

DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature

Eric Mitchell¹ Yoonho Lee¹ Alexander Khazatsky¹ Christopher D. Manning¹ Chelsea Finn¹

Abstract

The increasing fluency and widespread usage of large language models (LLMs) highlight the desirability of corresponding tools aiding detection of LLM-generated text. In this paper, we identify a property of the structure of an LLM’s probability function that is useful for such detection. Specifically, we demonstrate that text sampled from an LLM tends to occupy negative curvature regions of the model’s log probability function. Leveraging this observation, we then define a new curvature-based criterion for judging if a passage is generated from a given LLM. This approach, which we call DetectGPT, does not require training a separate classifier, collecting a dataset of real or generated passages, or explicitly watermarking generated text. It uses only log probabilities computed by the model of interest and random perturbations of the passage from another generic pre-trained language model (e.g., T5). We find DetectGPT is more discriminative than existing zero-shot methods for model sample detection, notably improving detection of fake news articles generated by 20B parameter GPT-NeoX from 0.81 AUROC for the strongest zero-shot baseline to 0.95 AUROC for DetectGPT. See ericmitchell.ai/detectgpt for code, data, and other project information.

1. Introduction

Large language models (LLMs) have proven able to generate remarkably fluent responses to a wide variety of user queries. Models such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), and ChatGPT (OpenAI, 2022) can convincingly answer complex questions about science, mathematics, historical and current events, and social trends.

¹Stanford University. Correspondence to: Eric Mitchell <eric.mitchell@cs.stanford.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

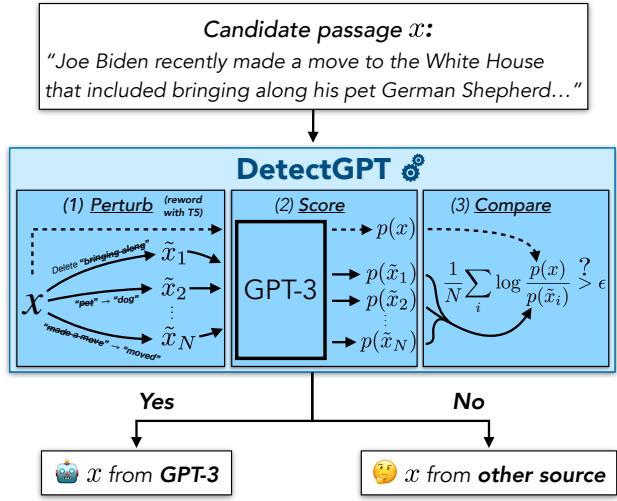


Figure 1. We aim to determine whether a piece of text was generated by a particular LLM p , such as GPT-3. To classify a candidate passage x , DetectGPT first generates minor **perturbations** of the passage \tilde{x}_i using a generic pre-trained model such as T5. Then DetectGPT **compares** the log probability under p of the original sample x with each perturbed sample \tilde{x}_i . If the average log ratio is high, the sample is likely from the source model.

While recent work has found that cogent-sounding LLM-generated responses are often simply wrong (Lin et al., 2022), the articulate nature of such generated text may still make LLMs attractive for replacing human labor in some contexts, notably student essay writing and journalism. At least one major news source has released AI-written content with limited human review, leading to substantial factual errors in some articles (Christian, 2023). Such applications of LLMs are problematic for a variety of reasons, making fair student assessment difficult, impairing student learning, and proliferating convincing-but-inaccurate news articles. Unfortunately, humans perform only slightly better than chance when classifying machine-generated vs human-written text (Gehrmann et al., 2019), leading researchers to consider automated detection methods that may identify signals difficult for humans to recognize. Such methods might give teachers and news-readers more confidence in the human origin of the text that they consume.

As in prior work (Jawahar et al., 2020), we study the

machine-generated text detection problem as a binary classification problem. Specifically, we aim to classify whether a *candidate passage* was generated by a particular *source model*. While several works have investigated methods for training a second deep network to detect machine-generated text, such an approach has several shortcomings, including a tendency to overfit to the topics it was trained on as well as the need to train a new model for each new source model that is released. We therefore consider the *zero-shot* version of machine-generated text detection, where we use the source model itself, without fine-tuning or adaptation of any kind, to detect its own samples. The most common method for zero-shot machine-generated text detection is evaluating the average per-token log probability of the generated text and thresholding (Solaiman et al., 2019; Gehrman et al., 2019; Ippolito et al., 2020). However, this zeroth-order approach to detection ignores the local structure of the learned probability function around a candidate passage, which we find contains useful information about the source of a passage.

