FastDetectGPT XS - Tiny Models for Machine-generated Text Detection

Christopher Adrian Kusuma and Shirshajit Sen Gupta
National University of Singapore

Abstract

Given the ever-increasing capabilities of large language models (LLMs) to generate coherent, error-free text, tools to detect generated text are in high demand. In this paper, we propose an improvement on the current state-of-the-art machine-generated text detection architecture, FastDetectGPT, by shrinking the backbone size and thereby also decreasing inference costs. Our FastDetectGPT XS has an average AUC score of 0.9756, beating DetectGPT's AUC of 0.8519 and FastDetectGPT's 0.9677.

1 Introduction

Large language models (LLMs) have shown rapid improvement in language generation capabilities and are now being incorporated into a wide variety of tasks with human levels of fluency (Liang et al., 2022; Yuan et al., 2022). For older models, humans can discern between real and machine-generated text fairly well (Ippolito et al., 2019; Dugan et al., 2020, 2023). At the same time, we are seeing the rise of abusive LLM usage from fake product reviews (Stiff and Johansson, 2022; Adelani et al., 2020) to plagiarism (Dehouche, 2021).

Differentiating human or machine-generated text can be seen as a binary classification problem, leading to research in building supervised classifiers and zero-shot detectors. RoBERTa-based supervised classifiers (Guo et al., 2023; Liu et al., 2023a,b) yielded high accuracy in detecting machine-generated text within their fine-tuned domain. However, they lack robustness to out-of-distribution texts (Bakhtin et al., 2019; Antoun et al., 2023) as they tend to overfit their training data. On the other hand, zero-shot detectors (Gehrmann et al., 2019; Mitchell et al., 2023; Bao et al., 2023) are typically more robust and do not suffer from overfitting. DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2023) utilise information from the local likelihood structure around an input text. They observe that perturbed machine texts generally have lower log-likelihood, while perturbed human texts may have higher or lower log-likelihood than the input text. Intriguingly, Mireshghallah et al. (2023) find that smaller language models are better universal detectors when coupled with the DetectGPT pipeline.

In this paper, we are interested in investigating the upper limit of a model size to maximise the accuracy of detecting machine-generated text.

Our main contributions are as follows:

- We determine the relation between the discriminator model size and model performance
- We make an in-depth exploration of the ability of cross-model generator-discriminator pairs
- FastDetectGPT XS, a state-of-the-art model for detecting generated text across multiple model architectures

2 Method

We are interested in a black-box zero-shot machine-generated text detector. In black-box settings, we do not know the generator model. Instead, we rely on a proxy model to evaluate the text. Fast-DetectGPT (Bao et al., 2023) requires a sampling model and a scoring model. Here, we tested two different settings: *1)* different sampling-scoring models, *2)* same sampling-scoring model.

2.1 Fast-DetectGPT

We follow the detection steps in Fast-DetectGPT (Bao et al., 2023): sampling, scoring, and comparing.

Sampling. The idea is that humans and machines tend to have different word choices during text generation. Under this assumption, the sampling can be done simply by inputting a text x into a *sampling model* and then introducing randomness in

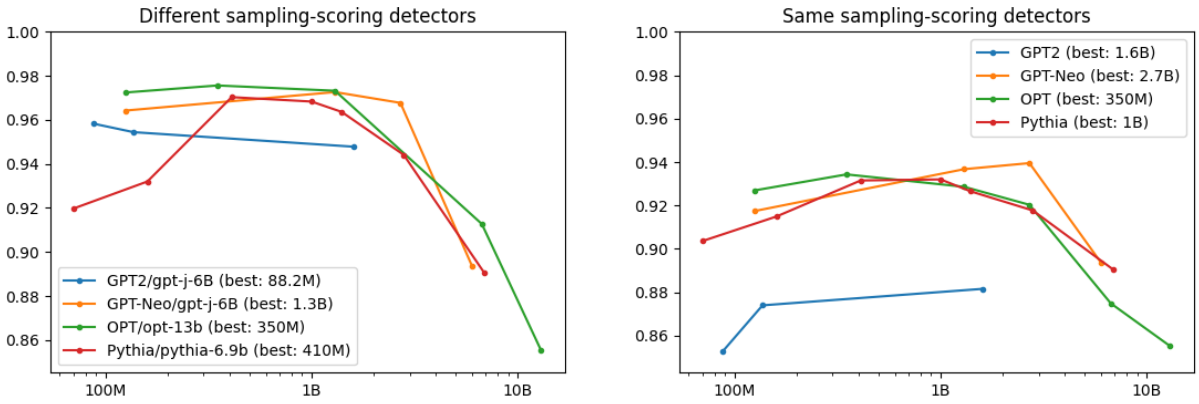

Figure 1: Detection results on text generated by small language models. *Left*: We use different sampling-scoring models as the detector. *Right*: We use same sampling-scoring models as the detector.

the sampling process.

Scoring. Unlike DetectGPT (Mitchell et al., 2023), where the perplexity of the text determines the score, the scoring process is conditioned on the input text. It calculates the score as the log-likelihood of the sampled text. To be precise, given an input text x, a sampling model φ, a scoring model θ, the likelihood is defined as:

$$p_\theta(\tilde{x}|x) = \prod_j p_\theta(\tilde{x}_j|x_{<j}) \tag{1}$$

then the score is calculated as:

$$d(x, \varphi, \theta) = \frac{\log p_\theta(x) - \tilde{\mu}}{\tilde{\sigma}} \tag{2}$$

where

$$\tilde{\mu} = \mathbb{E}_{\tilde{x}\sim\varphi(\tilde{x}|x)} \left[\log p_\theta(\tilde{x}|x)\right] \tag{3}$$

and

$$\tilde{\sigma}^2 = \mathbb{E}_{\tilde{x}\sim\varphi(\tilde{x}|x)} \left[(\log p_\theta(\tilde{x}|x) - \tilde{\mu})^2\right] \tag{4}$$

The score will be referred to as a sampling discrepancy.

Comparing. Given a set of human and machine texts, we calculate the sampling discrepancy for each text and compare the new input x score against the collected scores.

2.2 Smaller Models

Mireshghallah et al. (2023) discover that smaller language models are better universal detectors when coupled with the DetectGPT pipeline. We are interested in investigating the upper limit of a model size to maximise the accuracy of detecting machine-generated text. They determined that smaller models like OPT-125M outperform larger ones like GPTJ-6B and OPT-6.7B in detecting machine-generated text. We expand on their work by testing a wider range of models, as well as using an improved version of DetectGPT, i.e., FastDetectGPT.

3 Experiments

This section presents the datasets and models we used for the experiments. Then, we show the results measuring the detection accuracy as AUROC.

3.1 Datasets and Models

Datasets. We follow the technique used by Mitchell et al. (2023) to collect the human-machine text dataset. The texts are collected from several datasets to cover various domains, such as *XSum* for news articles (Narayan et al., 2018), *SQuAD* for Wikipedia contexts (Rajpurkar et al., 2016), *WritingPrompts* for story writing (Fan et al., 2018), and *PubmedQA* for biomedical research question answering (Jin et al., 2019). For each dataset, we randomly sample 500 human-written examples as human texts and generate an equal number of machine texts by prompting the generator model with the first 30 tokens of the human-written example. The generator models used here are divided into two groups: *1)* the small models (up to 20B parameters), including GPT2-XL, OPT-2.7, Neo-2.7, GPT-J, and NeoX; and *2)* the large models, including GPT-3, ChatGPT, and GPT-4.

Models. We use models varying in sizes from 88M up to 13B parameters as sampling and scoring models. These models include:

