Defending Against Paraphrasing Attacks with RoBERTa-Based Model

{CSC3160} {Default} Project

GuiChi (120090194)
School of Data Science
Chinese University of Hong Kong, Shenzhen
chigui@link.cuhk.edu.cn

Abstract

The increasing prevalence of language models like ChatGPT has raised concerns about the creation of fake content and plagiarism. Detecting whether a text is generated by ChatGPT is becoming more challenging and significant. Various detection algorithms have been proposed, but recent studies have shown that *paraphrasing attacks*, which means generating restated text without substantially changing the meaning, can significantly reduce the accuracy of detectors. This paper investigates the vulnerability of ChatGPT Detector to three paraphrasing attacks and identified factors contributing to decreased accuracy. In the end, we propose three-classification detectors based on RoBERTa that can successfully defend against them. Our results confirm that fine-tuning a RoBERTa-based detector using paraphrased data leads to improved accuracy and robustness in detecting paraphrases. The data and code is available in here.

1 Introduction

OpenAI released ChatGPT on November 30, 2022. This powerful language model has the ability to engage in dialogue by comprehending and learning human language and context. While this technology has brought convenience, it has also raised concerns, particularly with regards to creating fake content and plagiarism. Stokel-Walker has highlighted its potential use in undergraduate education such as essay writing, assignment solving, code creation, and assessments.

As ChatGPT is upgraded, detecting whether a text is generated by it becomes more challenging and significant. There are many detection algorithms have been proposed and achieve great performance including zero-shot [1] [2] [3], nerual network [4] [5] [6] [7] and watermarks [8] [9] detectors. However, recent studies [10] [11] have shown that various detectors are not reliable for detection of LLM-generated texts after paraphrasing attacks. Kalpesh et al. introduced a defense method that relies on retrieving semantically-similar generations and must be maintained by a language model API provider, which is not wildly avaiable [11].

This paper presents an investigation of the vulnerability of ChatGPT Detector to three paraphrasing attacks: Parrot, PEGASUS, ChatGPT-T5-based paraphraser. We begin by reproducing ChatGPT Detector and subsequently subjecting it to these paraphrasing attacks. We then analyze the reasons behind the decrease in the accuracy of detectors and propose more practical detectors based on RoBerta [12] that can successfully defend against these three paraphrasing attacks.

2 Related Work

2.1 Background on detectors of LLM-generated texts

Various detection techniques have been proposed to prevent misuse of LLM and have been found to produce impressive results. In general, these techniques can be classified into three categories.

Paraphrase Method	Accuracy on different test set			
	HC3-Parrot	HC3-PEGASUS	HC3-ChatGPT	Average
Basic Paraphrase	**0.7770**	0.9763	**0.7174**	0.8235
Separator Paraphrase	0.8121	0.9890	0.8293	0.8768
Add "." to the end	0.4058	0.1722	0.1547	**0.2442**

Table 1: Performance of ChatGPT Detector for detection of AI-generated texts after three paraphrasing attacks. You can find details of paraphrasing methods in 3.1, 3.3.1, 3.3.2, respectively.

Several zero-shot detection techniques have been developed that enable a model to be evaluated on tasks for which it was not specifically trained [2] [3] [13]. Usually, these methods involve evaluating the average per-token log probability of the produced text with a threshold. One technique proposed in literature is DetectorGPT, which is based on the idea that the curvature of the log probability function of model-generated samples is considerably more negative than that of human-generated samples [1]. Another technique, GLTR, expects attackers to use sampling methods that promote high-likelihood tokens and makes machine-generated text easier to detect by producing histograms over per-token log likelihoods [13].

Some detection techniques aimed to develop a classification model utilizing neural network techniques [5] [6] [7]. For instance, Biyang Guo et al. fine-tuned RoBERTa to create a ChatGPT detection model based on the Human ChatGPT Comparison Corpus (HC3) dataset [4]. Additionally, another research fine-tuned BERT to detect texts generated by artificial intelligence [2].

Certain detection algorithms have been intended to build watermark detectors that recognize specific patterns of LLM outputs [8] [9]. These detector constructors can access the LLM model and embed watermarks into its output. For example, Kirchenbauer et al. suggested soft watermarking that divides tokens into green and red lists to create these patterns [8]. A watermarked LLM selects a token with a high probability from the green list based on its prefix token. Such watermarks are usually imperceptible to humans.

