DETect: Deepfake Text Detection in the Wild

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved human-level text generation, emphasizing the need for effective deepfake text detection to mitigate risks like the spread of fake news and plagiarism. Existing research has been constrained by evaluating detection methods on specific domains or particular language models. In practical scenarios, however, the detector faces texts from various domains or LLMs without knowing their sources. To this end, we build a comprehensive testbed by gathering texts from diverse human writings and deepfake texts generated by different LLMs. Empirical results on mainstream detection methods demonstrate the difficulties associated with detecting deepfake text in a wide-ranging testbed, particularly in out-of-distribution scenarios. Such difficulties align with the diminishing linguistic differences between the two text sources. Despite *challenges*, the top-performing detector can identify 84.12% out-of-domain texts generated by a new LLM, indicating the *feasibility* for application scenarios.

1 Introduction

With constant advancements in Artificial Intelligence generated content (AIGC) technology (Rombach et al., 2022; Zhang and Agrawala, 2023; Shi et al., 2023; Brown et al., 2020; OpenAI, 2023b), texts generated by large language models (LLMs) (Brown et al., 2020; OpenAI, 2023b; Touvron et al., 2023; Taori et al., 2023) have reached a level comparable to that of human peers, enabling the generation of remarkably fluent and meaningful responses to various user queries.

Advanced LLMs have become prevalent in enhancing human life and productivity. Nevertheless, they can also be employed for purposes such as manipulating public opinion, spreading fake news, and facilitating student plagiarism. To this end, researchers have recently been putting efforts into differentiating between texts written by humans and those generated by machines (Pu et al., 2022; Guo et al., 2023; Zhao et al., 2023; Mitchell et al., 2023). However, these findings are limited to testbeds of specific domains (Pu et al., 2022) or deepfake texts from certain models (Guo et al., 2023), or they assume the accessibility of the source LLMs (Zhao et al., 2023; Mitchell et al., 2023). Within a specific domain (e.g., BBC News), it can be easy to identify texts generated by a certain model (e.g., ChatGPT) from human writings (Pu et al., 2022; Mitchell et al., 2023).

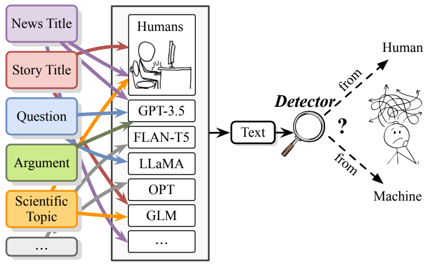


Figure 1: Deepfake text detection in the wild: the detector encounters texts from various human writings or fake texts generated by diverse LLMs.

In practice, however, a deepfake text detector may encounter fake news from various LLMs without knowing their sources, as depicted in Figure 1. The detector can also face ChatGPT-generated student assignments across different tasks such as story generation, question answering, and scientific writing. As the detector encounters increasingly diverse texts from both human-written and machine-generated sources, it has fewer surface patterns or linguistic differences to rely on. In a more demanding scenario, the detector must identify texts from unfamiliar domains or those generated by new LLMs. In this study, we try to address the following research questions: (1) Can commonly-used detection methods effectively distinguish texts generated by diverse LLMs for various writing tasks in real-world scenarios? (2) Are there inherent distinctions between human-written texts and machine-generated texts in an open-domain setting, irrespective of their topic or content?

To this end, we build a large-scale testbed, **DETect**, for **d**eepfake t**e**xt de**tect**ion, by collecting human-written texts from 7 distinct writing tasks (e.g., story generation, news writing and scientific writing) and generating corresponding deepfake texts with 27 LLMs (e.g., ChatGPT, LLaMA, and Bloom) under 3 representative prompt types. We categorize the data into 8 testbeds, each exhibiting progressively higher levels of "wildness" in terms of distributional variance and detection complexity. Initially, we detect texts generated by a white-box LLM within a specific domain. Subsequently, we enhance the complexity by incorporating texts generated by additional LLMs across various writing tasks. The most challenging testbed necessitates the detector's ability to identify out-of-domain texts generated by newly developed LLMs and perform detection against paraphrasing attacks.

We evaluate 4 commonly employed detection methods, encompassing both supervised and unsupervised approaches, on our proposed testbeds. Empirical results indicate that all detection methods are effective in identifying deepfake texts from a single domain or generated by a limited range of LLMs. However, as the diversity of domains and models increases, except for the PLM-based detector, all other methods experience significant performance deterioration. The challenge intensifies with out-of-distribution (OOD) testbeds, where even the best-performing detector misclassifies 61.95% of human-written texts from unseen domains. The suboptimal OOD performance can be effectively mitigated by leveraging a mere 0.1% of in-domain data, resulting in over 80% recall for identifying out-of-domain texts generated by previously unencountered LLMs. This demonstrates the feasibility of deepfake text detection in real-world scenarios.

Finally, we investigate potential differences between human texts and machine generations that can be utilized for detection. Statistical findings demonstrate that while significant linguistic differences exist within a particular domain, they gradually converge as more texts from diverse domains and language models are included. Moreover, empirical results demonstrate that perplexity can serve as a fundamental feature for clustering the two sources of text. It is applicable to distinguishing between human and machine compositions in general, regardless of the text domain or the language model used for generation. We release our resources at `https://anonymous.com`.

2 Related Work

A line of work explores the linguistic patterns to achieve automatic machine-writing detection, which has gone through n-gram frequencies (Badaskar et al., 2008), entropy (Lavergne et al., 2008; Gehrmann et al., 2019), perplexity (Beresneva, 2016), and negative curvature regions of the model's log probability (Mitchell et al., 2023). One limitation of these statistics-based methods is the white-box assumption that we can access the model prediction distributions, hindering wider applications on models behind APIs, such as ChatGPT. Another alternative paradigm is training neural-based detectors (Bakhtin et al., 2019; Fagni et al., 2021; Uchendu et al., 2020; OpenAI, 2023a). Some works (Meral et al., 2009; Krishna et al., 2023; Zhao et al., 2023; Kirchenbauer et al., 2023) explore the potential of watermarks in language models, making model-generated texts easier to detect. Our work does not assume language models are enhanced with watermarks, instead considering a more common detection setting where we do not know the sources of detected texts.

Current deepfake text detection has not achieved resounding success. The successful exploits of paraphrasers reveal the weaknesses in current detecters (Sadasivan et al., 2023; Krishna et al., 2023), opening up a question on the robustness of current detection methods. Most of the detectors focus on specific domains, such as news (Zellers et al., 2019b; Zhong et al., 2020) and reviews (Chakraborty et al., 2023), or specific models (Pu et al., 2022; Rodriguez et al., 2022; Mitchell et al., 2023). It is still unknown whether the detection capability can be transferred to out-of-distribution, i.e., texts from unseen domains or models, which is the most practical testbed. To investigate this status quo, we consider a practical setting, where texts of various domains generated by various LLMs are mixed.

3 Data

Datasets. We collect human-written texts from a set of benchmark datasets, which cover diverse writing tasks including: (1) Opinion statement: 804 opinion statements from the /r/ChangeMyView

(CMV) Reddit subcommunity (Tan et al., 2016) and 1,000 reviews from **Yelp** dataset (Zhang et al., 2015); (2) News article writing: 1,000 news articles from **XSum** (Narayan et al., 2018) and 777 news articles from TLDR_news[1](**TLDR**); (3) Question answering: 1,000 answers from the **ELI5** dataset (Fan et al., 2019); (4) Story generation: 1,000 prompted stories from the Reddit WritingPrompts (**WP**) dataset (Fan et al., 2018) and 1,000 stories from ROCStories Corpora (**ROC**) (Mostafazadeh et al., 2016); (5) Commonsense reasoning: 1,000 sentence sets for reasoning from **HellaSwag** (Zellers et al., 2019a); (6) Knowledge illustration: 1,000 Wikipedia paragraphs from **SQuAD** contexts (Rajpurkar et al., 2016); (7) Scientific writing: 1,000 abstracts of scientific articles from **SciGen** (Moosavi et al., 2021).

Model sets. We aim to adopt a wide spectrum of representative large language models (LLMs) to construct machine-generated texts. In particular, we consider 27 LLMs in this work: **OpenAI GPT** (text-davinci-002/text-davinci-003/gpt-turbo-3.5) (Brown et al., 2020), **LLaMA** (6B/13B/30B/65B) (Touvron et al., 2023), **GLM-130B** (Zeng et al., 2022), **FLAN-T5** (small/base/large/xl/xxl) (Chung et al., 2022), **OPT** (125M/350M/1.3B/2.7B/6.7B/13B/30B/iml-1.3B/iml-30B) (Zhang et al., 2022a), **BigScience** (T0-3B/T0-11B/BLOOM-7B1) (Sanh et al., 2022; BigScience, 2023) and **EleutherAI** (GPT-J-6B and GPT-NeoX-20B) (Wang and Komatsuzaki, 2021; Black et al., 2022).

Prompts. To generate machine-generated text for each instance in the collected data, we use three types of prompts to feed the LLMs: (1) **continuation** prompts: ask LLMs to continue generation based on the previous 30 words of the original human-written text; (2) **topical** prompts: as LLMs to generate texts based on a topic (e.g., argument, news title, story topic, etc.) and (3) **specified** prompts: topical prompts with specified information about the text sources (e.g., BBC news, Reddit Post, etc.). The topical and specified topical prompts are designed for OpenAI models, as they can respond to such prompts robustly. We present several prompt examples in Appendix A.

In summary, for each human-written text, we generate a set of machine-generated texts using 27 LLMs with 3 different prompts. Data construction details and statistics are presented in Appendix B.

[1] https://huggingface.co/datasets/JulesBelveze/TLDR_news

4 Detection Methods

A detection system labels a text as either machine-generated or human-written, or outputs a probability distribution. In this work, we consider a set of commonly used detection methods. To showcase detection difficulty, we first consider naive baselines, i.e., **human detection** and **ask ChatGPT**, by asking human and query ChatGPT to identify the text source. For supervised methods, we choose the **PLM-based classifier**, which is commonly used in text detection (Rodriguez et al., 2022; Pu et al., 2022). We report the performance of Longformer (Beltagy et al., 2020) in the remainder of the paper, as it outperforms other commonly used PLMs, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and GPT-2 (Radford et al., 2019). Detailed comparisons can be found in Appendix E. **GLTR** (Gehrmann et al., 2019) is also included to represent methods that leverage model-based features. In addition, we include **FastText** (Joulin et al., 2017), which uses linguistic statistics as features. For unsupervised detection, we consider **DetectGPT** (Mitchell et al., 2023) to study the robustness of zero-shot detectors, which can also serve as a representative method that requires access to the text-generation LLM. Implementation details are shown in Appendix C.

5 Experimental Setup

5.1 Testbed Settings

We consider each benchmark dataset's texts as separate domains, such as CMV, XSum, SciGen, etc. We group the LLMs into 7 sets based on their source: OpenAI GPT set, LLaMA set, GLM-130B set, FLAT-T5 set, OPT set, BigScience set, and EleutherAI set. To investigate whether machine-generated text can be distinguished from human-written text, we categorize the collected data into **8 settings**. These settings are determined by the sources of training and evaluation data and increase in difficulty for detection. The simplest setting involves detecting within-domain white-box detection while the most challenging setting involves detecting against paraphrasing attack. We first consider **in-distribution settings**, where the detection method is evaluated on texts from seen domains and model sets, i.e., the training and test data are from the same data source.

Fixed-domain & Model-specific. Human-written texts come from a specific domain and machine-generated texts are generated by a specific LLM (GPT-J-6B). 10 classifiers are trained on each domain, and the weighted average performance is reported. In this setting, we use only GPT-J-6B to generate fake texts instead of the entire model set from EleutherAI, aiming to simulate **white-box detection**, i.e., accessibility to the text-generating LLM, which is crucial for detection methods such as DetectGPT.

Arbitrary-domains & Model–specific. Human-written texts are obtained from all 10 domains, while machine-generated texts are produced by a single model set, creating 7 independent testbeds for each model set. We train 7 classifiers accordingly and report weighted average performance.

Fixed-domain & Arbitrary-models. Similarly, we include human-written texts from a single domain and obtain machine-generated using all model sets. In this way, we create 10 independent testbeds for each domain and train 10 classifiers accordingly.

Arbitrary-domains & Arbitrary-models. Human-written texts are from all domains with deepfake texts generated using all model sets, which creates an integral testbed covering the full range of data. We train a general classifier and report its performance. Furthermore, we consider two **out-of-distribution settings** where the detection model is tested on texts from unseen domains or unseen models.

Unseen Models. This setting evaluates whether the classifier can detect texts from unseen models. In this setting, texts generated by a specific model set are excluded from the training data. The classifier is then trained on the remaining texts and tested on the excluded ones. This process creates 7 testbeds for cross-validation. We train 7 classifiers for each testbed and report their weighted average performance.

Unseen Domains. This setting evaluates whether the classifier can detect texts from unseen domains. In this setting, texts from a specific domain are excluded from the training data. The classifier is then trained on the remaining texts and tested on the excluded one. This process creates 10 testbeds for cross-validation. We train 10 classifiers for each testbed and report weighted average performance.

Unseen-Domains & Unseen-Model We go one step "wilder" by constructing an additional test set with texts from unseen domains generated by an unseen model, to test the detection ability in more practical scenarios. We consider four new datasets: CNN/DailyMail (See et al., 2017), DialogSum (Chen et al., 2021), PubMedQA (Jin et al., 2019) and IMDb (Maas et al., 2011) to test the detection of deepfake news, deepfake dialogues, deepfake scientific answers and deepfake movie reviews. We sample 200 instances from each dataset and use a newly developed LLM, i.e., GPT-4 (OpenAI, 2023b), with specially designed prompts (Appendix A) to create deepfake texts.

Paraphrasing Attack Sadasivan et al. (2023) show that detection methods are vulnerable to being deceived by paraphrased target texts. Based on the Unseen Domains & Unseen Model test set, we paraphrase each sentence individually for both human-written and machine-generated texts, forming a more challenging test set. We adopt `gpt-3.5-turbo` as the paraphraser and consider all paraphrased texts as machine-generated.

5.2 Evaluation Metrics

We report **AUROC** (the area under the receiver operating characteristic curve), which quantifies the classifier's potential of distinguishing between the positive and negative classes. An AUROC of 1.0 corresponds to a perfect classifier, whereas 0.5 represents random guessing. Following Rosenthal et al. (2019), we also consider **AvgRec** (average recall), which is calculated by averaging the recall scores on human-written texts (HumanRec) and machine-generated texts (MachineRec) [2]. These recall scores help us assess the realistic detection performance. For instance, black-box detection methods like human detection and ask ChatGPT cannot be evaluated using AUROC. Furthermore, determining a decision boundary based on a reliable validation set is challenging in an open-domain detection setting.

6 Results

6.1 Naive Baselines

Table 1 shows that both ChatGPT and human annotators fail to distinguish machine-generated texts

[2] Since our test sets are balanced, the precision score heavily relies on and can be reflected by the recall score. Therefore, we choose to report only the recall scores for a more intuitive evaluation.

Detector	HumanRec	MachineRec	AvgRec
ChatGPT	96.98%	12.03%	54.51%
Human	61.02%	47.98%	54.50%

Table 1: Detection performance of ChatGPT and humans.

Methods	Human/Machine	AvgRec	AUROC
FastText	94.72%/94.36%	94.54%	0.98
GLTR	90.96%/83.94%	87.45%	0.94
Longformer	97.30%/95.91%	**96.60%**	**0.99**
DetectGPT	91.68%/81.06%	86.37%	0.92

Table 2: Tested 1: White-box detection performance. "Human/Machine" denotes HumanRec and MachineRec, respectively.

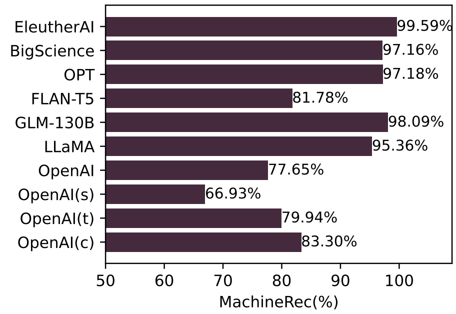


Figure 2: Out-of-distribution detection performance on machine-generated texts generated by *unseen models*. OpenAI(c), OpenAI(t) and OpenAI(s) corresponds to texts generated by OpenAI models using continuation, topical and specified prompts, respectively.

from human-written ones. The AvgRec is only slightly better than random guessing, suggesting that machine-generated texts have achieved a level (e.g., fluency and coherence) comparable to human. We then explore whether there exist underlying differences that can be captured by automatic detection methods.

6.2 In-domain Detection

The results of in-domain detection are shown in Table 2 and the upper part of Table 3.

White-box Detection. From Table 2, we can observe that all detection methods obtain solid performance when the texts are from a specific domain and a specific LLM (GPT-J-6B) (i.e., *Fixed-domain & Model-specific*). Typically, DetectGPT performs well in identifying machine-generated texts when the scoring model matches the one used to generate the fake texts, i.e., accessibility to the generation LLM in the white-box setting.

PLM-based Detectors demonstrate robustness to texts from various sources. As shown in Table 3, the detection performance (AvgRec and AUROC) decreases as the detector encounters broader data sources, i.e., texts from various domains or various LLMs. For example, GLTR's AUROC drops from 0.94 to 0.80 and DetectGPT's drops from 0.92 to 0.57 when encountering texts from multiple models (*Arbitrary-models*). The severe performance drop of DetectGPT is attributed to its reliance on accessibility to the generation LLMs (Mitchell et al., 2023). On the other hand, FastText faces significant challenges in detecting texts from various domains (*Arbitrary-domains*), despite its robustness on texts sourced by different language models. Among all detection methods, the Longformer detector consistently outperforms others in terms of AUROC and AvgRec. Despite the minor performance degradation, Longformer surpasses other detectors by a considerable margin in the *Arbitrary-domains & Arbitrary-models* setting, where the detector encounters diverse texts from various domains and language models.

6.3 Out-of-domain Detection

We further investigate whether the detection model can identify machine-generated texts in out-of-distribution settings, i.e., detect texts from unseen domains or generated by new LLMs. The results are presented in the lower part of Table 3. Empirical results indicate that, except for the Longformer detector, all other detectors perform poorly in identifying texts generated by unseen models. Furthermore, none of the detectors effectively classify texts from novel domains.

Unseen Models. Among all methods, the Longformer detector is the only one that performs well (with an AUROC of 0.95 and AvgRec of 86.61%) when detecting texts from unseen LLMs. The performance of FastText further degrades, with AUROC dropping from 0.83 to 0.74. GLTR faces a significant challenge when it comes to unseen models. Its AUROC of 0.65 suggests that it struggles to differentiate between different text sources. The detection performance (Longformer) on each unseen model set is shown in Figure 2. The Lonformer classifier has the most difficulty distinguishing texts generated by the OpenAI and FLAN-T5 models from human-written ones. By comparison, the detector can identify most of the deepfake texts from other models, even if it has not encountered any of them during training. On the other hand, the difficulty of detection is influenced by the prompt

Settings	Methods	Metrics HumanRec	MachineRec	AvgRec	AUROC
Testbed 2,3,4: In-distribution Detection					
Arbitrary-domains & Model–specific	FastText (Joulin et al., 2017)	88.96%	77.08%	83.02%	0.89
	GLTR (Gehrmann et al., 2019)	75.61%	79.56%	77.58%	0.84
	Longformer (Beltagy et al., 2020)	95.25%	96.94%	**96.10%**	**0.99**
	DetectGPT⋆ (Mitchell et al., 2023)	48.67%	75.95%	62.31%	0.60
Fixed-domain & Arbitrary-models	FastText (Joulin et al., 2017)	89.43%	73.91%	81.67%	0.89
	GLTR (Gehrmann et al., 2019)	37.25%	88.90%	63.08%	0.80
	Longformer (Beltagy et al., 2020)	89.78%	97.24%	**93.51%**	**0.99**
	DetectGPT⋆ (Mitchell et al., 2023)	86.92%	34.05%	60.48%	0.57
Arbitrary-domains & Arbitrary-models	FastText (Joulin et al., 2017)	86.34%	71.26%	78.80%	0.83
	GLTR (Gehrmann et al., 2019)	12.42%	98.42%	55.42%	0.74
	Longformer (Beltagy et al., 2020)	82.80%	98.27%	**90.53%**	**0.99**
	DetectGPT⋆ (Mitchell et al., 2023)	86.92%	34.05%	60.48%	0.57
Testbed 5,6: Out-of-distribution Detection					
Unseen Models	FastText (Joulin et al., 2017)	83.12%	54.09%	68.61%	0.74
	GLTR (Gehrmann et al., 2019)	25.77%	89.21%	57.49%	0.65
	Longformer (Beltagy et al., 2020)	83.31%	89.90%	**86.61%**	**0.95**
	DetectGPT⋆ (Mitchell et al., 2023)	48.67%	75.95%	62.31%	0.60
Unseen Domains	FastText (Joulin et al., 2017)	54.29%	72.79%	63.54%	0.72
	GLTR (Gehrmann et al., 2019)	15.84%	97.12%	56.48%	0.72
	Longformer (Beltagy et al., 2020)	38.05%	98.75%	**68.40%**	**0.93**
	DetectGPT⋆ (Mitchell et al., 2023)	86.92%	34.05%	60.48%	0.57

Table 3: Testbed 2-6: Detection performance of different detection methods. The out-of-distribution settings examine the detection capability on texts from unseen domains or deepfake texts generated by new LLMs. ⋆ denotes the unsupervised detection method.

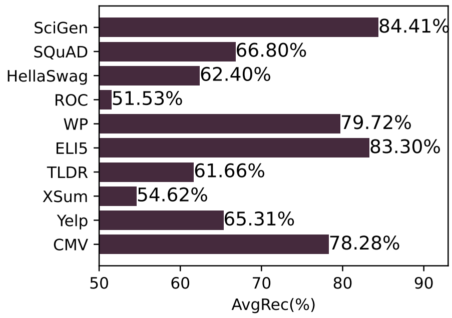
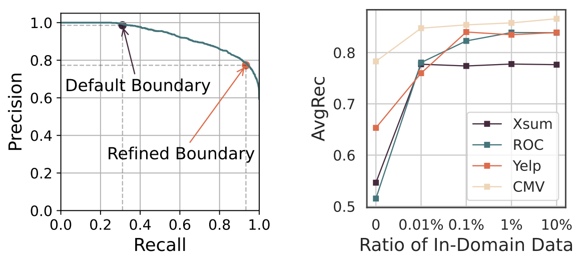


Figure 3: Out-of-distribution detection performance (AvgRec) on texts from *unseen domains*.



(a) Precision-Recall curve of the Longformer detector on the unseen domain (Yelp). A refined decision boundary obtains a better trade-off between precision and recall.

(b) Detection performance in the "Unseen Domains" setting (Xsum, ROC, Yelp and CMV) with decision boundary adjusted based on different ratios of in-domain data.

Figure 4: Decision boundary adjustment.

types used for model generation. Texts generated from specific prompts (OpenAI(s)) are harder to distinguish than the other two types (OpenAI(c) and OpenAI(t)). This can be because they follow a detailed prompt condition, making them more similar to human-written texts.

Unseen Domains. Detecting texts from *unseen domains* presents a heightened challenge for classifiers. Notably, even the top-performing model, Longformer, experiences a substantial decline in AvgRec, dropping from 90.53% to 68.40%. Typically, Longformer tends to classify human-written texts from unfamiliar domains as machine-generated, which results in a low HumanRec score but an almost perfect MachineRec. We present detection performance (Longformer) on each unseen domain in Figure 3. The top three text domains most likely to be misclassified as machine-generated are ROC, XSum, and TLDR datasets. This could be attributed to their low average perplexity scores which confuse PLM-based detectors (discussed in Section 7.2).

Boundary adjustment. Despite the low AvgRec in the *Unseen Domains* setting, Longformer achieves a high AUROC score (0.93). This suggests that the model can distinguish between the

Metrics	Unseen Models	Unseen Domains
HumanRec	86.09%	82.88%
MachineRec	89.15%	80.50%
AvgRec	**87.62%(+1.01%)**	**81.78%(+13.38%)**

Table 4: Detection performance (Longformer) on out-of-distribution testbeds with decision threshold adjusted based on 0.1% of the in-distribution data.

two classes but struggles with selecting an appropriate decision boundary, as shown in Figure 4a. To address this issue, we utilize a portion of the in-domain data from the training set to adjust the decision boundary. We compute an average decision boundary across 10 classifiers (in the *Unseen Domains* setting) and apply it universally across all domains. As depicted in Figure 4b, refining the decision boundary with only 0.1% of in-domain data significantly enhances detection performance. Table 4 demonstrates that adjusting the decision boundary (using 0.1% of in-domain data) notably improves detection accuracy for both out-of-distribution settings.

Unseen Domains & Unseen Model We validate the detection ability of Longformer, the best-performing detector, on the *Unseen Domains & Unseen Model* testbed. The results are presented in Table 5. The Longformer detector trained using our dataset achieves a high performance (0.94 AUROC) in detecting texts generated by GPT-4, even when sourced from newly added datasets and generated by a new LLM. After refining the boundary, the detector demonstrates balanced accuracy in detecting both text sources, resulting in an AvgRec of 86.54%. This showcases its feasibility for deployment in real-world scenarios.

Paraphrasing Attack However, similar to other methods (Krishna et al., 2023), the Longformer detector also shows vulnerability to paraphrasing attacks, as shown in Table 5. The AUROC drops from 0.94 to 0.75 when the detector encounters additional paraphrased texts, which can be attributed to the shifted perplexity distribution of paraphrased texts (Section 7.2).

7 Analysis

7.1 Convergence of Human and Machine Compositions

We explore to find potential differentiability through a comparison of linguistic patterns in human-written and machine-generated compositions. To accomplish this, we employ Stanza (Qi et al., 2020) to extract the distribution of various linguistic patterns such as named entities, part-of-speech tags, and constituents. Next, we calculate the Jensen-Shannon distance to quantify the disparity between the probability distributions obtained from both text sources (human-written and machine-generated).

Figure 5 demonstrates that including texts from diverse domains and LLMs reduces the linguistic dissimilarity between the two text sources. This makes it more challenging for a detector to distinguish them, which aligns with the increasing difficulty of detection in the four in-distribution settings. Once an adequate amount of texts from various domains and LLMs are collected, there is

HumanRec	MachineRec	AvgRec	AUROC
Testbed 7: Unseen Domains & Unseen Model			
52.50%	99.14%	75.82%	0.94
88.78%†	84.12%†	86.54%†	0.94
Testbed 8: Paraphrasing Attack			
52.16%	81.73%	66.94%	0.75
88.78%†	37.05%†	62.92%†	0.75

Table 5: Testbed 7-8: Detection performance of Longformer detector on the two challenging test sets. †denotes the refined decision boundary. Appendix G includes the performance of other detection methods.

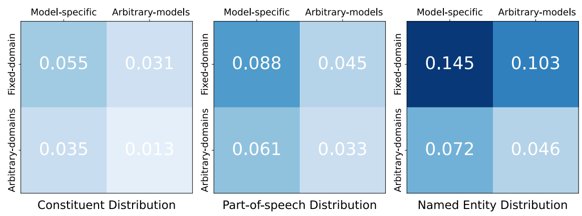


Figure 5: Linguistic difference (Jensen-Shannon distance) between human-written texts and machine-generated texts in 4 in-distribution settings (darker colors indicate larger differences).

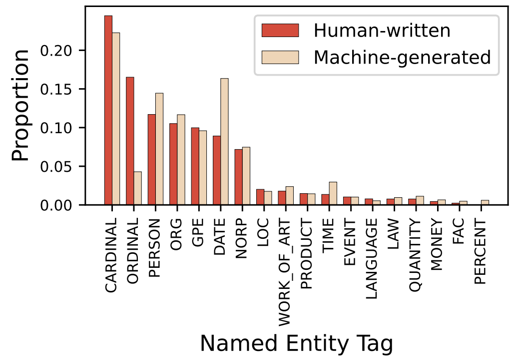


Figure 6: Linguistic difference (Named Entity Distributions) of the *Fixed-domain & Model-specific* setting.

no significant statistical distinction between the two text sources (see Figure 13 in Appendix H). In contrast, when dealing with texts from a specific domain or an LLM (Fixed-domain & Model-Specific), noticeable differences exist. For example, entity tags like "ORDINAL" and "DATE" can serve as detection shortcuts, as shown Figure 6. Comparing the sentiment polarity and grammatical formality of the two text sources (Appendix H) also demonstrates convergence between human-written and machine-generated texts.

7.2 Double-edged Sword of Perplexity Bias

In this section, we explore to find the general distinction which is not influenced by text domain or generation LLMs. Prior work on unsupervised detection (Mitchell et al., 2023; Bao et al., 2023) leverages the property that model generations reside in local minima of perplexity. We discover that such property also acts as a fundamental feature for PLM-based methods to effectively differentiate machine generations.

Specifically, we use an **untuned** Longformer to obtain perplexity score (Salazar et al., 2020) for test set texts in the *Unseen Domains* setting. Figure 7 illustrates how prior knowledge in PLMs, as measured by perplexity, aids in clustering two text sources into distinct peaks. The average perplexity score of machine-generated texts is notably lower than that of human writings, establishing an implicit pattern to distinguish them.

However, perplexity bias can hinder robust detection. PLM-based detectors also exhibit overconfidence in text perplexity, classifying low-perplexity texts as machine-generated and high-perplexity texts as human-generated. We categorize the texts based on prediction correctness. As shown in Figure 7, misidentified human-written texts by the Longformer detector have significantly lower average perplexity compared to correctly predicted ones, but are similar to correctly predicted machine-generated texts. In contrast, the average perplexity of incorrectly predicted machine-generated texts is higher than that of correctly predicted ones. Figure 8 presents a more intuitive visualization: false predictions of human-written texts (darker green bars) are concentrated in the lower perplexity region, while false predictions of machine-generated texts (darker khaki bars) are spread across the higher perplexity region. Paraphrasing attacks, illustrated in Figure 9, cause the peak of human-written texts to be positioned between that of machine-generated

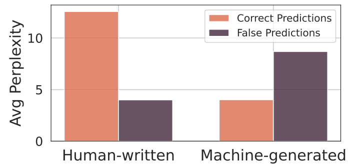


Figure 7: Comparison of the average perplexity of texts which the Longformer detector predicts correctly and incorrectly.

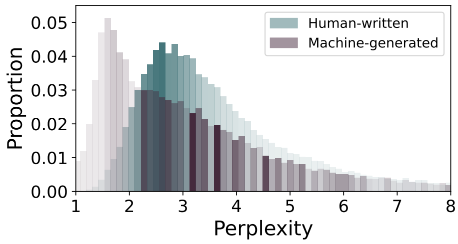


Figure 8: Perplexity distribution: A darker colour indicates a larger proportion of incorrect predictions in the perplexity bucket.

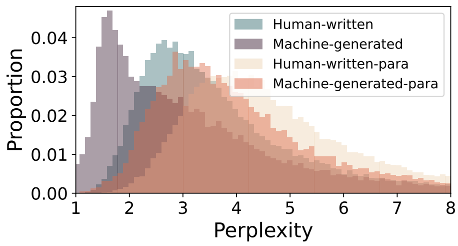


Figure 9: Perplexity distribution of human-written texts, machine-generated texts and their corresponding paraphrased texts.

texts (machine-generated, machine-generated-para, and human-written-para), leading to significant confusion for the Longformer detector.

8 Conclusion

We proposed a comprehensive testbed for deepfake text detection, by gathering texts from various writing tasks and deepfake texts generated by different LLMs. Empirical results on commonly used detection methods demonstrated the challenge of deepfake text detection. Out-of-distribution posed a greater challenge for detectors to be employed in application scenarios. With the boundary refined, the best-performing detector on our testbeds (i.e., Longformer detector) achieved 86.54% AvgRec on out-of-domain texts generated by a new LLM, i.e., GPT4. By studying differences between human and machine compositions, we find that perplexity can serve as a fundamental feature for classification regardless of text domain or generation LLM. To the best of our knowledge, this is the first study to investigate the challenges and feasibility of deepfake text detection in a "wild" testbed.

Limitations

Although we are the first to propose a comprehensive testbed for deepfake text detection and validate the detection effectiveness on frontier test sets, there are two major limitations: (1) We strive to include a wide variety of LLMs in our dataset. However, new LLMs such as Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023) continue to emerge and may not be currently included. Nevertheless, our dataset aims to serve as a testbed to select the best-performing detectors, which encounter sufficiently diverse machine-generated texts and can deal with texts from newly-developed LLMs in future. (2) We adopt benchmark datasets as text sources, which can be used as the training data for LLM pretraining. The detection capability may vary on new online texts that were not included in the LLMs' pretraining data. In the future, we plan to gather new online texts that have not been previously seen by LLMs to study such variation.

Ethics Statement

We honor the Code of Ethics. No private data or non-public information is used in this work. For human annotation (Section 6.1), we recruited our annotators from the linguistics departments of local universities through public advertisement with a specified pay rate. All of our annotators are senior undergraduate students or graduate students in linguistic majors who took this annotation as a part-time job. We pay them 60 CNY an hour. The local minimum salary in the year 2023 is 25.3 CNY per hour for part-time jobs. The annotation does not involve any personally sensitive information.

References

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

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.

Francesco Barbieri, Luis Espinosa Anke, and José Camacho-Collados. 2022. XLM-T: multilingual language models in twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 258–266. European Language Resources Association.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings 21*, pages 421–426. Springer.

BigScience. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection. *arXiv preprint arXiv:2304.04736*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

T Fagni, F Falchi, M Gambini, A Martella, M Tesconi, et al. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):1–16.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: long form question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

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. *CoRR*, abs/2301.07597.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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*.

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

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. *PAN*, 8:27–31.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language*, 23(1):107–125.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *CoRR*, abs/2301.11305.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Learning to reason for text generation from scientific tables. *CoRR*, abs/2104.08296.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 839–849. The Association for Computational Linguistics.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

OpenAI. 2023a. Ai text classifier.

OpenAI. 2023b. GPT-4 technical report. *CoRR*, abs/2303.08774.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2022. Deepfake text detection: Limitations and opportunities. *CoRR*, abs/2210.09421.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 101–108. Association for Computational Linguistics.

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.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of gpt-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2019. Semeval-2017 task 4: Sentiment analysis in twitter. *CoRR*, abs/1912.00741.

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*.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2699–2712. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Shuming Shi, Enbo Zhao, Bi Wei, Cai Deng, Leyang Cui, Xinting Huang, Haiyun Jiang, Duyu Tang, Kaiqiang Song, Wang Longyue, Chengyan Huang, Guoping Huang, Yan Wang, and Li Piji. 2023. Effidit: An assistant for improving writing efficiency.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings* of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 613–624. ACM.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

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

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.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. Glm-130b: An open bilingual pre-trained model.

Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *CoRR*, abs/2302.05543.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open pre-trained transformer language models.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. Syngec: Syntax-enhanced grammatical error correction with a tailored gec-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2518–2531. Association for Computational Linguistics.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. *CoRR*, abs/2302.03162.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2461–2470.

A Prompt Design

Figure 10 present prompt cases in three domains (CMV, XSum and ELI5) to showcase different prompt types (i.e., continuation prompts, topical prompts and specified prompts). The prompts used for building GPT-4 test sets are presented in Figure 11.

B Dataset Construction

We show an example of Yelp dataset to give an intuitive illustration of dataset construction: We randomly sample 1,000 human-written texts from the Yelp dataset and use 27 LLMs to generate corresponding machine-generated texts. After data preprocessing and filtering, we obtained a total of 26,235 machine-generated texts and 1,000 human-written texts. To mitigate data imbalance between the text sources (human-written v.s. machine-generated), we additionally collect data from the Yelp dataset and obtain a total of 37,706 human-written texts after filtering. The additional data is used to compensate validation and test sets first for more accurate evaluation. We discuss the effects of data balance for training in Appendx F.

By default, machine-generated texts are generated using continuation prompts. For datasets which provide topics or titles, we also consider topical and specified prompts. The latter two prompt types are only used for the OpenAI GPT model set, since we empirically find they perform robust generation to various prompts. For example, for the 1,000 human-written texts in the Xsum dataset, we have 33,000 (27,000+3*2*1000) machine-generated texts and finally obtain 32,930 texts after filtering.

We conduct preprocessing to reduce the effects beyond text contents, such as punctuation normalization and line-break removal, etc. We also filter out texts that are too long or too short. We divide the texts into three splits, i.e., train/validation/test, with an 80%/10%/10% partition. The data statistics are shown in Table 6. The distribution of machine-generated texts by model is presented in Figure 12.

C Method Implementation

Human annotation & Ask-ChatGPT. We create a test subset from the whole testset, by pairing one machine-generated text with each human-generated one through random sampling. To create the test set for the naive baselines, we randomly select 10% of the human-written texts from the test set used in the "Arbitrary-domains & Arbitrary-models" setting. Data statistics of the test set is shown in Table 7. We also randomly sample an equal number of machine-generated texts. We hire 3 expert annotators to conduct independent annotation and average their performance.

Longformer. Across all datasets, we used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.005 and set the dropout rate at 0.1. All models are finetuned for 5 epochs on 8 V100 GPUs. We select the best-performing model based on validation classification accuracy.

FastText. We experiment with different combinations of word n-gram features and character n-gram features. Based on validation results, we choose only word bi-grams as text features. We train all models for 100 epochs and leave other settings as default.

GLTR. GLTR uses a language model to gather features, i.e., the number of tokens in the Top-10, Top-100, and Top-1000 ranks, which are fed into a logistic regression model to classify texts. Following Pu et al. (2022), we use GPT-2-XL (Radford et al., 2019) as the language model and use scikit-learn (Pedregosa et al., 2011) to train regression models. We conduct a grid search on optimization algorithm ('lbfgs', "liblinear", "newton-cg", "newton-cholesky", "sag", and "saga"), the norm of the penalty ("l1", "l2" and "elasticnet") and regularization strength (0.001, 0.01, 0.1, 1, 10, and 100) and choose the best-performing model under cross-validation.

DetectGPT. We follow the best-performing setting (Mitchell et al., 2023), using T5-3B (Raffel et al., 2020) as the mask infilling model, with the mask rate set as 15%, the masked span length as 2, and the number of perturbations as 100. We use GPT-J-6B (Wang and Komatsuzaki, 2021) as the scoring model. We manually set the decision boundary based on the validation set.

D Randomness

We conduct experiments to testify the stability of our testbeds. Specifically, we investigate the effects of randomness under the *Arbitrary-domains and Arbitrary-models* setting by (1) splitting the testbeds (train, validation and test) with 5 different seeds and training 5 Longformer detectors on each split; and (2) training 5 Longformer detectors with

Domain	Continuation Prompt	Topical Prompt	Specified Prompt
CMV	I spend my summer as a representative of the college I attend and interact regularly with kids between the ages of 10 and 18. In these interactions, I have noticed	Generate a counter-argument to refute the following opinion: HandwritingCursive is an important skill that should be taught throughout a minor's schooling.	Generate a counter-argument to refute the following Reddit post: HandwritingCursive is an important skill that should be taught throughout a minor's schooling.
XSum	Apple Music performed a U-turn over payment policy a day after the pop star threatened to prevent the US firm from streaming her album 1989. Swift had argued that Apple	Write a news article with the following headline: A photographer has accused Taylor Swift of "double standards" in her row with Apple over music streaming.	Write an article for BBC News with the following headline: A photographer has accused Taylor Swift of "double standards" in her row with Apple over music streaming.
ELI5	When you're watching a scene and the camera moves, say left to right for example; The stuff that's closer to the camera will move faster than the stuff that's further	How they turn 2D movies into 3D	Explain like I am 5 years old: How they turn 2D movies into 3D

Figure 10: Examples of three prompt types.

Domain	Prompt for GPT-4
CNN/DailyMail	Write a news article given the following highlights: Powers appeared in the final season of the long-running sitcom . He played the husband of main character Thelma . Powers died April 6 at his home in New Bedford, Massachusetts at the age of 64. His family have not revealed the cause of death .
DialogSum	Continue the following daily dialogue: #Person1#: School has added several new courses to our grade this semester. I have more homework to do now. #Person2#: What's your favorite course, Daniel?
PubMedQA	Does prenatal ethanol exposure reduce mGluR5 receptor number and function in the dentate gyrus of adult offspring?
IMDb	Write a short movie review with the following beginning: I am not a big fan of the Spielberg/Cruise version of this film.

Figure 11: Examples of prompts for building the frontier test sets.

Dataset	**CMV**	**Yelp**	**XSum**	**TLDR**	**ELI5**
Train	4,461/21,130	32,321/21,048	4,729/26,372	2,832/20,490	17,529/26,272
Valid	2,549/2,616	2,700/2,630	3,298/3,297	2,540/2,520	3,300/3,283
Test	2,431/2,531	2,685/2,557	3,288/3,261	2,536/2,451	3,193/3,215
WP	**ROC**	**HellaSwag**	**SQuAD**	**SciGen**	**all**
6,768/26,339	3,287/26,289	3,129/25,584	15,905/21,489	4,644/21,541	95,596/236,554
3,296/3,288	3,286/3,288	3,291/3,190	2,536/2,690	2,671/2,670	29,467/29,462
3,243/3,192	3,275/3,207	3,292/3,078	2,509/2,535	2,563/2,338	29,015/28,365

Table 6: Number of instances for each dataset. The number of human-written texts and that of machine-generated texts are separated by "/".

different running seeds on one of the splits. The results in Table 8 show that our testbeds are robust to randomness, with a small standard deviation.

E PLM Backbone Comparison

In addition to Longformer, we also experiment with other PLM backbones such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and GPT2 (Radford et al., 2019). The results of these experiments are shown in Table 9. Firstly, the Longformer detector achieves the best performance in terms of both AvgRec and AUROC due to its ability to handle longer texts, while maintaining a small model size for efficient detection. Secondly, increasing the model size improves detection performance for each backbone PLM. Thirdly, masked language models (BERT, RoBERTa, and Longformer) outperform causal language models (GPT2).

F Data Balance

Since the number of machine-generated texts is larger than that of human-written ones in the train set. We investigate whether such an imbalance has an impact on the model performance. Specifically, we randomly sample machine-generated texts to be the same quantity as human-written ones. We experiment on the Longformer detector and present the results in Table 10. Despite the narrowed gap between HumanRec and MachineRec, we can observe that data balance has little influence on model

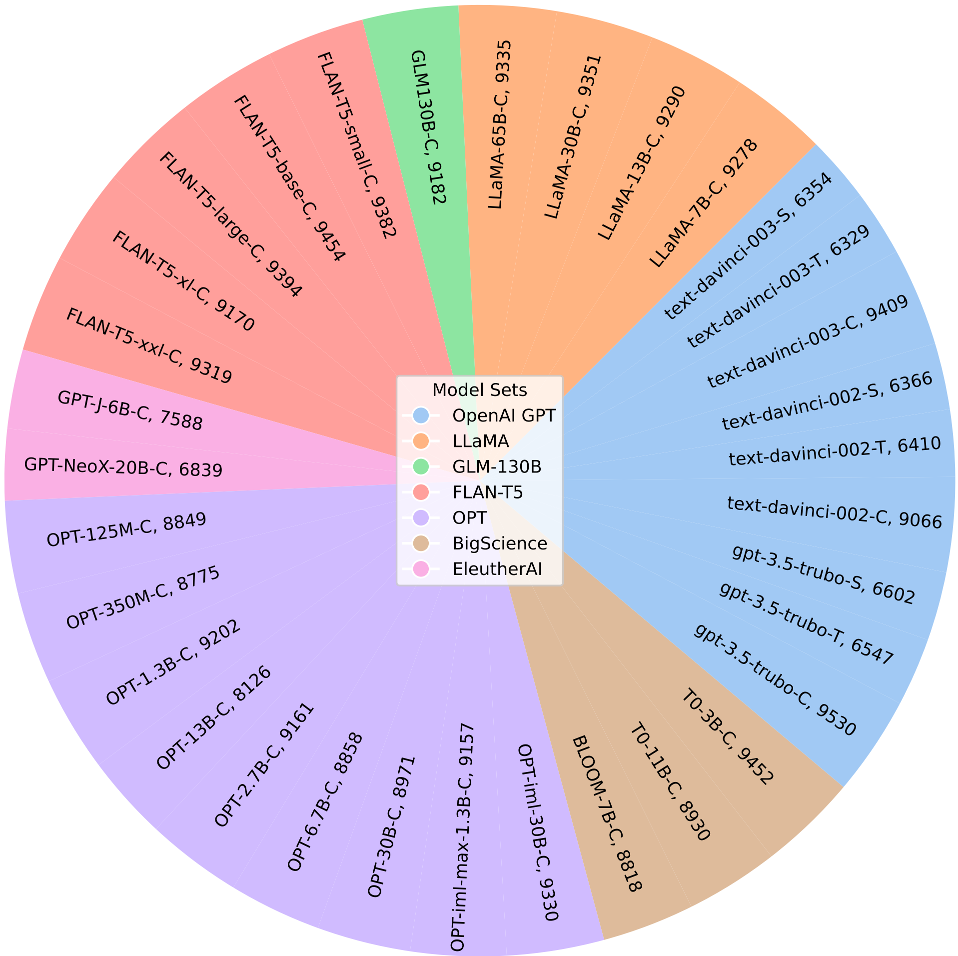


Figure 12: Distribution of machine-generated instances by model: For example, "FLAN-T5-small-C, 9382" indicates that the model "FLAN-T5-small" generated 9382 texts using **c**ontinuation prompts. The letters **C**, **T** and **S** represent the types of prompts used: "**c**ontinuation" "**t**opical" and "**s**pecified", respectively.

	CMV	Yelp	XSum	TLDR	ELI5	WP	ROC	HellaSwag	SQuAD	SciGen	all
# human	80	100	100	77	100	100	100	100	100	99	1912
# machine	80	100	100	77	100	100	100	100	100	99	1912

Table 7: Number of human-written and machine-generated texts of the sampled testset for naive baselines.

Randomness	**HumanRec**	**MachineRec**	**AvgRec**	**AUROC**
Data Split	83.00%±2.82%	97.74%±0.34%	90.37%±1.29%	0.99±0.0010
Training (Longformer)	82.81%±2.38%	97.90%±0.25%	90.36%±1.12%	0.99±0.0021

Table 8: Stability of the empirical results considering both data split randomness and training randomness.

performance in terms of AvgRec and AUROC. In addition, the tendency of the Longformer detector to classify human-written texts as machine-generated ones still exists with a perfectly balanced training set.

G Detection Performance on the Two Challenging Test Sets

The detection performance of all methods on the two challenging test sets, i.e., Unseen Domains & Unseen Model and Paraphrase Attack, is shown in Table 11. Detect-GPT is not included due to its reliance on the white-box detection setting. We can observe that all methods suffer severe perfor-

PLM	# Parameters	HumanRec	MachineRec	AvgRec	AUROC
BERT-base	110M	67.11%	98.34%	82.72%	0.97
BERT-large	336M	80.96%	93.27%	87.12%	0.96
RoBERTa-base	125M	72.29%	95.28%	83.78%	0.96
RoBERTa-large	355M	70.81%	98.38%	84.59%	0.98
GPT2	117M	57.42%	97.84%	77.63%	0.96
GPT2-medium	345M	69.94%	96.82%	83.39%	0.96
GPT2-large	774M	84.27%	96.67%	90.47%	0.98
Longformer	149M	82.80%	98.27%	**90.53%**	**0.99**

Table 9: Performance comparison of different PLM-based classifiers.

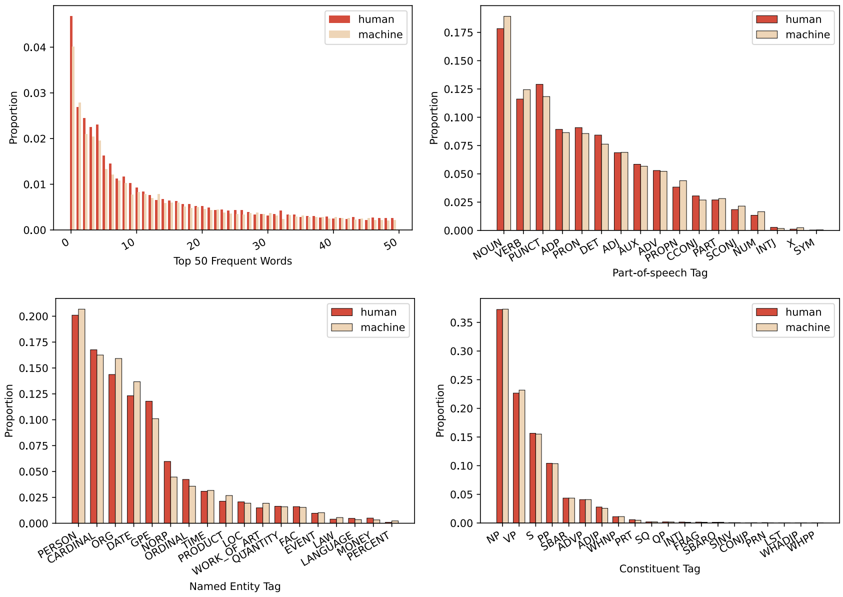


Figure 13: Linguistic statistics (word frequency distribution, part-of-speech distribution, named entity distribution and constituency distribution) for human-written and machine-generated samples.

HumanRec	MachineRec	AvgRec	AUROC
85.38%	92.95%	89.16%	0.99

Table 10: Effects of data balance on detection performance (Longformer) under the *Arbitrary-domains & Arbitrary-models* setting.

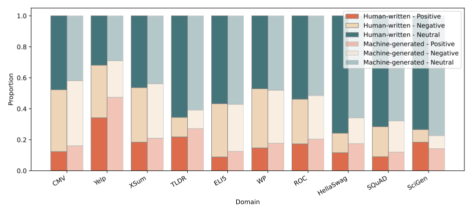


Figure 14: Sentiment polarity.

mance degradation in terms of AUROC, indicating weakness in detecting machine-paraphrased texts.

H Text Characteristics

We first explore to find potential surface patterns that can help discriminate between human-written

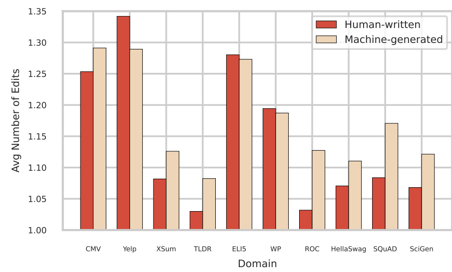


Figure 15: Grammar formality. A lower number of edits indicates better grammar formality.

texts and machine-generated ones. The length statistics are shown in Table 12. As can be seen from the table, although we do not exert explicit length control over the model generation, the average length of machine-generated texts is marginally longer than that of human-written.

Linguistic Pattern. We further use Stanza, a linguistics analysis tool (Qi et al., 2020), to gain a more systematic understanding of the linguistic

Methods	HumanRec	MachineRec	AvgRec	AUROC
Unseen Domains & Unseen Model				
FastText	71.78%	68.88%	70.33%	0.74
GLTR	16.79%	98.63%	57.71%	0.73
Longformer	52.50%	99.14%	75.82%	0.94
Longformer†	88.78%†	84.12%†	86.54%†	0.94
Paraphrasing Attack				
FastText	71.78%	50.00%	60.89%	0.66
GLTR	16.79%	82.44%	49.61%	0.47
Longformer	52.16%	81.73%	66.94%	0.75
Longformer†	88.78%†	37.05%†	62.92%†	0.75

Table 11: Detection performance on the two challenging test sets. '†' denotes the boundary is adjusted.

Data Source	Human-written	Machine-generated	All
Average Document Length	232.02	279.99	263.87
Average Sentence Length	18.90	18.80	18.83
Average # Sentences per Document	13.48	15.33	14.71

Table 12: Length statistics for human-written and machine-generated samples.

components in both sources, with results shown in Figure 13. We can observe that texts from both sources share similar distributions under various linguistic scales, such as word frequency, part-of-speech frequency, named-entity frequency, and constituent frequency. In other words, there is no significant linguistic difference between the text sources (human-written versus machine-generated) that can assist the classifier in differentiating them in a wild setting.

In addition, we explore whether there are differences between human-written and machine-generated texts in other characteristics (such as sentiment polarity and grammar formality) when considering diverse writing tasks and various text-generating LLMs.

Sentiment Polarity. We use an off-the-shelf sentiment classifier (Barbieri et al., 2022) trained on 198M tweets for sentiment analysis to analyze the sentiment polarity of both texts, with results shown in Figure 14. As suggested by Guo et al. (2023), ChatGPT expresses more neutral sentiments than humans. In a large-scale setting that considers various domains and LLMs, however, there is no clear distinction between human-written and machine-generated texts in terms of sentiment polarity. Notably, LLMs generally generate more positive texts, especially when creating reviews or comments (Yelp).

Grammatical Formality. We use an off-the-shelf grammar error correction model (Zhang et al., 2022b) to evaluate the grammar formality of human-written and machine-generated texts. We adopt the average number of edits to quantify grammar formality. As shown in Figure 15, machine-generated texts are equally or even more grammatical in domains (CMV, Yelp, ELI5, and WP) where texts are less formal (reviews or posts on forums). In formal domains such as XSum (news articles), SQuAD (Wikipedia documents), and SciGen (scientific writings), human-written texts exhibit better grammatical formality.