AI-Generated Text Detection via Contrastive Learning

Shicheng Wen Yinzhen Wang Yiming Tang Ran Liu Han Xiao
{wenshich, yinzhen, tangyimi, rliu0866, hxiao603}@usc.edu

Abstract

With the great success of generative language models, detecting generated text has become a research hotspot. However, current popular methods belong to white-box detection, which requires access to the model internals. Most of the existing black-box detection methods treat detection as a classification task using NLU models. In this paper, we propose a black-box detection method based on RoBERTa for GPT-2. We aim to further distinguish between AI-generated and human-written text by leveraging the loss of contrastive learning, and introduce adapters to save training time and computing resources. To demonstrate the effectiveness of our detector, we created new datasets for stories and news. The experimental results show that our detector achieves decent performance and can effectively detect GPT-2 generated text. Compared to simple RoBERTa classification, our method is more efficient and can effectively prevent overfitting. Code available at GitHub[1].

1 Introduction

Recent developments in natural language processing have led to the creation of large language models, such as GPT-4(OpenAI, 2023), which are capable of generating increasingly fluent and convincing machine-generated text. While these models offer many benefits and have the potential to be used in a variety of useful applications, they also pose a serious threat to the accuracy and reliability of information online. The ease with which anyone can access these models and generate large amounts of AI-generated text has led to an increase of fake news, rumors, and other forms of misinformation that may significantly mislead public opinion on essential social events. Furthermore, the usage of AI-generated text has also led to the rise in academic cheating and plagiarism, considering the difficulty to find the proof of using ai in the school work. Given the widespread impact of this issue, detecting AI-generated text has become a critical and urgent task. In this paper, we propose a method for automatically detecting AI-generated text from human-written text.

We conduct experiments on two human-written news-style datasets and two human-written story-style datasets, with negative instances generated by GPT-2(Radford et al., 2019). We compared the result from four settings: original RoBERTa-base, RoBERTa-base with triplet Loss, RoBERTa-base with a Mix-and-Match adapter, RoBERTa-base with the Mix-and-Match adapter and triplet loss. The experiments show that our method performs good accuracy on all datasets and the analysis further indicate that our model can distinguish ai-generated text and human written text.

2 Related Work

The field of text generation has seen remarkable advancements with large generative language models like ChatGPT(OpenAI, 2022) and GPT-2(Radford et al., 2019), which can produce highly realistic text that is often difficult to distinguish from the human-generated text. To detect AI-generated text for ensuring information security, Tang et al. (2023) suggested LLM-generated text detection can be categorized into two types, namely black box detection (lacking source model access and requiring the collection of text samples from both human-generated and machine-generated sources) and white box detection (the detector has full access to the target language model). The former involves data collection, detection feature selection, and classification models. The latter, on the other hand, comprises Post-hoc and Inference Time Watermarking.

White Box Detection Zellers et al. (2019) developed Grover, a model similar to GPT-2, for generating and discriminating news produced by

[1] https://github.com/wenshicheng97/detect-gpt2

neural networks as a defense mechanism for fake news. Gehrmann et al. (2019) visualized generated text by histograms of top-k counts, probability distribution counts, and top-10 entropy, providing straight and interparable comparison between machine- and human- generated text. Mitchell et al. (2023) has reached state-of-the-art on zero-shot machine-generated text detection problems by DetectGPT, which involved introducing perturbations to a given text passage and measuring perturbation discrepancy. Nevertheless, the approach is limited due to the assumption of access to a reasonable perturbation function. Moreover, it is more compute-intensive than other detection methods.

Black Box Detection Solaiman et al. (2019) introduced a sequence classifier based on fine-tuned RoBERTa to identify text created by GPT-2. Ippolito et al. (2019) has demonstrated fine-tuned BERT, bag of words, histogram-of-likelihood ranks with or without buckets, and total probability on different datasets generated by different sampling methods (top-k, nuclear, and random), truncation length, and choice of priming or no priming. Zhong et al. (2020) proposed a way to construct internal factual structure by going through entities inside documents and fit into a graph neural network. Then, it combines with word representations derived from RoBERTa (Liu et al., 2019), improving the detection performance at document level. Rodriguez et al. (2022) combined defender and adversary and shown that a few in-domain genuine and synthetic data is enough for good detection performance.

We introduce some methods to generate positive and negative samples for contrastive learning, using prompts to make AI continue to write a story, summarize some news or generate texts with some key words. We will use positive and negative samples to fine-tune a pre-trained language model (RoBERTa(Liu et al., 2019)) on the classification task. Considering the training time and resource allocation, we will use adapters with our fine-tuned model for shorter training time and lower resource for a light version of our method.