This paper poses a simple hypothesis: minor rewrites of *model-generated* text tend to have lower log probability under the model than the original sample, while minor rewrites of *human-written* text may have higher or lower log probability than the original sample. In other words, unlike human-written text, model-generated text tends to lie in areas where the log probability function has negative curvature (for example, near local maxima of the log probability). We empirically verify this hypothesis, and find that it holds true across a diverse body of LLMs, even when the minor rewrites, or *perturbations*, come from alternative language models. We leverage this observation to build DetectGPT, a zero-shot method for automated machine-generated text detection. To test if a passage came from a source model p_θ , DetectGPT compares the log probability of the candidate passage under p_θ with the average log probability of several perturbations of the passage under p_θ (generated with, e.g., T5; Raffel et al. (2020)). If the perturbed passages tend to have lower average log probability than the original by some margin, the candidate passage is likely to have come from p_θ . See Figure 1 for an overview of the problem and DetectGPT. See Figure 2 for an illustration of the underlying hypothesis and Figure 3 for empirical evaluation of the hypothesis. Our experiments find that DetectGPT is more accurate than existing zero-shot methods for detecting machine-generated text, improving over the strongest zero-shot baseline by over 0.1 AUROC for multiple source models when detecting machine-generated news articles.

Contributions. Our main contributions are: (a) the identification and empirical validation of the hypothesis that the curvature of a model’s log probability function tends to be significantly more negative at model samples than for human text, and (b) DetectGPT, a practical algorithm inspired by this hypothesis that approximates the trace of the log

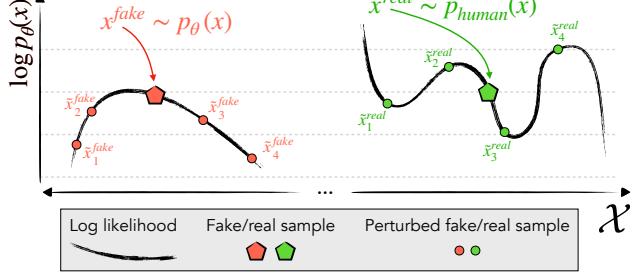


Figure 2. We identify and exploit the tendency of machine-generated passages $x \sim p_\theta(\cdot)$ (**left**) to lie in negative curvature regions of $\log p(x)$, where nearby samples have lower model log probability on average. In contrast, human-written text $x \sim p_{\text{real}}(\cdot)$ (**right**) tends not to occupy regions with clear negative log probability curvature; nearby samples may have higher or lower log probability.

probability function’s Hessian to detect a model’s samples.

2. Related Work

Increasingly large LLMs (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2022; Zhang et al., 2022) have led to dramatically improved performance on many language-related benchmarks and the ability to generate convincing and on-topic text. GROVER (Zellers et al., 2019) was the first LLM trained specifically for generating plausible news articles. Human evaluators found GROVER-generated propaganda at least as trustworthy as human-written propaganda, motivating the authors to study GROVER’s ability to detect its own generations by fine-tuning a detector on top of its features; they found GROVER better able to detect GROVER-generated text than other pre-trained models. However, Bakhtin et al. (2019); Uchendu et al. (2020) note that models trained explicitly to detect machine-generated text tend to overfit to their training distribution of data or source models.

Other works have trained supervised models for machine-generated text detection on top of neural representations (Bakhtin et al., 2019; Solaiman et al., 2019; Uchendu et al., 2020; Ippolito et al., 2020; Fagni et al., 2021), bag-of-words features (Solaiman et al., 2019; Fagni et al., 2021), and hand-crafted statistical features (Gehrman et al., 2019). Alternatively, Solaiman et al. (2019) notes the surprising efficacy of a simple zero-shot method for machine-generated text detection, which thresholds a candidate passage based on its average log probability under the generative model, serving as a strong baseline for zero-shot machine-generated text detection in our work. In our work, we similarly use the generating model to detect its own generations in a zero shot manner, but through a different approach based on estimating local curvature of the log probability around the sample rather than the raw log probability of the sample itself. See Jawahar et al. (2020) for a complete survey on machine-

generated text detection. Other work explores watermarks for generated text (Kirchenbauer et al., 2023), which modify a model’s generations to make them easier to detect. Our work does not assume text is generated with the goal of easy detection; DetectGPT detects text generated from publicly available LLMs using standard LLM sampling strategies.

The widespread use of LLMs has led to much other contemporaneous work on detecting LLM output. Sadasivan et al. (2023) show that the detection AUROC of the an detector is upper bounded by a function of the TV distance between the model and human text. However, we find that AUROC of DetectGPT is high even for the largest publicly-available models (Table 2), suggesting that TV distance may not correlate strongly with model scale and capability. This disconnect may be exacerbated by new training objectives other than maximum likelihood, e.g., reinforcement learning with human feedback (Christiano et al., 2017; Ziegler et al., 2020). Both Sadasivan et al. (2023) and Krishna et al. (2023) show the effectiveness of paraphrasing as a tool for evading detection, suggesting an important area of study for future work. Liang et al. (2023) show that multi-lingual detection is difficult, with non-DetectGPT detectors showing bias against non-native speakers; this result highlights the advantage of zero-shot detectors like DetectGPT, which generalize well to any data generated by the original generating model. Mireshghallah et al. (2023) study which proxy scoring models produce the most useful log probabilities for detection when the generating model is not known (a large-scale version of our Figure 6). Surprisingly (but consistent with our findings), they find that smaller models are in fact *better* proxy models for performing detection with perturbation-based methods like DetectGPT.

