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

Eric Mitchell [1] **Yoonho Lee** [1] **Alexander Khazatsky** [1] **Christopher D. Manning** [1] **Chelsea Finn** [1]

Stanford University

ICML'23

Problem

- Large language models are convincing but unreliable
  - Half of model-generated sentences are **not fully supported citations.**
  - One quarter of citations **do not support** the associated model-generated claim.

- We're still tempted to use them anyway!

FUTURISM | JAN 19 by JON CHRISTIAN

CNET Secretly Used AI on Articles That Didn't Disclose That Fact, Staff Say

"They use AI to rewrite the intros every two weeks or so because Google likes updated content. Eventually it gets so mangled that about every four months a real editor has to look at it and rewrite it."

Lawyers blame ChatGPT for tricking them into citing bogus case law

AP

BY LARRY NEUMEISTER
Published 8:25 PM PDT, June 8, 2023

Share

NEW YORK (AP) — Two apologetic lawyers responding to an angry judge in Manhattan federal court blamed ChatGPT Thursday for tricking them into including fictitious legal research in a court filing.

Attorneys Steven A. Schwartz and Peter LoDuca are facing possible punishment over a filing in a lawsuit against an airline that included references to past court cases that Schwartz thought were real, but were actually invented by the artificial intelligence-powered chatbot.

Schwartz explained that he used the groundbreaking program as he hunted for legal precedents supporting a client's case against the Colombian airline Avianca for an injury incurred on a 2019 flight.

https://apnews.com/article/artificial-intelligence-chatgpt-courts-e15023d7e6fdf4f099aa122437dbb59b

Motivation

It would be helpful to know when we're reading LM-generated text.

But how?

Detecting LM-generated text

Initial ideas

Option 1: Train a second LM specifically for detection

1. Gather lots of data from human sources and the model(s) of interest
2. Train a binary classifier to distinguish between human/LM text
3. Hope it generalizes well

\+ Powerful, expressive model

\- Inconvenient (data collection, training)

\- Can overfit to domain, model, language, etc.

Detecting LM-generated text

Initial ideas

Option 1: Train a second LM specifically for detection

Option 2: Use the source LM itself to detect its generations “zero-shot”

1. Given a candidate passage, compute the log probability of each token
2. If avg. log probability is high or avg. rank of observed tokens is low, we probably have a model sample

\+ No training or data collection!

\- Not so accurate in practice

Detecting LM-generated text

An alternative strategy

Can we improve **zero-shot** detectors, retaining their **convenience**?

Idea: leverage the structure of the model's log probability function **around** the candidate passage

Hypothesis:

