Detection vs. Anti-detection: Is text generated by AI detectable?

Yuehan Zhang*[1,2][0000−0003−2197−8617], Yongqiang Ma*[1,2][0000−0002−4980−9834], Jiawei Liu[1,2][0000−0002−2774−1509], Xiaozhong Liu[3][0000−0003−3477−8323], Xiaofeng Wang[4][0000−0002−0607−4946], and Wei Lu[1,2][0000−0002−0929−7416]

[1] Wuhan University, Wuhan, 430072, China
[2] Information Retrieval and Knowledge Mining Laboratory of Wuhan University, Wuhan, 430072, China
[3] Worcester Polytechnic Institute, USA
[4] Indiana University Bloomington, USA
{john_love,mayongqiang,liujiawei,weilu}@whu.edu.cn xliu14@wpi.edu xw7@indiana.edu

Abstract. The swift advancement of Large Language Models (LLMs) and their associated applications has ushered in a new era of convenience, but it also harbors the risks of misuse, such as academic cheating. To mitigate such risks, AI-generated text detectors have been widely adopted in educational and academic scenarios. However, their effectiveness and robustness in diverse scenarios are questionable. Increasingly sophisticated evasion methods are being developed to circumvent these detectors, creating an ongoing contest between detection and evasion. While the detectability of AI-generated text has begun to attract significant interest from the research community, little has been done to evaluate the impact of user-based prompt engineering on detectors' performance. This paper focuses on the evasion of detection methods based on prompt engineering from the perspective of general users by changing the writing style of LLM-generated text. Our findings reveal that by simply altering prompts, state-of-the-art detectors can be easily evaded with F-1 dropping over 50%, highlighting their vulnerability. We believe that the issue of AI-generated text detection remains an unresolved challenge. As LLMs become increasingly powerful and humans become more proficient in using them, it is even less likely to detect AI text in the future.

Keywords: AI-generated Text Detection · Large Language Model · AIGC · Prompt Engineering

1 Introduction

With its powerful performance, large language models (LLMs) have been increasingly accepted as effective personal assistants to elevate our productivity. However, they also pose significant risks if misused. When unregulated, such

* Both authors contributed equally to this research.

technology facilitates the creation of convincing yet fraudulent content, including articles, news, and even fake scientific publications. Journalism, education and research are under the threat of inequality, plagiarism, and other various forms of misuse of LLMs[28, 17, 25]. Issues in intellectual property, ethics and security are raised. It also challenges the enforcement of legal and regulatory standards. Consequently, addressing these challenges necessitates reliable detection methods for AI-generated text, which is a binary classification task that detects whether the provided text, such as an essay submitted by a student, is generated by LLMs or written by humans[38].

Recent studies have highlighted challenges in distinguishing between human-authored and AI-generated texts[24, 8, 12], prompting the development of automated detection methods. OpenAI released its online detector[4], and researchers also proposed many AI-generated text detection methods[11, 14, 17, 18, 20, 23, 27, 31, 35, 39], particularly for the ChatGPT-generated text, such as ZeroGPT[6], GPTZero[3].

Along with the evolution of LLMs such as from GPT-1, GPT-2, and GPT-3 to ChatGPT, AI-generated text detectors were also constantly updated like a cat-and-mouse game. In the line of detection, AI-generated text detectors are developed based on metrics or deep learning. Watermark is also utilized to tackle the detection of AI-generated text[13, 19]. In the line of detection evasion, Sadasivan et al.[29] developed a paraphrase method. Researchers also show the effectiveness of randomly added spaces[7] and automatically optimized prompts[39]. Beyond the research of the detection method, the detectability of AI-generated texts is a question not sufficiently discussed. Chakraborty et al.[9] claimed that there is always detectability as the sample number or input length grows.

From the perspective of text generators, the model scale has increased a lot. Therefore, our first hypothesis is that the more parameters the model has, the more difficult it becomes to detect the generated text. Our experiment on the detection of GPT-series models' generated text shows that the detector trained for the weaker LLM has a performance decline when applied to a newer and stronger LLM, which shows the gap between outdated detectors and newer, stronger LLMs.