3 Problem Description

In this research, we investigate the problem of detecting AI-generated text within a black box few-shot framework. The few-shot nature of this problem implies that the detection process necessitates merely a limited number of samples derived from both human-authored and AI-generated texts. Operating under black box settings signifies that we are unable to access the original text generation model.

To address this challenge, we propose the adoption of a RoBERTa-based classification model, which is fine-tuned using a labeled dataset specifically curated for this task. This approach aims to provide an effective and efficient means of identifying AI-generated text, even when confronted with a paucity of available samples and without direct access to the underlying generative model.

4 Methodology

Concurrent with the release of the GPT-2 output dataset[2], Radford et al. (2019) introduced a detector model grounded in the RoBERTa architecture. The original model employed the pre-trained RoBERTa-For-Sequence-Classification module from the HuggingFace library, effectively transforming the problem of discerning AI-generated text into a binary classification task.

4.1 Triplet Loss

In order to examine additional potential avenues, we propose a novel model that modifies the loss function by incorporating a combined loss, defined as:

$$\mathcal{L} = \mathcal{L}_c + \gamma \mathcal{L}_t \tag{1}$$

Where $\mathcal{L}_c$ is the original classification loss and $\mathcal{L}_t$ is the triplet loss as follows:

$$\mathcal{L}_t = \sum_{i=1}^{N} [\mathcal{S}(t_i, f_i) - \mathcal{S}(f_i, \hat{f}_i) + \alpha] \tag{2}$$

Wheres N denotes the batch size, t_i symbolizes the human-authored text, f_i represents the AI-generated text, and $\hat{f}_i$ corresponds to the text within the batch exhibiting the lowest consine similarity score $\mathcal{S}$ in relation to f_i. Our primary objective is to map texts belonging to the same category, whether human-authored or AI-generated, to a proximate feature space. Concurrently, we strive to map texts with analogous content but divergent categories to distinct and distant feature spaces.

To accommodate the triplet loss within the fine-tuning process, we establish a one-to-one correspondence between the "same content" human-authored and AI-generated texts, subsequently

[2] https://github.com/openai/gpt-2-output-dataset

forming positive and negative sample pairs during the construction of the training dataset. This approach involves concatenating each sample pair into a single, unified sample throughout the training data, effectively enabling the model to learn the relationships between human-authored and AI-generated texts with similar content.

4.2 Adapter

In order to enhance the performance of the classification model, it may be necessary to finetune the model, which could result in an extended duration for the training process as all parameters are updated. Furthermore, a prior classification experiment utilizing the RoBERTa baseline demonstrated considerable overfitting with an increase in epoch numbers. To address this issue, we implemented the Mix-and-Match Adapter (He et al., 2021) for a unified and parameter-efficient finetuning approach.

5 Experiment

5.1 Datasets

We collected human-written text data from various datasets, including CNN Daily Mails[3], long-text dataset CC-News dataset[4], Wiki Plots[5] and short-text dataset ROC stories[6]. To ensure the similarity in content between human-authored and AI-generated text pairs, we initially truncated the original data, retaining only a small portion as the prompt. Subsequently, we input this truncated prompt into the GPT2-XL model, which generated the remaining text. As a result, the content of these human-AI text pairs should exhibit a high degree of similarity, given that both the human-written and AI-generated texts share the same foundational prompt.

For longer text sources such as CNN Daily Mail, CC-News, and Wiki Plots, we opted to use the first 50 tokens as the prompt for the continuation of writing. To maintain a comparable length with the original text, we limited the generated text to the length of the original text, with a tolerance of plus or minus 20 tokens. The approach ensures that the AI-generated text remains contextually similar and relatively close in length to the original human-authored text.

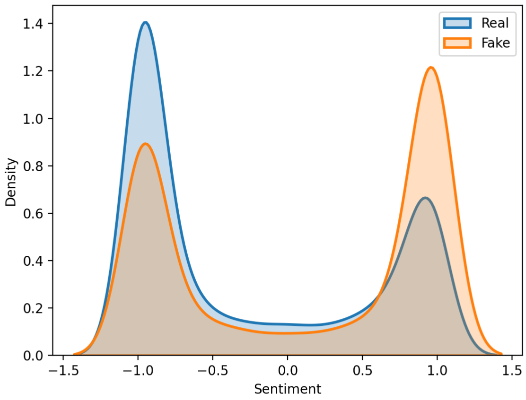

Figure 1: Density distribution plot on Wiki Plots sentiment scores

Each story in the ROC Stories dataset consists of approximately 4 to 5 sentences, rendering them considerably shorter in comparison. Consequently, we selected the first sentence of each story to serve as the prompt for generating the AI-generated text.

5.2 Training

We have four training settings: 1) RoBERTa-base with the original loss, our baseline 2) RoBERTa-base combined with the original loss and the triplet loss 3) RoBERTa-base with a Mix-and-Match adapter 4) RoBERTa-base with the new loss and the adapter. We apply the Adam optimizer with learning rate of 2e-5 and weight decay of 1e-8. We use accuracy to evaluate the performance.

