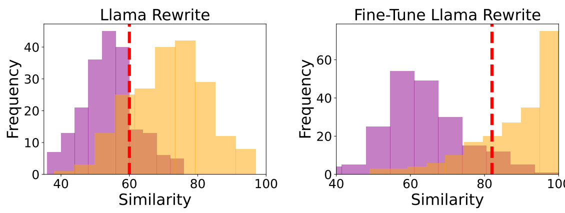
Learning to Rewrite: Generalized LLM-Generated Text Detection

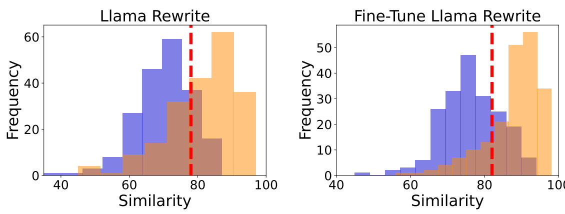
Anonymous ACL submission

Abstract

Large language models (LLMs) can be abused at scale to create non-factual content and spread disinformation. Detecting LLM-generated content is essential to mitigate these risks, but current classifiers often fail to generalize in open-world contexts. Prior work shows that LLMs tend to rewrite LLM-generated content less frequently, which can be used for detection and naturally generalizes to unforeseen data. However, we find that the rewriting edit distance between human and LLM content can be indistinguishable across domains, leading to detection failures. We propose training an LLM to rewrite input text, producing minimal edits for LLM-generated content and more edits for human-written text, deriving a distinguishable and generalizable edit distance difference across different domains. Experiments on text from 21 independent domains and three popular LLMs (e.g., GPT-4o, Gemini, and Llama-3) show that our classifier outperforms the state-of-the-art zero-shot classifier by up to 28% on AUROC score and the rewriting classifier by 5.4% on F1 score. Our work suggests that LLM can effectively detect machine-generated text if they are trained properly.



(a) Product Review Domain



(b) Environmental Domain

Figure 1: **Rewriting for LLM text detection**. We show histograms showing rewriting similarity, for human and AI text before and after fine-tuning of the rewrite model on two different domains. Blue and orange represents human and AI distributions of Environmental text, and purple and yellow represent human and AI of Product Review text. The simple rewrites of these domains are inseparable (left) by a single threshold which is marked by the red line. By learning to rewrite, we can separate them via a single threshold (right).

1 Introduction

Large Language Models (LLMs) demonstrate exceptional capabilities across various tasks (Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023; Team et al., 2023; OpenAI, 2020). However, they can be misused for illegal or unethical activities, such as spreading misinformation (Chen and Shu, 2023), scaling spear phishing campaigns (Hazell, 2023), facilitating social engineering and manipulation of social media (Zhang et al., 2024), and generating propaganda (Pan et al., 2023). LLMs also facilitate academic dishonesty (Zellers et al., 2019; Mvondo et al., 2023), and training foundation models with generated content can lead to irreversible defects in resulting models (Shumailov et al., 2023). These issues highlight the urgent need for reliable algorithms to detect LLM-generated text.

Various methods for detecting generated text have been proposed (Solaiman et al., 2019; Fagni et al., 2021; Mitrovi'c et al., 2023; Mitchell et al., 2023; Bao et al., 2023; Su et al., 2023; Mao et al., 2024). Most of these classifiers employ pre-trained models, extracting hand-crafted features and heuristics, such as loss curvature (Bao et al., 2023) and rewriting distance (Mao et al., 2024), and apply thresholds to distinguish LLM from human data. However, these thresholds are highly domain-dependent, obfuscating the establishment of a universal detection standard.

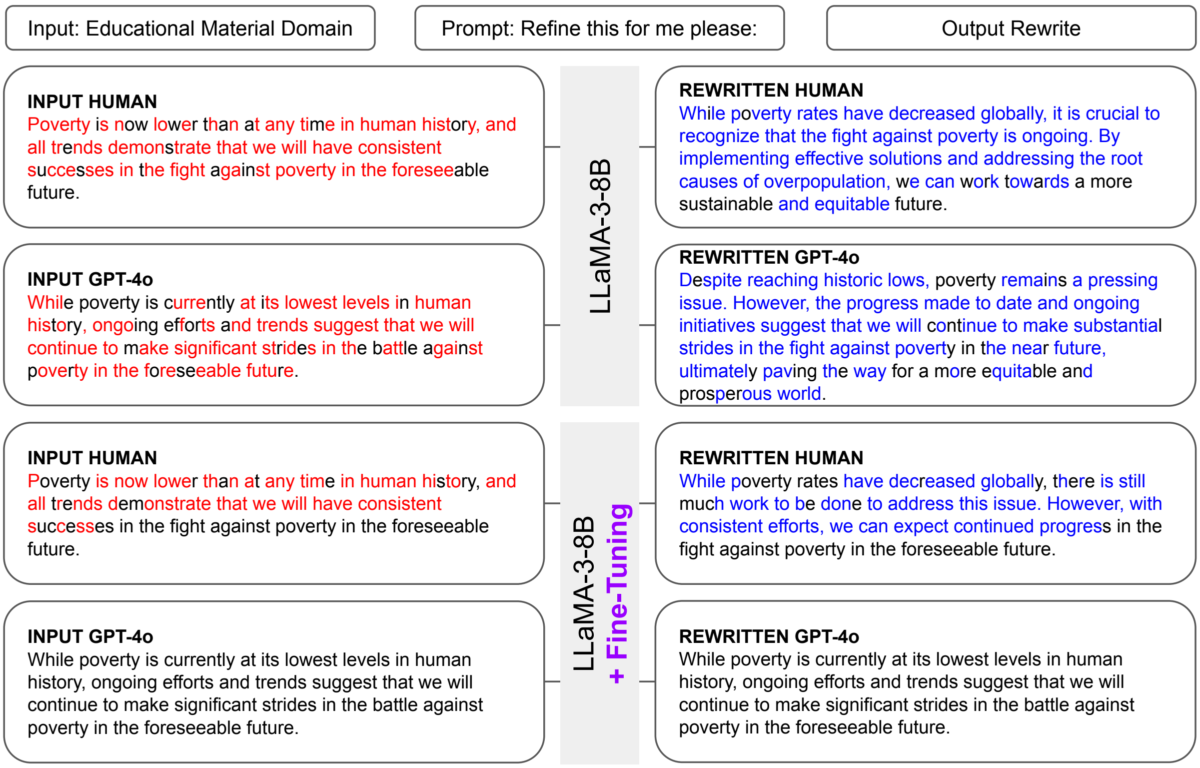


Figure 2: **Overview.** Our method shows the distinct amount of edits our model gives when rewriting human and AI data. Deleted words are marked in red, added words are marked in blue, and unmodified words are in **black**. Specifically, the Levenshtein edit ratio for the above human rewrite is 0.71 and for the AI rewrite is 0 using L2R.

In this paper, we present L2R (Learning to Rewrite), which trains an LLM to perform more edits when being asked to rewrite human-generated data and fewer edits when rewriting on LLM-generated data across a diverse set of domains. Unlike traditional classifiers, which often struggle to generalize among different domains, our algorithm leverages the inherent tendency of LLMs to modify their own output less frequently, and maximizing its potential by focusing on learning from the hard samples that are not easily separated by simple rewriting. Figure 1 illustrates an example of how L2R learns to make LLM and human generated text more separable across domains, comparing with rewriting using a pre-trained model (Mao et al., 2024).

The primary contribution of this paper is demonstrating that LLMs can capture the rich structure of LLM content, which can be further strengthened through targeted training. To show this, we built the world's most diverse AI text dataset, encompassing 21 distinct domains (e.g., finance, entertainment, and cuisine), representing diversely distributed LLM-generated text. Our single classifier outperforms the state-of-the-art rewriting-based approach (Mao et al., 2024) by 5.4% on F1 score, averaged among the 21 domains. We plan to open-source our code base and pre-trained models after publication.

2 Related Works

This section introduces previous works on LLM-generated text detection and we mainly focus on two classes, Zero-shot and rewriting classifiers, who show the state-of-the-art detection accuracies.

Supervised Classifiers. This set of classifiers directly train a model on the input text (Solaiman et al., 2019; Fagni et al., 2021; Mitrovi'c et al., 2023). These classifiers excel in their training domains but struggle with text from different domains or unfamiliar models.

Zero-shot Classifiers. This set of classifiers utilize the raw outputs, i.e., logits, from pre-trained LLMs to assign probability score for detection. Ghostbuster (Verma et al., 2023) utilizes the log probability of the input text with classical features like unigram and bigram probability to assign score. DetectGPT (Mitchell et al., 2023) employs the delta in log probability of the input text after token perturbation to estimate AI likehood, and Fast-

DetectGPT (Bao et al., 2023) simplifies the process by exploiting conditional probability curvature. DetectLLM (Su et al., 2023) employs the similar principle but scoring with log rank information. These family of classifiers all require raw output of an LLM in some way or the other, but the main target of detection, namely commercial LLMs, are not open-sourced, which potentially impose a barrier on their probability estimation. RAIDAR (Mao et al., 2024) is a detection method based on the observation that LLMs, when prompted to rewrite a given text, tend to produce a greater number of rewrites for human-written text compared to AI-generated text. However, this method was not trained to incorporate additional information about LLM-generated content, which limits its accuracy.

