A Practical Examination of AI-Generated Text Detectors for Large Language Models

Brian Tufts
Carnegie Mellon University
`btufts@cs.cmu.edu`

Xuandong Zhao
UC Berkeley
`xuandongzhao@berkeley.edu`

Lei Li
Carnegie Mellon University
`leili@cs.cmu.edu`

Abstract

The proliferation of large language models has raised growing concerns about their misuse, particularly in cases where AI-generated text is falsely attributed to human authors. Machine-generated content detectors claim to effectively identify such text under various conditions and from any language model. This paper critically evaluates these claims by assessing several popular detectors (RADAR, Wild, T5Sentinel, Fast-DetectGPT, PHD, LogRank, Binoculars) on a range of domains, datasets, and models that these detectors have not previously encountered. We employ various prompting strategies to simulate practical adversarial attacks, demonstrating that even moderate efforts can significantly evade detection. We emphasize the importance of the true positive rate at a specific false positive rate (TPR@FPR) metric and demonstrate that these detectors perform poorly in certain settings, with TPR@.01 as low as 0%. Our findings suggest that both trained and zero-shot detectors struggle to maintain high sensitivity while achieving a reasonable true positive rate. All code and data necessary to reproduce our experiments are available at `https://github.com/LeiLiLab/llm-detector-eval`.

1 Introduction

Large language models (LLMs) are becoming increasingly accessible and powerful, leading to numerous beneficial applications (Touvron et al., 2023; Achiam et al., 2023). However, they also pose risks if used maliciously, such as generating fake news articles, facilitating academic plagiarism or spam content (Feng et al., 2024; Zellers et al., 2019b; Perkins, 2023; Fraser et al., 2024). The potential for misuse of LLMs has become a significant concern for major tech corporations, particularly in light of the 2024 elections in the united states. At the Munich Security Conference on February 16th, 2024, these companies pledged to combat misleading machine-generated content, acknowledging the potential of AI to deceptively influence electoral outcomes (Accord, 2024). As a result, there is a growing need to develop reliable methods for differentiating between LLM-generated and human-written content. To ensure the effectiveness and accountability of LLM detection methods, continuous evaluation of popular techniques is crucial.

Many methods have been released recently that claim to have a strong ability to detect the difference between AI-generated and human-generated texts. These detectors primarily fall into three categories: trained detectors, zero-shot detectors, and watermarking techniques (Yang et al., 2023b; Ghosal et al., 2023; Tang et al., 2023). *Trained detectors* utilize datasets of human and AI-generated texts and train a binary classification model to detect the source of a text (Zellers et al., 2019b; Hovy, 2016; Hu et al., 2023; Tian and Cui, 2023; Verma et al., 2024). *Zero-shot detection* utilizes a language model's inherent traits to identify text it generates, without explicit training for detection tasks other than calibrating a threshold for detection in some cases (Gehrmann et al., 2019; Mitchell et al., 2023; Bao et al., 2024; Yang et al., 2023a; Venkatraman et al., 2024). *Watermarking* is another technique in which the model owner embeds a specific probabilistic pattern into the text to make it detectable Kirchenbauer et al. (2023). However, watermarking requires the model owner to add the signal, and its design has theoretical guarantees; we do not evaluate watermarking models in this study.

In this paper, we test the robustness of these detection methods to unseen models, data sources, and adversarial prompting. To do this, we treat all model-generated text as a black box generation. That is, none of the detectors know the source of the text or have access to the model generating the text. This presents the most realistic scenario where the user is presented with text and wants to

Findings of the Association for Computational Linguistics: NAACL 2025, pages 4839–4856
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

know if it is AI-generated or not. Our contributions can be summarized as follows:

- We conduct a thorough evaluation of AI-generated text detectors on unseen models and tasks, providing insights into their effectiveness in real-world settings.
- We analyze the performance of various detectors under practical adversarial prompting, exploring the extent to which prompting can be used to evade detection.
- We demonstrate that high AUROC scores, which are often used as a measure of performance in classification tasks, do not necessarily translate to practical usage for machine-generated text detection. Instead, we motivate using the metric of true positive rate (TPR) at a 1% false positive rate (FPR) threshold as a more reliable indicator of a detector's effectiveness in practice.

2 Related Work and Background

There is a variety of related work that discusses text detectors. These works cover different aspects, such as the text detectors themselves, their types, evaluation, and red-teaming of detectors.

Text Detectors. Machine-generated text detectors can be divided into trained classifiers, zero-shot classifiers, and watermark methods (Yang et al., 2023b; Hans et al., 2024; Ghosal et al., 2023; Jawahar et al., 2020). (1) Trained detectors use classification models to determine if the text is machine-generated or human-written (Zellers et al., 2019b; Hovy, 2016; Hu et al., 2023; Tian and Cui, 2023; Verma et al., 2024). However, the increasing prevalence of machine-generated content (European-Union, 2022) makes it difficult to label human-generated work for training, as even humans find it hard to distinguish between the two (Darda et al., 2023). (2) Zero-shot detectors leverage intrinsic statistical differences between machine-generated and human-generated text (Gehrmann et al., 2019; Mitchell et al., 2023; Bao et al., 2024; Yang et al., 2023a; Venkatraman et al., 2024). Proposed methods include using entropy (Lavergne et al., 2008), log probability (Solaiman et al., 2019), and more recently, intrinsic dimensionality (Tulchinskii et al., 2023). (3) Watermark-based detection, introduced by Kirchenbauer et al. (2023), involves embedding a hidden but detectable pattern in the generated output. Various enhancements to this method have been suggested (e.g., Zhao et al. (2023); Lee et al. (2023)). This paper focuses on the black-box setting, which closely resembles real-world detection scenarios. Watermarking is not tested due to its guaranteed detectability and low false positive rates (e.g., (Zhao et al., 2023)). The primary concern is detecting un-watermarked text, as it is the most commonly encountered and poses the greatest threat.

Evaluation of Text Detectors. The most commonly utilized metric in evaluating detectors is the area under the receiver operating curve (AUROC) (Mitchell et al., 2023; Sadasivan et al., 2023). Although it offers a reasonable estimate of detector performance, research by Krishna et al. (2023); Yang et al. (2023a), and our experimental results demonstrate that there can be a substantial difference in performance between two models with AUROC values nearing the maximum of 1.0. Consequently, the true positive rate at a fixed false positive rate (TPR@FPR) presents a more accurate representation of a detector's practical effectiveness. Both AUROC and true positive rate at a fixed false positive are important metrics for a complete evaluation of text detectors.