6 Result & Discussion

6.1 Stories

As shown in Table 1, the performance under each setting is extremely good. The accuracy scores are around 99%. The best performance on Wiki Plots reach the best performance with MaM adapter and with Triplet loss separately and ROC Stories have the best only with triplet loss. Since GPT-2 is trained on Web text, it is hard for GPT-2 to mimic story. Thus, there could be a huge difference between human-writing style and GPT2-writing style in stories. In order to analyze the style, we conduct sentiment analysis between real and fake text using Natural Language Toolkit (NLTK).

We plot the distribution of the compound sentiment score in Figure 1. Zero score stands for neutral sentiment. Positive score represents positive

[3] https://huggingface.co/datasets/cnn_dailymail/viewer/3.0.0/train

[4] https://huggingface.co/datasets/cc_news

[5] https://github.com/markriedl/WikiPlots

[6] https://huggingface.co/datasets/wza/roc_stories/tree/main

Method	Dataset			
	CNN Daily Mail	CC News	Wiki Plots	ROC Stories
RoBERTa	98.50	**96.4**	99.82	99.62
RoBERTa+MaM Adapter	94.2	96.31	**99.97**	99.55
RoBERTa+Triplet Loss	**98.8**	95.92	**99.97**	**99.71**
RoBERTa+MaM Adapter+Triplet Loss	97.3	95.5	99.91	99.54

Table 1: Validation accuracy(%) evaluated on four training settings on datasets including CNN Daily Mail, CC News, Wiki Plots, and ROC Stories. RoBERTa with triplet loss shows the best performance on most datasets.

Method	Dataset			
	CNN	CC	Wiki	ROC
RoBERTa	98.8	93.9	99.97	99.80
MaM Adapter	95.2	95.83	99.91	99.73

Table 2: Validation accuracy(%) of ablation studies on different datasets with RoBERTa baseline + triplet loss and RoBERTa + triplet loss + MaM adapter

sentiment and vice versa. There is a clear distribution difference between real stories and fake stories. Real stories express more negative sentiments than positive, whereas more fake stories have positive sentiment. In the sentiment perspective, we conclude the writing styles between human and AI on stories are significantly different.

6.2 News

Table 1 shows the performances on two news dataset are not as good as on story datasets. Though CNN Daily Mail has the best performance with triplet loss, the CC News even cannot outperform the baseline. It could be that GPT-2 has seen news data through pre-training. It can handle news text well and form a human-like style on writing news text. We also notice that some generated news list author names at the end, which is completely the same format as human-written news. With the similar writing style and same format as real news, GPT-2 can generate fake news hard for RoBERTa to recognize.

6.3 Ablation Studies

In the combined loss, our triplet loss defined that t corresponds to human-written text and f denotes AI-generated text. We conducted an ablation study by interchanging the roles of t and f, wherein t represents AI-generated text and f represents human-written text. The results, displayed in Table 2, reveal that the exchange of t and f has a minimal impact on the performance of the model on story datasets, and decreased the overall performance on news datasets.

7 Conclusion

In this paper, we proposed a new idea for improving the performance of the RoBERTa model by adding adapters and Triplet Loss to enhance the model's ability to distinguish between AI-generated text and human-written text. We added adapters to RoBERTa for parameter-efficient fine-tuning and used Triplet Loss to constrain the model to learn more accurate text representations. From the results, we can see that the model performs well on stories, but it still needs further improvement for news domain and the domains the generative model is trained on.

8 Future Work

Dataset domain It is important to select data domain for AI-detection task. On the one hand, if there is a domain the generative model is not trained on, the discriminative model can recognize them easily by simply tuning the model. On the other hand, if there is a domain the generative model is trained on and making the discriminative model hard to recognize, then more methods needed to improve the performance.

Generative model To improve our text-generation model, we can explore alternative models (GPT-3.5, GPT-4, Alpaca(Taori et al., 2023), etc). GPT-2 is outdated and lacks diversity. By using GPT-4 or GPT-3.5, we can access larger and more diverse datasets. Alternatively, we can use Alpaca, which is specifically designed to generate high-quality text for specific domains such as news or finance. More challenging data can make the task more meaningful. If data is easy to be recognized by a model, then the model can be easily fooled by finer fake text generated by more powerful models. This specialized design may allow for better results in these specific domains.

References

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. *CoRR*, abs/1906.04043.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Human and automatic detection of generated text. *CoRR*, abs/1911.00650.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. arXiv:2301.11305.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue. *OpenAI blog*.

OpenAI. 2023. Gpt-4 technical report.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting LLM-generated texts.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *CoRR*, abs/1905.12616.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. *CoRR*, abs/2010.07475.