The problem of machine-generated text detection echoes earlier work on detecting deepfakes, artificial images or videos generated by deep nets, which has spawned substantial efforts in detection of fake visual content (Dolhansky et al., 2020; Zi et al., 2020). While early works in deepfake detection used relatively general-purpose model architectures (Güera & Delp, 2018), many deepfake detection methods rely on the continuous nature of image data to achieve state-of-the-art performance (Zhao et al., 2021; Guarnera et al., 2020), making direct application to text difficult.

3. The Zero-Shot Machine-Generated Text Detection Problem

We study zero-shot machine-generated text detection, the problem of detecting whether a piece of text, or *candidate passage* x , is a sample from a *source model* p_θ . The problem is zero-shot in the sense that we do not assume access to human-written or generated samples to perform detection. As in prior work, we study a ‘white box’ setting (Gehrman et al., 2019) in which the detector may evaluate the log prob-

Algorithm 1 DetectGPT model-generated text detection

```

1: Input: passage  $x$ , source model  $p_\theta$ , perturbation function  $q$ ,  
number of perturbations  $k$ , decision threshold  $\epsilon$   
2:  $\tilde{x}_i \sim q(\cdot | x)$ ,  $i \in [1..k]$  // mask spans, sample replacements  
3:  $\tilde{\mu} \leftarrow \frac{1}{k} \sum_i \log p_\theta(\tilde{x}_i)$  // approximate expectation in Eq. 1  
4:  $\hat{\mathbf{d}}_x \leftarrow \log p_\theta(x) - \tilde{\mu}$  // estimate  $\mathbf{d}(x, p_\theta, q)$   
5:  $\tilde{\sigma}_x^2 \leftarrow \frac{1}{k-1} \sum_i (\log p_\theta(\tilde{x}_i) - \tilde{\mu})^2$  // variance for normalization  
6: if  $\frac{\hat{\mathbf{d}}_x}{\sqrt{\tilde{\sigma}_x^2}} > \epsilon$  then  
7:   return true // probably model sample  
8: else  
9:   return false // probably not model sample

```

ability of a sample $\log p_\theta(x)$. The white box setting **does not** assume access to the model architecture or parameters. Most public APIs for LLMs (such as GPT-3) enable scoring text, though some exceptions exist, notably ChatGPT. While most of our experiments consider the white box setting, see Section 5.2 for experiments in which we score text using models other than the source model. See Mireshghallah et al. (2023) for a comprehensive evaluation in this setting.

The detection criterion we propose, DetectGPT, also makes use of generic pre-trained mask-filling models in order to generate passages that are ‘nearby’ the candidate passage. However, these mask-filling models are used off-the-shelf, without any fine-tuning or adaptation to the target domain.

4. DetectGPT: Zero-shot Machine-Generated Text Detection with Random Perturbations

DetectGPT is based on the hypothesis that samples from a source model p_θ typically lie in areas of negative curvature of the log probability function of p_θ , unlike human text. In other words, if we apply small perturbations to a passage $x \sim p_\theta$, producing \tilde{x} , the quantity $\log p_\theta(x) - \log p_\theta(\tilde{x})$ should be relatively large on average for machine-generated samples compared to human-written text. To leverage this hypothesis, first consider a perturbation function $q(\cdot | x)$ that gives a distribution over \tilde{x} , slightly modified versions of x with similar meaning (we will generally consider roughly paragraph-length texts x). As an example, $q(\cdot | x)$ might be the result of simply asking a human to rewrite one of the sentences of x , while preserving the meaning of x . Using the notion of a perturbation function, we can define the *perturbation discrepancy* $\mathbf{d}(x, p_\theta, q)$:

$$\mathbf{d}(x, p_\theta, q) \triangleq \log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot | x)} \log p_\theta(\tilde{x}) \quad (1)$$

We state our hypothesis more formally as the Local Perturbation Discrepancy Gap Hypothesis, which describes a gap in the perturbation discrepancy for model-generated text and human-generated text.

Perturbation Discrepancy Gap Hypothesis. *If q produces samples on the data manifold, $\mathbf{d}(x, p_\theta, q)$ is positive and large with high probability for samples $x \sim p_\theta$. For human-written text, $\mathbf{d}(x, p_\theta, q)$ tends toward zero for all x .*