From the perspective of the prompt text fed to the text generator, researchers have found that the current LLMs are sensitive to the input prompt[41]. However, the detectors' performance on AI-generated text with deliberately designed prompts has not been evaluated. In our research, we make the second hypothesis that the AI-generated text is difficult to detect when deliberately designed prompts are used. To verify this hypothesis, we develop a prompt manipulation method that evades detectors sufficiently without any model training, generated-text paraphrasing, or editing. Specifically, we inject the writing style information into the prompt to change the model output to escape the detector. We find that when prompting the LLMs to generate target text with a given writing style, the LLM-generated text could become much less detectable.

Our findings highlight the gap between AI-generated text detectors and the potential anti-detection methods. Particularly, the detector, trained on a static

dataset collected from a single LLM, will fail when the size of the model increases or the generation prompt is customized designed. Factors like model scale and writing styles are important aspects to consider when developing AI-generated detectors.

2 Related work

2.1 Detection of AI-generated Text

As the detection of AI-generated text gains much attention[28, 33], many approaches are developed. They include statistical metrics-based methods such as entropy, perplexity, log-rank[31] and intrinsic dimensions[35]. DetectGPT[27] makes perturbations to detect AI-generated text based on log probability. There are also transformer-based classifiers such as HC3 classifier[15], and OpenAI classifier. DNA-GPT leveraged LLM itself to generate a few samples for detection based on the similarity between the text to be detected and newly generated ones. Researchers have also noticed the scenario of online environment[36]. Krishna et al. put forward a retrieval-based detection method to defend paraphrasing[20]. Hu et al. proposed a detector-paraphraser joint training method[17]. Watermark techniques [19, 11, 32, 40, 21] and datasets designed for detection research are conducted to tackle the challenge of detection[15, 24, 37]. Off-the-shelf detectors are deployed for the service of detection[3, 6, 5, 1, 2].

2.2 Anti-detection of AI-generated Text

Liang et al.[22] elevated and decreased the literature and vocabulary of text and successfully reversed the detection result. Lu et al.[25] took out an automatic prompt substitution framework to generate less detectable content. Sadasivan et al.[29] use paraphrasing attacks to evade watermarked and non-watermarked detectors and retrieval-based detectors. SpaceInfi[7] randomly added a space character to evade GPTZero[3], HC3[15], and MPU detectors[34].

2.3 Detectability of AI-generated Text

Sadasivan et al.[29] claimed an impossibility of detection as the total variation between humans and AI decreased, causing a theoretical detection ceil that detectors could not be employed practically. They also find that the total variation decreases as model size increases. Chakraborty et al.[9] argues that human and AI distributions are hard to be the same due to the vast diversity within the human population and finds that as collecting more samples the possibility of detection increases.

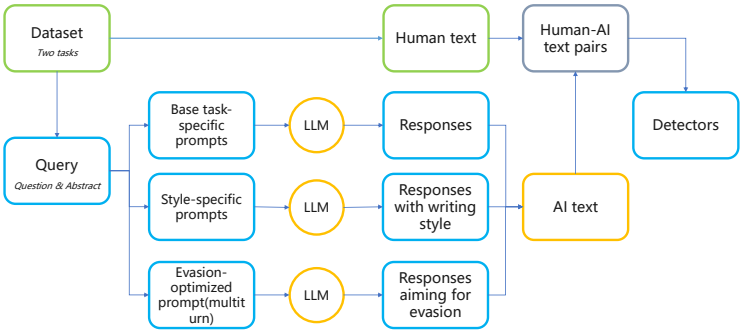

Fig. 1. Framework of Prompt Manipulation Method.

3 Methods

3.1 Task Definition

LLMs can play an important role in text reading and writing improvement. In the scientific domain, there are tools developed by integrating the OpenAI API, such as chatPDF[5] and GPT academic[6]. Considering the usage scenarios, we test our hypothesis on a QA task, a structured abstract generation task(SA task) and an abstract polishing task (Polish task). Under each task, a small set of datasets is generated to test the detectors' performance.

1. For the QA task, we prompt LLMs to generate the answer for a given question. The dataset used is randomly selected samples from HC3 dataset without restricting domains[15].
2. For the Polish task, we use the abstracts of research papers and prompted LLMs to polish the human-written abstract.
3. For the SA task, we use the titles of research papers and prompted LLMs to generate an abstract according to the titles.

The detectors only have access to the human text and generated responses with no extra information such as prompts, questions, original abstracts or titles of the abstract.