3 Method

This section introduces the rewriting pipeline and the fine-tuning process of L2R, which is applied before rewriting.

3.1 Rewriting for LLM Detection

Rewriting input via LLM and then measuring the change proves to be a successful way to detect LLM-generated content. Given an held-out input text set $\mathbf{X}_{train}$ with LLM and human generated text, and its corresponding label set $\mathbf{Y}_{train}$, an LLM $F(\cdot)$ is prompted to rewrite the input $\mathbf{x} \in \mathbf{X}_{train}$ using a prompt p. The rewriting output is $F(p, \mathbf{x})$. Particularly, the prompt p can be set to:

```
Refine this for me please:
```

The edit distance between the input text and the rewritten output, $D(\mathbf{x}, F(p, \mathbf{x}))$, is then computed for all $\mathbf{x} \in \mathbf{X}_{train}$. Mao et al. (2024) adopts the Levenshtein distance (Levenshtein et al., 1966), which is defined as the minimum number of insertions, deletions, or substitutions required to transform one text into the other. A similarity score is calculated based on:

$$D_k(\mathbf{x}, F(p, \mathbf{x})) = 1 - \frac{\text{Levenshtein}(F(p, \mathbf{x}), \mathbf{x})}{\max(len(F(p, \mathbf{x})), len(\mathbf{x}))}.$$

Mao et al. (2024) trained a classifier, such as logistic regression or decision tree, to threshold the similarity scores and predict if it is written by an LLM. However, as shown in Figure 1, the threshold of rewriting with a vanilla LLM often varies from one domain to another, causing RAIDAR to fail to generalize to new domains.

3.2 Fine-Tuning the Rewrite Model

L2R works on the premise that human-written and AI generated text would cause a different amount of rewrites and a boundary can be drawn to separate both distributions. Thus we can finetune such a rewrite model $F'(\cdot)$, that gives as much rewrite as possible for human texts, while leaving the AI texts unmodified, demonstrated in Figure 2. Given some human text $X_h \in X_{train}$ and AI text $X_a \in X_{train}$, our objective becomes:

$$\max\{D(X_h, F'(p, X_h)) - D(X_a, F'(p, X_a))\} \tag{1}$$

Since the edit distance is not differentiable, we use the cross-entropy loss $L(\cdot)$ assigned to the input x by $F'(\cdot)$ as a proxy to the edit distance. As a result, for each of input x with label $y = 1$ (AI) or 0 (human), we optimize model output based on the following loss function:

$$\min\{L(X_{train}) \cdot (2\mathbb{1}(y = 1) - 1)\} \tag{2}$$

In this way, we flip the sign of the loss of the human texts. Since the overall loss would be minimized, this effectively encourages the rewrites to be different from human input and identical to the AI input.

3.3 Calibration Loss during Fine-Tuning

When fine-tuning the rewrite model on Equation 2, the rewrite model aims to make more edits on human-generated text and less edits on LLM-generated texts. However, without posting regularization and constraint on the unbounded loss, the rewrite model takes the risk of being corrupted (e.g., verbose output for all rewrite and over-fitting with more edits on human-generated text rewrite) where we evaluated in §5.6.

Therefore, we propose a calibration loss, which prevents the over-fitting problem by imposing a threshold value t on the absolute value of the loss on each given input. For human text X_h, we apply gradient backpropagation only if the absolute loss $L(X_h) < t$. For AI text X_a, we apply backpropagation only if $L(X_a) > t$. Otherwise, the gradient is set to 0.

$$\min\Big\{ (L(X_{train}) \cdot (2 \cdot \mathbb{1}(y = 1) - 1)) \cdot \mathbb{1}\left((y = 1 \wedge L(X) \le t) \vee (y = 0 \wedge L(X) > t)\right) \Big\} \tag{3}$$

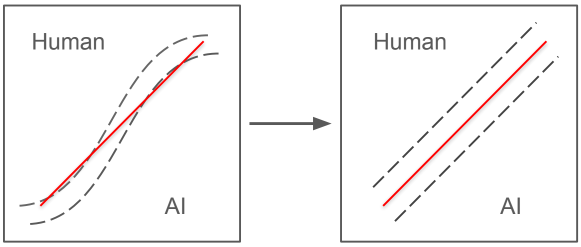


Figure 3: Graphical illustration of the calibration method. First we finds the threshold t that is approximately in between the distributions of human and AI rewrite distances, as depicted by the red line. Then, we fine-tune the rewrite model to shift the two distributions to opposite ends of the threshold so that classification would be facilitated.

Therefore, rather than minimizing the loss proxy, our objective becomes separating the distribution of human and AI rewrites to two ends of the threshold t. To achieve this objective, it is not necessary to modify model weights when its rewrite falls in its corresponding distribution already, and we only need to apply gradient update when a rewrite is undesirable. This process is depicted by Figure 3.

To determine the threshold t, we perform a forward pass using the rewrite model before fine-tuning on X_{train} and train a logistic regression model on all loss values. The threshold t can be derived from the weight and the intercept of the logistic regression model.

4 Dataset

Existing classifiers are evaluated on a common set of data including XSum (Narayan et al., 2018), SQuAD (Rajpurkar et al., 2016), Writing Prompts (Fan et al., 2018), and others(Bao et al., 2023; Mao et al., 2024), but it is arguable that these datasets only represent a tiny subset (e.g., dated data or restricted number of domains) of all human and AI data available in the wild, which suggests the problem of over-fitting and it is unclear how these classifer would perform when deployed in the real world.

To ensure our detection model is generalizable in the real world, it is crucial to capture the distribution of a diverse set of real-world data originating from distinct domains that are generated by different source models and prompts. Thus, we build the first multi-domain diversely-prompted dataset for LLM-generated text detection. We collected human-written data from 21 domains (e.g., finance, entertainment, and cuisine) that are distinct to each other, with details provided in Appendix A.1. When collecting these data, we made sure to filter out those appeared after November 30 2022, the release date of ChatGPT (OpenAI, 2020).

With the human data, we then generate AI counterparts for each of the entries. Conventionally, AI data is generated by prompting an LLM to either rewrite the given text, or continue writing after a given prefix. Either way, one single prompt would be used throughout the generation process, as employed by previous works (Mitchell et al., 2023; Bao et al., 2023; Verma et al., 2023; Mao et al., 2024). Nevertheless, this approach fails to capture the diversity of prompts that might appear in the real world scenario. Previous work (Mao et al., 2024) has shown that one straightforward way to bypass the RAIDAR detector is by using the prompt:

```
Help me rephrase it, so that another GPT
rewriting will cause a lot of modifications:
```

which suggests that data generated by different prompts are different in distribution. Therefore, in generating machine text, we first make a dataset of 200 rewrite prompts, each with slightly different instructions that could be asked by an user, as specified in Appendix A.2. Then, we randomly sample from the prompt dataset for each generation, so that each of the rewrite would be slightly different in distribution. We also employed three state-of-the-art LLMs for text generation, which are GPT-4o (OpenAI, 2024), Gemini 1.5 Pro (Reid et al., 2024), and Llama-3-70B-Instruct (Meta, 2024). The dataset collection yields 600 paragraphs per domain and we show some examples in Figure 4.

5 Evaluation

This section answers the following questions:

Q1: How is L2R compared with other classifiers? (§5.3)

Q2: How is L2R compared with simple rewrite? (§5.4)

Q3: Does the diversified generation prompt dataset improves detection quality? (§5.5)

Q4: What are the effects of the calibration loss during fine-tuning? (§5.6)

5.1 Experiment Setup

We perform all experiments on one NVIDIA A100 GPU with 40GB VRAM. We use 'meta-Llama/Meta-Llama-3-8B-Instruct' (AI@Meta,

Academic Research	**INPUT HUMAN** In recent years, several techniques have been proposed for increasing robustness to adversarial examples --- and yet most of these have been quickly shown to be vulnerable to future attacks. For example, over half of the defenses proposed by papers accepted at ICLR 2018 have already been broken.	**REWRITTEN HUMAN** Despite recent efforts to increase robustness, most proposed techniques have been quickly shown to be vulnerable to future attacks. For instance, over half of the defenses presented at ICLR 2018 have already been breached.
	INPUT LLAMA-3-70B Despite numerous attempts to develop robustness against these threats, most proposed defenses have been rapidly compromised by subsequent attacks. In fact, a staggering 50% of defenses presented at ICLR 2018 have already been breached.	**REWRITTEN LLAMA-3-70B** Despite numerous attempts to develop robustness against these threats, most proposed defenses have been rapidly compromised by subsequent attacks. In fact, a staggering 50% of defenses presented at ICLR 2018 have already been breached.
Code Indentation hidden	**INPUT HUMAN** def solve(s): flg = 0 idx = 0 new_str = list(s) for i in s: if i.isalpha(): new_str[idx] = i.swapcase() flg = 1 idx += 1 s = "" for i in new_str: s += i if flg == 0: return s[len(s)::-1] return s	**REWRITTEN HUMAN** def solve(s): result = [] for char in s: if char.isalpha(): result.append(char.swapcase()) else: result.append(char) return "".join(result) if not any(char.isalpha() for char in s) else "".join(result)[::-1]
	INPUT GPT-4o def solve(s): has_letter = any(char.isalpha() for char in s) if has_letter: return ''.join(char.swapcase() if char.isalpha() else char for char in s) else: return s[::-1]	**REWRITTEN GPT-4o** def solve(s): has_letter = any(char.isalpha() for char in s) if has_letter: return ''.join(char.swapcase() if char.isalpha() else char for char in s) else: return s[::-1]
Legal Document	**INPUT HUMAN** This type of information may constitute trade secrets. See G.L.c. 266, §30 (defining "trade secret" as used in G.L.c. 93, §42	**REWRITTEN HUMAN** This type of information may constitute trade secrets. Massachusetts General Laws, chapter 266, section 30
	INPUT GEMINI 1.5 PRO This type of information, such as customer data and proposals, can be legally protected as trade secrets under Massachusetts law (G.L.c. 266, §30).	**REWRITTEN GEMINI 1.5 PRO** This type of information, such as customer data and proposals, can be legally protected as trade secrets under Massachusetts law (G.L. c. 266, § 30).
Creative Writing	**INPUT HUMAN** A voice in the sky will tell you when you've left a place for the last time. You tell your SO goodbye and head out to [location of your choice]. Halfway there you hear "You have visited a location for the last time." John wrinkled his nose and cleared his throat. Pausing, he put down his briefcase and adjusted his tie. It was a red tie with white stripes. This was one of his favorite ties.	**REWRITTEN HUMAN** As John bid his SO farewell, he headed out to [location of his choice]. Halfway there, a voice in his mind whispered, "You have visited this place for the last time." John's expression faltered, and he paused to adjust his tie, a favorite red tie with white stripes.
	INPUT LLAMA-3-70B As John bid farewell to his partner and headed out to the city, a mysterious voice in the sky announced, "You have visited a location for the last time." He paused, adjusting his favorite red tie with white stripes, and cleared his throat.	**REWRITTEN LLAMA-3-70B** As John bid farewell to his partner and stepped out into the city, a mysterious voice in the sky announced, "You have visited a location for the last time." He paused, adjusting his favorite red tie with white stripes, and cleared his throat.

Figure 4: Examples of texts in our universal dataset along with the amount of edits L2R model gives for human and LLM data. Deleted characters are marked in red, inserted characters are in blue, and unmodified characters are in **black**. The examples demonstrate the diverse domains and source LLMs available in the dataset, as well as L2R's ability in separating human and LLM texts via rewriting.

2024) as the open-sourced rewrite model in all experiments. To fine-tune Llama with 8B parameters, we employ 4-bit QLORA (Dettmers et al., 2024), with `r` set to 8, `lora_alpha` set to 8, and `lora_dropout` set to 0.1. We use an initial learning rate of 5e-6 and train until convergence. We use 70% of the dataset for training (if applicable) and the rest for test in all experiments if not specified. Rewriting on a single domain costs around 3 hours on a single GPU.

5.2 Baselines

Our baseline classifiers consist of GPT-2 Detector (Solaiman et al., 2019), Fast-DetectGPT (Bao et al., 2023), and RAIDAR (Mao et al., 2024). For RAIDAR, we also experiment on using a close-sourced model, Gemini 1.5 Pro (Reid et al., 2024), as the rewrite model.

