Distinguishing between Machine-Generated and Human-Written Text:

Evaluating an AI Text Classifier Across Different Combinations of Topic and Writing Style

Northeastern University
CS6220: Natural Language Processing
Alex Leon, , David Dada, Sarthak Khandelwal

Abstract:

Neural network architectures like the transformer have become widespread in the machine learning domain. This has given birth to systems such as ChatGPT which utilizes GPT-3 to perform multiple tasks including question answering, text generation, and text summarization. However, this advancement has also led people to misuse such architectures, which is putting forth a wrong impression of the advancement of AI. Hence, the detection of such systems has become imperative. With time, several supervised and zero-shot techniques have been introduced to address this problem, providing excellent scores on test data.

This study evaluates a zero-shot method called DetectGPT [1]. The model achieved a better score with the XSum, SQuad, and WritingPrompts datasets compared to supervised and other zero-shot methods. This study validates the method's application to other writing styles (such as Reviews and Book descriptions) and presents its performance. We achieved comparable results on our new datasets using the DetectGPT algorithm, as the results achieved on the original datasets.

Introduction:

Large language models (LLMs) are considered some of the most profound applications of ANI to date. In large part, the widespread popularity of LLM-based applications like chatGPT can be attributed to the transformer architecture. The success of the chatGPT application has been demonstrated in a variety of NLP tasks including text generation and question-answering. The impact to student learning needs to be further assessed, as students can use such applications' question-answering abilities to easily answer questions with little-to-no intellectual challenge. Further, students can use the text generation abilities of these LLM-based applications to create convincing, human-like written essays in a matter of moments. There may be a large-scale need for instructors to identify if a given piece of text was created by an AI, or by a human. DetectGPT is a method developed by a group of professors from Stanford that can determine if a given piece of text was generated by a machine (such as chatGPT) or was written by a human.

Contributions. Here, we extend the original DetectGPT paper. We apply the method on new datasets and compare the results to the ones achieved by the original authors, verifying the algorithm works well on new datasets (especially ones that have a different style of text). Further, we document the original source code and provide a document outlining how we ran our code (and tuned associated hyperparameters).

Literature Review:

Increasingly, large Language Model (LLM) models have gained much attention in the natural language processing (NLP) community due to their impressive performance on various language-related benchmarks and the ability to generate convincing and on-topic text. GPT-3, developed by OpenAI, is one such LLM model that has received significant attention due to its large size and impressive performance on various language tasks.

One significant application of LLM models is machine-generated text detection, which has been studied by several researchers. GROVER, developed by Zellers et al. in 2019, was the first LLM trained specifically for generating realistic-looking news articles [3]. 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.

Several other works have also explored supervised models for machine-generated text detection, including models based on neural representations, bag-of-words features, and hand-crafted statistical features. Solaiman et al. noted 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 [4].

However, models trained explicitly to detect machine-generated text tend to overfit their training distribution of domains or source models, as noted by Bakhtin et al. and Uchendu et al [5]. In addition, the problem of machine-generated text detection echoes earlier work on detecting deep-fakes, artificial images, or videos generated by deep nets, which has spawned substantial efforts in the detection of fake visual content.

Other work explores watermarks for generated text. Kirchenbauer et al. (2023) have proposed a technique for modifying the generations of a language model to include watermarks, making them easier to detect [6]. In contrast, our approach with DetectGPT does not assume that the generated text is intentionally modified for easy detection. Instead, we focus on detecting text that is generated using standard sampling strategies from publicly available LLMs.

Dataset and Metrics:

In this study, we mainly use three datasets that highlight different writing styles on which the model has been evaluated. Following is the description of each dataset.

1. **MedQuad:** This dataset contains 47,457 question-answer pairs [2]. The dataset is populated with 12 NIH websites making a total of 37 types of Question Answer pairs. Several additional annotations were also provided in the original dataset but were out of context for this task and thus not taken into account. Furthermore, we combined question-answer pairs into single strings that were then sent to the model for performing further tasks in the detection process.
2. **Amazon Reviews:** 500,000 short paragraph reviews of Amazon food products in the fine food category. It also includes reviews from other categories. For the purposes of this paper, only the product review paragraph was used.
3. **Google Books:** 1300 book titles, and descriptions among other metadata from the Google Books API. For the purposes of this paper, only the book descriptions were used.