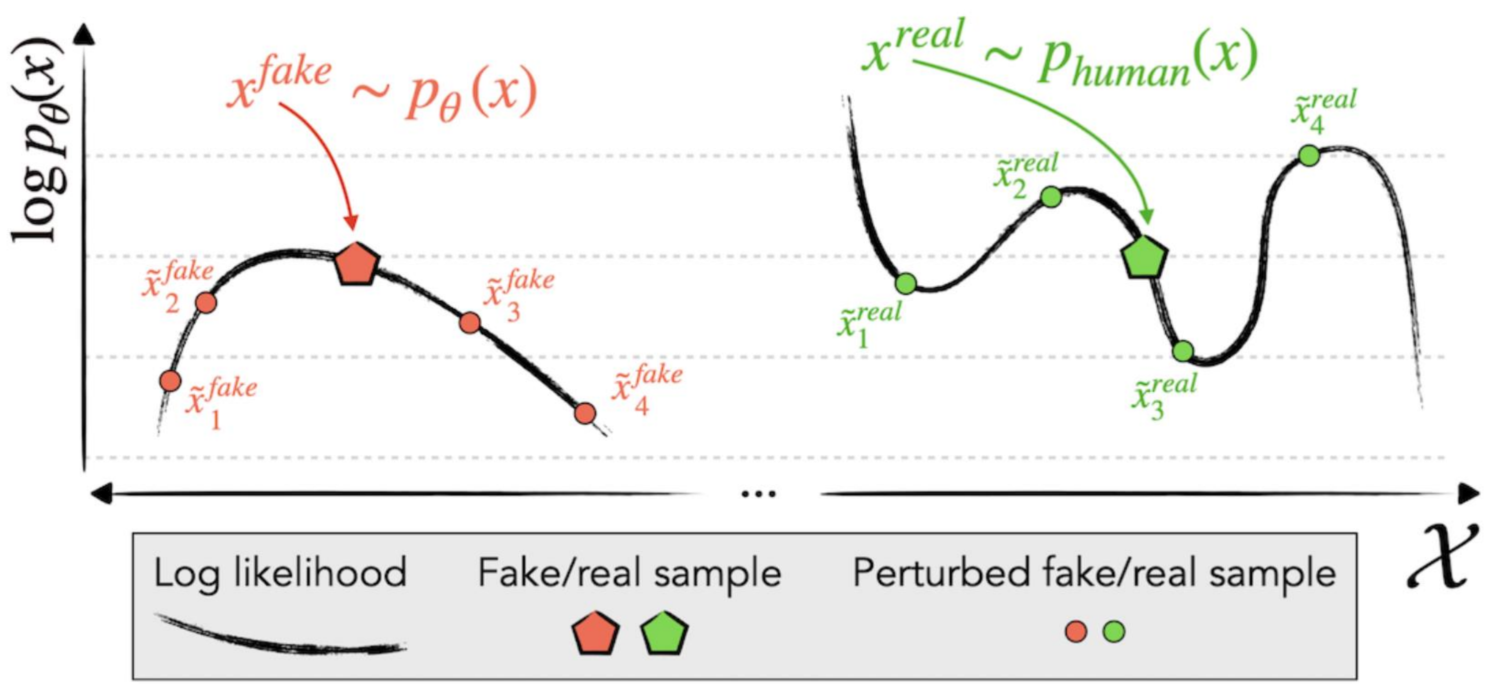
Model samples lie near **local maxima** of the model's log probability function

"If we slightly rephrase model-generated text, the log probability tends to drop"

DetectGPT

The Perturbation Discrepancy Gap Hypothesis

"The perturbation discrepancy is larger for model samples than for human text"



DetectGPT

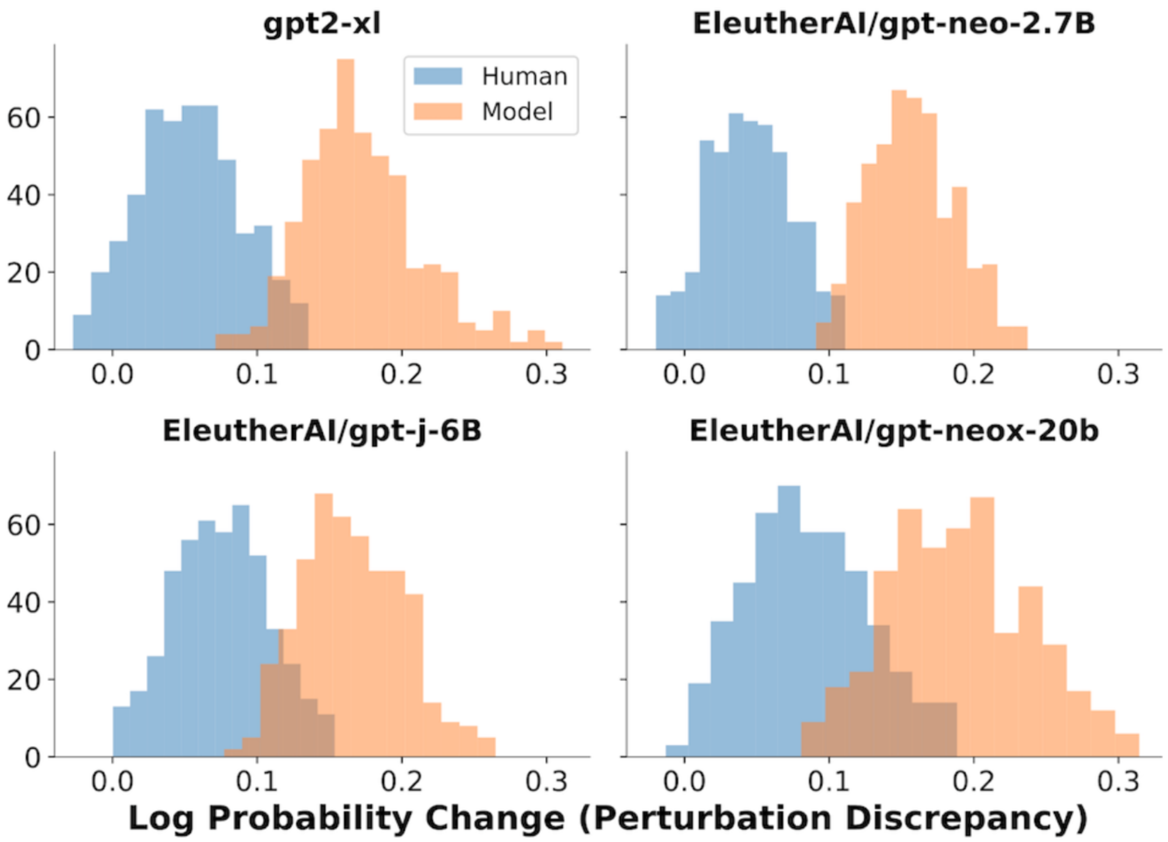
The perturbation discrepancy

"How much does the logprob of a sample x drop when I **perturb** (rephrase) it, on average over many **perturbations**?"

$$\mathbf{d}\left(x, p_\theta, q\right) \triangleq \underbrace{\log p_\theta(x)}_{\text{logprob of } x} - \underbrace{\mathbb{E}_{\overbrace{\tilde{x} \sim q(\cdot|x)}^{\text{perturbation of } x}} \log p_\theta(\tilde{x})}_{\text{avg logprob of perturbations to } x}$$

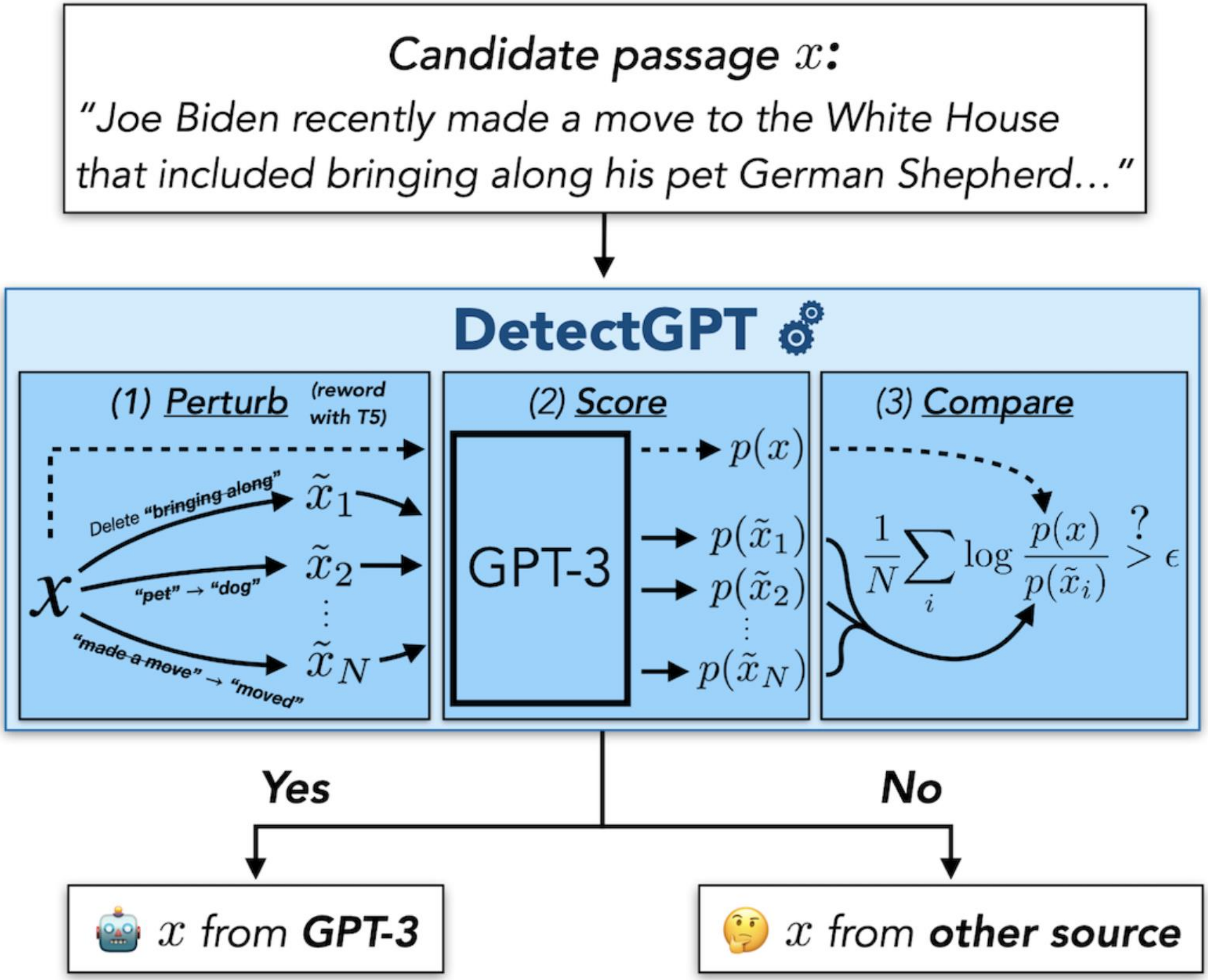
Testing the hypothesis

Computing the perturbation discrepancy for many human-written and model-generated texts:



Perturbations are generated by randomly masking 2-word spans and sampling replacement with T5-3B

DetectGPT algorithm overivew



Experiments

Method	XSum GPT-2	XSum OPT-2.7	XSum Neo-2.7	XSum GPT-J	XSum NeoX	XSum **Avg.**	SQuAD GPT-2	SQuAD OPT-2.7	SQuAD Neo-2.7	SQuAD GPT-J	SQuAD NeoX	SQuAD **Avg.**	WritingPrompts GPT-2	WritingPrompts OPT-2.7	WritingPrompts Neo-2.7	WritingPrompts GPT-J	WritingPrompts NeoX	WritingPrompts **Avg.**
$\log p(x)$	0.86	0.86	0.86	0.82	0.77	0.83	0.91	0.88	0.84	0.78	0.71	0.82	0.97	0.95	0.95	0.94	0.93*	0.95
Rank	0.79	0.76	0.77	0.75	0.73	0.76	0.83	0.82	0.80	0.79	0.74	0.80	0.87	0.83	0.82	0.83	0.81	0.83
LogRank	0.89*	0.88*	0.90*	0.86*	0.81*	0.87*	0.94*	0.92*	0.90*	0.83*	0.76*	0.87*	0.98*	0.96*	0.97*	0.96*	**0.95**	0.96*
Entropy	0.60	0.50	0.58	0.58	0.61	0.57	0.58	0.53	0.58	0.58	0.59	0.57	0.37	0.42	0.34	0.36	0.39	0.38
DetectGPT	**0.99**	**0.97**	**0.99**	**0.97**	**0.95**	**0.97**	**0.99**	**0.97**	**0.97**	**0.90**	**0.79**	**0.92**	**0.99**	**0.99**	**0.99**	**0.97**	0.93*	**0.97**
Diff	0.10	0.09	0.09	0.11	0.14	0.10	0.05	0.05	0.07	0.07	0.03	0.05	0.01	0.03	0.02	0.01	-0.02	0.01

Table 1. AUROC for detecting samples from the given model on the given dataset for DetectGPT and four previously proposed criteria (500 samples used for evaluation). From 1.5B parameter GPT-2 to 20B parameter GPT-NeoX, DetectGPT consistently provides the most accurate detections. **Bold** shows the best AUROC within each column (model-dataset combination); asterisk (*) denotes the second-best AUROC. Values in the final row show DetectGPT's AUROC over the strongest baseline method in that column.

Evaluate various zero-shot detectors on **news**, **wikipedia-style articles**, and **creative writing**

DetectGPT is consistently most discriminative

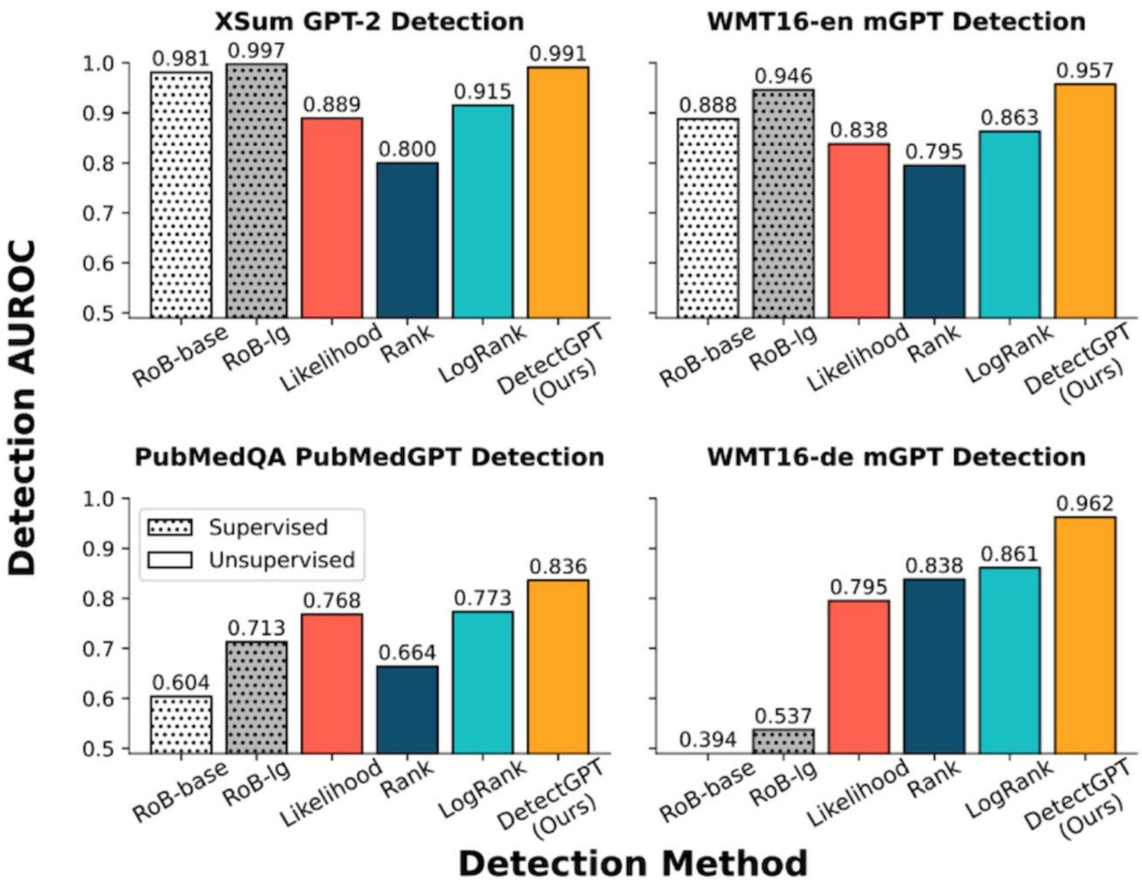
Results averaged across 6 models from **1.5B to 20B**

Method	XSum top-p	XSum top-k	SQuAD top-p	SQuAD top-k	WritingPrompts top-p	WritingPrompts top-k
$\log p(x)$	0.92	0.87	0.89	0.85	**0.98**	0.96
Rank	0.76	0.76	0.81	0.80	0.84	0.83
LogRank	0.93*	0.90*	0.92*	0.90*	**0.98**	**0.97**
Entropy	0.53	0.55	0.54	0.56	0.32	0.35
DetectGPT	**0.98**	**0.98**	**0.94**	**0.93**	**0.98**	**0.97**

Table 3. AUROC for zero-shot methods averaged across the five models in Table 1 for both top-k and top-p sampling, with $k = 40$ and $p = 0.96$. Both settings enable slightly more accurate detection, and DetectGPT consistently provides the best detection performance. See Appendix Tables 4 and 5 for complete results.

Experiments

DetectGPT generalizes to diverse text distributions

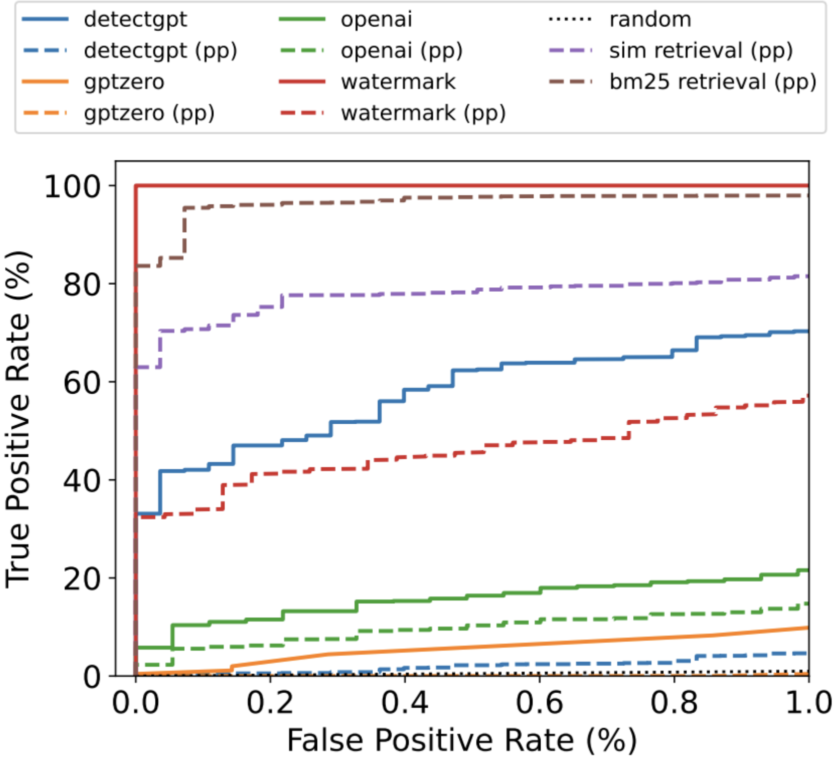


For **news articles in English**, DetectGPT is as good or better than existing detectors

For **biomedical** texts or news articles in **German**, DetectGPT outperforms by a larger margin

Related works

Metric →	Sim ↑	Detection Accuracy ↓				
Detector →		Watermarks	DetectGPT	OpenAI	GPTZero	RankGen
GPT2-1.5B	-	100.0	70.3	21.6	13.9	**13.5**
+ DIPPER 20L	99.2	97.1	28.7	19.2	9.1	15.8
+ DIPPER 40L	98.4	85.8	15.4	17.8	7.3	18.0
+ DIPPER 60L	96.9	68.9	8.7	**13.3**	7.1	19.8
+ DIPPER 60L, 60O	94.3	**57.2**	**4.6**	14.8	**1.2**	28.5
OPT-13B	-	99.9	14.3	11.3	8.7	**3.2**
+ DIPPER 20L	99.1	96.2	3.3	11.8	5.4	5.2
+ DIPPER 40L	98.6	84.8	1.2	11.6	3.8	6.6
+ DIPPER 60L	97.1	63.7	0.8	**9.1**	6.3	9.3
+ DIPPER 60L, 60O	94.6	**52.8**	**0.3**	10.0	**1.0**	13.5
GPT-3.5-175B, davinci-003	-	-	26.5*	30.0	7.1	**1.2**
+ DIPPER 20L	97.6	-	12.5*	20.6	4.3	1.7
+ DIPPER 40L	96.7	-	8.0*	22.4	4.8	2.0
+ DIPPER 60L	94.2	-	7.0*	**15.6**	6.1	3.9
+ DIPPER 60L, 60O	88.4	-	**4.5***	**15.6**	**1.8**	7.3
Human Text	-	1.0	1.0	1.0	1.0	1.0



How AI Detection at GPTZero works

GPTZero's technology uses deep learning to keep pace with AI advancements to deliver precise, reliable results that help you understand and interpret the origin of a piece of text.

Input Text
GPTZero accepts copy and pasted text, docx, pdf, and image files, analyzing up to 50 files at a time.

Deep Learning
We employ an end-to-end deep learning approach, trained on text datasets from the web, education, and AI- generated from a range of LLMs.

Sentence Classifier
A sentence-by-sentence classification model determines the probability and confidence that a text was created by AI.

Paraphraser Shield
We defend against tools looking to exploit AI detectors. Our model shields against common methods to bypass AI detection, such as paraphrasing and homoglyph attacks.

Output Result
You can view easy-to-interpret results in our dashboard, with premium features to detect AI vocabulary, plagiarism, and citeable sources.

Thanks