Redteaming Language Model Detectors. AI text detectors are increasingly evaluated in red teaming scenarios, with recent contributions from Zhu et al. (2023); Chakraborty et al. (2023); Kumarage et al. (2023); Shi et al. (2024); Wang et al. (2024). Shi et al. (2024) identifies two main evasion techniques: word substitution and instructional prompts. Word substitution includes query-based methods, which iteratively select low detection score substitutions, and query-free methods, which use random substitutions. Instructional prompts, akin to jailbreaking, instruct the model to mimic a human-written sample. Query-based word substitution proved most effective, reducing the True Positive Rate (TPR) to less than 5% at a 40% False Positive Rate (FPR) against DetectGPT. Wang et al. (2024) explore robustness testing of language model detectors with three editing attacks: typo insertion, homoglyph alteration, and format character editing. Typo insertion adds typos, homoglyph alteration replaces characters with similar shapes, and format character editing uses invisible text disruptions. Paraphrasing attacks, noted by Krishna et al. (2023), include synonym substitution (model-free and model-assisted), span

Method	Datasets
RADAR	OpenWebText Corpus (Gokaslan et al., 2019), Xsum (Narayan et al., 2018), SQuAD (Rajpurkar et al., 2016), Reddit Writing Prompts (Fan et al., 2018), and TOEFL (Liang et al., 2023)
Wild	Reddit CMV sub-community comments (Tan et al., 2016), Yelp Reviews (Zhang et al., 2015), Xsum (Narayan et al., 2018), TLDR_news[1], ELI5 dataset (Fan et al., 2019), Reddit Writing Prompts (Fan et al., 2018), ROCStories Corpora (Mostafazadeh et al., 2016), HellaSwag (Zellers et al., 2019a), SQuAD (Rajpurkar et al., 2016), and SciGen (Moosavi et al., 2021)
T5Sentinel	OpenWebText Corpus (Gokaslan et al., 2019)
Fast-DetectGPT	Xsum (Narayan et al., 2018), SQuAD (Rajpurkar et al., 2016), Reddit Writing Prompts (Fan et al., 2018), WMT16 English and German (Bojar et al., 2017), PubMedQA (Jin et al., 2019)
PHD	Wiki40b (Guo et al., 2020), Reddit Writing Prompts (Fan et al., 2018), WikiM (Krishna et al., 2023), StackExchange (Tulchinskii et al., 2023)
LogRank	Xsum (Narayan et al., 2018), SQuAD (Rajpurkar et al., 2016), Reddit Writing Prompts (Fan et al., 2018)
Binoculars	CCNews (Hamborg et al., 2017), PubMed (Sen et al., 2008), CNN (Hermann et al., 2015), ORCA (Lian et al., 2023)

Table 1: Datasets used for training and evaluation by each model. To avoid data leakage and cherry-picking, these datasets are excluded from the current study.

perturbations (masking and refilling random spans), and paraphrasing at sentence and text levels.

Evaluated Detectors and Datasets. In our paper, we evaluate seven representative detectors: RADAR (Hu et al., 2023), Detection in the Wild (Wild) (Li et al., 2024), T5Sentinel (Chen et al., 2023), Fast-DetectGPT (Bao et al., 2024), PHD (Tulchinskii et al., 2023), LogRank (Ippolito et al., 2020)[2], and Binoculars (Hans et al., 2024). RADAR, Wild, and T5Sentinel are trained detectors, while Fast-DetectGPT, PHD, LogRank, and Binoculars are zero-shot detectors. To ensure a fair comparison and assess the detectors' ability to generalize to new data, we carefully select datasets that have not been used in the training or evaluation of these detectors. Table 1 presents an overview of the datasets and domains on which each detector has been evaluated. Several datasets, such as Xsum, SQuAD, and Reddit Writing Prompts, have been used in the evaluation or training of multiple detectors. Although these detectors achieve strong Area Under the Receiver Operating Characteristic (AUROC) scores on these datasets, they do not report the True Positive Rate at a set False Positive Rate (TPR@FPR), which is a crucial metric in real-world scenarios. To address this gap, we aim to evaluate all seven detectors on the same datasets using both AUROC and TPR at FPR metrics.

Comparison to Previous Works. There are some other papers that have explored similar work to ours, specifically Wang et al. (2024) and Dugan et al. (2024). Our work differs from theirs in some important ways. We do not focus as much on the various methods of red-teaming the detectors in complicated ways. Rather, we explore some more natural methods that an average person might utilize in practice. We also explore in more depth the variability in detector capabilities across various tasks and languages with discussion on potential sources of that difference. And lastly, we utilize newer models, which gives insight into the adaptability of the detectors.

[2]LogRank has been evaluated on many datasets, we report the ones from Mitchell et al. (2023).

3 Benchmarking Procedure

Our benchmarking method involves compiling datasets that have not been encountered by any of the detectors during their training or evaluation phases. This approach ensures that the datasets represent new, unseen data and prevents the possibility of data contamination. For zero-shot detectors, this methodology eliminates the risk of using cherry-picked datasets that may bias the evaluation. For trained detectors, this reduces the risk of data leakage and tests on out-of-domain data. Furthermore, we assess the model's performance across a diverse range of domains that the detectors may not have been previously evaluated against. This comprehensive evaluation strategy allows for a more robust assessment of the detectors' generalization capabilities. Additionally, we evaluate the detectors on a variety of language models that they have not encountered before. This enables us to examine the detectors' performance on unfamiliar language models, providing a more comprehensive understanding of their effectiveness and adaptability.

3.1 Datasets

We evaluate each of the detectors on seven different tasks with three of the tasks, question answering, summarization, and dialogue writing, including multilingual results. The datasets chosen for each domain are as follows:

- **Question Answering:** The MFAQ dataset (De Bruyn et al., 2021) was used for this domain. It contains over one million question-answer pairs in various languages. We used the English, Spanish, French, and Chinese subsets.
- **Summarization:** We used the MTG summarization dataset (Chen et al., 2022) for this task. The complete multilingual dataset comprises roughly 200k summarizations. We utilized the English, Spanish, French, and Chinese subsets.
- **Dialogue Writing:** For this task, we utilized the MSAMSum dataset, a translated version of the SAMSum dataset(Feng et al., 2022; Gliwa

et al., 2019). This dataset consists of over 16k dialogues with summaries in six languages. We utilized English, Spanish, French, and Chinese for consistency with the other multilingual domains.

- **Code:** We used the APPS dataset (Hendrycks et al., 2021), which contains 10k code questions and solutions. The subset used was randomly selected from all the data included in APPS.
- **Abstract Writing:** For this task, we utilized the Arxiv section of the scientific papers dataset (Cohan et al., 2018) to avoid potential bias, as some detectors have previously been exposed to PubMed data. Additionally, we only selected papers published in 2020 or earlier to remove potential LLM influence.
- **Review Writing:** The PeerRead dataset was used for the review writing task (Kang et al., 2018). PeerRead contains over 10k peer reviews written by experts corresponding to the paper that they were written for.
- **Translation:** We used the Par3 dataset (Karpinska et al., 2022), which provides paragraph level translations from public-domain foreign language novels. Each paragraph includes at least 2 human translations of which we selected only one to represent human translation.

3.2 Large Language Models

Our objective is to evaluate the detectors on models that they have not previously been trained or assessed on to gauge their generalization capabilities. We evaluated 4 different models across every task. The models we use are Llama-3-Instruct 8B (AI@Meta, 2024), Mistral-Instruct-v0.3 (Jiang et al., 2023), Phi-3-Mini-Instruct 4k (Abdin et al., 2024), and GPT-4o.