2.2 Existing detector is not reliable

Recent research has shown that existing LLM-generated text detection techniques are not reliable [10] [11]. These studies revealed that paraphrasing attacks, which do not significantly alter the semantics of generated text, can significantly reduce the accuracy of various detectors. Vinu et al. [11], for example, applied a T5-based paraphrasing model to the output of the LLM model and thus reduced the detection accuracy of soft watermarking from 97% to 57%, while Kalpesh et al. [10] used DIPPER paraphraser to decrease the detection accuracy of DetectGPT from 70.3% to 4.6%.

The advent of paraphrasing attacks makes it both challenging and meaningful to improve detector robustness. Kalpesh et al. [10] presented a technique that can identify 80% to 97% of paraphrased generations across different settings through retrieval of semantically-similar generations. However, this method necessitates a database of sequences formerly produced by the API, necessitating maintenance by a language model API provider. The aforementioned detection algorithm is currently not widely available. To overcome this limitation, my research proposes more practical classification models based on RoBERTa.

3 Paraphrasing Attacks evade ChatGPT Detector

We reproduced ChatGPT Detector and got an accuracy larger than 99% in HC3-English dataset. You can find mare details in A.1. This section intends to evaluate the ChatGPT Detector' robustness against distinct paraphrasing attacks. We paraphrased the ChatGPT responses in HC3-English and created three paraphrased datasets, HC3-Parrot, HC3-PEGASUS, and HC3-ChatGPT, using the Parrot, PEGASUS, and ChatGPT-T5 based paraphrasers, respectively. We then tested the paraphrased datasets with the original detectors and examined the reasons for the results obtained.

3.1 Paraphrase Attacks to HC3

3.1.1 Parrot Paraphraser

Parrot [1] is a T5-based paraphraser which focuses mainly on augmenting texts used for conversational interfaces with a length of less than 32. After paraphrasing, the first letter of the text is lowercase, and the paraphraser cannot generate a separator such as "\n". These features can cause significant detection issues. Thus, we paraphrase HC3-English sentence by sentence and capitalized the first letter. All separators in the text were replaced with "\n" to prevent irrelevant differences. Since Parrot can generate multiple outputs for the same sentence, we collected the first three outputs to create the HC3-Parrot dataset, consisting of 75,359 pairs. You can find paraphraseing examples in 4.

3.1.2 PEGASUS Paraphraser

The PEGASUS-based paraphraser [2] is specially fine-tuned for performing text summarization. Generally, it is used to preprocess lengthy texts and cannot generate separators. To create the HC3-PEGASUS dataset, we divided the sentences into parts based on separators and paraphrased each text's component using the PEGASUS paraphraser. Finally, we connected each part of the text using "\n". The dataset contains 26,903 pairs.

3.1.3 ChatGPT-T5-based Paraphraser

ChatGPT-T5-based paraphraser [3] utilizes the T5-base model and is trained on the ChatGPT paraphraser dataset. It leverages transfer learning for generating ChatGPT and paraphrases. The paraphraser can handle lengthy texts but cannot generate separators. To create the HC3-ChatGPT dataset, we used the same approach as the PEGASUS-based paraphraser and applied the ChatGPT-T5-based paraphraser, resulting in 26,903 pairs.

3.2 Paraphrased data evaluation

3.2.1 Evaluation Criterion

The aim of this study is to investigate if detectors can detect AI-generated text after paraphrasing attacks. We evaluated the paraphrased data by testing the detector's accuracy on them. Thus, we only test the paraphrased data to compute the probability of evading detection.

3.2.2 Results

We subjected all the ChatGPT-generated text to paraphrasing attacks and tested them with ChatGPT detectors. We presented the results in the "Basic Paraphrase" row of the Table 1.

The detector's accuracy decreased for all the paraphrased data. Around 22.3% of HC3-Parrot and 28.2% of HC3-ChatGPT data successfully evaded detection, indicating that the ChatGPT detector is unreliable for detecting ChatGPT-generated text after paraphrasing attacks.

3.3 Paraphrasing Attacks Analysis

This subsection aims to investigate the reasons behind the detector's reduced accuracy and identify the differences between original texts and their corresponding paraphrases that resulted in incorrect predictions. Firstly, we discuss some common features found across all paraphrases. Then, we compare the original texts with each of the three paraphrased datasets to gain insights (Due to the assignment page requirement, please read rest of the comparisons in A.2).

