Baseline Analysis of DetectGPT, a Zero-Shot Machine-Generated Text Detection using Probability Curvature

Lauren Okamoto
Princeton University
lokamoto@princeton.edu

Tyler Benson
Princeton University
tb19@princeton.edu

AJ Askergali
Princeton University
aigerima@princeton.edu

Abstract

DetectGPT is a zero-shot machine-generated text detection model by Mitchell et al. (Mitchell et al., 2023). In our paper, we first reproduce the results of Mitchell et al. on the XSum dataset. Then, we experiment changing two hyperparameters of the model: the masked percentage and the temperature of the mask filling model. Our code can be found at https://github.com/laurenok24/cos484-detect-gpt. We found that increasing temperature drastically in mask filling reduced the performance of DetectGPT to 50%, while decreasing temperature did not affect performance much. Additionally, 15% masking percentage achieved the best performance, with ablations from that value only reducing perfomance marginally.

1 Introduction

Large language models (LLMs) in natural language processing have become increasingly smarter at responding to user queries accurately and fluently. Some of these models, such as GPT-3 and ChatGPT, can answer complex questions about an incredibly wide range of topics. The responses are so well crafted that they can be indistinguishable from human responses.

In particular, ChatGPT was released to the public in November 2022, and its usage has exploded such that there were over 100 million users worldwide by January 2023. With such widespread use, concerns have arisen as to whether certain written work is human or machine generated. For example, whether an essay is student-written or generated by AI.

When classifying human-written text from machine-generated text, humans only do slightly better than random chance. This is where automated detection methods come into play, since they may be able to pick up on signals difficult for humans to see.

In their study, "DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature," Mitchell et al. define a new curvature-based criterion for determining whether a passage is generated from a given large language model or not (Mitchell et al., 2023). They demonstrated that texts generated from a language model usually occupy negative curvature regions of the model's log probability distribution (local maxima). Using this observation, they generated random perturbations of the text and compared the log probability of the candidate text with the average of the log probabilities of the perturbations. They found that if the perturbations had log probabilities that were exclusively lower than that of the candidate text, the candidate text was likely machine-generated.

In this paper, we reproduced the baselines in Mitchell et al.'s study, and explored specifically the impact of the choice of the mask percentage and temperature hyperparameters on the results.

2 Background and Related Work

One of the most important parts of DetectGPT is how the perturbed texts are being generated. First, a certain percentage of the original text is masked. Then, a mask-filling model fills in the masked words to create the perturbed text. Higher mask percentages lead to greater changes (i.e. larger perturbations) of the text, which should result in the perturbed text being further away from the original text on the log likelihood curve. See Figure 1.

2.1 Masked Percentage

For DetectGPT, Mitchell et al. masked 15 percent of words in each text for the purpose of generating the perturbed text. Although used for a different purpose, the 15 percent masking rate has been widely used for training masked learning models (MLMs), and has been the default rate for training MLMs regardless of other factors. However, this masking rate may not be universally 15 per-

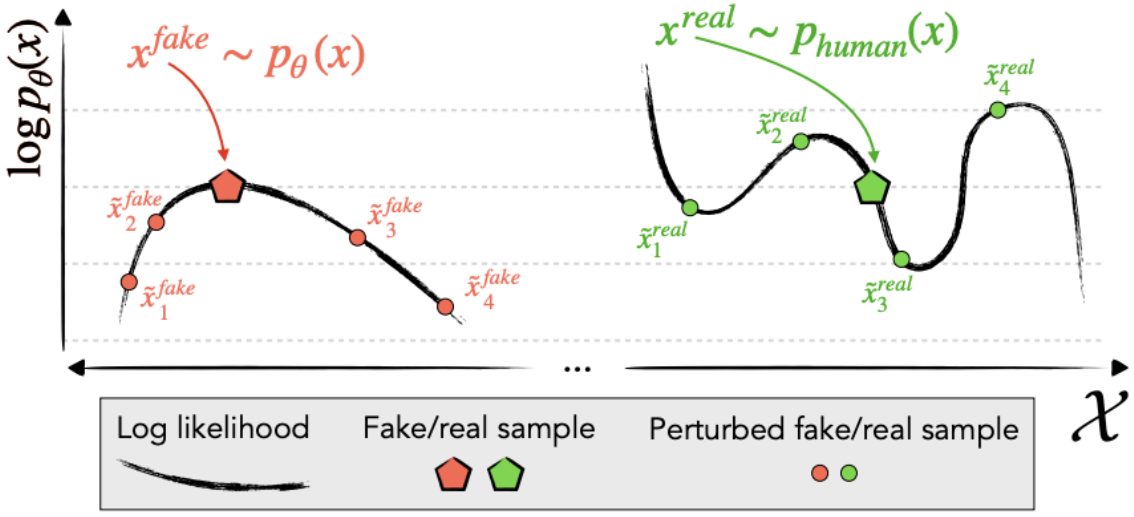

Figure 1: Taken directly from DetectGPT paper. They identify and exploit the tendency of machine-generated passages (left) to lie in negative curvature regions of the log probability curve, where nearby samples have lower model log probability on average. In contrast, human-written text (right) tends not to occupy regions with clear negative log probability curvature.

cent. This hypothesis was explored in Wettig et al.'s study titled "Should you mask 15 percent in masked language modeling?" (Wettig et al., 2023). The 15 percent masking rate came from the reasoning that "models cannot learn good representations when too much text is masked, and training is inefficient when too little is masked." (Wettig et al., 2023) In other words, 15 percent was just accepted as the default, and the impact of different masking rates under different conditions was left unexplored. Wettig et al. experimented with different model sizes and different masking strategies. They found that larger models should adopt higher masking rates, and that the default uniform masking strategy does better on higher masking rates than more sophisticated masking strategies (e.g. PMI Masking). In particular, they surprisingly found that 40 percent does better than 15 percent for BERT-large size models.

2.2 Temperature

For DetectGPT, Mitchell et al. do not adjust the hyperparameters of the mask-filling model T5. In particular, they keep the temperature at 1 for all their experiments. Temperature is a parameter used in NLP models to increase or decrease the "confidence" a model has in its most likely response. Higher temperatures make the model more "creative," while lower temperatures make the model more "confident." In general, neural networks produce class probabilities by applying a softmax function to produce a probability vector.

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

where z is the logit vector, q is the probability vector, and T is the temperature parameter (Shenk, 2018).

3 Experiments

3.1 Baseline Replication

We first sought to show that we were able to reproduce the results of Mitchell et al.'s paper successfully. In Mitchell et al.'s study, they vary the model that generates the passage, the scale of that model, the dataset, and the number of perturbations generated from one passage. They also compare the results to different zero-shot methods for machine-generated text detection (that also leverage the predicted token-wise conditional distributions of the source model for detection).

We decided to only reproduce the results from their experiment using the XSum dataset (Narayan et al., 2018), the GPT2-xl model (Radford et al., 2019) for passage generation, and T5-3B (Raffel et al., 2020) for mask filling. We varied the number of perturbations generated from each passage (1, 10, 100), and also compared each of the resulting

Model	AUROC
perturb=1_d	0.8155
perturb=1_z	0.8155
perturb=10_d	0.9676
perturb=10_z	0.9456
perturb=100_d	0.9879
perturb=100_z	**0.9887**

Model	AUROC
Likelihood	0.8791
Rank	0.7905
Log Rank	0.9085
Entropy	0.5642
RoBERTa-base	0.9676
RoBERTa-large	**0.9949**

Table 1: The first table shows DetectGPT models with varying numbers of perturbed texts. We also vary the normalization of the perturbation discrepancy (d: non-normalized, z: normalized). The second table compares the results of the DetectGPT model to different zero-shot methods for machine-generated text detection.

AUROC (area under the ROC curve) scores to other machine-generated text detection models. We used the normalized version of the perturbation discrepancy (by dividing by the standard deviation of the observed values) in the experiments in addition to the non-normalized version. See the results in Table 1. Additionally, we plotted each ROC curve for visual reference in Figure 2.

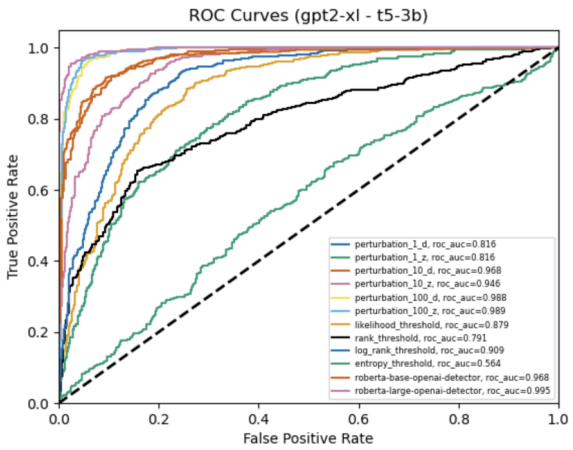

Figure 2: The ROC curves from the models in Table 1

Our AUROC results are very similar to Mitchell et al.'s, with AUROC values within 0.5 percent of the original paper. Results from Mitchell et al.'s paper on the XSum dataset is shown in Figure 3. The models that perform the best for both our results and the original results are RoBERTa-large and DetectGPT, which is the expected outcome.

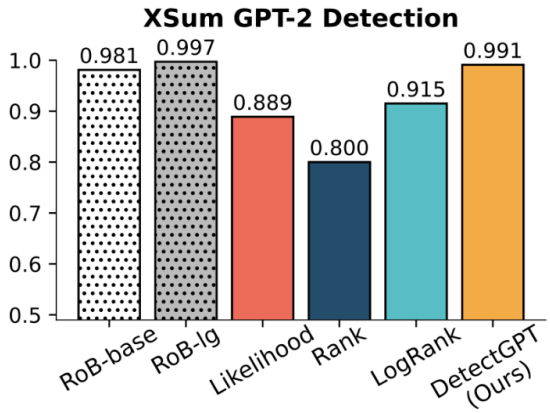

Figure 3: Results from original DetectGPT paper. Figure taken directly from paper.

Next, we take a closer look at the log likelihoods. We show results from the DetectGPT model with 100 perturbations and with normalized perturbation discrepancies in Figure 4. For the human-generated texts and its perturbations, we see that the difference between their distributions is very slim. For the machine-generated texts and its perturbations, we see that the difference between their distributions is larger than with the human-generated texts. This observation is also shown in the third histogram of the log-likelihood ratios. We additionally see that the distribution for the machine-generated text ratios is strictly above zero, aligning with the hypothesis of the DetectGPT paper that texts generated from a machine usually occupy negative curvature regions of the model's log probability distribution. We see that this is not the case with the human-generated texts, since there are likelihood ratios below zero.

The models with differing perturbations (normalized or not normalized) follow the same trends, and the same conclusion can be drawn from them. Thus, our replicated AUROC scores and log likelihood values align with the results of the original paper. This gives a validation that our research findings for when we extend the baseline (experiment with varying the masking percentage and temperature) are reliable.

3.2 Baseline Extension Results

3.2.1 Masked Percentage

We first varied the masking percentage for generating the perturbed text. We show results from masking rates of 5%, 15%, 25%, and 30%. We use

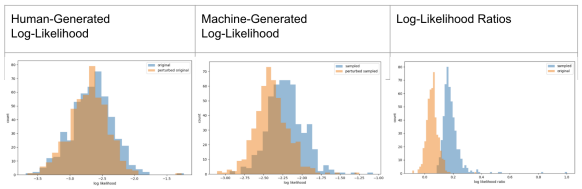

Figure 4: **Left:** Log-likelihood distribution of the original human-generated text (blue) and the distribution of the average log-likelihood of the 100 perturbations of a text (orange). **Middle:** Log-likelihood distribution of the sampled machine-generated text (blue) and the distribution of the average log-likelihood of the 100 perturbations of a sampled text (orange). **Right:** This histogram sums up the previous two histograms by comparing the log likelihood ratios of the human (orange) and machine (blue) generated text compared with their perturbations.

Masked Percentage	AUROC
5%	0.998892
15%	0.999576
25%	0.99946
30%	0.999352

Table 2

GPT2-small (Radford et al., 2019) as the passage generation model, and T5-3B as the mask filling model. We generate 100 perturbations for each text and normalize the perturbation discrepancy. The resulting AUROC scores are shown in Table 2.

We see that there isn't much variation in AUROC score as we change the masking percentage. The highest AUROC came from a masking percentage of 15% with 0.99957. The lowest AUROC came from a masking percentage of 5% with 0.99889, which is just 0.0007 lower than the highest AUROC. This is an interesting outcome. It suggests that DetectGPT works even with extremely small changes to the original texts. It also suggests that even with larger changes to the original texts, the accuracy of DetectGPT does not improve. Perhaps there is a threshold for how much a text needs to change, and then after that threshold is passed the model does not improve. Since we tested with masking percentages as low as 5%, this threshold must be extremely low (lower than 5% of the text changed).

Next, we take a closer look at the log-likelihood distributions in Figure 5. The shape of the distribution of the log likelihood ratios for both human-generated texts (orange) and machine-generated texts (blue) are relatively similar across the different mask percentages. However, we see that as the masking percentage gets bigger, the difference between the text and the perturbations gets larger. For the human-generated texts, we see almost no change in the log likelihood distributions across masked percentages. However, for the machine-generated texts, we see that as the masked percentage increases, the more the orange histogram separates from the blue histogram. We can think of the regions of the histogram that are overlapping as regions where the model is more "uncertain" in its classifications. As the masked percentage increases, the smaller the region of "uncertainty" becomes. In other words, as the masked percentage increases, the more confident the model becomes in its predictions. However, models that are more confident are not necessarily better classifiers, as we can see in the resulting AUROC scores.

Temperature	AUROC
0.01	0.990772
0.1	0.990716
1	0.999576
10	0.501648
100	0.462984

Table 3

3.2.2 Temperature

Then, we experimented with varying the temperature of the T5-3B model (mask filling model). We show results from using temperatures 0.01, 0.1, 1, 10, 100. The resulting AUROC scores are shown in Table 3. The log likelihood distributions are shown in Figure 6.

With temperatures **lower** than the default of one, both the human-generated and machine-generated log likelihood distributions shift to the left. Additionally, the machine-generated log likelihood distributions (blue and orange) start to overlap more, resembling the pattern we've seen for the human-generated log likelihood distributions. This likely happens because sampling more "realistic" texts with lower temperature could create a perturbation that is more likely to be outputted by the GPT model than the original sampled text. If we are sampling more perturbed texts that have a higher log probability than the original sample, then that resembles what the model thinks should happen to a human-generated text. Looking at the log-likelihood ratios, we see that the orange and blue are still relatively separated, meaning DetectGPT

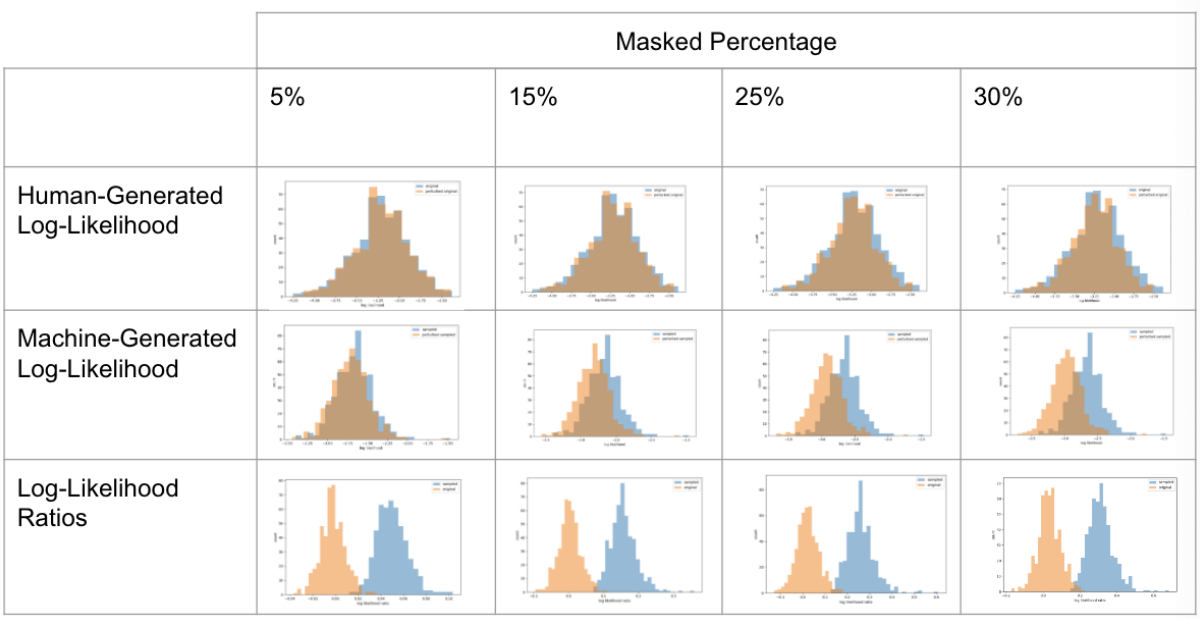

Figure 5: **Human-Generated Log-Likelihood:** Log-likelihood distributions of the original human-generated text (blue) and the distribution of the average log-likelihood of the 100 perturbations of a text (orange). **Machine-Generated Log-Likelihood:** Log-likelihood distributions of the sampled machine-generated text (blue) and the distribution of the average log-likelihood of the 100 perturbations of a sampled text (orange). **Log-Likelihood Ratios:** These histograms sum up the previous two histograms by comparing the log likelihood ratios of the human (orange) and machine (blue) generated text with their perturbations.

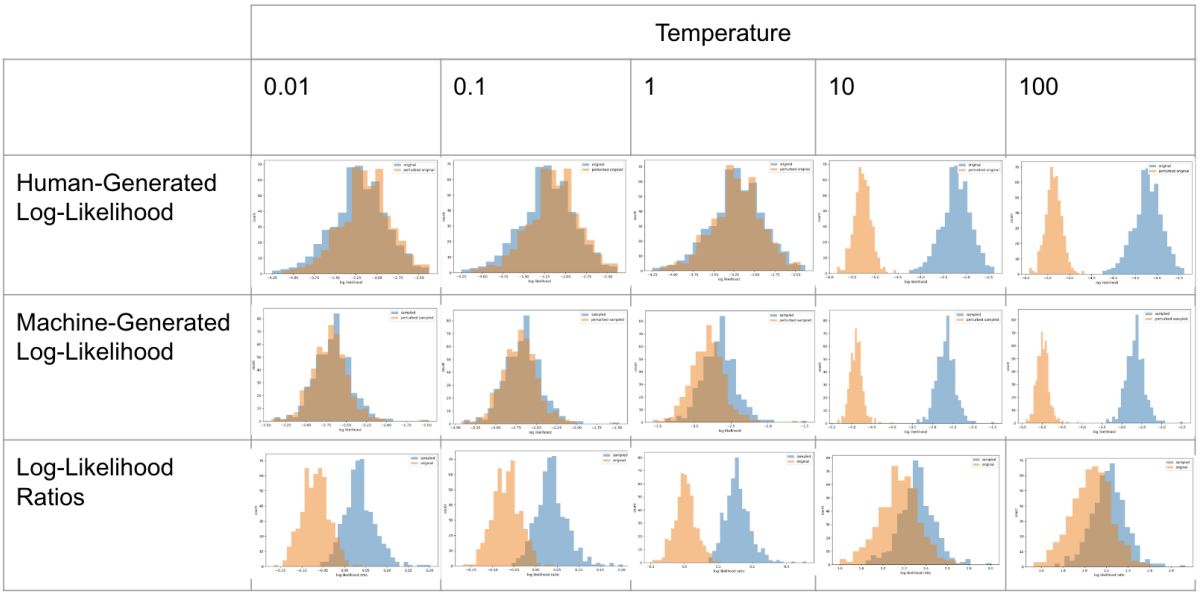

Figure 6: Same as Figure 5 except with varying temperature.

should still be able to predict well. This is observed since the AUROC scores for temperatures 0.01 and 0.1 are still above 99%.

With temperatures **higher** than the default of one, both the human-generated and machine-generated log likelihood distributions shift to the right. In addition to shifting, the blue and orange separate for the human-generated texts, resembling the pattern we've seen for the machine-generated distributions. This likely happens because sampling more "random" texts with higher temperature creates perturbed texts that have generally make less "sense" than the original text and so are generally out of the "local" region and substantially lower on the

log-likelihood curve. This makes it difficult for the model to tell the difference between human and machine generated texts as the perturbed texts have substantially lower log-likelihoods than both of them at these higher temperatures. In other words, the classification scheme of the model requires the the perturbed texts to be generally "worse" (lower log-likelihood) than the machine generated texts, while being sometimes "better" (higher log-likelihood) than the human generated texts. However, at higher temperatures, perturbed texts are generally "worse" than both human and machine generated texts, so the classification scheme of the model no longer works. Looking at the log-likelihood ratios, we see that the orange and blue are relatively overlapping, meaning it should be harder for DetectGPT to predict well. This is observed since the AUROC scores for temperatures 10 and 100 drop to around 50% (as good as random guessing).

4 Conclusion

We have replicated the work of Mitchell et al., and successfully arrived at similar results as the original paper. In addition to reproducing DetectGPT, we experimented changing a couple hyperparameters of the mask-filling model T5-3B–masked percentage and temperature. We found that as mask percentage increased, the difference between the text and the perturbation log likelihoods got larger. This told us that the confidence of the model's output with the higher masking percentage was higher than with the lower masking percentage. However, because the AUROC dropped slightly from a masking percentage of 15% to 30%, we concluded that "confidence" doesn't equate to a more accurate model. At temperature values lower than one, we found that the log likelihood ratio histograms of the texts and its perturbations shifted to the left. In other words, the lower the temperature, the more DetectGPT predicts a given text as human-generated. At temperature values greater than one, the log likelihood ratio histograms of the texts and its perturbations shifted to the right. Additionally, the higher the temperature, the less DetectGPT is able to distinguish human from machine-generated text.

5 Future Work

Originally for our project, we were also supposed to experiment changing the mask filling model itself from T5 to BART. For DetectGPT, Mitchell et al. used the mask filling model T5-3B for almost all experiments (except for GPT-NeoX and GPT-3 experiments where they use T5-11B). BART is a sequence-to-sequence model, like T5, but instead is implemented with a bidirectional encoder over corrupted text (corrupted by adding an arbitrary noising function) and a left-to-right decoder (reconstructing the original text). It would have been interesting to see how using models like BART-base and BART-large would affect the results of DetectGPT. Unfortunately as we were trying to implement the BART model, we ran into many unexpected bugs that we were unable to fix before the deadline. If we had more time, it would be great to see the results of this experiment. Additionally, we jump from a temperature of 1 to 10, and don't experiment with anything in between. For future work, we would like to see the results of DetectGPT using, for example, 2 as the temperature. Perhaps there is a point where DetectGPT actually improves by increasing temperature slightly instead of drastically.

Acknowledgements

We would like to thank Professor Danqi Chen for teaching us so much in this class, and also our advisor Samyak Gupta for his constant support and advice throughout the project.

References

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

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*, abs/1808.08745.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Justin Shenk. 2018. What is temperature in lstm (and neural networks generally)?

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling?