DetectGPT

Zero-Shot Machine-Generated Text Detection using Probability Curvature

Hamed Mahdavi

April 26, 2023

Motivation: Misuses of LLMs

- Students use LLMs to write their homework. This impairs student learning and makes a fair assessment of their homework hard.
- LLMs make research plagiarism easier. Paraphrased plagiarized articles are hard to detect using older methods.
- LLMs can ease propaganda: assume a scenario when a large group of online bot social media accounts starts to generate incorrect information about a topic.
- With these possibilities, one needs to ask, is it possible to distinguish outputs of LLMs with human written text?

Problem Definition

Problem Definition: Generated Text Detection

Given a candidate passage and a source model, we want to detect if the candidate passage is generated by a source model with a high probability.

Possible Approaches

- Training another network for binary classification of generated text:
  - We need to train a new classification model for each source model.
  - The classifier might overfit to the topics it was trained on.
- Zero-shot Approach (Solaiman et al. 2019): One can look at model outputs such as the average log-probability of generated results to decide for each sample.
  - Does not need to train other models or gather a dataset.
  - But ignores the local structure of the learned probability function around a candidate passage.

Proposed Solution

- **Facts:**
  - We know that sampling from Language models follows a local search procedure (e.g. beam search) to find high-probability samples
  - Solaiman et al. (2019) method works well in practice.
- **Hypothesis:**
  - LLMs generate samples near local maxima of log-probability. Perturbing a sample significantly reduces its log probability.

Formal Statement of Hypothesis

- We denote a perturbation function with $q(\cdot|x)$. q can be the result of asking a human to rewrite one of the sentences of x while preserving the meaning of x.
- the notion of a perturbation function, we can define the perturbation discrepancy $\mathbf{d}\,(x, p_\theta, q)$:

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

Hypothesis

If q produces samples on the data manifold, $\mathbf{d}\,(x, p_\theta, q)$ is positive 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.

Proposed Solution: Visualization

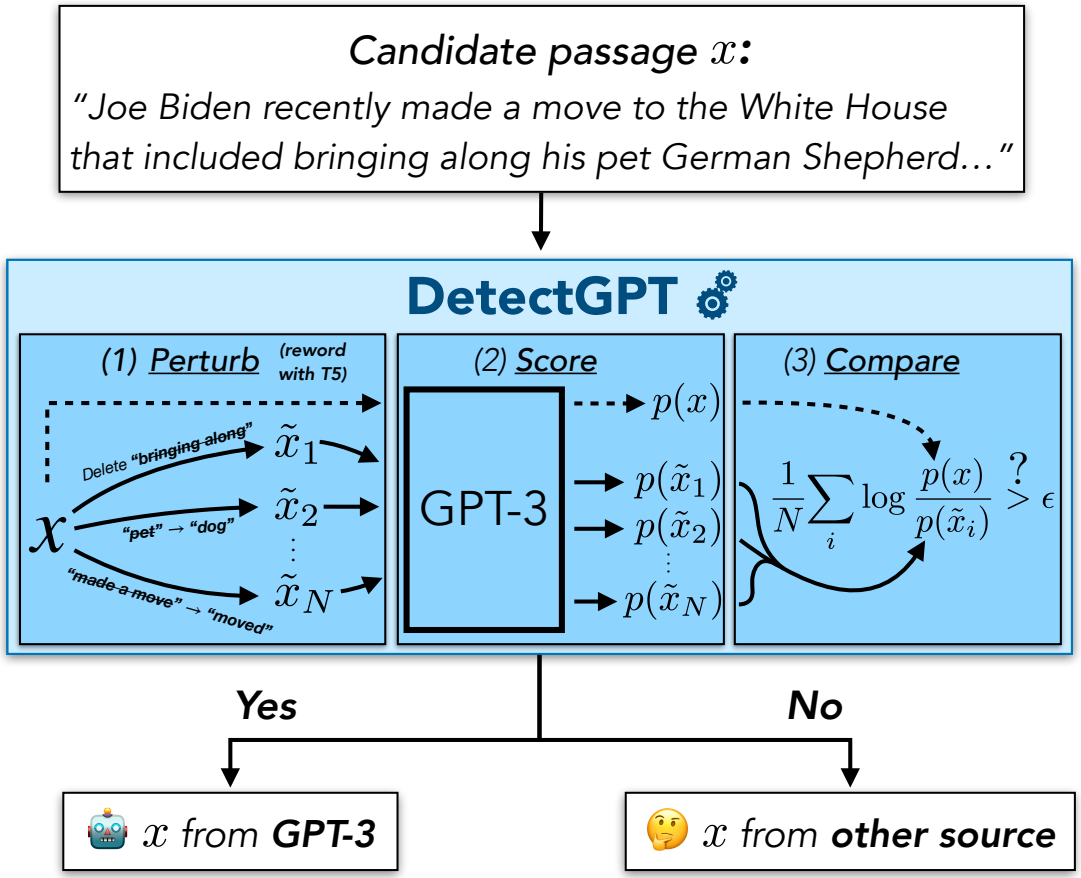

Figure: DetectGPT compares the log probability under p of the original sample x with each perturbed sample $\tilde{x}_i$. If the average log ratio is high, the sample is likely from the source model.

Hypothesis: Visualization

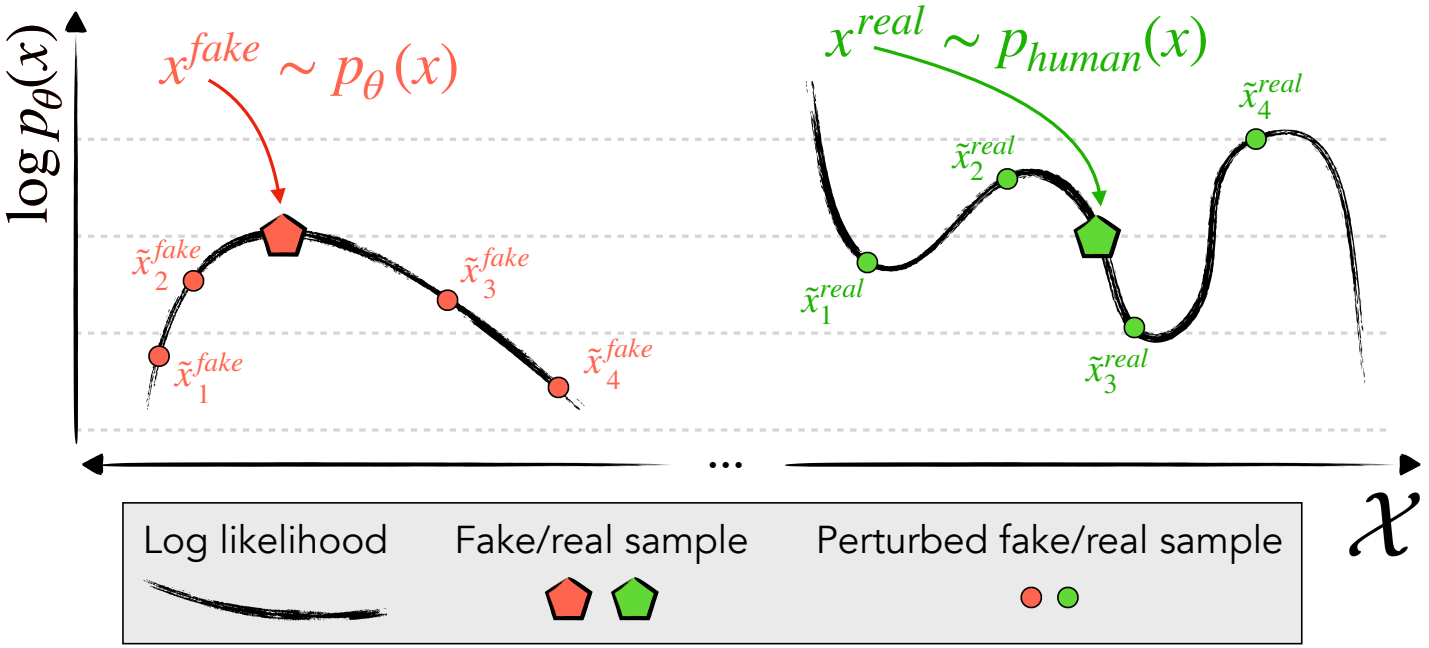

Figure: Human written text is not a result of greedy local search optimization, so if we perturb it, the log-probability of the perturbed text does not drop.

Testing Hypothesis: Using other LLMs

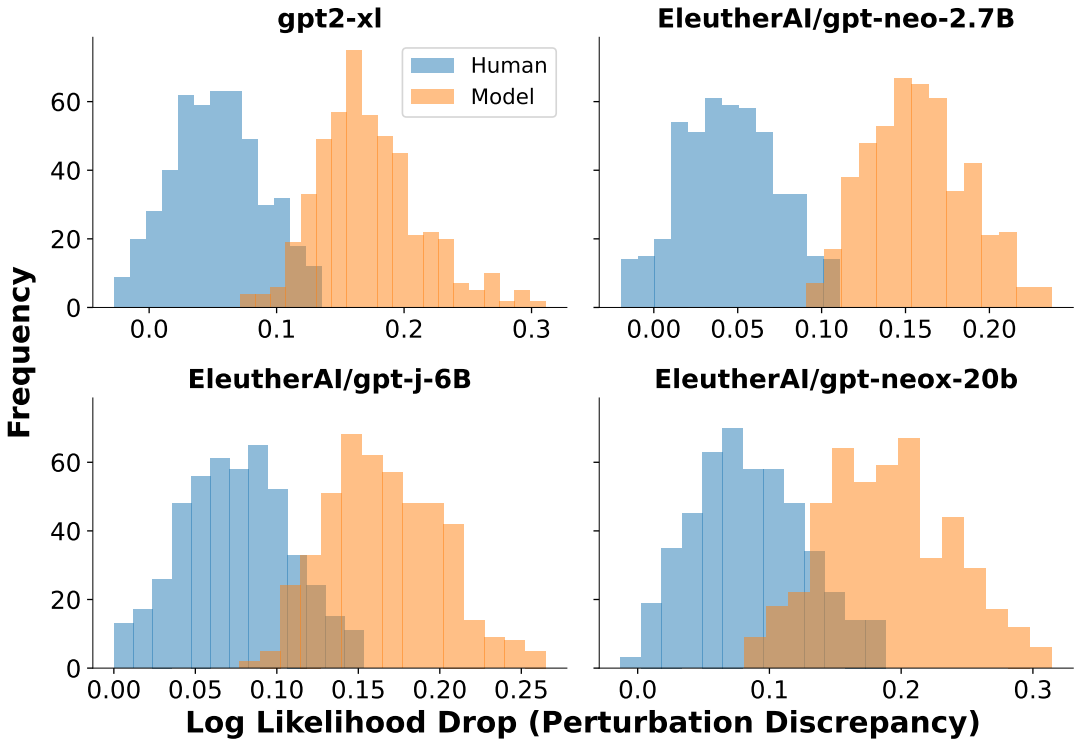

Figure: One can test the stated hypothesis by modeling q with a masked language model with T5. Here 500 news articles from XSum dataset have been used for real data.4 LLMs have been used to sample models from XSum's articles based on their first 30 tokens

Interpretation of the Perturbation Discrepancy as Curvature

- The perturbation discrepancy approximates a measure of the local curvature of the log probability function near the candidate passage.
- If z_is are i.i.d with mean zero and variance 1, then the following is a random estimation of A matrix's trace

$$\mathrm{tr}(A) = \mathbb{E}_{\mathbf{z}} \mathbf{z}^\top A \mathbf{z}$$

- We can approximate second-order directional derivative with:

$$\mathbf{z}^\top H_f(x)\mathbf{z} \approx \frac{f(x + h\mathbf{z}) + f(x - h\mathbf{z}) - 2f(x)}{h^2}$$

Interpretation of the Perturbation Discrepancy as Curvature (cont.)

- Combining two equations from the last slide we get:

$$-\operatorname{tr}(H)_f(x) \approx 2f(x) - \mathbb{E}_{\mathbf{z}}[f(x+\mathbf{z}) + f(x-\mathbf{z})]$$

- If the noise distribution is symmetric, then we have:

$$\frac{-\operatorname{tr}(H)_f(x)}{2} \approx f(x) - \mathbb{E}_{\mathbf{z}} f(x+\mathbf{z})$$

Results: AUROC for Detecting Samples

	XSum						SQuAD						WritingPrompts					
Method	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	**Avg.**	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	**Avg.**	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	**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

Figure: DetectGPT's AUROC consistently outperforms four previously proposed criteria across various models and datasets, with 500 samples used for evaluation. Bold indicates the best AUROC, and an asterisk (*) denotes the second-best. The final row shows DetectGPT's AUROC compared to the strongest baseline method in each column, ranging from 1.5B to 20B parameters in GPT models.

Results: Supervised Machine Generated Text Detection

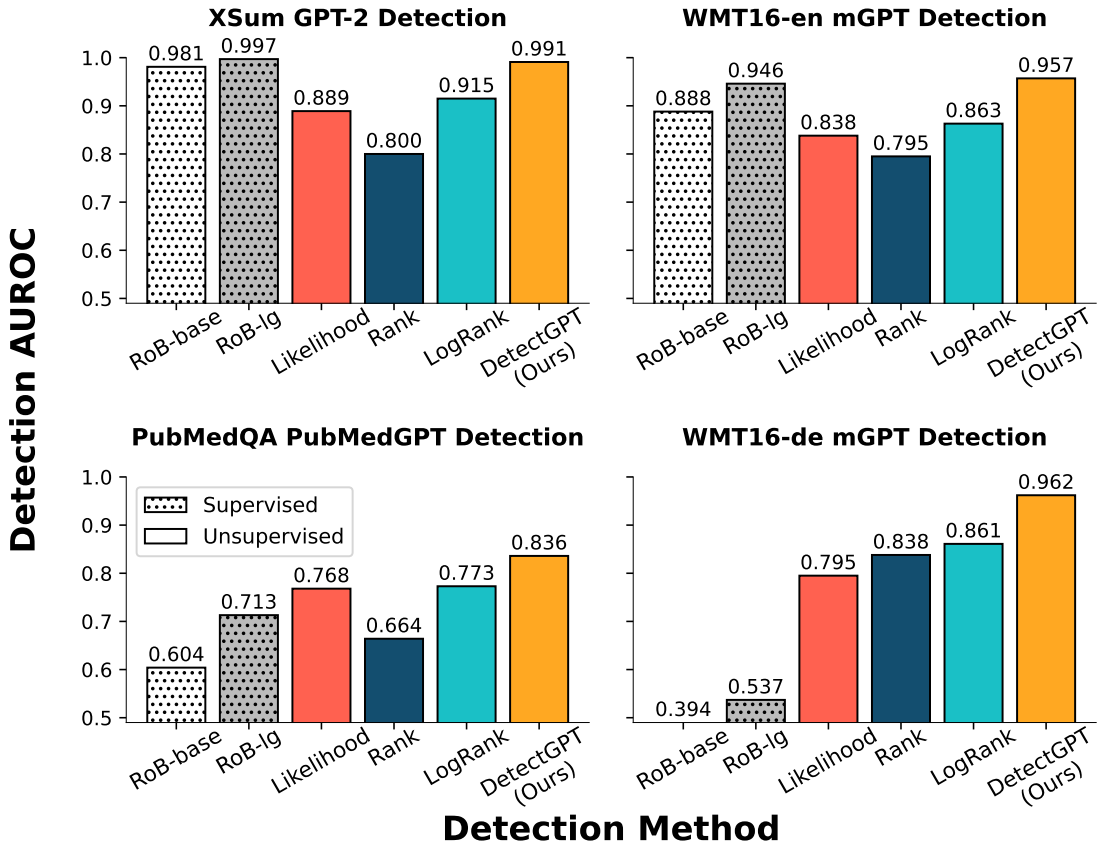

Figure: Supervised models work well on in-distribution text, DetectGPT performs better in out-of-distribution data.

Results: DetectGPT vs Supervised

	PubMedQA	XSum	WritingP	Avg.
RoBERTa-base	0.64	**0.92**	**0.92**	0.83
RoBERTa-large	0.71	**0.92**	0.91	**0.85**
$\log p(x)$	0.64	0.76	0.88	0.76
DetectGPT	**0.84**	0.84	0.87	**0.85**

Figure: DetectGPT and supervised models for detecting machine-generated text have similar AUROC for GPT-3 generations. Supervised models outperform DetectGPT in detecting typical text like news articles.

Results: Capacity of Mask Filling and Detection Quality

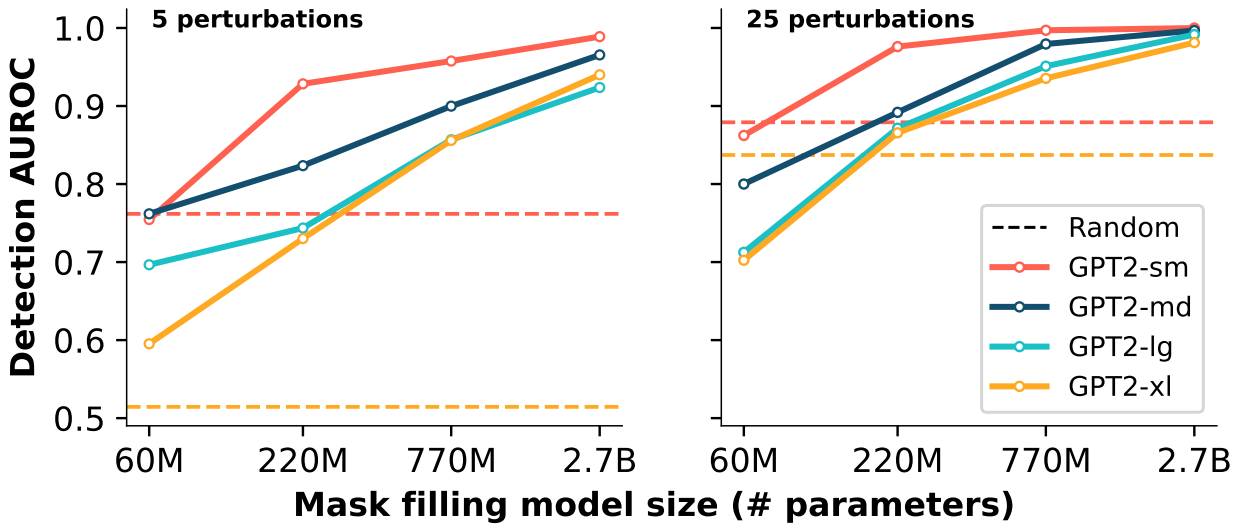

Figure: The AUROC scores on 200 SQuAD contexts are presented as curves. The mask-filling model's capacity is associated with detection performance across different source model scales. Random mask filling with uniform sampling performs badly, indicating that the perturbation function must generate samples on the data manifold.

Results: The effect of the number of perturbations

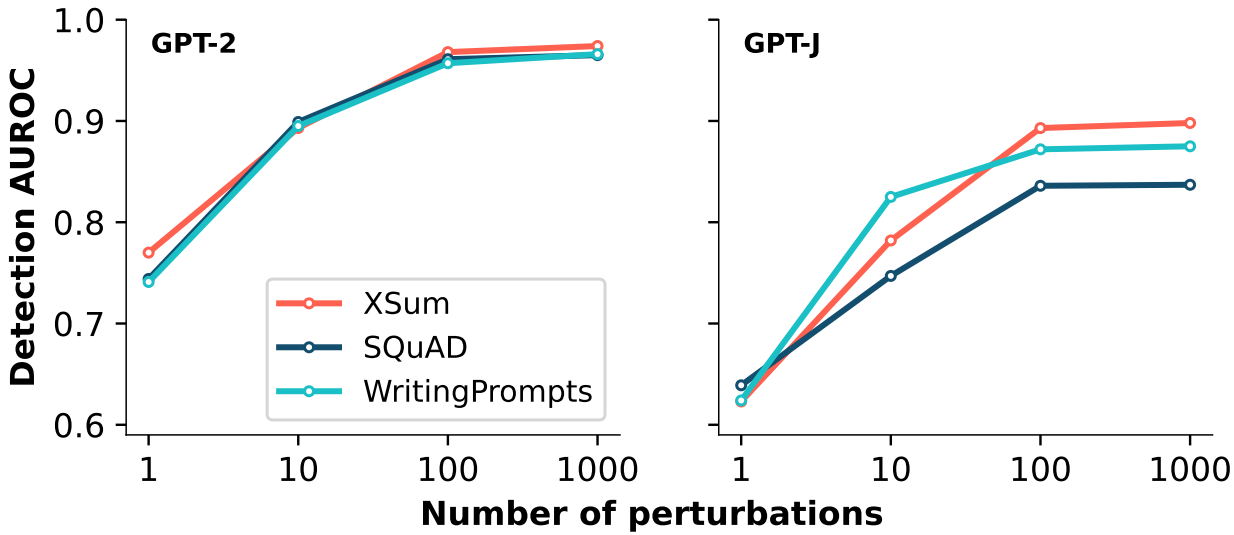

Figure: DetectGPT's discrimination power for classification, as measured by auROC (left) and auPR (right), is significantly improved by averaging up to 100 perturbations. The fills are sampled from T5-large, and the impact of varying the number of perturbations is evaluated.

Results: Scoring Quality

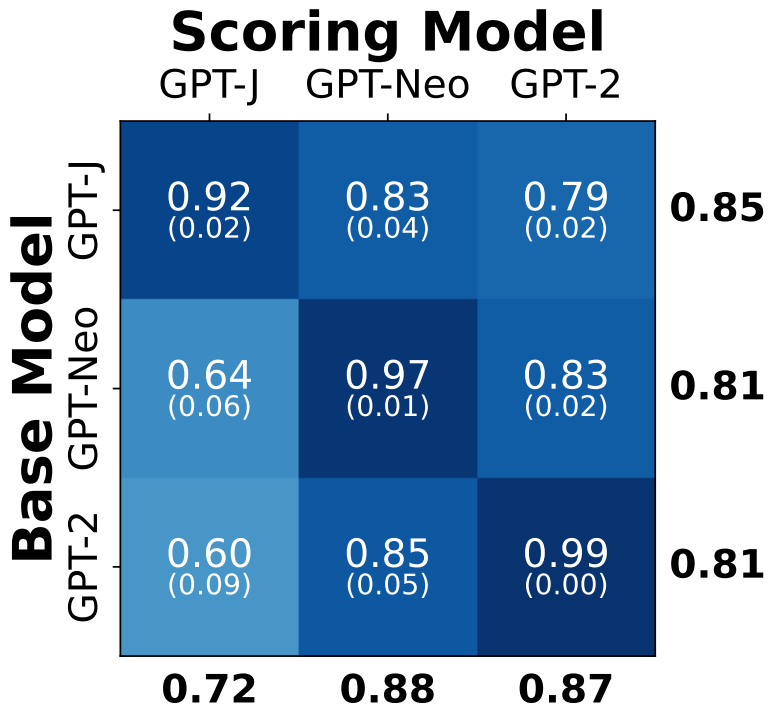

Figure: DetectGPT works well in scoring samples generated by the same model (diagonal), but column means suggest that some models (such as GPT-Neo and GPT-2) may be better scorers than others (such as GPT-J). Mean (standard error) AUROC over XSum, SQuAD, and WritingPrompts is represented by white values, while row/column mean is shown by black value

Results: Varying Edit Fractions

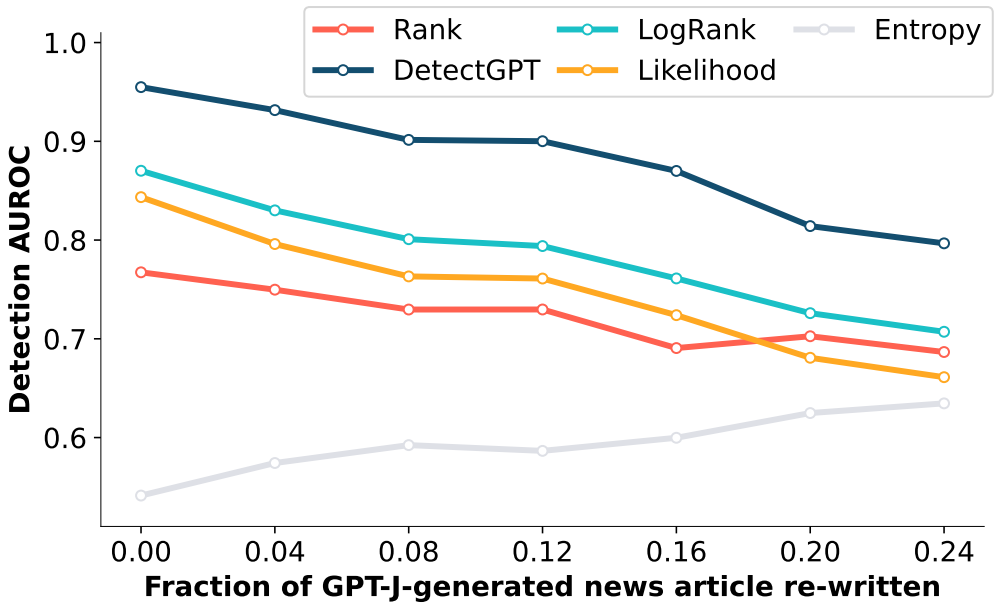

Figure: The performance of four top-performing methods degrades as revisions become heavier when simulating human edits to machine-generated text by replacing different fractions of model samples with T5-3B generated text. DetectGPT consistently performs the best in this experiment conducted on the XSum dataset. The simulation is achieved by masking out random five-word spans until r% of text is masked.

Discussion

- Accessing to probability outputs of a language model can be expensive (e.g. GPT3 API).
- The paper does not discuss the social impacts of their method. Does this work well equally for native and non-native English speakers?
- What is the effect of adding a prompt before generating text?

Questions

Any Questions?