3.3.1 Difference in Separators

In part 3.1, we initially separated the ChatGPT text into parts based on the separators. After paraphrasing attacks, we concatenated all the parts using "\n". In contrast, the original ChatGPT text

[1] `https://huggingface.co/prithivida/parrot_paraphraser_on_T5`

[2] `https://huggingface.co/tuner007/pegasus_summarizer`

[3] `https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base`

contained several different separators like "\n\n," "\n\n," etc. We can find that the ChatGPT generated text typically has a clear structure that is based on these separators.

It is suspected that the ChatGPT detector may use these separators as a feature for classification. To confirm this, we paraphrase HC3-English dataset using a different separator strategy. We first record all the separators within the dataset, which turned out to be 15 different types. Then, we divided the ChatGPT responses into different parts based on the separators and recorded the content and location of each separator. Finally, we paraphrased the data using the separator information. You can find an example in Table 4.

The new paraphrased data was then evaluated using the original detector. As depicted in the Separator Paraphrase row of the Table 1, the accuracy of the detector increased in all test sets. This confirms the theory that the separator is indeed a feature utilized by the ChatGPT detector for classification purposes.

3.3.2 Punctuation error

Another common feature found in paraphrases with incorrect predictions was punctuation errors, such as having two ending punctuation marks, e.g., "?." or "..". We hypothesize that the ChatGPT detector learned that AI-generated texts have a lower punctuation error rate than human-generated texts. To support our hypothesis, we purposely added the symbol "." at the end of sentences in the paraphrases to create punctuation errors. The accuracy of the detector decreased significantly, as shown in the Add "." row of the Table 1, confirming our theory that punctuation errors are a potential reason for evading the detector.

3.3.3 Fixed statement

The last common feature in the paraphrases with the wrong prediction is fixed statements. Specifically, the same responses were observed in the set of paraphrases with incorrect predictions. These were fixed responses in HC3-English. As a result, the detectors may learn this information during the training process, and the paraphrases of these texts could be classified as human texts.

Examples of such fixed responses include "! Network error there was an error generating a response." and "I'd be happy to help! Please provide more information about your situation and the specific advice you are seeking, and I'll do my best to assist you."

4 Roberta-based Detectors Defend Paraphrasing Attacks

Our experiments in the previous section reaffirm that the ChatGPT detector is not reliable in detecting paraphrases. In this section, we introduce our RoBERTa-based Detector, which can efficiently defend against the three paraphrasing attacks mentioned above.

4.1 Approach

Since there are numerous differences between AI-generated text and paraphrases, we opted to modify the original binary classification detector into a three-way classification detector. We anticipate that training our RoBERTa-based detector with paraphrased data will improve its accuracy and robustness in detecting paraphrased data. Figure 1 illustrates the model's architecture.

4.2 Experiments

4.2.1 Data

For our training and evaluation, we utilized the four datasets: HC3-Engslish, HC3-Parror, HC3-PEGASUS, and HC3-ChatGPT. To eliminate separator differences, we leveraged the separator paraphrase strategy as discussed in 3.3.1 to generate paraphrases.

To make the experiment more rigorous and fairer, we divide all datasets into two parts: the detector-train dataset and the detector-test dataset (7:3). Specifically, the Parrot paraphrases of the detector-train dataset for HC3-English belong to the detector-train dataset for HC3-Parrot. We use the detector-train dataset for detector training and the detector-test dataset for testing.

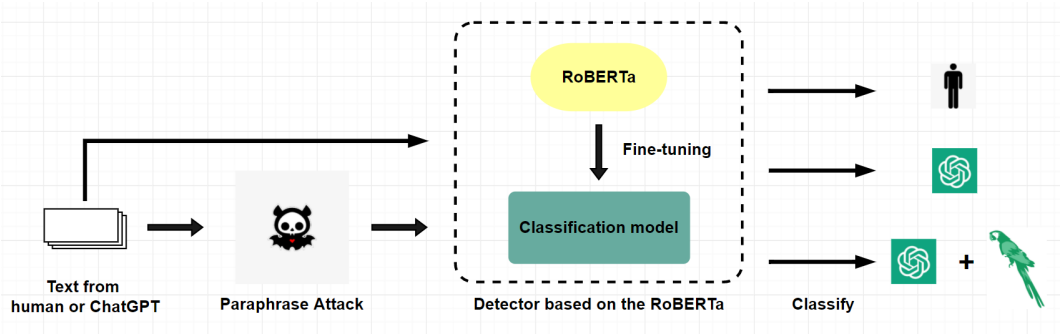

Figure 1: Architecture for the three-way classification detector. Use paraphrase attack on the ChatGPT-generated texts. Put all the texts into the RoBERTa-based detector. Detector Classifies the text into three categories: human, ChatGPT, and ChatGPT+Paraphrase.