In the evaluation of a binary classification model, the area under the receiver operating characteristic curve (AUC/ROC) is a widely used performance metric. The AUC/ROC measures the model's ability to distinguish between the positive and negative classes, where the true positive rate (sensitivity) is plotted against the false positive rate. The closer the AUC/ROC score is to 1, the better the model is at distinguishing between the two classes.

This evaluation is particularly useful in scenarios where the cost of false positives and false negatives is high, such as in medical diagnoses or fraud detection. The high AUC/ROC score achieved by GPT-2 suggests its potential as a valuable tool in these fields. Overall, the use of AUC/ROC as a performance metric and the impressive results obtained by GPT-2 in this study highlights the importance of robust evaluation techniques in machine learning and the potential of advanced models like GPT-2 in various fields.

Methodology

DetectGPT is a zero-shot machine-generated text detection algorithm meaning that it does not require data for its training. The approach behind DetectGPT is pretty straightforward. This algorithm works using the inference from the log curvature of a difference between the candidate passage and the mean of its several perturbations. This difference is termed a *perturbation discrepancy*. The algorithm works in a white box setting where we do not assume access to model architectures or parameters that can be further fine-tuned.

In this project, we aimed to develop a zero-shot model for detecting machine-generated text. The proposed model, DetectGPT, utilises the Generative Pretrained Transformer (GPT-2) architecture as the source model and a Text-to-Text Transfer Transformer (T5) model as the mask-filling model. To evaluate the performance of DetectGPT, we used three datasets, namely MedQuad, Amazon Reviews, and Google Books. MedQuad is a dataset of medical text, Amazon Reviews is a dataset of product reviews, and Google Books is a dataset of books in various genres. We selected these datasets to cover a diverse range of text genres. The performance of DetectGPT was compared with other state-of-the-art zero-shot machine-generated text detection algorithms. These algorithms were selected based on their popularity and reported performance in the literature. The comparison was made based on the AUROC (Area under the ROC curve) score.

Algorithm 1 DetectGPT model-generated text detection

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

Fig.1: DetectGPT Algorithm

To perform the machine-generated text detection, the candidate passage was input to the DetectGPT model along with the source model (pθ), perturbation function (q), number of perturbations (k), and decision threshold (e). The function q(•|x) was used to perform perturbations on the candidate passage using the T5 mask-filling model. The perturbation discrepancy (d) was calculated as the difference between the log probability of the candidate passage and the expected value of log probabilities of the perturbations of the candidate passage under the mask-filling model. If the ratio of perturbation discrepancy and standard deviation of the distribution was greater than the decision threshold, we classified the passage as machine-generated text; otherwise, it was classified as human-written text.

In this algorithm, we take two versions of a candidate passage. The first one is from the original dataset and the second one is created using a sample from the original dataset and combining it with text generation from a source model. We use several source models for experimentation such as gpt2, gpt2-medium, and gpt2-large in this study to visualise the impact on DetectGPT's performance by changing the source model. We also compared mask-filling models such as t5-small and t5-large to apply masking on a percentage of words in the corpus to include randomness. After careful consideration, we selected GPT-2 as the source model and t5-large as the mask-filling model. Inspired by the original paper, we have also kept the temperature parameter to 1. This parameter identifies the randomness in the generated text. It's usually between 0 and 2. The more it is close to 2, the more random the output-generated text would be. Here also, we have used two different masking models i.e. t5-large and t5-small for the same purpose.

Experiments and Results

Using AUC/ROC, the original paper benchmarks GPT2 to have achieved around 97-99% accuracy on the datasets. Our results returned 93%, 94%, and 76% accuracy on the MedQuad, Google Books API, and Amazon Food Reviews datasets, respectively.

Method	GPT-2
logp(x)	**0.86**
Rank	0.79
LogRank	**0.89***
Entropy	0.60
DetectGPT	**0.99**

Fig 2. XSum Dataset [1]

Method	GPT-2
logp(x)	**0.83**
Rank	0.76
LogRank	**0.88***
Entropy	0.50
DetectGPT	**0.97**

Fig 3. SQuAD Dataset [1]

Method	GPT-2
logp(x)	**0.82**
Rank	0.80
LogRank	**0.90***
Entropy	0.58
DetectGPT	**0.99**

Fig 4. Writing Prompts [1]

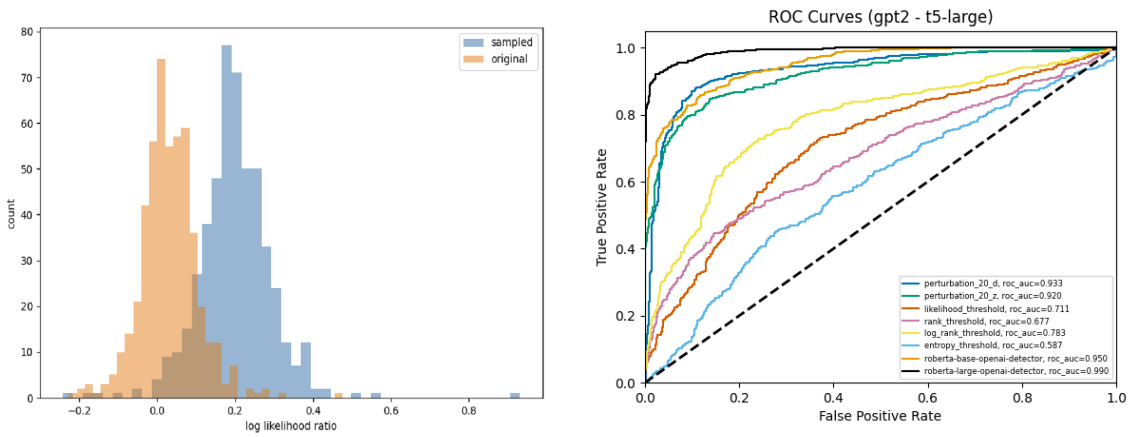

Fig. 5, MedQuad Dataset, ROC 0.93

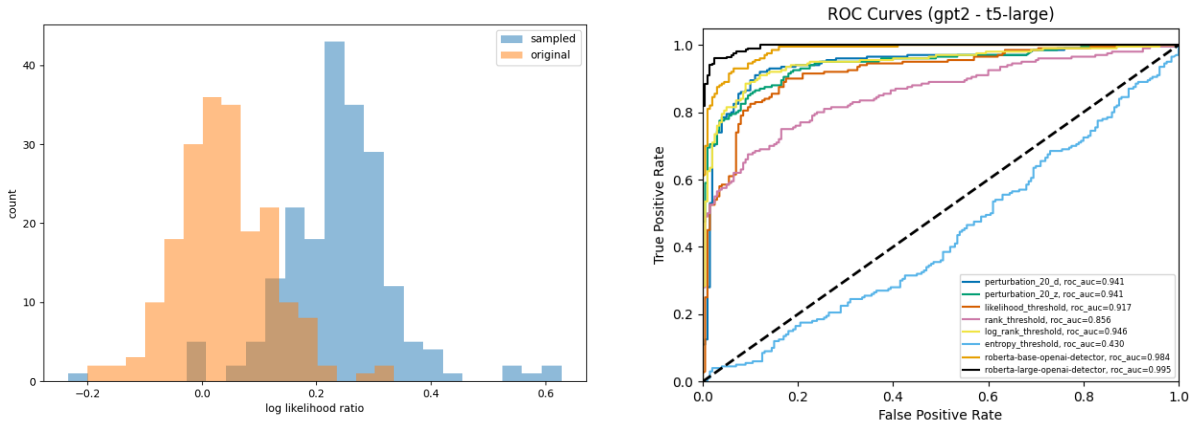

Fig. 6, Google Books API Dataset, ROC 0.94

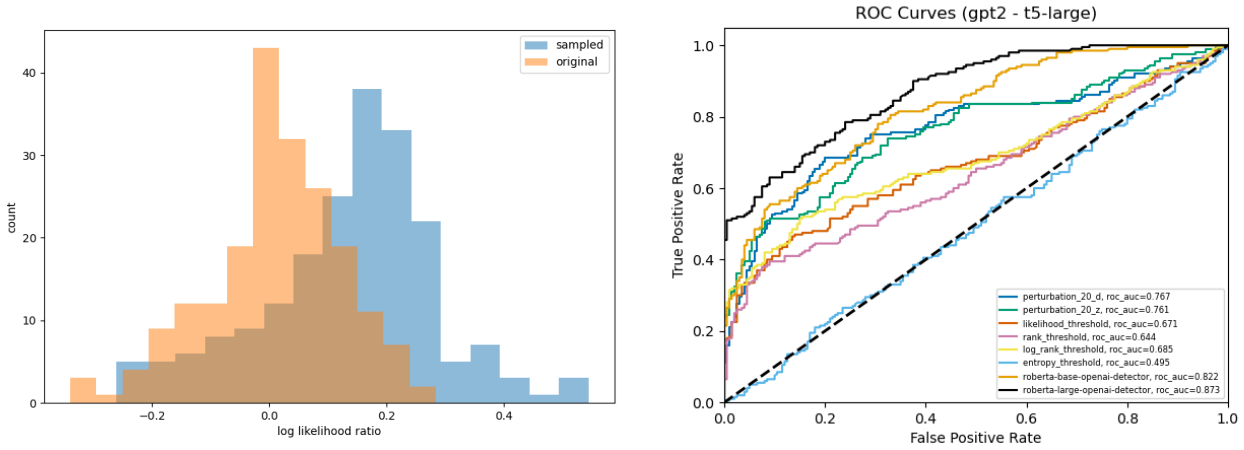

Fig. 7, Amazon Food Reviews Dataset, ROC 0.76

Conclusion and Future Work

Our experiments showed that GPT-2 outperformed GPT-2 Large and GPT-2 Medium in detecting generated text. This could point to the GPT-2 model generalising well, as opposed to overfitting.

We faced significant computational limitations, even with cloud computing, which prevented us from testing the t5-3b model. Instead, we used t5-large, which outperformed the t5-small and t5-medium. Because of this, we expect the t5-3b to outperform t5-large. This could also be the reason for the 76% accuracy with the amazon reviews, among other factors such as the length of text of each sample used, and the nature of the medium used.

We chose to use 20 perturbations, as after 100, the performance gains were negligible. We used a smaller set of samples (200) than was used in the original paper (500). This is because the zero-shot technique produces the same amount of data points across multiple datasets.

Moving forward, we suggest that future work explores improving the GPT-2 model using ensemble methods and further investigating the relationship between prompting and detection [1]. Another domain of exploration is verifying that negative log curvature that can be found in generated text, may also be present in other mediums such as audio, video, and images[1]. The final and original future work idea is exploring if other genres of text affect the model's predictive accuracy.

Bibliography

[1] Mitchell, Eric, et al. "DetectGPT: Zero-shot machine-generated text detection using probability curvature." arXiv preprint arXiv:2301.11305 (2023).

[2] Ben Abacha, Asma, and Dina Demner-Fushman. "A question-entailment approach to question answering." BMC bioinformatics 20.1 (2019): 1-23.

[3] Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. Defending against neural fake news. In Neural Information Processing Systems, 2019.

[4] Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., and Wang, J. Release strategies and the social impacts of language models, 2019. URL https://arxiv.org/ftp/arxiv/papers/1908/1908.09203.pdf

[5] Bakhtin, A., Gross, S., Ott, M., Deng, Y., Ranzato, M., and Szlam, A. Real or fake? Learning to discriminate machine from human generated text. arXiv, 2019. URL http://arxiv.org/abs/1906.03351. cite arxiv:1906.03351.

[6] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models, 2023. URL https://arxiv.org/abs/2301.10226.