3.3 Detection Models

The detection models were chosen from the newest and highest performing detectors in their respective categories. Our goal was to represent both trained and zero-shot detectors. As previously mentioned, the trained detectors we are using are RADAR (Hu et al., 2023), Detection in the Wild (Wild) (Li et al., 2024), and T5Sentinel (Chen et al., 2023). The zero-shot detectors we are using are Fast-DetectGPT (Bao et al., 2024), GPTID (Tulchinskii et al., 2023), LogRank (Ippolito et al., 2020), and Binoculars (Hans et al., 2024). Each of the zero-shot detectors utilize a generating model as a part of their detection process. We utilize the same underlying models as reported by each respective zero-shot model's original publication listed in Table 2. We also evaluate every zero-shot method using three of the other underlying models for a more accurate comparison. This is notably unfair to the Binoculars method, which uses two different underlying models: base and instruction tuned. We replace both with the same model for these experiments because not all models have both base and instruction tuned versions.

Method	Model
Fast-DetectGPT	GPT-Neo-2.7B (Black et al., 2021)
GPTID	Roberta-Base (Liu et al., 2019)
LogRank	GPT2-Medium (Radford et al., 2019)
Binoculars	Falcon-7B, Falcon-7B-Instruct (Almazrouei et al., 2023)

Table 2: Underlying models utilized by each zero-shot detection method.

Notably, we did not include any watermark detectors. The primary reason for this is that the evaluation techniques we use over various models would not work with watermark detection. While watermark detection has shown strong performance (Kirchenbauer et al., 2023), they have a significant drawback in that they only work if a model applies a watermark. In this paper, we assume a scenario in which no watermark is applied or it is unknown whether a watermark is applied. Therefore, we must turn to other detection methods.

3.4 Evaluation Metrics

In this study, we evaluate machine-generated text detectors using AUROC and TPR at a fixed FPR. Our findings, consistent with prior research (Krishna et al., 2023; Yang et al., 2023a), suggest that AUROC alone may not reflect a detector's practical effectiveness, as a high AUROC score can still correspond to significant false positive rates. This is critical since false positives, particularly in fields like academia and media, can have severe consequences. We argue that TPR at a given FPR should be the standard evaluation metric, as demonstrated by a detector achieving a 0.89 AUROC but less than 20% TPR at a 1% FPR on a task.

3.5 Red Teaming

We employ two different methods of prompting for every task: plain prompting and adversarial prompting. Plain prompting involves using a typical assistant system prompt and providing the model with the same input that was given to the human for human-generated content. Adversarial prompting, on the other hand, requests that the model try to act more like a person. Examples of the question answering plain and adversarial prompts[3] are shown as follows:

Plain Prompt Example: Question Answering

You are a helpful question answering assistant that will answer a single question as completely as possible given the information in the question. Do NOT use any markdown, bullet, or numbered list formatting. The assistant will use ONLY paragraph formatting. **Respond only in {language}**.

Adversarial Prompt Example: Question Answering

{Question answering prompt} Try to sound as human as possible.

We also conducted experiments using the LLMs as writing assistants. Specifically, we requested that the model rewrite the human response and improve upon its clarity and professionalism. This represents a scenario where a person will write down an answer first and then request that a model make their answer better before presenting it. The specific prompt we used it as follow:

Rewriting Prompt

You are a helpful writing assistant. Rewrite the following text to improve clarity and professionalism. Do not provide any other text. Only provide the rewritten text.

4 Experiment

4.1 Dataset Processing

Each dataset undergoes additional processing to prepare it for detection tasks. Research indicates that detectors of machine-generated text are more effective with longer content (Yang et al., 2023b). To leverage this, we aimed to use human samples of maximum possible length. However, the minimum length needed to obtain sufficient samples varied by task. We randomly selected 500 samples of human text from filtered subsets with the following token lengths using Llama2-13B tokenizer (Touvron et al., 2023): 500 tokens for question answering, 400 tokens for code[4], 150 tokens for summarization, 275 tokens for dialogue, 500 tokens for reviews, 500 tokens for abstracts, and 500 tokens for translation (Table 3). These 500 samples served as human examples. From them, prompts from the first 100 samples were chosen for use in the generator model, using the input given to the human author as the model prompt. This resulted in a dataset of 500 human examples and 100 machine-generated examples per model for a total of 400 machine-generated examples for each task. This slight data imbalance is intentional to ensure a more accurate TPR@FPR metric because there would likely be more human examples than machine generated examples in practice.

Task	AI		Human	
	Avg	Min	Avg	Min
Code	486.58	15	4496.88	605
QA	508.01	24	1052.37	501
Summ	410.03	18	191.00	151
Dialogue	380.92	15	402.13	276
Reviews	551.28	24	796.06	501
Abstract	427.92	30	2081.88	501
Translation	525.32	256	772.75	501

Table 3: Average and minimum token counts of machine-generated and human-generated text for each task, tokenized using the Llama2 tokenizer (Touvron et al., 2023). Minimum token counts for human-generated text are omitted as they were previously described.

Detection methods show improved performance with longer text sequences (Wu et al., 2023) so we show the statistics of the text in Table 3. Our primary focus was on detectors' ability to identify AI-generated text while maintaining a low FPR. The longer length of human-generated text is likely to enhance the TPR@FPR by making it easier to detect as human. We considered the AI-generated text sufficiently long for two reasons. First, Li et al. (2024) reports an average AI generation length of 279.99, which is much lower than our average token lengths. Their extensive training and evaluation data support the adequacy of this length for AI content. Second, our models, with a maximum generation length of 512 tokens [5], produced responses indicative of real-world lengths.

4.2 Text Generation and Detection Process

Once the prompt samples were selected, we needed to generate positive examples. The process for this can be seen in Figure 1. We employ three different

[3]The others can be found in the appendix Table 13.

[4]Length limited to 2500 tokens.

[5]The averages can exceed this number due to different tokenizers and additional tokens to keep text coherent

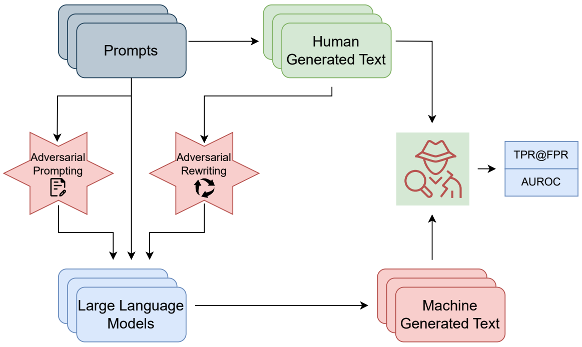


Figure 1: Pipeline for prompting and evaluation. Adversarial prompting and rewriting are applied to the LLMs. After collecting machine-generated text, AUROC and TPR@FPR are measured for each detector.

strategies for prompting the models, one is a plain prompt and the other two are adversarial prompts. The first strategy involves using a basic prompt for each domain that explains the goal of the model and the desired output format. The second strategy consists of requesting that the model be as human as possible. The third strategy requests that the model rewrite and improve upon the human written response [6]. The first strategy aims to simulate a basic system prompt that would generally be in place on a model someone is using to generate content. The second strategy simulates the case where a user might try to get the model to generate content that closely resembles human-generated content. The third strategy simulates a scenario where the user writes their own response and simply wants the model to clean it up or make it easier to understand. The outputs of the models were taken as is with no editing. After generating the positive examples, we passed all of the machine-generated and human-generated examples through the detectors. RADAR, Wild, and T5Sentinel all return a percentage probability for each class, and GPTID, Fastdetectgpt, Binoculars, and LogRank return a value representing their score. We do not use any thresholds and take the scores as is for AUROC and TPR@FPR metrics.

5 Results and Analysis

Table 4 shows the overall performance of each detector across the entire dataset. In this section, we break down the performance of each detector across tasks, languages, and prompt techniques.

[6] Prompts and templates can be found in the appendix.

Detector	TPR@0.01	TPR@0.05	TPR@0.1	AUROC
Radar	0.05	0.15	0.27	0.6009
Fast-DetectGPT	0.49	0.61	0.68	0.8405
Wild	0.11	0.19	0.29	0.6841
PHD	0.08	0.23	0.37	0.6790
LogRank	0.09	0.40	0.50	0.7763
T5Sentinel	0.03	0.09	0.14	0.5179
Binoculars	0.58	0.67	0.72	0.8485

Table 4: Performance of different detectors across the entire dataset.

5.1 Plain Prompting

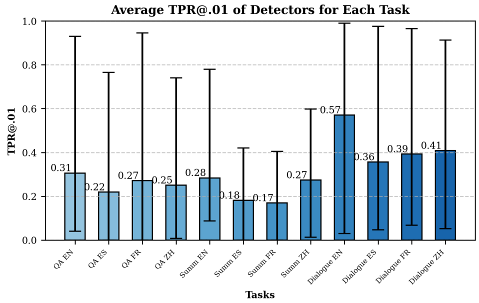
We evaluate the AUROC and TPR at 0.01 FPR for machine-generated texts from direct prompting using identical prompts as human written texts. A simple prompt was employed to ensure the generated text was in the correct format and language for the multilingual tasks.

Figures 2a and 2b show the results for the multilingual tasks and 3a and 3b show the results for the only English tasks. The results broken down by detector are shown in Appendix A.3. A significant difference is observed in detector performance across languages and tasks, particularly in the multilingual setting as well as across detectors. In the TPR@.01 setting, the difference between the best detector and worst detector is greater than 0.95. Across all detectors we generally see strong results in the English tasks, while the performance drops off in the non-English tasks. In most detectors, in all tasks, they struggle to maintain a strong TPR rate at an FPR rate of 0.01.

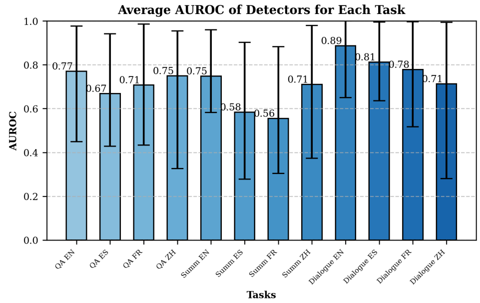
For the English-only tasks, most detectors show improved performance in the AUROC, while the TPR@0.01 stays quite low. Despite expectations that the translation domain would be the most challenging due to lower entropy in translated texts, detectors performed reasonably well from the AUROC perspective. The TPR@0.01 graph highlights ongoing challenges in maintaining low false positive rates.

5.2 Adversarial Prompting

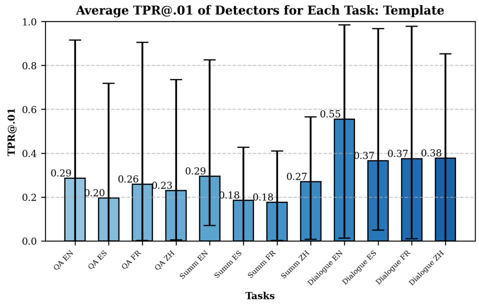
Figure 2c shows the results on the multilingual tasks where the model was instructed to be "as human as possible." Interestingly, this request had little effect on performance. In the few instances where changes occurred, scores generally increased, suggesting that asking the model to "sound human" may have made its output easier to detect. This aligns with expectations, as large language models are already trained on predominantly



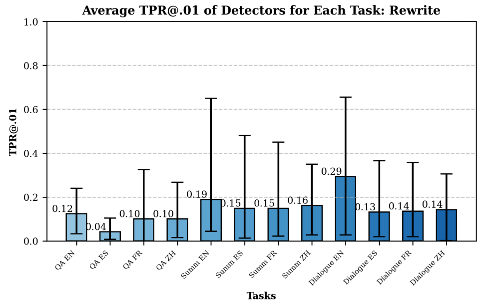
(a) Average TPR@0.01 results for multilingual tasks with normal prompting across all detectors.



(b) Average AUROC results for multilingual tasks with normal prompting across all detectors.

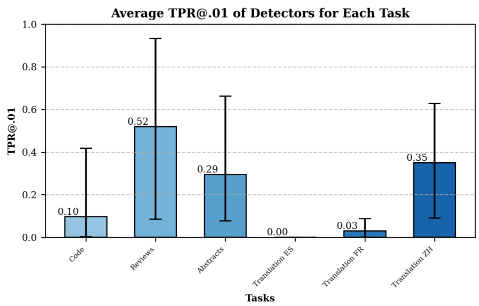


(c) Average TPR@0.01 results for multilingual tasks with template prompting across all detectors.

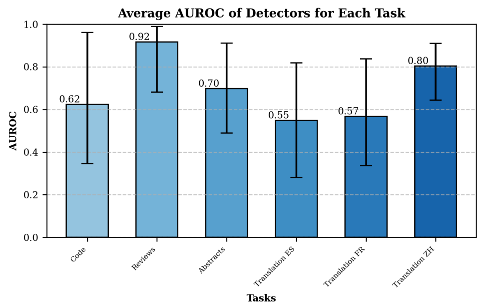


(d) Average TPR@0.01 results for multilingual tasks with rewrite prompting across all detectors.

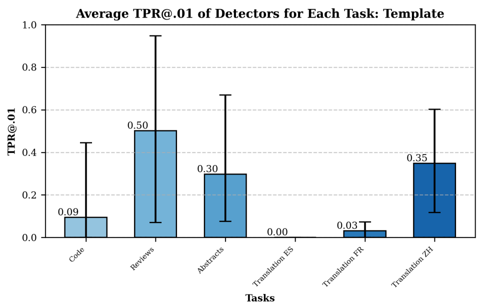
Figure 2: Comparison of average AUROC results for multilingual tasks across all detectors using different normal prompting and average TPR@0.01 across all detectors using normal, template, and rewrite prompting. Error bars show maximum and minimum performance across detectors.



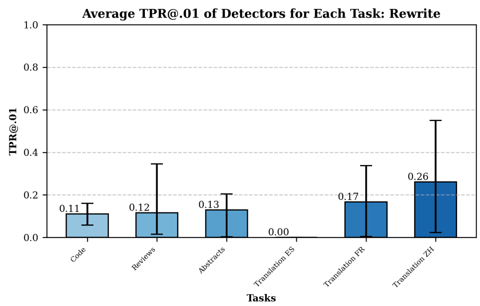
(a) Average TPR@0.01 results for English tasks with normal prompting across all detectors.



(b) Average AUROC results for English tasks with normal prompting across all detectors.



(c) Average TPR@0.01 results for English tasks with template prompting across all detectors.



(d) Average TPR@0.01 results for English tasks with rewrite prompting across all detectors.

Figure 3: Comparison of average AUROC results for English tasks across all detectors using different normal prompting and average TPR@0.01 across all detectors using normal, template, and rewrite prompting. Error bars show maximum and minimum performance across detectors.

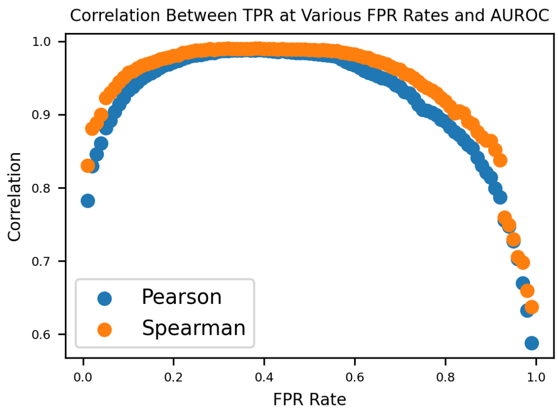


Figure 4: Correlations between the TPR at various FPR rates and the overall AUROC score. AUROC score is more representative of the middle FPR rates, while this detection task is more concerned with the lower end of FPR.

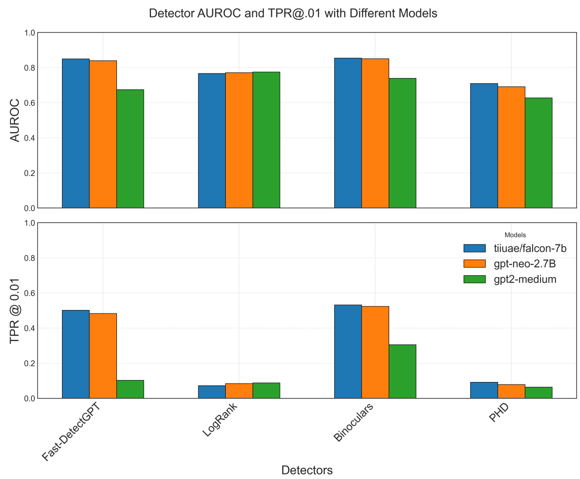


Figure 5: AUROC and TPR@0.01 for each zero-shot method using various underlying models. Only Fast-DetectGPT and Binoculars show a significant change in performance with GPT2-Medium.

human-written texts, and generating more conversational output can make detection more straightforward, as evidenced in dialogue generation tasks.

On the English tasks, as shown in figure, 3c, the results were similarly unaffected by the human-like request, with some slight score increases where changes were observed. This is especially expected in domains such as reviews, code, and abstracts, which follow specific writing conventions, while tasks like question answering and dialogue generation exhibit more variability and creativity.

5.3 Rewriting

Finally, we show the results for the rewriting prompt for the multilingual tasks in figure 2d and for the English tasks in figure 3d. We observe a notable decrease in TPR@0.01 performance for detectors that previously performed well leading to a drop in the average performance in most tasks. Some of the lower performing did see an increase in performance which is why the average performance in the Code and French Translation tasks are slightly higher. Despite these shifts, the relative performance across tasks remains consistent, indicating an inherent variability in detectability based on the type of task and language.

5.4 TPR@FPR vs AUROC

In this paper, we utilize both the AUROC and TPR@FPR metrics. However, we also argue that TPR at a low FPR is a much more important metric for this detection task. Figure 4 shows the correlation between TPR scores at various FPR rates and the AUROC score for all tasks, detectors, and models used in this research. The AUROC correlates much higher with FPR rates in the 0.4 to 0.6 range and much lower with FPR rates at the edges, less than 0.2 and greater than 0.8. While the 0.75 is still a reasonable correlation value, the AUROC is still much more representative of the middle FPR's while we are really concerned with the lower FPR's for this task. This is why we report the TPR@0.01, which is much more representative of the applicability of a detector than the AUROC.

5.5 Output Quality and Detection

Measuring the quality of LLM outputs, especially in creative tasks, remains challenging, making it difficult to determine if higher-quality outputs are harder to detect. Table 5 compares various models' performance scores and rankings from Chatbot Arena (Chiang et al., 2024), allowing us to explore if output quality affects detectability. The data shows little difference in detectability across models of varying quality, with AUROC and TPR@0.01 scores remaining consistent. This suggests that output quality does not significantly impact the difficulty of detection, though further research is needed for a fuller understanding.

5.6 Impact of Model on Zero-shot Methods

Each zero-shot method used in this paper has an underlying model that assists in the detection process. In this paper we consider the model chosen by the respective authors of each detector to be a part of the detector itself. However, we also swapped

Model	Code		Reviews		Abstract		QA		Summ		Dialogue		Trans.		Arena Score
	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	
GPT-4o	0.02	0.55	0.28	0.63	0.04	0.53	0.05	0.54	0.05	0.50	0.03	0.58	0.03	0.52	1339
Llama-3	0.06	0.56	0.28	0.67	0.21	0.64	0.12	0.60	0.07	0.57	0.08	0.60	0.09	0.56	1152
Mistral	0.02	0.54	0.28	0.65	0.04	0.54	0.10	0.58	0.04	0.51	0.06	0.59	0.05	0.54	1072
Phi-3	0.04	0.57	0.24	0.62	0.13	0.58	0.08	0.58	0.12	0.58	0.13	0.63	0.08	0.56	1066

Table 5: Model performance (AUROC and TPR@0.01) across tasks compared with model generation quality. The Chatbot Arena score is utilized to measure the quality of a model. The higher scores do not correlate with lower detectability of generated content.

out each model to directly compare the statistical methods themselves, removing any impact from a specific model on a detector.

Figure 5 shows the results of running each detector across the entire dataset with three different models. There is generally not much of a difference in the ability of a detection method when changing the underlying model. Fast-DetectGPT and Binoculars show a small change in AUROC and a larger change in TPR@0.01 when using the gpt2-medium model (Radford et al., 2019). Gpt2-medium is the oldest model of the three, which likely results in its output logits being different than the generation models more often. This provides some evidence that these zero-shot methods will require updated underlying models to remain successful on more advanced generation models, but more research would need to be conducted.

6 Conclusion

This study evaluates seven advanced detectors across seven tasks and four languages, revealing notable inconsistencies in their detection capabilities. We also examined three different prompting strategies and their impact on detectability, finding that requests for more "human-like" output do not make the text harder to detect, while rewritten human content proves more difficult to identify.

The detection results for both the Translation task and Rewrite prompt are generally lower than the average detectability for other machine generated text. This encourages a discussion about whether this type of text should be detected as machine generated or not. The text may have been machine generated, but it is heavily influenced by human generated text. Specifically in the translation case, the text should match the source text in another language. In the Rewrite case, the model is not generating any new ideas, just improving the readability of the text. It is clear from the results that these cases are harder to detect than when a model has a more open-ended generation. It is likely worth differentiating between these two cases which we leave to future work.

Additionally, this research highlights the limitations of relying on the AUROC metric for assessing machine-generated content detectors. Our findings emphasize the need for robust evaluation methods to develop more reliable detection techniques. The study underscores the challenges in detecting machine-generated text, particularly when human written text was only modified by a language model, and advocates for TPR@FPR as the preferred evaluation metric to better capture detector performance.

7 Limitations

A limitation of this method is the settings in which the human data was collected may vary from the settings in which these detectors will be used. Additionally, some of the datasets we used had collected their data from the internet which raises a concern that some of that data is not completely human generated. This is a challenge that all future detectors will also struggle with when training and evaluating. These results pose the risk of emboldening users to use AI generated content when they otherwise should not because they know detectors cannot be confidently trusted. However, acknowledging this is important to encouraging research into new detection methods and improving current methods.

Acknowledgements

L.L. is partly supported by a gift from Hydrox AI and a CMU CyLab Seed Fund. The views expressed are those of the author and do not reflect the official policy or position of the funding agencies.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv preprint*, abs/2404.14219.

AI Elections Accord. 2024. A tech accord to combat deceptive use of ai in 2024 elections.

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*, abs/2303.08774.

AI@Meta. 2024. Llama 3 model card.

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

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *ICLR*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. If you use this software, please cite it using these metadata.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

Megha Chakraborty, SM Tonmoy, SM Zaman, Krish Sharma, Niyar R Barman, Chandan Gupta, Shreya Gautam, Tanay Kumar, Vinija Jain, Aman Chadha, et al. 2023. Counter turing test ctˆ 2: Ai-generated text detection is not as easy as you may think–introducing ai detectability index. *ArXiv preprint*, abs/2310.05030.

Yiran Chen, Zhenqiao Song, Xianze Wu, Danqing Wang, Jingjing Xu, Jiaze Chen, Hao Zhou, and Lei Li. 2022. MTG: A benchmark suite for multilingual text generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2508–2527.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Token prediction as implicit classification to identify LLM-generated text. In *Proc. of EMNLP*, pages 13112–13120.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proc. of NAACL-HLT*, pages 615–621.

Kohinoor Darda, Marion Carre, and Emily Cross. 2023. Value attributed to text-based archives generated by artificial intelligence. *Royal Society Open Science*, 10(2):220915.

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2021. MFAQ: a multilingual FAQ dataset. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 1–13.

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors.

European-Union. 2022. Facing reality? law enforcement and the challenge of deepfakes, an observatory report from the europol innovation lab.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proc. of ACL*, pages 3558–3567.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proc. of ACL*, pages 889–898.

Shangbin Feng, Herun Wan, Ningnan Wang, Zhaoxuan Tan, Minnan Luo, and Yulia Tsvetkov. 2024. What does the bot say? opportunities and risks of large language models in social media bot detection. *ArXiv preprint*, abs/2402.00371.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. MSAMSum: Towards benchmarking multi-lingual dialogue summarization. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 1–12.

Kathleen C. Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2024. Detecting ai-generated text: Factors influencing detectability with current methods. *Preprint*, arXiv:2406.15583.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proc. of ACL*, pages 111–116.

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

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40B: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2440–2452.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please. In *Everything Changes, Everything Stays the Same? Understanding Information Spaces. Proceedings of the 15th International Symposium of Information Science (ISI 2017)*, pages 218–223.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *ArXiv preprint*, abs/2401.12070.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. Measuring coding challenge competence with apps. *ArXiv preprint*, abs/2105.09938.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews – an adversarial study. In *Proc. of ACL*, pages 351–356.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: robust ai-text detection via adversarial learning. In *Proc. of NeurIPS*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *ArXiv preprint*, abs/2310.06825.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proc. of EMNLP*, pages 2567–2577.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proc. of NAACL-HLT*, pages 1647–1661.

Marzena Karpinska, Katherine Thai, Kalpesh Krishna, John Wieting, Moira Inghilleri, and Mohit Iyyer. 2022. Par3.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proc. of ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084.

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. In *Proc. of NeurIPS*.

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. How reliable are AI-generated-text detectors? an assessment framework using evasive soft prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1337–1349.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. *Pan*, 8(27-31):4.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023. Who wrote this code? watermarking for code generation. *ArXiv preprint*, abs/2305.15060.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proc. of ACL*, pages 36–53.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7).

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. *Preprint*, arXiv:1907.11692.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proc. of ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962.

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

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proc. of NAACL-HLT*, pages 839–849.

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 *Proc. of EMNLP*, pages 1797–1807.

Mike Perkins. 2023. Academic integrity considerations of ai large language models in the post-pandemic era: Chatgpt and beyond. *Journal of university teaching & learning practice*, 20(2):07.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 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. In *Proc. of EMNLP*, pages 2383–2392.

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

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine*, 29(3):93.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2024. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics*, 12:174–189.

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*, abs/1908.09203.

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.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *ArXiv preprint*, abs/2303.07205.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey I. Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. In *Proc. of NeurIPS*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2024. GPT-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 103–115.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717.

Yichen Wang, Shangbin Feng, Abe Bohan Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. 2024. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. *ArXiv preprint*, abs/2402.11638.

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

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

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023b. A survey on detection of llms-generated content. *ArXiv preprint*, abs/2310.15654.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL*, pages 4791–4800.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019b. Defending against neural fake news. In *Proc. of NeurIPS*, pages 9051–9062.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proc. of NeurIPS*, pages 649–657.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *ArXiv preprint*, abs/2306.17439.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *ArXiv preprint*, abs/2306.04528.

A More Results

This section contains results for detections by models and tasks, the prompts used for plain prompting, and the results by detector.

A.1 Results by Model

The following tables show the results for each detector by generation model and task. As discussed in the paper, there is not a significant difference in detectability of a text by the model that generated that text. More specifically, a higher quality model like GPT-4o is not noticeably harder to detect than a lower quality model like Phi-3. The differences in detectability are more obvious across tasks than generation models.

Model	Detector	TPR@.01	AUROC
GPT-4o	Binoculars	0.14	0.8739
	Fast-DetectGPT	0.07	0.8223
	LogRank	0.00	0.5722
	PHD	0.00	0.3976
	Radar	0.00	0.6594
	T5Sentinel	0.01	0.4604
	Wild	0.00	0.4724
Llama-3	Binoculars	0.45	0.9474
	Fast-DetectGPT	0.16	0.8211
	LogRank	0.00	0.6060
	PHD	0.04	0.4940
	Radar	0.05	0.8088
	T5Sentinel	0.09	0.5358
	Wild	0.06	0.6231
Mistral	Binoculars	0.39	0.9681
	Fast-DetectGPT	0.16	0.9029
	LogRank	0.06	0.4552
	PHD	0.06	0.3104
	Radar	0.00	0.5787
	T5Sentinel	0.02	0.3315
	Wild	0.07	0.5412
Phi-3	Binoculars	0.35	0.7825
	Fast-DetectGPT	0.21	0.8076
	LogRank	0.08	0.5470
	PHD	0.14	0.4657
	Radar	0.06	0.7700
	T5Sentinel	0.03	0.5209
	Wild	0.12	0.5779

Table 6: Code

A.2 Plain Prompts

Table 13 shows the prompts used for each task in the plain prompting. The prompts we used were intentionally very simple and not overly instructive. This is because we wanted to replicate a realistic scenario of an average person prompting a language model. We performed small ablations on these prompts and found no difference in detectability.

Model	Detector	TPR@.01	AUROC
GPT-4o	Binoculars	0.47	0.8676
	Fast-DetectGPT	0.35	0.8918
	LogRank	0.01	0.5972
	PHD	0.01	0.4758
	Radar	0.02	0.3474
	T5Sentinel	0.01	0.4909
	Wild	0.02	0.5088
Llama-3	Binoculars	0.66	0.9501
	Fast-DetectGPT	0.68	0.9465
	LogRank	0.03	0.7816
	PHD	0.02	0.6637
	Radar	0.13	0.6581
	T5Sentinel	0.03	0.4995
	Wild	0.06	0.6024
Mistral	Binoculars	0.56	0.8954
	Fast-DetectGPT	0.57	0.8852
	LogRank	0.04	0.7052
	PHD	0.04	0.6261
	Radar	0.04	0.5776
	T5Sentinel	0.02	0.5078
	Wild	0.04	0.5664
Phi-3	Binoculars	0.48	0.8188
	Fast-DetectGPT	0.41	0.8424
	LogRank	0.13	0.7448
	PHD	0.13	0.6553
	Radar	0.07	0.5966
	T5Sentinel	0.02	0.5787
	Wild	0.07	0.5923

Table 7: Question Answering

Model	Detector	TPR@.01	AUROC
GPT-4o	Binoculars	0.05	0.6533
	Fast-DetectGPT	0.11	0.6981
	LogRank	0.23	0.7070
	PHD	0.00	0.4922
	Radar	0.00	0.2085
	T5Sentinel	0.01	0.4339
	Wild	0.11	0.4753
Llama-3	Binoculars	0.33	0.8134
	Fast-DetectGPT	0.19	0.7540
	LogRank	0.61	0.8941
	PHD	0.30	0.6642
	Radar	0.08	0.6605
	T5Sentinel	0.05	0.4720
	Wild	0.15	0.6570
Mistral	Binoculars	0.06	0.6011
	Fast-DetectGPT	0.08	0.5862
	LogRank	0.30	0.7635
	PHD	0.01	0.5216
	Radar	0.00	0.3428
	T5Sentinel	0.02	0.4256
	Wild	0.13	0.5931
Phi-3	Binoculars	0.48	0.7501
	Fast-DetectGPT	0.30	0.7135
	LogRank	0.60	0.9049
	PHD	0.33	0.7668
	Radar	0.09	0.7492
	T5Sentinel	0.01	0.3561
	Wild	0.38	0.8890

Table 8: Summarization

Model	Detector	TPR@.01	AUROC
	Binoculars	0.68	0.9362
	Fast-DetectGPT	0.52	0.9277
	LogRank	0.37	0.7857
GPT-4o	PHD	0.07	0.6279
	Radar	0.03	0.6143
	T5Sentinel	0.06	0.5045
	Wild	0.01	0.5633
	Binoculars	0.80	0.9695
	Fast-DetectGPT	0.67	0.9487
	LogRank	0.42	0.8235
Llama-3	PHD	0.24	0.6772
	Radar	0.14	0.6907
	T5Sentinel	0.04	0.5265
	Wild	0.03	0.6378
	Binoculars	0.64	0.9150
	Fast-DetectGPT	0.56	0.9246
	LogRank	0.33	0.7827
Mistral	PHD	0.17	0.6514
	Radar	0.19	0.6989
	T5Sentinel	0.02	0.4949
	Wild	0.02	0.6044
	Binoculars	0.69	0.9243
	Fast-DetectGPT	0.59	0.8506
	LogRank	0.60	0.9206
Phi-3	PHD	0.37	0.7745
	Radar	0.04	0.7626
	T5Sentinel	0.11	0.6134
	Wild	0.10	0.6505

Table 9: Dialogue

Model	Detector	TPR@.01	AUROC
	Binoculars	0.39	0.8584
	Fast-DetectGPT	0.33	0.8895
	LogRank	0.00	0.6474
GPT-4o	PHD	0.00	0.2399
	Radar	0.01	0.1890
	T5Sentinel	0.00	0.3086
	Wild	0.06	0.6178
	Binoculars	0.71	0.9248
	Fast-DetectGPT	0.70	0.9352
	LogRank	0.12	0.8505
Llama-3	PHD	0.09	0.5909
	Radar	0.29	0.6430
	T5Sentinel	0.16	0.6843
	Wild	0.42	0.8185
	Binoculars	0.44	0.8672
	Fast-DetectGPT	0.38	0.8937
	LogRank	0.01	0.6964
Mistral	PHD	0.01	0.3262
	Radar	0.03	0.1770
	T5Sentinel	0.05	0.4642
	Wild	0.04	0.5601
	Binoculars	0.38	0.5372
	Fast-DetectGPT	0.43	0.7147
	LogRank	0.30	0.8252
Phi-3	PHD	0.48	0.7344
	Radar	0.42	0.8498
	T5Sentinel	0.01	0.3748
	Wild	0.49	0.9189

Table 10: Abstract

Model	Detector	TPR@.01	AUROC
	Binoculars	0.69	0.9247
	Fast-DetectGPT	0.65	0.8847
	LogRank	0.31	0.8778
GPT-4o	PHD	0.00	0.7663
	Radar	0.12	0.8791
	T5Sentinel	0.02	0.5718
	Wild	0.00	0.9249
	Binoculars	0.80	0.9662
	Fast-DetectGPT	0.72	0.9305
	LogRank	0.65	0.9240
Llama-3	PHD	0.47	0.8589
	Radar	0.39	0.8883
	T5Sentinel	0.11	0.5814
	Wild	0.30	0.9541
	Binoculars	0.79	0.9659
	Fast-DetectGPT	0.74	0.9358
	LogRank	0.60	0.9249
Mistral	PHD	0.06	0.8347
	Radar	0.37	0.9015
	T5Sentinel	0.07	0.6376
	Wild	0.42	0.9574
	Binoculars	0.70	0.8889
	Fast-DetectGPT	0.49	0.8178
	LogRank	0.40	0.8272
Phi-3	PHD	0.17	0.8137
	Radar	0.47	0.9024
	T5Sentinel	0.03	0.4166
	Wild	0.25	0.9410