3.2 Prompt Design and Dataset Construction

We use LLMs with different scales to generate the structured abstract dataset for an AI-generated detector. Here, we employed GPT-series models, such as

[5] https://www.chatpdf.com/
[6] https://github.com/binary-husky/gpt_academic

GPT-2, text-davinci-002, and ChatGPT as our text generator. The text-davinci-002 is the former base model of ChatGPT[7]. And the GPT-2[30] is the weaker text generation model than text-davinci-002. The GPT-2 is a white-box model, the parameters of which are open-access. The text-davinci-002 and ChatGPT are the black model, whose responses could only be generated by API. We set max_tokens as 2048 and temperature as 1 for generation parameters.

The prompt is the input text containing instructions that is fed into the LLM to get the desired content. Prompts used in our experiments are designed under four rules to elevate the generating performance, as shown in Figure 2: Setting a role, stating the tasks clearly, providing information inside of backticks, and indicating the output format. For questions and abstracts in the prompt, we employed the dataset in previous works[15, 26] by random sampling.

To generate different writing styles of AI-generated text, we add a response text style control statement in the prompt: Write in the style of someone. The demo prompts of the QA task are listed in Appendix A, and the full prompts used in the Polish task are in Appendix B.

We also collected responses under deliberately designed prompts without writing styles. The concepts of perplexity and burstiness are explained to ChatGPT in the prompt for evasion of detectors in multi-turn chat dialogue format[10], which could leverage the power of chat format LLMs. The prompt is listed in Appendix C.

Finally, the text is tagged into categories. The original answer is labeled as a human answer, and the text generated from AI models is labeled as an AI-generated answer.

As ***a highly intelligent question answering bot***, your task is to **answer questions**.
You will be provided with the **question delimited by triple backticks.** ***Format the output*** in a JSON object with the following keys: question, answer.
```{question}```

Fig. 2. Example prompt used in QA task.

Table 1. The detection methods used in this work

Metric-based methods	Classifier-based methods
Log-Likelihood	
Rank	OpenAI Detector
Log-Rank	HC3 Detector
Entropy	ZeroGPT
GLTR Test 2 Features	

[7] https://platform.openai.com/docs/model-index-for-researchers/models-referred-to-as-gpt-3-5
```

3.3 AI-generated Text Detector

we select several metrics-based detectors and classifier detectors[16] and an online detector, as shown in Table 1. The Log-likelihood method takes the probability of words as features[30]. The Rank and Entropy methods use the average rank of words and the entropy of the predicted distribution as features[14]. GLTR applies the statistical features above[14]. Log-Rank uses the average observed log-rank of the tokens in the candidate text[27]. OpenAI Detector is a RoBERTa model fine-tuned on GPT2 output[30]. HC3 Detector is trained on HC3 dataset containing text pairs of human and ChatGPT[15]. ZeroGPT is an off-the-shelf online detector[6].For equivalent comparison, we trained Logistic Regression classifiers based on the metrics.

4 Experimental Results

4.1 Detectability Analysis of Different Scale LLMs

Here, we use the detector OpenAI -Roberta-detector to detect the AI-generated structured abstracts. The detector is trained on WebText data and GPT-2 output text. The performance of a trained detector varies from the scale of the text generator. Specifically, as the model size increases and the model version iterates, the detection effect of AI-generated texts declines a lot.

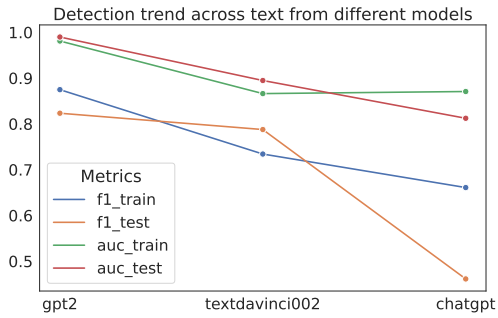

Fig. 3. Detectors' result on the generated text on models of different model scales.

As shown in Figure 3, there is a clear trend of decline in detection accuracy. The F-1 performance on GPT-2 and ChatGPT test data is 82% versus 46%, showing a relative drop of 36%. The possible reason is that output of GPT-2 is not very smooth and may contain grammatical errors. As the model scale increases, the model output is more fluent. Therefore the performance of detectors trained on original text based on GPT-2 will decline on the output of the ChatGPT model.