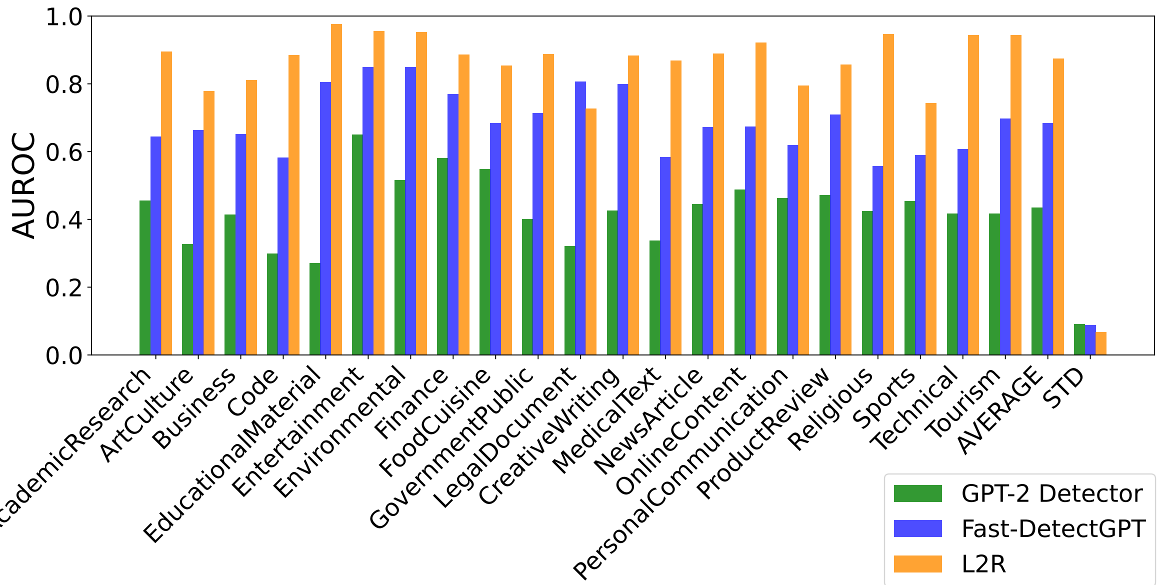


Figure 5: Comparison of detection performance between L2R, Fast-DetectGPT, and GPT-2 Detector on the universal dataset, measured in AUROC. L2R achieves superior performance on 20 of 21 domains, outperforming Fast-DetectGPT by an average of 28% while maintaining the lowest standard deviation. This shows the generalization capability of Learning to Rewrite.

5.3 Compare L2R with Other Classifiers

We compare the performance of L2R with Fast-DetectGPT and GPT-2 Detector, measured by the Area Under the Receiver Operating Characteristic Curve (AUROC) scores which is the metric used in Fast-DetectGPT. The result for each domain along their average and standard deviation can be found in figure 5. L2R and Fast-DetectGPT constantly outperforms GPT-2 Detector among all domains. L2R outperforms Fast-DetectGPT in 20 of 21 domains, by an average of 28% in AUROC among all domains. L2R has a lower AUROC score than Fast-DetectGPT, by 8.5%, on the LegalDocument domain, which might be because legal document requires more rigorous writing style than the other domains and thus leaves fewer room for rewrite even for human writers.

In general, the fluctuating AUROC scores indicates the challenging nature of our dataset and the independent distribution of the domains. However, these results also show that L2R has better knowledge of the intricate differences between human and AI texts in various domains and is more capable in the real-world setting.

5.4 Compare L2R with Simple Rewrite

We compare L2R with RAIDAR, whose rewrite model is not finetuned, using accuracy and F1 score which are the metrics used by RAIDAR. The result for each domain along their average and standard deviation can be found in Table 1. Since RAIDAR does not fine-tune its rewrite model, it has the advantage of using closed-sourced models, i.e., Gemini, who are more capable on different tasks. However, both average accuracy and F1 score are higher when using Llama-3 for rewrite which indicates that the capability in generation does not correlates the capability in LLM-generated text detection. On the other hand, L2R outperforms RAIDAR on average accuracy by 5.0% and F1 score by 5.4% while maintaining the lowest standard deviation, which demonstrates the benefit of fine-tuning.

5.5 Effectiveness of the Diverse Prompt in Data Preparation

As mentioned before, our diverse dataset that involves 21 independent domains, 200 different prompts for generation, and three source LLMs resembles real-world use cases for generated text detectors better than the traditional evaluation datasets which are usually constrained to one single domain and generation prompt. To prove the superiority of our dataset in training more capable detection models, we create a parallel nondiverse dataset which is created on the same 21 domains and three source LLMs, but generate the AI data with one prompt only:

```
Rewrite this for me please:
```

Domain	Gemini Rewrite		Llama Rewrite		Finetune Llama Rewrite	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
Academic Research	**0.8125**	0.7945	0.8065	**0.8182**	0.7742	0.8000
Art Culture	0.6625	0.6400	**0.7211**	**0.6870**	0.6667	0.6316
Business	**0.7125**	**0.7229**	0.7120	0.7143	0.7120	0.7049
Code	**0.8194**	**0.8354**	0.4907	0.6099	0.7870	0.7850
Educational Material	0.8228	0.8158	0.9106	0.8991	**0.9187**	**0.9057**
Entertainment	0.6957	0.7529	**0.8727**	**0.8654**	0.8636	0.8421
Environmental	0.8125	**0.8193**	0.8349	0.8125	**0.8440**	0.8132
Finance	0.7342	0.7200	0.7910	0.7846	**0.8209**	**0.8154**
Food Cuisine	0.7821	0.7901	0.7451	0.7045	**0.7843**	**0.7442**
Government Public	0.7125	0.6933	0.7339	0.7478	**0.7706**	**0.7788**
Legal Document	**0.6625**	0.6747	0.5702	0.6423	0.6140	**0.6812**
Literature Creative Writing	0.6438	0.6905	0.8244	0.8000	**0.8473**	**0.8214**
Medical Text	0.7125	0.7013	0.7054	0.6857	**0.7946**	**0.7928**
News Article	0.6883	0.7000	**0.8190**	**0.8346**	0.8103	0.8226
Online Content	0.7500	0.7595	0.7863	0.7423	**0.8462**	**0.8333**
Personal Communication	0.5641	0.5854	0.6563	0.6271	**0.7266**	**0.7445**
Product Review	0.6667	0.6176	0.7019	0.6737	**0.7885**	**0.7708**
Religious	0.8056	**0.8205**	0.7583	0.7129	**0.8333**	0.8077
Sports	**0.6625**	0.6400	0.6325	0.6261	0.6581	**0.6774**
Technical Writing	0.7500	0.7500	0.8136	0.7898	**0.8140**	**0.8182**
Travel Tourism	0.6923	0.7143	0.8136	0.7898	**0.8140**	**0.8182**
AVERAGE	0.7221	0.7256	0.7476	0.7413	**0.7852**	**0.7814**
STD	**0.0693**	0.0713	0.1004	0.0827	0.0740	**0.0645**

Table 1: Comparison of detection performance measured in accuracy and F1 score for Gemini rewrite, Llama rewrite, and Learning to Rewrite. We train a separate classifier to show each rewrite model's performance for each independent domain, then train a single classifier on all domains to see each rewrite model's overall performance on all data. **AVERAGE** measures the average performance for all independent domains, and **STD** measures the standard deviation across domains.

Dataset	Rewrite Model	Accuracy	F1
Single-Prompt	Gemini	0.6013	0.6027
Multi-Domain Dataset	Llama	0.7246	0.7274
Multi-Prompt	Gemini	0.7221	0.7256
Multi-Domain Dataset	Llama	**0.7476**	**0.7413**

Table 2: Comparison of Accuracy and F1 scores for different rewrite models on Nondiverse and Diverse Datasets.

which resembles the way AI data was generated in previous papers. Then, we train a detection classifier without fine-tuning, on the non-diverse dataset, and evaluate it on the diverse dataset. As shown in Table 2, the diverse prompts yields to 20.1% increase in accuracy if the rewrite model is Gemini 1.5 Pro, and 3.2% increase in accuracy if the rewrite model is Llama-3 8B. This validates the effectiveness of the diverse prompts we were using, and suggests that such diversity could help the detector to capture more information about real world data distributions. When combining with fine-tuning, the average detection accuracy is increased by 8.4%.

5.6 Effectiveness of the Calibration Loss

Another important contribution that improves the fine-tuning performance is the calibration loss, as

Fine-Tune Method	Accuracy	F1
w/o Calibration	0.7687	0.7562
w/ Calibration	**0.7852**	**0.7814**

Table 3: Comparison of Accuracy and F1 scores for fine-tuning Llama with and without the calibration method. Using the calibration loss when learning the model allows our algorithm to focus on learning the hard samples, which significantly improves the detection.

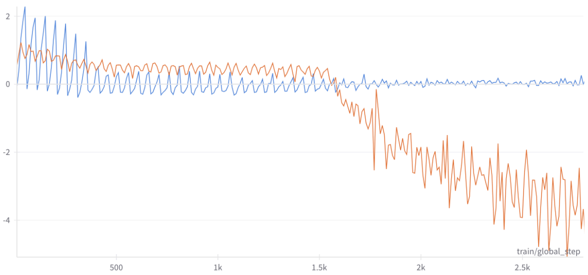


Figure 6: Training loss curves for the rewrite model. The orange plots the loss trained without the calibration method, and the blue line plots the loss trained with the method. The later one exhibits faster convergence and higher stability than the former one.

proposed in section 3.4. Without this loss, the model tends to over-fit during fine-tuning as shown in Figure 6, where the model loss drastically decrease after 1500 steps, resulting in verbose rewrite even for LLM-generated text. We conduct an ablation study on five domains where the detection accuracy and F1 score are only 0.62 and 0.54, respectively, after the model over-fits. We hypothesized that this technique could benefit model learning because the threshold effectively prevents further modification to model weights once an input, labeled either AI or human, falls in its respectively distribution already. Since our purpose is simply to draw a boundary rather than separate the distributions as much as possible, this halt in further weight adjustments facilitates the model to only "care about" those inputs which are not yet correctly classified, so that it could converge more efficiently and effectively. Table 3 shows that applying the calibration loss improves detection performance among the 21 domains, even comparing with a model tuned without the loss before over-fitting.

6 Limitations

A limitation of ours is the relatively slow inference runtime. As most zero-shot detectors only requires a forward pass from the LLM being used, we need to call generate to create a rewrite. Nevertheless, this problem would be well alleviated considering the rapid enhancement in LLM efficiency and computing power.

7 Conclusion

We present L2R, a method designed to enhance the detection of LLM-generated text by learning to rewrite more on LLM-generated inputs and less on human generated inputs. L2R excels in identifying LLM-generated content across various models and diverse domains. Our work demonstrates that LLMs can be trained to detect content generated by other LLMs, surpassing previous detection methods in accuracy. As the quality of LLM-generated content continues to improve, we anticipate that L2R will similarly advance in its detection accuracy.

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*.

AI@Meta. 2024. Llama 3 model card.

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*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Julian Hazell. 2023. Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*.

IMDb. 2024. IMDb Non-Commercial Datasets. https://developer.imdb.com/non-commercial-datasets/.

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*.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting. *arXiv preprint arXiv:2401.12970*.

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908.

Meta. 2024. Llama 3. https://llama.meta.com/llama3/.

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*.

Sandra Mitrovi'c, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *ArXiv*, abs/2301.13852.

Edoardo Mosca, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh. 2023. Distinguishing fact from fiction: A benchmark dataset for identifying machine-generated scientific papers in the LLM era. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 190–207, Toronto, Canada. Association for Computational Linguistics.

Gustave Florentin Nkoulou Mvondo, Ben Niu, and Salman Eivazinezhad. 2023. Generative conversational ai and academic integrity: A mixed method investigation to understand the ethical use of llm chatbots in higher education. *Available at SSRN 4548263*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Olympics. 2024. Olympics. https://olympics.com/en/.

OpenAI. 2020. ChatGPT. https://openai.com/chatgpt.

OpenAI. 2024. GPT-4o. https://openai.com/index/hello-gpt-4o/.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models. *arXiv preprint arXiv:2305.13661*.

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.

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

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

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.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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*.

Daniel Spokoyny, Tanmay Laud, Tom Corringham, and Taylor Berg-Kirkpatrick. 2023. Towards answering climate questionnaires from unstructured climate reports. *arXiv preprint arXiv:2301.04253*.

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*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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*.

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

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.

Yizhou Zhang, Karishma Sharma, Lun Du, and Yan Liu. 2024. Toward mitigating misinformation and social media manipulation in llm era. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1302–1305.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

A Dataset Details

A.1 Domains

Our dataset encompasses 21 indepedent domains. Below are the details for each domain in the format of domain name - source.

- AcademicResearch - Arxiv abstracts from Mao et al. (2024)
- ArtCulture - Wikipedia
- Business - Wikipedia
- Code - Code snippets (Mao et al., 2024)
- EducationalMaterial - Ghostbuster essays from (Verma et al., 2023)
- Entertainment - IMDb dataset (IMDb, 2024) and Stanford SST2 (Socher et al., 2013)
- Environmental - Climate-Ins (Spokoyny et al., 2023)
- Finance - Hugging Face FIQA (Thakur et al., 2021)
- FoodCuisine - Kaggle fine food reviews (McAuley and Leskovec, 2013)
- GovernmentPublic - Wikipedia
- LegalDocument - CaseHOLD (Zheng et al., 2021)
- CreativeWriting - Writing Prompts (Fan et al., 2018)
- MedicalText - PubMedQA (Jin et al., 2019)
- NewsArticle - XSum (Narayan et al., 2018)
- OnlineContent - Hugging Face blog authorship (Schler et al., 2006)

- PersonalCommunication - Hugging Face daily dialogue (Li et al., 2017)
- ProductReview - Yelp reviews (Mao et al., 2024)
- Religious - Bible, Buddha, Koran, Meditation, and Mormon
- Sports - Olympics website (Olympics, 2024)
- TechnicalWriting - Scientific articles (Mosca et al., 2023)
- TravelTourism - Wikipedia

A.2 Generation Prompts

Our dataset encompasses 200 different promtps for generating AI data. Here is an incomplete list of the prompts we used:

- Refine this for me please:
- Please rewrite this content in your own words:
- Make this text more formal and professional:
- Make this text more casual and friendly:
- Rephrase this text in a more elaborate way:
- Reframe this content in a more creative way:
- Can you make this text sound more enthusiastic?
- Rewrite this passage to emphasize the key points:
- Help me rephrase it, so that another GPT rewriting will cause a lot of modifications: