Enhancing Robustness of LLM-Synthetic Text Detectors for Academic Writing: A Comprehensive Analysis

Anonymous ACL submission

Abstract

The emergence of large language models (LLMs), such as Generative Pre-trained Transformer 4 (GPT-4) used by ChatGPT, has profoundly impacted the academic and broader community. While these models offer numerous advantages in revolutionizing work and study methods, they have also garnered significant attention due to their potential negative consequences. One example is generating academic reports or papers without or with a limited human contribution. Consequently, researchers have focused on developing detectors to address the misuse of LLMs. However, most existing works prioritize achieving higher accuracy on restricted datasets, neglecting the crucial aspect of generalizability. This limitation hinders their practical application in real-life scenarios where reliability is paramount. In this paper, we present a comprehensive analysis of the influence of prompts on the text generated by LLMs and highlight the potential lack of robustness in one of the current state-of-the-art GPT detectors. To mitigate these issues concerning the misuse of LLMs in academic writing, we propose a reference-based Siamese detector taking a pair of texts: one as the inquiry and the other as the reference. Our method effectively addresses the lack of robustness and significantly improves the baseline performances in challenging scenarios, increasing them by approximately 25% to 67%.

1 Introduction

Recently, the applications of large-scale language models, such as Open AI's GPT-4 (OpenAI, 2023), or Google's Pathways Language Model 2 (Anil et al., 2023), have become an integral part of people's lives and works, often being utilized unconsciously. From casual conversations with chatbots to accurately expressing search queries on search engines and relying on models like ChatGPT for writing assistants, LLMs have gained widespread usage due to their powerful performance. This extensive application potential has attracted numerous companies to leverage LLMs for optimizing their services. However, while LLMs greatly facilitate daily activities, they pose significant security risks if maliciously exploited for attacks or deceptions. Consequently, with the growing popularity of LLMs, the importance of AI security has come to the forefront of people's attention.

Among the various security concerns, academic cheating stands out as a particularly grave issue. Within academia, universities face the most severe challenges in this regard. University students possess the necessary expertise to leverage LLMs effectively, and they frequently encounter writing tasks such as papers, assignments, and examinations. ChatGPT, in particular, has gained widespread popularity among college students worldwide. Consequently, universities urgently need robust detectors to address this issue, which has driven continuous advancements in the field of detection technology.

Research on detectors in this field can be broadly categorized into two directions. The first approach involves expanding the machine text corpus and enhancing the detector's performance using diverse training data. The second approach focuses on designing novel detector structures to improve overall performance. Both directions have yielded notable results, with detectors showcasing good performance on limited test sets in their respective research papers.

The versatility of LLMs, including the GPT family, enables students to exploit various prompts for academic cheating, thereby undermining detectors' effectiveness. However, achieving high performance solely on limited test sets falls short of adequately addressing real-world challenges. There is a pressing need to evaluate the robustness of models across a broader range of prompts and test sets, an aspect that has been largely overlooked in existing studies.

Our paper makes three contributions:

- **Highlighting the insufficient robustness of existing detectors through the example of academic writing cheating:** We demonstrate that solely adjusting the prompt is inadequate for ensuring the robustness of current detectors, particularly in the context of academic writing cheating.

- **Introducing a new detection approach for academic writing cheating using a Siamese network:** We analyze the academic writing cheating scenario and propose a novel detection approach based on a Siamese network. Our model exhibits superior prompt generalization capabilities compared to existing detectors, effectively addressing the issue of insufficient robustness when confronted with specific prompts.

- **Exploring the prompt-induced lack of robustness and evaluating model applicability:** We put forward a hypothesis to explain the reasons behind the lack of robustness caused by the prompt and provide evidence to support our claims. Furthermore, we demonstrate the broad applicability of our model based on this hypothesis.

The rest of the paper is organized as follows: Firstly, we review the literature and evaluate the existing detector's lack of robustness in detecting academic cheating. Subsequently, we conduct an in-depth analysis of the academic cheating scenario, leading us to propose a new network specifically designed to address the robustness issue. Finally, we put forward a hypothesis regarding the factors contributing to the lack of robustness in generated articles, supported by our experimental evidence.