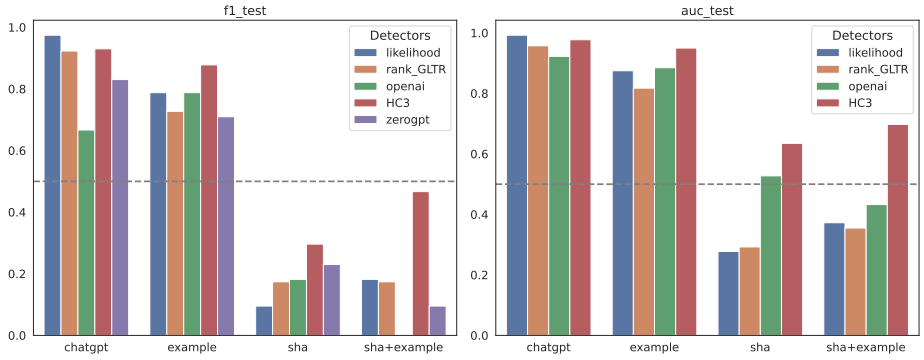

Fig. 4. Detectors' result on QA task with original prompt, prompt with ***Shakespeare*** writing style guidance in author name, author name plus example, and mere example format.

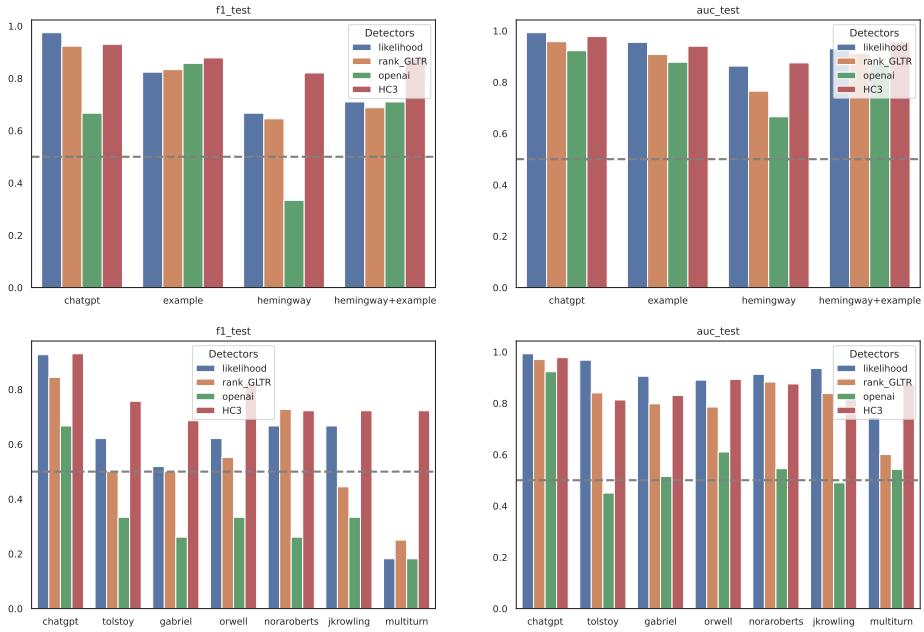

Fig. 5. The upper subfigures: Detectors' result on QA task with original prompt, prompt with ***Hemingway*** writing style guidance in author name, author name plus example and mere example format. The lower subfigures: Detectors' result on QA task with prompts of different authors and multi-turn prompt designed for evasion.

4.2 Detectability Analysis of Different Writing Styles of AI-generated Text

QA task: After adding a writing style to the generated text, it becomes undetectable. When writing style is explicitly appointed by an author name, the

detectors fall under a random classifier, as shown in Figure 4. The Roberta detector and ZeroGPT detector get worse when the author name and example writing style text are explicitly referred to. The one-shot example prompt without a specified author name only weakens detectors a little.

While changing writing styles, we find that different styles have different effect levels. Although some writing styles are weaker than others, it could still weaken the detectors to state that they are not usable. As shown in Figure 5, the F-1 value of Roberta-base-openai-detector also falls under 50%.