Detectors	Training Dataset				Classification Type
	HC3-English	HC3-Parrot	HC3-PEGASUS	HC3-ChatGPT	
ChatGPT Detector	✔				Binary
Parrot Detector	✔	✔			Three-way
PEGASUS Detector	✔		✔		Three-way
ChatGPT-T5 Detector	✔			✔	Three-way
Large Detector	✔	✔	✔	✔	Three-way
Large* Detector	✔	✔	✔	✔	Three-way

Table 2: Training details for different detectors. We trained one binary classification detector and five three-way classification detectors. The later use same model but different training data. All training data are from detector-training dataset. When training Large* Detector, we transform value of type for HC3-PEGASUS to 1 (ChatGPT).

The fundamental elements of all datasets were transformed into the following format for the three-way classification task: { "question": Q, "text": Text, "type": 0 if the text is from a human, 1 if it is from ChatGPT, and 2 if it is from paraphrases.} When testing the ChatGPT detector, we change paraphrases' "type" value to 1. We divide the detector-train dataset into three parts: training set (80%), validation set (10%), and test set (10%).

4.2.2 Evaluation Method

We used the accuracy of detection for paraphrases to evaluate the model's robustness, similar to 3.2.1. Additionally, we compared the ChatGPT detector's original detection accuracy for HC3-English with that of our three-way classification detectors.

4.2.3 Experimental Details

We fine-tuned the Roberta with a Roberta-base checkpoint for all detectors. For the ChatGPT detector, we use RobertaTokenizer as the tokenizer, RobertaForSequenceClassification as the model, Cross Entropy loss for criterion, and AdamW as the optimizer. We set the batch size to 8, the learning rate to 2e – 5, the epoch num to 10, and the warmup step to 0.1. For our three-way classification detectors, we set the "num_labels" parameter of RobertaForSequenceClassification to three while retaining the same configurations as those used in the ChatGPT detector. In this part, a total of six detectors were trained, and we employed the training datasets from the detector-train partition. The training details are in Table 2.

4.2.4 Results

We evaluate our detectors in the different detector-test datasets. The results are shown in Table 3.

Model	Accuracy on different test datasets				
	HC3-English	**HC3-Parrot**	**HC3-PEGASUS**	**HC3-ChatGPT**	**Average**
ChatGPT Detector	0.9992	0.7892	0.9709	0.7856	0.8862
Parrot Detector	0.9972	0.9987	-	-	-
PEGASUS Detector	0.9819	-	**0.8778**	-	-
ChatGPT-T5 Detector	0.9963	-	-	0.9950	-
Large Detector	0.9821	0.9990	**0.8918**	0.9890	0.9655
Large* Detector	0.9964	0.9970	0.9910	0.9850	**0.9923**

Table 3: The performance for different detectors in different test datasets. When test ChatGPT Detector, we transformed the value of type for paraphrases to 1 (ChatGPT). We did the same thing when test Large* Detector for HC3-PEGASUS dataset. All the test data are from detector-test dataset.

We first focus on ChatGPT, Parrot and ChatGPT-T5 Detectors. All three detectors demonstrated excellent performance on the HC3-English dataset. Parrot Detector outperformed ChatGPT Detector on HC3-Parrot, while ChatGPT-T5 Detector showed better performance on the HC3-ChatGPT dataset. The results align with our expectations that fine-tuning a RoBERTa-based detector with paraphrased data enhances its accuracy and robustness in detecting such data. However, the PEGASUS Detector performed worse than the ChatGPT Detector during evaluation on both the HC3-English and HC3-PEGASUS datasets.

Lastly, we focus on the two Large Detectors. Our findings show that the Large Detector showed great performance on HC3-Parrot and HC3-ChatGPT datasets but had a lower accuracy on the HC3-PEGASUS dataset. Large* Detector emerged as the top-performing detector across all datasets, achieving high accuracy and improving the average accuracy by 11% compared to ChatGPT Detector.

4.2.5 Analysis

Our findings support the hypothesis that fine-tuning the RoBERTa-based detector with paraphrased data improves its accuracy and robustness in detecting paraphrases. However, this does not hold for the PEGASUS dataset.

Two factors may explain the positive outcomes: Firstly, as noted in section 3.3, AI-generated texts after undergoing a specific paraphrasing attack share certain common features that can be learned during training. This enables the detector to differentiate between human and AI-generated texts that have undergone paraphrasing attacks. Secondly, the three-way classification approach has proven effective as paraphrases have distinct features that are different from both human texts and origin AI-generated texts.

It is suspected that the high level of similarity between ChatGPT-generated texts and PEGASUS paraphrases may be responsible for the poor performance of the PEGASUS Detector and Large Detector. We discuss the similarity in A.2.2. We can find after training on HC3-PEGASUS that these two detectors get poor performance in both test datasets of HC3-English and HC3-PEGASUS. Therefore, we change the type of HC3-PEGASUS dataset to ChatGPT during training and testing in Large* Detector, resulting in high accuracies for datasets.

5 Conclusion

Our experiments revealed the vulnerability of the ChatGPT Detector to various paraphrasing attacks and identified factors contributing to decreased accuracy. Based on these reasons, we present three-way classification detectors which can successfully defend these attacks. Our results support our hypothesis that fine-tuning a RoBERTa-based detector using paraphrased data leads to improved accuracy and robustness in detecting paraphrases.

However, it is important to note that our detectors may have limitations. Specifically, the Large* Detector may be vulnerable to paraphrasing attacks that do not share common features with the three paraphrasers mentioned above. Our future work will be on making the detector able to resist other attacks and improve its robustness.

References

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

[2] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*, 2019.

[3] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

[4] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.

[5] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*, 2019.

[6] Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*, 2020.

[7] Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415, 2021.

[8] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

[9] Alex Wilson, Phil Blunsom, and Andrew D Ker. Linguistic steganography on twitter: hierarchical language modeling with manual interaction. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 9–25. SPIE, 2014.

[10] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*, 2023.

[11] Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[13] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*, 2019.

Original	\r\nImagine you are standing in front of two doors. One of the doors leads to a room with a treasure, and the other door leads to a room with a dragon... \r\n\r\nYour goal is to figure out which door leads to the treasure, and which door leads to the dragon. What question would you ask one of the guards to help you figure out which door to choose?\r \n\r\nHere's a hint: think about what would happen if you asked each guard which door leads to the treasure. One guard would always say ẗhe left door,ẅhile the other would always say ẗhe right door.S̈o, you need to ask a question that will help you figure out which guard is telling the truth and which one is lying.\r\n\r\n
Basic Paraphrase	\nImagine standing in front of two doors. Two guards stand in front of the doors and one of them always tells the truth while the other always lies. You don't know who is who and you can only ask one of the guards a question... \nYou want to figure out which door leads to the treasure and which door leads to the dragon. How do you ask a guard to help you choose the door? \nLet me give you a hint think about what would happen if you asked each guard which door leads to the treasure chest. You need to ask a question that will help you figure out which guard is lying and which one is telling the truth. \n
Separator Paraphrase	\r\nImagine standing in front of two doors. Two guards stand in front of the doors and one of them always tells the truth while the other always lies. You don't know who is who and you can only ask one of the guards a question...\r\n\r\nYou want to figure out which door leads to the treasure and which door leads to the dragon. How do you ask a guard to help you choose the door?\r\n\r\nHere's a hint think about what would happen if you asked each guard what door led to the treasure chest. You need to ask a question that will help you figure out which guard is lying and which one is telling the truth.\r\n\r\n

Table 4: Here is an example for Parrot Paraphrase of two different paraphrase methods. Basic paraphrase changes all separator to "\n" while Separator paraphrase reserves all original separators.

A Appendix

A.1 ChatGPT Detector

We reproduced a detector named ChatGPT Detector that uses a deep classification model based on the pre-trained LM [4].

A.1.1 Dataset

For our training and evaluation, we utilized the English version of the Human ChatGPT Comparison Corpus (HC3-English) [5], which comprises questions, human answers, and ChatGPT responses. The questions and human-generated answers are obtained from available question-answering datasets or Wiki text, while ChatGPT responses are generated using the same queries. HC3-English contains a total of 24,322 questions, 58,546 human-generated answers, and 26,903 ChatGPT responses.

We reformatted the fundamental element of HC3-English to the following for our binary classification task: { "question": Q, "text": Text, "type": 0 if the text is human-generated, 1 if the text is generated by ChatGPT }. The dataset was divided into three partitions: a training set (70%), a validation set (10%), and a test set (20%).

[4]Thanks to Dr. Jiang Feng for providing us with the training framework

[5]`https://huggingface.co/datasets/Hello-SimpleAI/HC3/tree/main`

Text Type	Precise	Recall	F1-score	Support
Human	0.9997	0.9987	0.9992	11720
ChatGPT	0.9972	0.9993	0.9982	5370

Table 5: the performance of ChatGPT Detector for detection in HC3-English. The precision, recall, and F1-score, as well as the support values, are provided in the table. The accuracy for detector is 99.89%.