- **GPT-2** DistilGPT2, GPT2, and GPT2-XL: GPT-based models with 88M, 124M, and

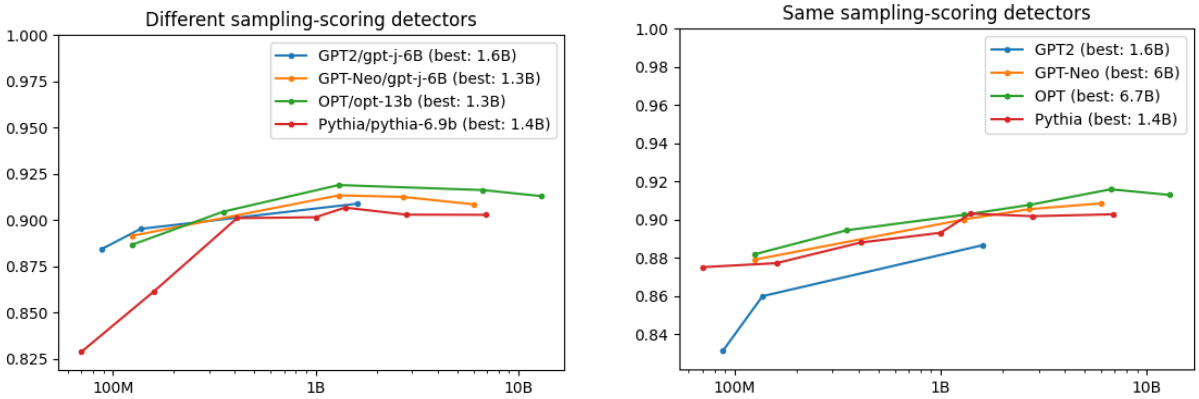

Figure 2: Detection results on text generated by large language models. *Left*: We use different sampling-scoring models as the detector. *Right*: We use the same sampling-scoring models as the detector.

1558M parameters, respectively.

- **GPT-Neo** Neo-125M, Neo-1.3B, Neo-2.7B, and GPT-j: GPT-Neo is an open-source implementation of GPT-3 trained on The Pile dataset.
- **OPT** OPT-125M, OPT-350M, OPT-1.3B, OPT-2.7B, OPT-6.7B, and OPT-13B: Open Pre-trained Transformer models by Meta AI.
- **Pythia** Pythia-70M, Pythia-160M, Pythia-410M, Pythia-1B, Pythia-1.4B, Pythia-2.8B, and Pythia-6.9B: Models trained with consistent data order across scales(Biderman et al., 2023).

3.2 Results

Figure 1 reports the results, averaged across all datasets, for different sampling-scoring models (left) and the same sampling-scoring models (right) for text generated by the small language models. We observe that the performance increases as the model size increases until a certain point, after which the performance decreases. Typically, the performance drops when the model size is larger than 1B parameters, indicating these models are less generalisable for detecting text from various generator models. Another interesting observation is that different sampling-scoring models outperform the same ones, indicating the necessity for a better sampling model. Similar observations are also seen in Figure 2 for text generated by the large language models. However, the turning points are larger for all model families. This suggests that slightly larger models are better at detecting text generated by a large language model, such as ChatGPT. However, we do not see a significant drop, as seen in the case of small language models. Perhaps this drop can be observed if we consider a much larger model (more than 13B parameters) as the scoring model.

4 Discussion

Log-Likelihoods. To better understand this behaviour, we randomly select a few human texts and their respective machine version (GPT2-XL) and plot the log-likelihoods (Figure 3) and sampling discrepancy (Figure 4) on two different settings: large scoring model (left) and small scoring model (right). We make the following observations:

- A large scoring model often gives a lower log-likelihood to generations from a smaller language model relative to its sampled generations. Consequently, its sampling discrepancies shown in Figure 4 (left) are distributed around zero.
- Human texts' log-likelihood fluctuates around their sampled mean. As a result, the sampling discrepancies are distributed around zero.
- A smaller scoring model correctly gives machine texts a higher log-likelihood, leading to better separation between human and machine sampling discrepancies.

Robustness to Attacks. We subject the FastDetectGPT XS model to *paraphrasing* and *decoherence* attacks, following the work from (Bao et al., 2023). Given the prevalence of GPT-detectors like GPTZero (Tian and Cui, 2023), bad actors may paraphrase text generated by ChatGPT to avoid detection. To simulate this scenario we run our generated text through a T5-based paraphraser. As

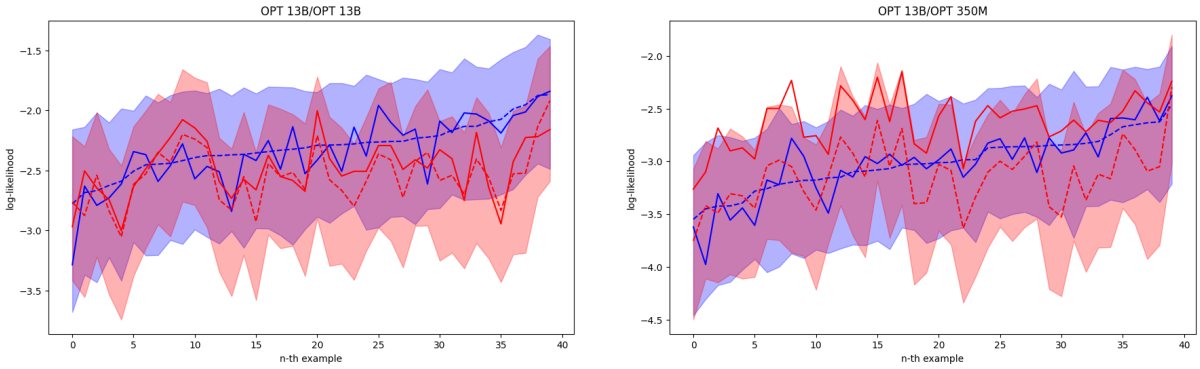

Figure 3: Log-likelihood of human and machine input texts and their sampled distribution. The solid lines represent the input sentences, with blue being the human and red being the machine-generated inputs. These inputs are then sampled 10000 by the sampling model, creating the distributions.

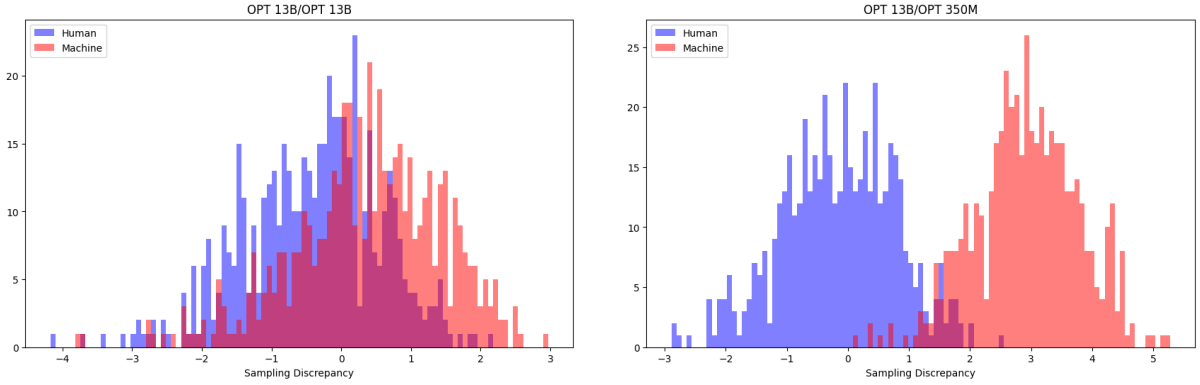

Figure 4: Sampling distribution discrepancy of human and machine texts. The largest sampling distribution discrepancy is observed between the OPT13B-OPT350M models, creating a cleaner decision boundary.

expected, all models show a decline in accuracy. This is most likely due to a drop in coherence caused by the paraphrasing process. Even here, our OPT13B-OPT125M showed a strong performance, with improvements of about 6-10%.

For the decoherence attack, we randomly transpose adjacent words in sentences longer than 20 words. This attack attempts to simulate artificial mistakes an attacker may make to bypass a detector, similar to intentional spelling errors in spam emails. OPT350M, a model $8\times$ smaller than GPT-Neo, consistently outperforms said model by 3-10%.

5 Conclusion

LLMs have gained popularity in a variety of tasks with human levels of fluency. However, there are numerous misuses of LLM, from fake texts to plagiarism, increasing the need to differentiate between human and machine texts to avoid such negative usages. We thereby propose FastDetectGPT-XS, a zero-shot detector robust to different generator models, paraphrasing attacks, and performs cost-effective inference using conditional probability curvature on smaller models.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In *Advanced information networking and applications: Proceedings of the 34th international conference on advanced information networking and applications (AINA-2020)*, pages 1341–1354. Springer.

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model-generated text: Is chatgpt that easy to detect? *ArXiv*, abs/2306.05871.

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*, abs/1906.03351.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text

via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Nassim Dehouche. 2021. Plagiarism in the age of massive generative pre-trained transformers (gpt-3). *Ethics in Science and Environmental Politics*, 21:17–23.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. Roft: A tool for evaluating human detection of machine-generated text. *arXiv preprint arXiv:2010.03070*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

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

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. 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. *ArXiv*, abs/2301.07597.

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

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023a. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *ArXiv*, abs/2304.07666.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023b. On the detectability of chatgpt content: Benchmarking, methodology, and evaluation through the lens of academic writing.

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

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

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

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

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383.

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

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, pages 841–852.