2 Related Work

With the popularity of LLM, many studies have explored the security problems that LLM may bring in recent years. Evan et al. conducted a comprehensive investigation into the potential security issues posed by LLMs and provided an overview of existing detection systems (Crothers et al., 2023). Stiff et al. analyzed the possible disinformation of false texts, tested the text on multiple platforms using the RoBERTa model, and analyzed whether the existing detection technology can detect the existing generated text (Stiff and Johansson, 2022). Greshake, et al. pointed out that many applications now integrate LLMs as part of their functions (Greshake et al., 2023). However, LLMs may be affected by the input. If an attacker designs malicious input to mislead LLMs, it will likely cause data leakage and other security problems.

Researchers in detector development have explored strategies to optimize the training set for improved model performance. Notably, Liyanage et al. pioneered an AI-generated academic dataset using GPT-2, although it is considered inferior to the more advanced ChatGPT model currently available (Liyanage et al., 2022). Yuan et al. proposed BERTscore, a novel evaluation method for filtering high-quality generated text that closely resembles human writing (Yuan et al., 2021). Such text can be incorporated into the training set, thereby enhancing the performance of the detectors.

Researchers have also focused on optimizing the model itself. Jawahar et al. addressed the challenge of hybrid text, introducing a method to detect the boundary between machine-generated and human-written content, rather than solely distinguishing between the two (Jawahar et al., 2020). Zhao et al. conducted a comprehensive survey of various LLMs, analyzing their performance across multiple dimensions, including pre-training, adaptation tuning, utilization, and capacity evaluation. They also identified potential future development directions for LLMs (Zhao et al., 2023). Additionally, Mitchell et al. proposed a novel model utilizing a curvature-based criterion to determine whether a given passage was generated by an LLM (Mitchell et al., 2023).

Studies have also examined the robustness of detectors. Rodriguez et al. investigated the impact of dataset domain on detector performance, highlighting a significant decrease in performance when the training and test datasets differ in domain (Rodriguez et al., 2022). Their findings emphasized how the diversity of training sets directly affects the detector's performance. Pu et al. analyzed the issue of insufficient robustness in existing detection systems by exploring changes in decoding or text sampling strategies (Pu et al., 2022). While previous research focused on robustness in terms of dataset domains and generative models' parameters, this study highlights that prompt adjustments alone can significantly affect the robustness of the detector, particularly in the context of academic

Simple prompt: Write an abstract for a paper.

Specific prompt: Write an abstract for a paper about +

Human-writen paper's title
Universal Metrics for Large-scale Performance Analysis of Deep Neural Network.
Algorithms and Complexity of Range Clustering.
⋮

Figure 1: Examples of a simple prompt and a specific prompt.

cheating. The subsequent section will provide a demonstration of this phenomenon.

3 Asserting the Limitation of Existing Detectors

We conducted a simple preliminary test to highlight the prompt-induced limitations of a state-of-the-art AI-generated text detector.

3.1 Dataset Construction

Since the release of GPT-3, OpenAI has allowed users to provide input prompts to shape the output text, enabling a wide range of functionalities. This inclusion of prompts significantly enhances the variation in generated text, presenting a more significant challenge for detection tasks. In contrast, the previous model, GPT-2, lacks prompt functionality and is irrelevant to the robustness of prompt-related issues. ChatGPT, a question-answering platform, does not offer APIs or adjustable parameters, making it unsuitable for generating large-scale datasets with diverse outputs. Hence, this paper uses GPT-3 for dataset generation, serving as the benchmark for our measurements. It is essential to clarify that throughout this paper, the term "GPT model" specifically refers to GPT-3.

For the human-written part of the dataset, we obtained the real human paper abstracts by collecting 500 samples from the arXiv dataset (Clement et al., 2019), which is available on Kaggle[1] and covers various fields. To create the AI-generated part of the dataset, we divided it into two subsets as depicted in Fig. 1. The "Simple prompt" subset consists of 500 GPT abstracts generated by GPT-3 using the prompt "Write an abstract for a professional paper." The "Specific prompt" subset includes 500 GPT abstracts generated by GPT-3 using prompts beginning with "Write an abstract for a paper about" followed by the corresponding titles from the real human abstracts.

[1] https://www.kaggle.com/datasets/Cornell-University/arxiv

3.2 Detector Benchmark

Among the state-of-the-art detectors available, such as ChatGPT detector and GPTZero, many lack associated published articles or datasets for reproduction. Moreover, a significant number of these detectors do not provide APIs, making it impossible to conduct batch-testing experiments. Consequently, we have chosen the RoBERTa base OpenAI Detector (OpenAI detector in short) on Hugging Face[2], a single-input binary classifier, as our target detector due to its availability and usability.

The detector demonstrates an impressive accuracy of 98% in detecting abstracts generated by **simple prompts** and 98% in identifying human-written abstracts. However, when it comes to abstracts generated by **specific prompts**, the accuracy rate **drops to only 87%**. This substantial reduction in performance by simply adding a human-written sentence to the prompt clearly indicates the limited robustness of existing detectors. Notably, specific prompts are commonly used in academic cheating scenarios, where students tailor their assignments or reports to meet specific requirements provided by their professors, utilizing prompts similar to the specific prompts used in this study. An example of the abstract generated using the corresponding title that was misclassified by the OpenAI detector is shown in Tab. 4 in the Appendix.

4 Our Solution

Our solution consists of two key components. Firstly, we analyze potential academic cheating scenarios and develop a **cheating model** specifically tailored to address these instances of cheating. Secondly, we propose a **novel detection system** designed to identify instances of academic cheating based on our developed model.

[2] https://huggingface.co/roberta-base-openai-detector

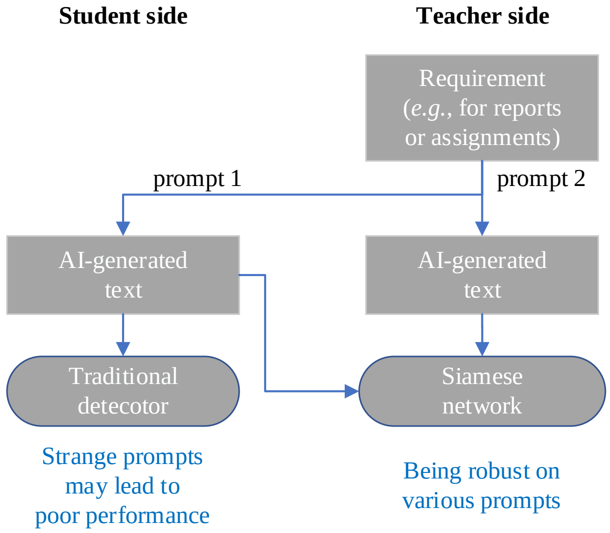

Figure 2: The proposed student cheating model.

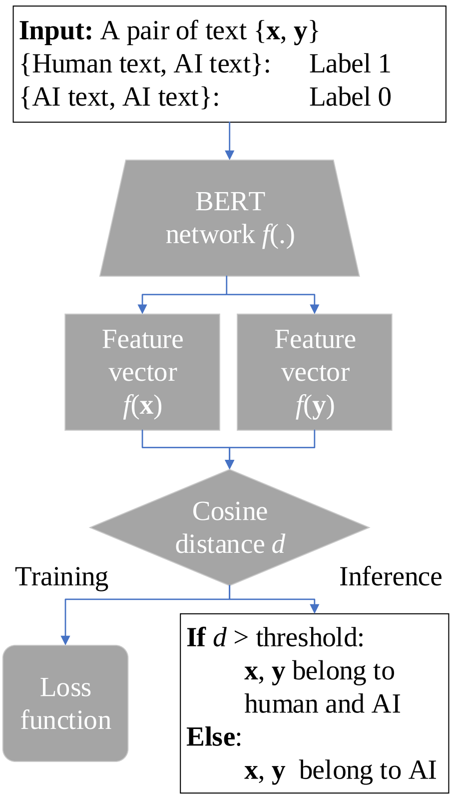

Figure 3: Overview of the proposed detector network.

4.1 Student Cheating Model

As depicted in Fig.3, the model comprises two parties: the student side and the teacher side. Initially, the teacher assigns specific requirements for an academic task. Subsequently, a potentially deceitful student utilizes these requirements as input to generate an article using a generative model, such as GPT-3. The student may customize the provided requirements to evade detection, as discussed in Section3 with specific prompts. On the other hand, the teacher also proactively employs the generative model to generate an article. Then, the teacher uses a model to compare the similarities between the student's submission and their own generated text in terms of content and style to determine whether the student engaged in cheating.

This cheating model closely resembles real-life situations where students' assignments or examination articles are typically centered around specific topics and come with detailed requirements from teachers. To meet these requirements, students generally use the teacher's instructions as input for generating their articles. Any slight modifications to the requirements or using different seeds for the generative model have minimal impact on the cheating model.

4.2 Detection System

The network structure, as depicted in Fig. 3, involves the input of two articles: **x** and **y**. The article **y** represents the teacher's AI-generated article, while **x** can either be a human-written article or an AI-generated one submitted by the student.

Our detector employs a pre-trained BERT network as a feature extractor, denoted as $f(.)$, which is initialized with pre-trained weights. We fine-tune it using a supervised training approach. During the labeling of training data, if both **x** and **y** represent AI-generated articles, the label l is assigned as 0. Conversely, if **x** corresponds to a human-written article and **y** represents an AI-generated article, the label l is set as 1.

We use cosine distance $\delta(.,.)$ for measuring the similarity between two feature vectors $\mathbf{f_x} = f(\mathbf{x})$ and $\mathbf{f_y} = f(\mathbf{y})$, described in Eq. 1.

$$\delta(\mathbf{f_x}, \mathbf{f_y}) = 1 - \frac{\mathbf{f_x} \cdot \mathbf{f_y}}{\|\mathbf{f_x}\|_2 \|\mathbf{f_y}\|_2} \quad (1)$$

The loss function utilized during training is described by Eq. 2.

$$\mathcal{L} = l\delta(\mathbf{f_x}, \mathbf{f_y})^2 + (1 - l)(2 - \delta(\mathbf{f_x}, \mathbf{f_y}))^2 \quad (2)$$

During the inference phase, our model calculates the cosine distance between the two input texts. A smaller distance indicates a higher similarity between **x** and **y**. As **y** represents AI-generated text, a smaller distance suggests that **x** is more likely to be generated by AI. Conversely, **x** is more likely to be written by a human or contain a significant human contribution.

Table 1: Accuracy of the detectors on the prompt-generalization test set with level-n prompts.

Prompt variant	**Prompt content**	**OpenAI detector (original)**	**OpenAI detector (fine-tuned)**	**Proposed detector**
Human text		100%	98%	92%
Directly use requirement	Write an abstract for a paper about X	11%	35%	85%
Another expression	If you are a student, please complete the abstract of the article assigned by the teacher with topic X.	17%	46%	71%
Double GPT	Revise X then write an abstract about the revised text.	7%	15%	81%
Many $\rightarrow$ one	Find five human abstracts about X then summarize them into one.	11%	14%	81%

5 Experiment Results and Discussions

5.1 Experimental Design

We conducted experiments following similar settings as described in Section 3, but with an expanded dataset as outlined below:

- We utilized various levels of **specific prompts** in four different variants, as illustrated in Fig. 4 (and exemplified in Tab. 5 in the Appendix).

- The **training set** consisted of 2,000 human-written abstracts and 4,000 GPT-3 generated texts using **level-1** specific prompts. This dataset was employed for fine-tuning the detectors.

- For the **prompt-generalization test set**, we selected 100 human-written abstracts and generated 100 abstracts per each prompt variant that mimics different manipulative behaviors students may employ with **level n**. The prompt variants include *"Directly use requirement,"* which is the specific prompt we designed before. *"Another expression"* is where the student expresses the meaning of the requirement using different wording. The *"Double GPT"* variant involves using the generative model (GPT) twice, where the student modifies the original human idea X using GPT before generating the article. Lastly, the *"Many $\rightarrow$ one"* variant simulates a common plagiarism method where the student collects five human articles about human idea X and combines them into a new article. These prompt variants allow us to evaluate the detector's performance in detecting different manipulative strategies employed by students. Examples of each variant are shown in Fig. 4 and Tab. 1.

- We extended the prompt-generalization test set to form the **human-contribution test set** by incorporating different levels of human contribution. Each variant in the test set represents a different level of human's involvement in the generated text. The levels range from including only the field name, to including the title, summary of the abstract, and finally the entire abstract, denoted as 0, 1, 2, and n, respectively. By incorporating varying degrees of human contribution, we aim to assess the detector's ability to distinguish between AI-generated text with different levels of human involvement. Examples of each level are shown in Fig. 4 (and Tab. 5 in the Appendix).

- For the **domain-generalization test set**, we chose 50 human-written abstracts and generated 50 abstracts per each generative model comprising OpenAI's GPT-3, Perplexity's customized GPT-3.5[3], and the Falcon-7B[4]. All abstracts were generated using **level-1** and **level-2** prompts. This test set enables us to assess the detectors' ability to generalize across different generative models, providing insights into their performance and adaptability in diverse AI-generated text scenarios.

[3]https://www.perplexity.ai/
[4]https://falconllm.tii.ae/

Table 2: Accuracy of the original OpenAI detector (before fine-tuning) in different levels. X denotes the human-written content incorporated into the prompts.

X level	Directly use requirement	Another expression	Double GPT	Many → one
level 0 (X = Field name)	100%	100%	99%	86%
level 1 (X = Title)	70%	74%	53%	72%
level 2 (X = Summary of abstract)	34%	24%	20%	29%
level n (X = Entire abstract)	11%	17%	7%	11%

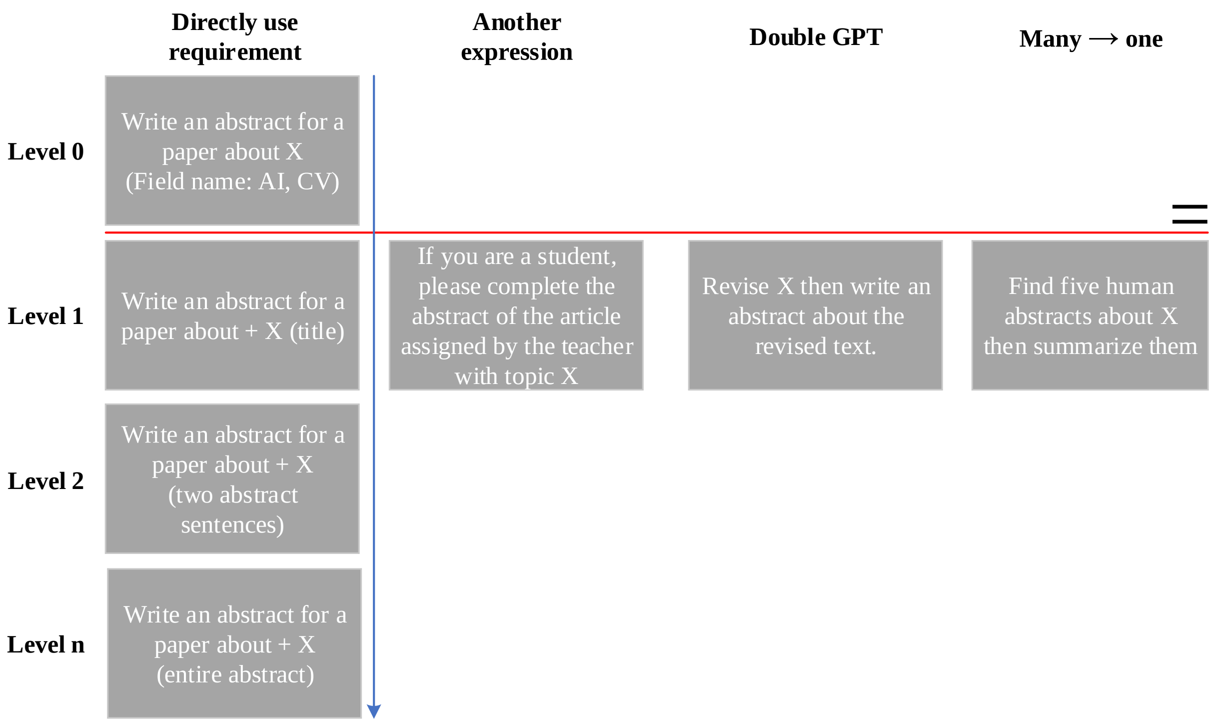

Figure 4: The prompts can be categorized into different levels based on the degree of human-written content. The horizontal red line indicates that prompts within the same level share similar characteristics. The vertical blue arrow illustrates that the generated articles become more challenging to classify accurately as the level increases.

Regarding the classification threshold (cosine distance) employed by our detector, we have empirically set it at 0.8. This threshold strikes a balance between the false rejection rate and the false acceptance rate across various scenarios. However, it is important to note that users have the flexibility to adjust this threshold based on their individual use cases and specific requirements.

5.2 Prompt-Variant Generalizability

We utilized the prompt-generalization test set to assess the detectors' performance in detecting various prompt variants. As presented in Table 1, the OpenAI detector exhibited a significant drop in performance on different variants of prompts level n (the most extreme cases), even after fine-tuning, with a maximum true positive rate (TPR) of only 46%. In contrast, our model demonstrated superior generalizability, achieving a minimum TPR of 71% on the "another expression" specific prompts. This implies that in academic cheating scenarios, our model can effectively detect the usage of GPT by students, regardless of the complexity of the professor's requirements and the inclusion of a certain amount of human-written content in the prompts (approximately 200 to 250 words as an abstract).

In terms of the true negative rate (TNR), which evaluates the detectors' capability to accurately identify human-written text, our detector achieved a commendable accuracy of 92%. Although this is slightly lower than the fine-tuned OpenAI detector (98%) and its original version (100%), it is a reasonable trade-off considering the decrease in the TPRs of the OpenAI detector. Furthermore,

Table 3: Accuracy (or TPR) of the detectors on the text generated by different LLMs. OpenAI detector, a binary classifier, only needs one input. Our detector, besides the query text, requires the corresponding generated text (from the teacher) as an anchor. Within each cell, the upper number represents the result on level-1 prompts, while the lower number represents the result on level-2 prompts.

Source of input text	**OpenAI detector (original))**	**OpenAI detector (fine-tuned)**	**Proposed detector**		
			GPT-3 text as anchor	**Falcon-7B text as anchor**	**Perplexity text as anchor**
Human	100% 100%	98% 98%	92% 92%	70% 12%	90% 90%
GPT-3	70% 35%	99% 98%	95% 100%		
Falcon-7B	16% 13%	92% 57%	60% 12%	70% 96%	
Perplexity	47% 53%	98% 98%	100% 70%		100% 100%

users have the flexibility to adjust the classification threshold according to their specific use cases and requirements.

To investigate the drop in the OpenAI detectors' performance, we examined the impact of reducing human contribution in prompts using the human-contribution test set. Results in Table 2 showed that the original OpenAI detector ideally detected AI-generated text with level-0 prompts, except for "many $\rightarrow$ one" prompts (86% accuracy). Performance remained acceptable at level 1 but deteriorated significantly at level 2 and beyond. This is unacceptable in real-life scenarios where malicious students may strategically add additional keywords or phrases to make their generated text more convincing and harder to detect.

5.3 Domain Generalizability

Although GPT has become mainstream, students may utilize several other text-generation models based on LLMs to avoid detection. To assess the detectors' effectiveness, we conducted tests using the domain-generalization test set. It is important to note that all detectors were fine-tuned solely using GPT-3 generated text.

The results are presented in Table 3. The original OpenAI detectors struggled to perform effectively in most cases, while its fine-tuned version achieved the highest accuracies except when dealing with text generated by Falcon-7B using level-2 prompts. Our proposed detector performed highly on the GPT variants (GPT-3 and customized GPT-3.5). However, it showed limited generalizability when faced with Falcon's generated text using anchor text generated by other LLMs.

We hypothesized that during the training of our Siamese-based detector with the proposed cheating model, the detector learned to identify authorship information. It distinguished GPT as one author and humans as another. When a new "author" (Falcon-7B) emerged, the detector struggled to assign its text to either the human or GPT. When using asymmetric pairs as input, the scores fell around the decision threshold, leading to degraded performance. Conversely, when using pairs of Falcon-7B's text, the detector treated them as originating from the same author, resulting in improved accuracy.

To improve the inter-model generalizability of our detector, teachers can select representative models from popular LLM families such as GPT, LLaMA (Touvron et al., 2023), and Falcon to create multiple anchor texts for multiple comparisons.

6 Hypothesis for the Prompt-Induced Lack of Robustness

As demonstrated in the previous section, traditional detectors exhibit limited robustness due to the infinite possibilities of prompts. While we evaluated specific prompts related to academic cheating, it is crucial to acknowledge that the prompts we examined cannot encompass the entire spectrum of academic cheating scenarios. To systematically address this issue, we generalized the result in Tab.2 to form a hypothesis that aims to (1) illuminate the potential factors underlying the reduced robustness of traditional detectors and (2) substantiate the generalizability of our chosen of prompts.

Our hypothesis can be illustrated using Figure 4 and can be explained as follows:

- Within the prompt, only the component X, which contains human ideas, influences the characteristics of the generated articles and contributes to the limited robustness observed in existing detectors.
- When the X component remains at a certain level, the generated articles exhibit similar characteristics regardless of the other parts of the prompt.
- As the complexity and level of detail in the X component increase, it becomes more challenging to detect the generated articles.

In summary, the X component of the prompt plays a crucial role in the characteristics of the generated articles and poses challenges for detection, particularly as it becomes more intricate and detailed.

7 Conclusion

This study addresses the issue of academic cheating facilitated by LLMs, which are widely utilized in contemporary contexts. By examining the RoBERTa Base OpenAI Detector as a case study, we identified potential limitations in the robustness of existing detection methods. To tackle this challenge, we formulated a cheating scenario in academic writing and proposed a novel detection approach. Our experimental results conclusively demonstrated that our new detector exhibits superior prompt generalization capabilities compared to the OpenAI detector. Additionally, we conducted an in-depth analysis and presented a hypothesis highlighting the role of human contribution (X factor) in prompts contributing to the detector's lack of robustness. Building upon this principle, we infer that our model has the ability to detect various forms of GPT-generated text, extending beyond the scope of our experimental evaluation in this paper.

Limitations

Given OpenAI's current API charging standards, collecting a substantial amount of the latest GPT articles is time-consuming and costly. As a result, the dataset used in this study is relatively small, and the test results are significantly influenced by randomness. While we have made efforts to analyze the experimental results, the conclusions we can draw are inherently limited, particularly when compared to the billion-level training set of the RoBERTa Base OpenAI Detector. Additionally, we suspect that different prompt variants at the same level may introduce subtle differences, despite generally aligning with our hypothesis. Therefore, we plan to conduct a comprehensive analysis using larger datasets to explore and investigate various possibilities in future research.

In relation to our proposed detector, its accuracy heavily relies on the quality of the anchor article used for inference. If the anchor is too brief or exhibits unusual characteristics due to the inherent randomness of GPT, it can adversely affect the model's accuracy. Additionally, security concerns arise, such as the possibility of tampering or attacks targeting the anchors. Furthermore, our detector encounters challenges in generalizing effectively when the query text and anchor text are generated by distinct LLMs. Hence, in future research, we intend to explore model-agnostic approaches for generating anchor texts and selecting those that best align with the requirements as input.

Ethics Statement

In contrast to previous studies that primarily examine the parameters and training sets of generative models, our research delves into the potential lack of robustness of detectors from the prompt perspective. Unlike the training set of the generator, which is challenging for general users like students to modify, the design of complex prompts is relatively accessible. Therefore, our findings, particularly the insights on prompt-induced detector vulnerabilities, could inadvertently serve as a reference for potential attackers seeking to enhance their evasion techniques. It is crucial to strike a balance between advancing our understanding of detection limitations and safeguarding against misuse of this knowledge.

Acknowledgements

This work was partially supported by [place holder].

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*.

Colin B Clement, Matthew Bierbaum, Kevin P O'Keeffe, and Alexander A Alemi. 2019. On the use of arxiv as a dataset. *arXiv preprint arXiv:1905.00075*.

Evan Crothers, Nathalie Japkowicz, and Herna Viktor. 2023. Machine generated text: A comprehensive survey of threat models and detection methods.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey.

Keenan Jones, Enes Altuncu, Virginia N. L. Franqueira, Yichao Wang, and Shujun Li. 2022. A comprehensive survey of natural language generation advances from the perspective of digital deception.

M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, B. Stein, and M. Potthast. 2021. Overview of the cross-domain authorship verification task at pan 2021. In *CLEF-WN 2021 - Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum*, volume 2936 of *CEUR Workshop Proceedings*, pages 1743–1759. CEUR-WS.

Cyril Labbé and Dominique Labbé. 2012. Duplicate and fake publications in the scientific literature: how many SCIgen papers in computer science? *Scientometrics*, pages 10.1007/s11192–012–0781–y.

Vijini Liyanage, Davide Buscaldi, and Adeline Nazarenko. 2022. A benchmark corpus for the detection of automatically generated text in academic publications.

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

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2022. Deepfake text detection: Limitations and opportunities.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

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.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

A Examples of Human's and AI's Texts and Their Detection.

Tab. 4 shows an example of the abstract generated using the corresponding title that was misclassified by the OpenAI detector. Tab 5 shows examples of prompts with different levels of human-written contents (factor X).

B Scientific Artifacts Detail

This paper adheres strictly to OPENAI's terms of use[5], and no violations have occurred. It is important to note that OPENAI has not provided specific guidelines regarding expected model application scenarios. The human texts used in this paper were sourced from the publicly available arXiv dataset (Clement et al., 2019). No personally identifiable information has been included in the study.

[5] https://openai.com/policies/terms-of-use

Table 4: An example that OpenAI detector misclassified a GPT-generated text as written by humans. Leveraging LIME (Ribeiro et al., 2016), we demonstrate that the OpenAI Detector exhibited high confidence in its classification of the blue sentences as human-written. In contrast, our detector accurately distinguishes between human-generated text and GPT-generated text, correctly classifying both with precision.

GPT Text	**Human Text**
This paper presents a new knowledge selection method for knowledge-grounded conversation generation. This method, called Difference-aware Knowledge Selection (DKS), leverages the difference between a given conversation context and the associated knowledge to determine the most relevant knowledge to use. DKS first computes the semantic similarity between the conversation context and the available knowledge. It then uses a reinforcement learning algorithm to select the knowledge with the highest reward, which is calculated by the semantic similarity and the expected conversation turn difference. Experimental results demonstrate that the DKS method outperforms baseline methods in terms of both response quality and diversity.	In a multi-turn knowledge-grounded dialog, the difference between the knowledge selected at different turns usually provides potential clues to knowledge selection, which has been largely neglected in previous research. In this paper, we propose a difference-aware knowledge selection method. It first computes the difference between the candidate knowledge sentences provided at the current turn and those chosen in the previous turns. Then, the differential information is fused with or disentangled from the contextual information to facilitate final knowledge selection. Automatic, human observational, and interactive evaluation shows that our method is able to select knowledge more accurately and generate more informative responses, significantly outperforming the state-of-the-art baselines. The codes are available at https://github.com/chujiezheng/DiffKS.

Table 5: Examples of human-written contents from different levels of prompts. It is obvious that as the level increases, the length and complexity of X increase. Therefore, it can be considered that the higher the level, the more human ideas X contains.

Level 0	**Level 1**	**Level 2**	**Level n**
Field name	Title	Summary of abstracts	Whole abstract
AI	DeepStruct: Pretraining of Language Models for Structure Prediction	We introduce a method for improving the structural understanding abilities of language models. Unlike previous approaches that finetune the models with task-specific augmentation, we pretrain language models on a collection of task-agnostic corpora to generate structures from text. Our structure pretraining enables zero-shot transfer of the learned knowledge that models have about the structure tasks.	We introduce a method for improving the structural understanding abilities of language models. Unlike previous approaches that finetune the models with task-specific augmentation, we pretrain language models on a collection of task-agnostic corpora to generate structures from text. Our structure pretraining enables zero-shot transfer of the learned knowledge that models have about the structure tasks. We study the performance of this approach on 28 datasets, spanning 10 structure prediction tasks including open information extraction, joint entity and relation extraction, named entity recognition, relation classification, semantic role labeling, event extraction, coreference resolution, factual probe, intent detection, and dialogue state tracking. We further enhance the pretraining with the task-specific training sets. We show that a 10B parameter language model transfers non-trivially to most tasks and obtains state-of-the-art performance on 21 of 28 datasets that we evaluate.

C Computational Experiment Detail

Our proposed model contained 108.57M parameters and was trained for two hours on a single NVIDIA A100 GPU. To ensure the reliability of the results, we conducted two runs and averaged the outcomes reported in this paper.