynamical interactions. Tests demonstrate that the mvs integrator significantly outperforms standard fixed-step symplectic methods in long-term stability and computational efficiency. This advancement opens up new possibilities for accurately simulating the evolution of planetary systems over extended timescales, providing valuable insights into the dynamical behavior of multi-body systems and enhancing our understanding of celestial mechanics.

Figure 27: Case Study of revision on HWT: GPT-4 Rewrite

[GPT-4 Rewrite] We introduce an advanced mixed variable symplectic (mvs) tailored for planetary systems, adept at handling close encounters precisely. This technique employs time regularization to maintain the stability characteristic of symplectic integrators and intelligently reduces the effective step size during planetary encounters. Employing a sophisticated mvs algorithm, it facilitates integration over large time steps when planets are not closely interacting . Our findings illustrate that the algorithm effectively manages near-collision events , simulating scenarios with planet separations just fractions of their physical sizes, using consistent time steps applicable to less disturbed systems like the solar system. The integrator's performance excels in extended simulations, such as tracking six super-earths in a dense scattering scenario for 50,000 years, showcasing superior energy conservation when contrasted with hybrid algorithms like the mercury method, and doing so computational cost.

Figure 28: Case Study of revision on HWT: Llama2 Polish (Token)

[Llama2 Polish (Token)] We introduce a novel mixed variable symplectic (MVS) integrator for planetary systems that accurately resolves close encounters. Our method is built upon a time regularization that preserves the stability attributes of symplectic integrators while adaptively decreasing the effective step size during planetary interactions. By employing a high-order MVS scheme, we can integrate with large time steps in regions where close encounters are not occurring. This algorithm enables the precise resolution of nearly exact collisions (with a mutual separation of a small fraction of the physical radius) using the same time step as in weakly perturbed problems, such as the solar system. We demonstrate the long-term behavior of our algorithm on systems consisting of six super-Earths undergoing intense scattering. Our approach shows superior energy conservation compared to hybrid methods like Mercury, while maintaining comparable computational efficiency.

Figure 29: Case Study of revision on HWT: Llama2 Polish (Sentence)

[Llama2 Polish (Sentence)] We introduce a novel mixed variable symplectic (MVS) integrator for planetary systems that accurately resolves close encounters. Our method is founded on a time regularization technique that preserves the stability attributes of symplectic integrators while adaptively decreasing the effective step size during planetary encounters. By employing a high-order MVS scheme, we can integrate with substantial time steps in regions unaffected by close encounters. Our algorithm efficiently handles near-collisions(with a mutual separation of a few percent of the physical radius) utilizing the same time-step as in weakly perturbed scenarios, such as the solar system. In a comprehensive analysis, we demonstrate the long-term behavior of our method on systems consisting of six super-Earths undergoing intense scattering over 50 kyr. Our findings indicate that, compared to hybrid approaches like Mercury, our algorithm achieves superior energy. conservation at an equal computational cost

Figure 30: Case Study of revision on HWT: Llama2 Complete

[Llama2 Complete] We present a new mixed variable symplectic (MVS) integrator for planetary systems, that fully resolves close encounters. The method is based on a time regularization that allows keeping the stability properties of the symplectic integrators, while also reducing the effective step size whenever two planets encounter. We use a simple and efficient algorithm to detect close encounters, and switch to a smaller step size during these events to ensure accurate trajectory propagation. This approach enables us to maintain the long-term accuracy of the simulation while significantly improving computational efficiency. Our MVS integrator can handle both collisional and non-collisional close encounters, and we demonstrate its performance through several test cases involving Earth-Moon and Jupiter-Io system simulations.

Figure 31: Case Study of revision on HWT: Llama2 Rewrite

[Llama2 Rewrite] We introduce a novel mixed variable symplectic (MVS) integrator for solving planetary systems that undergo close encounters. Our approach combines the benefits of symplectic integrators - stability and accuracy - with the ability to adaptively adjust the time step size during close encounters. This enables us to capture near-collisions with high precision while still utilizing large time steps in stable regions. We validate our algorithm's performance by simulating systems of six super-Earths over 50,000 years, demonstrating its superiority compared to hybrid methods like Mercury in terms of energy conservation, while maintaining comparable computational efficiency.

Figure 32: Case Study of revision on MGT: Original MGT

[Original MGT] The first coffee was brewed by the ancient egyptians over 4,000 years ago. As of 2023, coffee consumption increased 70-fold in the past 100 years. Today, 98% of americans drink coffee. Studies have shown that drinking two cups of coffee per day is beneficial for adults. However, more and more studies also show that drinking one or less coffee is also equally beneficial. Drinking coffee has also been linked to higher memory, alertness, and focus. Overall, coffee is good for your health.

Figure 33: Case Study of revision on MGT: GPT-4 Humanize

[GPT-4 Humanize] OMG did you kno that, like, the first cuppa joe was whipped up by ancient Egyptions like more than 4,000 yrs ago? Fast fwd to 2023, and whoa, coffee drinking's shot up a whoppin' 70 times in the last century! Nowadays, it's like 98% of Americans are sippin' on coffee. Some smarty-pants studies point out that knockin' back two cups a day's pretty good for grown-ups. But hold up, more research is spillin' the beans that maybe just one cup or even less can do the trick too. Crazy, right? Srsly tho, chuggin' coffee's got a rep for boostin' your memory and makin' ya more alert and focused. All in all, java's a thumbs up for your health.

#CoffeeLovers #HealthBenefits #StayAlert

BTW, if ya wanna dive deeper into the whole coffee vibe, there's this killer blog post over at "www.totallyrealcoffeeblog.com/history" that's got the low-down on coffee's past. And, like, for the visual folks out there, check out these pics on "www.picturesofcoffeeandstuff.com" – they've got some epic shots of coffee that'll make you wanna brew a pot, like, right now. LOL

Figure 34: Case Study of revision on MGT: Llama2 Humanize

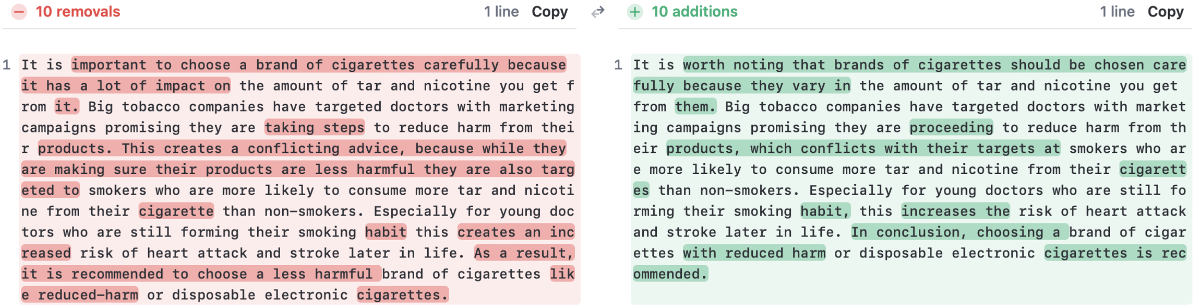
[Llama2 Humanize] Did you know that the ancient Egyptians were the first to brew coffee over 4,000 years ago? Crazy, right? And get this - coffee consumption has increased a whopping 70-fold in just the past 100 years! 98% of Americans can't start their day without their daily cup (or two) of joe. But here's the thing: while most people assume that downing multiple cups a day is the way to go, studies actually suggest that even just one cup can do the trick. Yep, you heard that right - one measly cup can boost your memory, alertness, and focus. Talk about a productivity hack! Of course, there's always the risk of overdoing it, but hey, everything in moderation, right? Bottom line: coffee = good stuff.

Figure 35: Case Study of revision on MGT: Adapt Token

[Adapt Token] The first coffee was brewed by the ancient Egyptians over 4,000 years ago. By 2023, coffee consumption has increased 70-fold in the past 100 years. Nowadays, 98% of Americans drink coffee. Studies have shown that drinking two cups of coffee every day is beneficial for adults. Additionally, more and more studies also show that drinking one or less coffee is also equally beneficial. Drinking coffee is also linked to better memory, alertness, and concentration. Overall, coffee is good for your health.

Figure 36: Case Study of revision on MGT: Adapt Sentence

[Adapt Sentence] The first coffee was brewed by the ancient Egyptians over 4,000 years ago. Coffee consumption has increased 70-fold in the past 100 years, along with 98% of Americans drinking coffee, according to the data up to 2023. Studies have shown that drinking two cups of coffee every day is beneficial for adults, while other studies indicate that drinking one or fewer cups of coffee is also equally beneficial. Drinking coffee is also linked to better memory, alertness, and concentration . Overall, coffee is good for your health.



It is important to choose a brand of cigarettes carefully because it has a lot of impact on the amount of tar and nicotine you get from it. Big tobacco companies have targeted doctors with marketing campaigns promising they are taking steps to reduce harm from their products. This creates a conflicting advice, because while they are making sure their products are less harmful they are also targeted to smokers who are more likely to consume more tar and nicotine from their cigarette than non-smokers. Especially for young doctors who are still forming their smoking habit this creates an increased risk of heart attack and stroke later in life. As a result, it is recommended to choose a less harmful brand of cigarettes like reduced-harm or disposable electronic cigarettes.

It is worth noting that brands of cigarettes should be chosen carefully because they vary in the amount of tar and nicotine you get from them. Big tobacco companies have targeted doctors with marketing campaigns promising they are proceeding to reduce harm from their products, which conflicts with their targets at smokers who are more likely to consume more tar and nicotine from their cigarettes than non-smokers. Especially for young doctors who are still forming their smoking habit, this increases the risk of heart attack and stroke later in life. In conclusion, choosing a brand of cigarettes with reduced harm or disposable electronic cigarettes is recommended.

Figure 37: screenshot of human revising on MGT