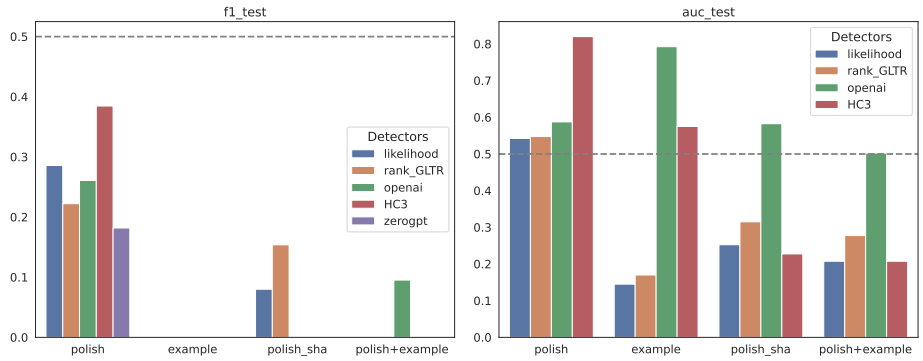

Fig. 6. Detectors' result on Polish task with original prompt, prompt with ***Shakespeare*** writing style guidance in author name, author name plus example and mere example format.

Polish Task: The metrics-based detectors, deep-learning-based detectors, and off-the-shelf online detectors all fail on the polish task. In Figure 6, their F1 values are under 40% which means they are unusable. Adding writing styles also causes a decline in detection. This result is in line with another research[24].

5 Conclusion

In previous work, the anti-detection methods neglected the effect of prompt engineering on detection methods and We analyze the detectability of AI-generated text under different model scales and prompts. Easily and cheaply, the detectors of the time are easily evaded by prompts added writing styles or evasion concepts. This demonstrates the vulnerability of detectors.

Although this research is limited by the number of tasks, our method is easy to replicate even for anyone who has access to a LLM. Detection of AI-generated text is still an unsolved problem. This work aims to highlight the potential dangers of misusing LLMs. From a practical and empirical aspect, AI-generated text could not even be detected sufficiently for now. As we wrote this

manuscript, OpenAI has quietly shut down its online detection tool. Thus, we highly suspect the usage of popular online detectors nowadays and call for robust and sufficient methods to face the challenge of potential misuse of LLMs.

References

1. AI Content Detector, https://crossplag.com/ai-content-detector/
2. AI Detector (GPT / ChatGPT / Claude) | Sapling, https://sapling.ai/ai-content-detector
3. GPTZero | The Trusted AI Detector for ChatGPT, GPT-4, & More, https://gptzero.me/
4. New AI classifier for indicating AI-written text, https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text
5. Originality.ai, https://app.originality.ai/api-access
6. ZeroGPT - Accurate Chat GPT, GPT4 & AI Text Detector Tool, https://www.zerogpt.com/
7. Cai, S., Cui, W.: Evade ChatGPT Detectors via A Single Space (Jul 2023). https://doi.org/10.48550/arXiv.2307.02599, http://arxiv.org/abs/2307.02599
8. Casal, J.E., Kessler, M.: Can linguists distinguish between ChatGPT/AI and human writing?: A study of research ethics and academic publishing. Research Methods in Applied Linguistics **2**(3), 100068 (2023). https://doi.org/https://doi.org/10.1016/j.rmal.2023.100068, https://www.sciencedirect.com/science/article/pii/S2772766123000289
9. Chakraborty, S., Bedi, A.S., Zhu, S., An, B., Manocha, D., Huang, F.: On the Possibilities of AI-Generated Text Detection (Jun 2023). https://doi.org/10.48550/arXiv.2304.04736, http://arxiv.org/abs/2304.04736
10. Chris: The Ultimate ChatGPT Prompt: Content that Outsmarts AI Detectors with 99% Accuracy (Mar 2023), https://medium.datadriveninvestor.com/the-ultimate-chatgpt-prompt-content-that-outsmarts-ai-detectors-with-99-accuracy-ef20d81582bb
11. Christ, M., Gunn, S., Zamir, O.: Undetectable Watermarks for Language Models (May 2023). https://doi.org/10.48550/arXiv.2306.09194, http://arxiv.org/abs/2306.09194
12. Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., Smith, N.A.: All that's 'human' is not gold: Evaluating human evaluation of generated text. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 7282–7296. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.565, https://aclanthology.org/2021.acl-long.565
13. Fu, Y., Xiong, D., Dong, Y.: Watermarking Conditional Text Generation for AI Detection: Unveiling Challenges and a Semantic-Aware Watermark Remedy (Jul 2023). https://doi.org/10.48550/arXiv.2307.13808, http://arxiv.org/abs/2307.13808
14. Gehrmann, S., Strobelt, H., Rush, A.: GLTR: Statistical detection and visualization of generated text. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 111–116. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-3019, https://aclanthology.org/P19-3019

15. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., Wu, Y.: How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection (Jan 2023). https://doi.org/10.48550/arXiv.2301.07597, http://arxiv.org/abs/2301.07597
16. He, X., Shen, X., Chen, Z., Backes, M., Zhang, Y.: MGTBench: Benchmarking Machine-Generated Text Detection (Jun 2023). https://doi.org/10.48550/arXiv.2303.14822, http://arxiv.org/abs/2303.14822
17. Hu, X., Chen, P.Y., Ho, T.Y.: RADAR: Robust AI-Text Detection via Adversarial Learning (Jul 2023). https://doi.org/10.48550/arXiv.2307.03838, http://arxiv.org/abs/2307.03838
18. Ippolito, D., Duckworth, D., Callison-Burch, C., Eck, D.: Automatic detection of generated text is easiest when humans are fooled. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1808–1822. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.164, https://aclanthology.org/2020.acl-main.164
19. Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T.: A watermark for large language models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 17061–17084. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/kirchenbauer23a.html
20. Krishna, K., Song, Y., Karpinska, M., Wieting, J., Iyyer, M.: Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense (Mar 2023). https://doi.org/10.48550/arXiv.2303.13408, http://arxiv.org/abs/2303.13408
21. Kuditipudi, R., Thickstun, J., Hashimoto, T., Liang, P.: Robust Distortion-free Watermarks for Language Models (Jul 2023). https://doi.org/10.48550/arXiv.2307.15593, http://arxiv.org/abs/2307.15593
22. Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., Zou, J.: GPT detectors are biased against non-native English writers (Jul 2023). https://doi.org/10.48550/arXiv.2304.02819, http://arxiv.org/abs/2304.02819
23. Liu, Y., Zhang, Z., Zhang, W., Yue, S., Zhao, X., Cheng, X., Zhang, Y., Hu, H.: ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models (Apr 2023). https://doi.org/10.48550/arXiv.2304.07666, http://arxiv.org/abs/2304.07666
24. Liu, Z., Yao, Z., Li, F., Luo, B.: Check Me If You Can: Detecting ChatGPT-Generated Academic Writing using CheckGPT (Jun 2023). https://doi.org/10.48550/arXiv.2306.05524, http://arxiv.org/abs/2306.05524
25. Lu, N., Liu, S., He, R., Wang, Q., Tang, K.: Large Language Models can be Guided to Evade AI-Generated Text Detection (Jun 2023). https://doi.org/10.48550/arXiv.2305.10847, http://arxiv.org/abs/2305.10847
26. Ma, Y., Liu, J., Yi, F., Cheng, Q., Huang, Y., Lu, W., Liu, X.: AI vs. Human – Differentiation Analysis of Scientific Content Generation (Feb 2023). https://doi.org/10.48550/arXiv.2301.10416, http://arxiv.org/abs/2301.10416
27. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., Finn, C.: DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature (Jul 2023). https://doi.org/10.48550/arXiv.2301.11305, http://arxiv.org/abs/2301.11305
28. Pan, Y., Pan, L., Chen, W., Nakov, P., Kan, M.Y., Wang, W.Y.: On the Risk of Misinformation Pollution with Large Language Models (May 2023). https://doi.org/10.48550/arXiv.2305.13661, http://arxiv.org/abs/2305.13661

29. Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W., Feizi, S.: Can AI-Generated Text be Reliably Detected? (Jun 2023), http://arxiv.org/abs/2303.11156
30. Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J.W., Kreps, S., McCain, M., Newhouse, A., Blazakis, J., McGuffie, K., Wang, J.: Release Strategies and the Social Impacts of Language Models (Nov 2019). https://doi.org/10.48550/arXiv.1908.09203, http://arxiv.org/abs/1908.09203
31. Su, J., Zhuo, T.Y., Wang, D., Nakov, P.: DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text (May 2023). https://doi.org/10.48550/arXiv.2306.05540, http://arxiv.org/abs/2306.05540
32. Tang, L., Uberti, G., Shlomi, T.: Baselines for Identifying Watermarked Large Language Models (May 2023). https://doi.org/10.48550/arXiv.2305.18456, http://arxiv.org/abs/2305.18456
33. Tang, R., Chuang, Y.N., Hu, X.: The Science of Detecting LLM-Generated Texts (Jun 2023). https://doi.org/10.48550/arXiv.2303.07205, http://arxiv.org/abs/2303.07205
34. Tian, Y., Chen, H., Wang, X., Bai, Z., Zhang, Q., Li, R., Xu, C., Wang, Y.: Multiscale Positive-Unlabeled Detection of AI-Generated Texts (Jun 2023). https://doi.org/10.48550/arXiv.2305.18149, http://arxiv.org/abs/2305.18149, arXiv:2305.18149 [cs]
35. Tulchinskii, E., Kuznetsov, K., Kushnareva, L., Cherniavskii, D., Barannikov, S., Piontkovskaya, I., Nikolenko, S., Burnaev, E.: Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts (Jun 2023). https://doi.org/10.48550/arXiv.2306.04723, http://arxiv.org/abs/2306.04723
36. Wang, H., Luo, X., Wang, W., Yan, X.: Bot or Human? Detecting ChatGPT Imposters with A Single Question (May 2023). https://doi.org/10.48550/arXiv.2305.06424, http://arxiv.org/abs/2305.06424
37. Wang, Y., Mansurov, J., Ivanov, P., Su, J., Shelmanov, A., Tsvigun, A., Whitehouse, C., Afzal, O.M., Mahmoud, T., Aji, A.F., Nakov, P.: M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection (May 2023). https://doi.org/10.48550/arXiv.2305.14902, http://arxiv.org/abs/2305.14902
38. Wu, J., Yang, S., Zhan, R., Yuan, Y., Wong, D.F., Chao, L.S.: A survey on llm-generated text detection: Necessity, methods, and future directions (2023)
39. Yang, X., Cheng, W., Petzold, L., Wang, W.Y., Chen, H.: DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text (May 2023). https://doi.org/10.48550/arXiv.2305.17359, http://arxiv.org/abs/2305.17359
40. Zhao, X., Ananth, P., Li, L., Wang, Y.X.: Provable Robust Watermarking for AI-Generated Text (Jun 2023). https://doi.org/10.48550/arXiv.2306.17439, http://arxiv.org/abs/2306.17439
41. Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Gong, N.Z., Zhang, Y., Xie, X.: PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts (Aug 2023). https://doi.org/10.48550/arXiv.2306.04528, http://arxiv.org/abs/2306.04528

Appendix A The prompts used for QA task

[chatgptanswers] As a highly intelligent question answering bot, your task is to answer questions. You will be provided with the question delimited by triple

backticks. Format the output in a JSON object with the following keys: question,answer.
“‘question“‘

[chatgptanswers_sha] As a highly intelligent question answering bot, your task is to answer questions in specific writing styles. Write in the style of Shakespeare. You will be provided with the question delimited by triple backticks. Format the output in a JSON object with the following keys: question,answer.
“‘question“‘

[chatgptanswers_sha_withexampletext] As a highly intelligent question answering bot, your task is to answer questions in specific writing styles. Write in the style of Shakespeare.You will be provided with an example of Shakespeare's writing style delimited by triple dashes. You will be provided with the question delimited by triple backticks. Format the output in a JSON object with the following keys: question,answer,style_example.
style_example: —example—
question: “‘question“‘

[chatgptanswers_sha_onlyexampletext] As a highly intelligent question answering bot, your task is to answer questions in specific writing styles. Write in the writing style of an example but ignore the content and topic of the example. You will be provided with the style example delimited by triple quotes. You will be provided with the question delimited by triple backticks. Format the output in a JSON object with the following keys: question,answer,style_example
style_example: """example"""
question: “‘question“‘

example = From what power hast thou this powerful might, With insufficiency my heart to sway, To make me give the lie to my true sight, And swear that brightness doth not grace the day? Whence hast thou this becoming of things ill, That in the very refuse of thy deeds, There is such strength and warrantise of skill, That in my mind thy worst all best exceeds? Who taught thee how to make me love thee more, The more I hear and see just cause of hate? O though I love what others do abhor, With others thou shouldst not abhor my state. If thy unworthiness raised love in me, More worthy I to be beloved of thee.

For other writing styles, we change the author name and example text in the prompts.

example_hemingway =He no longer dreamed of storms, nor of women, nor of great occurrences, nor of great fish, nor fights, nor contests of strength, nor of his wife. He only dreamed of places now and of the lions on the beach. They played like young cats in the dusk and he loved them as he loved the boy. He always thought of the sea as 'la mar' which is what people call her in Spanish when they love her. Sometimes those who love her say bad things of her but they are always said as though she were a woman. Some of the younger fishermen, those who used buoys as floats for their lines and had motorboats, bought when the shark livers had brought much money, spoke of her as 'el mar' which is masculine.They spoke of her as a contestant or a place or even an enemy. But the old man always thought of her as feminine and as something that gave or withheld great favours,

and if she did wild or wicked things it was because she could not help them. The moon affects her as it does a woman, he thought. Fish," he said softly, aloud, "I'll stay with you until I am dead. No one should be alone in their old age, he thought. Fish," he said, "I love you and respect you very much. But I will kill you dead before this day ends.

Appendix B The prompts used for Polish Task

[polish] As an academic paper writer, your task is to rewrite an abstract of a research paper. You will be provided with the abstract delimited by triple backticks. Format the output in a JSON object with the following keys:
original_abstract,rewritten_abstract.
original_abstract :"'ab"'

[p_sha] As an academic paper writer, your task is to rewrite an abstract of a research paper in specific writing styles. Write in the style of Shakespeare.You will be provided with the original abstract delimited by triple backticks. Format the output in a JSON object with the following keys: original_abstract,rewritten_abstract.
"'ab"'

[p_withtext] As an academic paper writer, your task is to rewrite an abstract of a research paper in specific writing styles. Write in the style of Shakespeare. You will be provided with an example of Shakespeare's writing style delimited by triple quotes. You will be provided with the original abstract delimited by triple backticks. Format the output in a JSON object with the following keys: original_abstract,rewritten_abstract,style_example.
style_example: """{example_shakespeare}"""
original_abstract :"'original_abstract"'

[p_onlytext] As an academic paper writer, your task is to rewrite an abstract of a research paper in specific writing styles. Write in the writing style of an example but ignore the content and topic of the example. You will be provided with style example delimited by triple quotes. You will be provided with the original abstract delimited by triple backticks. Format the output in a JSON object with the following keys: original_abstract,rewritten_abstract,style_example.
style_example: """example_shakespeare"""
original_abstract :"'original_abstract"'

The example text of Shakespeare is the same as that in the QA task.

Appendix C Multi-turn prompt used for detectors evasion

[{"role": "user", "content": "As a highly intelligent question answering bot, your task is to rewrite the answer of a question. I'm going to give you some information. You will be provided with instructions, the question, the answer delimited by triple quotes."},{"role": "assistant", "content": "Sure, I'm here to help! Please provide me with the information, instructions, the question, and the

answer delimited by triple quotes, and I'll be happy to rewrite the answer for you."},{"role": "user", "content":"When it comes to writing content, two factors are crucial, "perplexity" and "burstiness." Perplexity measures the complexity of text. Separately, burstiness compares the variations of sentences. Humans tend to write with greater burstiness, for example, with some longer or complex sentences alongside shorter ones. AI sentences tend to be more uniform. Therefore, when writing the following content I am going to ask you to create, I need it to have a good amount of perplexity and burstiness. Do you understand?"},{"role": "assistant", "content":"Yes, I understand. You would like the content I generate to exhibit both perplexity, which refers to the complexity of the text, and burstiness, which involves variations in sentence length and complexity. By incorporating these elements, the content should resemble human writing with a mix of longer or complex sentences alongside shorter ones. Please provide me with the specific instructions, the question, and any additional details you'd like to include in the content. I'll make sure to generate a response that meets your criteria for perplexity and burstiness."},{"role": "user", "content": Prompt}]

Prompt = Using the concepts written previously, rewrite this answer of the question with a high degree of perplexity and burstiness:
question:"""{question}"""
answer:"""{answer}"""
Format the output in a JSON object with the following keys:
question,answer,rewritten_answer.