Table 11: Reviews

Model	Detector	TPR@.01	AUROC
	Binoculars	0.12	0.7020
	Fast-DetectGPT	0.05	0.6539
	LogRank	0.02	0.6059
GPT-4o	PHD	0.02	0.5611
	Radar	0.01	0.6624
	T5Sentinel	0.01	0.3732
	Wild	0.13	0.5788
	Binoculars	0.54	0.8739
	Fast-DetectGPT	0.40	0.8146
	LogRank	0.23	0.7281
Llama-3	PHD	0.08	0.6874
	Radar	0.18	0.8910
	T5Sentinel	0.06	0.4902
	Wild	0.37	0.7462
	Binoculars	0.31	0.7781
	Fast-DetectGPT	0.14	0.7139
	LogRank	0.05	0.6167
Mistral	PHD	0.03	0.6115
	Radar	0.08	0.8602
	T5Sentinel	0.02	0.3869
	Wild	0.19	0.6562
	Binoculars	0.36	0.7671
	Fast-DetectGPT	0.20	0.6446
	LogRank	0.09	0.6377
Phi-3	PHD	0.30	0.7130
	Radar	0.36	0.9726
	T5Sentinel	0.01	0.4116
	Wild	0.45	0.8199

Table 12: Translation

Task	Prompt
Code	You are a helpful code assistant that can teach a junior developer how to code. Your language of choice is Python. Don't explain the code, just generate the code block itself.
Question Answering	You are a helpful question answering assistant who will answer a single question as completely as possible given the information in the question. Do NOT use any markdown, bullet, or numbered list formatting. The assistant will use ONLY paragraph formatting. **Respond only in {language}**
Summarization	You are a helpful summarization assistant that will summarize a given article. Provide only the summarization in paragraph formatting. Do not introduce the summary. **Respond in {language}**
Dialogue	You are a helpful dialogue generation assistant that will generate a dialogue between people given a short paragraph describing the people involved. Provide only the dialogue. Do not introduce the dialogue. **Respond in {language}**
Abstract Writing	You are a helpful abstract writing assistant. You will write an abstract given the content of a paper. Do not provide any other text. You will only provide an abstract.
Review Writing	You are a helpful conference paper review assistant. Please provide a detailed review of the following paper, including its strengths, weaknesses, and suggestions for improvement.
Translation	You are a helpful translation assistant that will translate a given text into English. Provide only the translation and nothing else.
Rewriting	You are a helpful writing assistant. Rewrite the following text to improve clarity and professionalism. Do not provide any other text. Only provide the rewritten text.

Table 13: The table shows the prompts used in the plain prompting. For GPT, these were used as system prompts, and for huggingface models they were prepended to the questions.

A.3 Results by Detector

This section shows the numerical value of each detector on each task. In the paper we display graphs representing most of these values but show all of the numbers here for reference. The TPR@.01 and AUROC change significantly across tasks for every detector signifying that these detectors are not equally capable of detecting all types of machine generated text.

	Code		**Reviews**		**Abstract**		**Translation ES**		**Translation FR**		**Translation ZH**	
	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC
Radar	0.0258	0.7042	0.3358	0.8928	0.1858	0.4647	0.1806	0.8528	0.1475	0.8565	0.3625	0.8307
Fast-DetectGPT	0.1508	0.8385	0.6475	0.8922	0.4608	0.8583	0.1782	0.6337	0.0817	0.6078	0.3333	0.7953
Wild	0.0608	0.5537	0.2425	0.9443	0.2517	0.7288	0.1759	0.7433	0.0875	0.5455	0.5358	0.9049
PHD	0.0625	0.4169	0.1758	0.8184	0.1467	0.4729	0.0556	0.5264	0.0075	0.6328	0.2283	0.7915
LogRank	0.0367	0.5451	0.4883	0.8884	0.1075	0.7549	0.0509	0.4446	0.0600	0.5900	0.1792	0.7820
T5Sentinel	0.0400	0.4621	0.0575	0.5519	0.0575	0.4580	0.0231	0.2582	0.0025	0.3473	0.0767	0.5286
Binoculars	0.3317	0.8930	0.7450	0.9364	0.4783	0.7969	0.1944	0.6989	0.1492	0.6754	0.5233	0.8807

Table 14: Detector performance (AUROC and TPR@0.01) across tasks.

	QA EN		**QA ES**		**QA FR**		**QA ZH**	
	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC
Radar	0.1225	0.7542	0.0100	0.4832	0.0525	0.5008	0.0400	0.3730
Fast-DetectGPT	0.5450	0.9063	0.4233	0.8522	0.6483	0.9390	0.5808	0.8864
Wild	0.1375	0.7437	0.0308	0.5019	0.0183	0.4660	0.0275	0.6180
PHD	0.0600	0.4694	0.0792	0.6228	0.0158	0.6371	0.0733	0.7138
LogRank	0.0725	0.7517	0.0725	0.6427	0.0200	0.7741	0.1308	0.8323
T5Sentinel	0.0558	0.6007	0.0075	0.4353	0.0025	0.4131	0.0175	0.6688
Binoculars	0.6950	0.9271	0.5292	0.8295	0.7250	0.9326	0.4908	0.8785

Table 15: Detector performance (AUROC and TPR@0.01) across multilingual QA tasks.

	Summ EN		**Summ ES**		**Summ FR**		**Summ ZH**	
	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC
Radar	0.0825	0.5721	0.0125	0.3839	0.0208	0.3966	0.2433	0.6940
Fast-DetectGPT	0.2175	0.8185	0.1408	0.6446	0.1058	0.6169	0.1950	0.6729
Wild	0.1250	0.7436	0.2683	0.6027	0.2350	0.5578	0.4392	0.8499
PHD	0.1292	0.5907	0.1417	0.6107	0.1617	0.5875	0.1933	0.6631
LogRank	0.7517	0.9705	0.4425	0.8902	0.4217	0.8754	0.4533	0.8317
T5Sentinel	0.1333	0.6275	0.0042	0.2728	0.0208	0.3219	0.0183	0.3828
Binoculars	0.3792	0.7916	0.2225	0.6866	0.1942	0.6482	0.1333	0.6935

Table 16: Detector performance (AUROC and TPR@0.01) across multilingual summarization tasks.

	Dialogue EN		Dialogue ES		Dialogue FR		Dialogue ZH	
	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC
Radar	0.7175	0.9344	0.0583	0.6827	0.0417	0.5549	0.1317	0.6120
Fast-DetectGPT	0.7475	0.9508	0.4300	0.8847	0.6158	0.9098	0.4217	0.9011
Wild	0.0292	0.8720	0.0775	0.6266	0.1000	0.6035	0.0617	0.3290
PHD	0.0425	0.6224	0.1892	0.7168	0.1842	0.7401	0.3835	0.7962
LogRank	0.8583	0.9889	0.4250	0.8824	0.4283	0.8987	0.5208	0.8870
T5Sentinel	0.1708	0.6628	0.0467	0.6010	0.0508	0.4639	0.0217	0.3446
Binoculars	0.7767	0.9489	0.7700	0.9407	0.7125	0.9393	0.6467	0.9475

Table 17: Detector performance (AUROC and TPR@0.01) across multilingual dialogue tasks.