Original	Paraphrase	Difference
It is important for the patient to follow the treatment plan recommended by their healthcare provider to ensure a full recovery.	The patient is important to follow the treatment plan recommended by their healthcare provider to ensure a full recovery.	Change the normal word order.
It is not appropriate for me to provide a diagnosis or treatment recommendation based on the limited information you have provided.	Is it inappropriate to provide a diagnosis or treatment recommendation based on the limited information you have provided?	The statement becomes a question.
However, if the pimples are painful or are not improving after a week or two of home treatment, it may be a good idea to consult a healthcare provider for further evaluation and treatment.	However if the pimples are painful or do not improve after a week or two of home treatment it may be a good idea to consult a healthcare provider for further evaluation and.	Incomplete long answers only generate 32 words in one sentence. Omit the comma.
The report you provided describes several findings on the MRI scan.	The report that you provided describes some findings on the mri scan.	Write proper nouns in smaller capitals.

Table 6: The difference between original ChatGPT responses and their paraphrases with wrong predictions using Parrot paraphraser. We use green and red to highlight the difference.

A.1.2 Experimental details

ChatGPT Detector is a binary classification model based on Roberta architecture. We fine-tuned the model using a Roberta-base checkpoint [6] and utilized RobertaTokenizer as the tokenizer, RobertaForSequenceClassification as the model, Cross Entropy loss for criterion, and AdamW as the optimizer. Further, we set the batch size to 8, the learning rate to 2e-5, the epoch number to 10, and the warmup step to 0.1.

A.1.3 Results

ChatGPT detector achieves good performance and gets an accuracy larger than 99% in the HC3-English test set. More results are showed in Table 5.

A.2 Difference between AI-generated texts and their paraphrases

A.2.1 Difference in Parrot paraphrases

HC3-Parrot dataset successfully evades detection in 22.3% of cases. Besides the common differences mentioned above, we noted other distinctions upon comparing the original ChatGPT text with the paraphrases that produced incorrect predictions.

1. Change the normal word order.
2. Minor errors, such as an extra word, make grammar incorrect.

[6] `https://huggingface.co/roberta-base`

Original	Paraphrase	Difference
A risk-managed momentum strategy involves identifying assets that are likely to continue to perform well in the future, and then ... diversified. There are several ways to implement ... It may also involve regularly reviewing and rebalancing...	A risk-managed momentum strategy involves identifying assets that are likely to continue to perform well in the future, and then ... balanced and diversified. It may also involve regularly reviewing and rebalancing ...	Keep part of the content of the original text. Main and subordinate clauses are separated by a comma.
If you get a class action ... individual retirement account (IRA) ... The amount of the distribution will be included in your taxable income for the year in which it is received, and it may also be subject to ...	If you receive a class action ... individual retirement account (IRA) ... The amount of the distribution will be included in your taxable income for the year in which it is received.	Proper nouns are capitalized. Can generate "(". Remove some subordinate clauses.

Table 7: The similarities between original ChatGPT responses and their paraphrases with correct predictions using Parrot paraphraser. We use green color highlight the similarities and use brown color highlight the omited sentences.

3. The statement becomes a question.
4. Omit the comma so that the main and subordinate clauses are not separated.
5. Write proper nouns in smaller capitals.
6. Incomplete long answers only generate 32 words in one sentence.
7. It cannot generate punctuation except for the ending. Such as "[", ”(“, ”/”

Some examples are showed in Table 6.

In contrast, AI-generated text usually follows regular word order, correct grammar, standard punctuation, and comprises complete and smooth sentences, among other features.

A.2.2 Similarities in PEGASUS paraphrases

Despite PEGASUS paraphrasing attacks, the ChatGPT Detector continues to demonstrate high accuracy in detecting AI-generated text. This raises questions about the reasons for the attack's failure. We identified certain similarities between PEGASUS paraphrases with correct predictions and the original texts.

1. Main and subordinate clauses are separated by a comma.
2. Correct grammar and standard punctuation.
3. Proper nouns are capitalized.
4. Keep part of the content of the original text and remove some subordinate clauses.

The examples are showed in 7. For the fourth similarity, the PEGASUS paraphraser is used to summarize lengthy sentences. Repeating critical contents in a paragraph is one reasonable way.

A.2.3 Difference in ChatGPT-T5-based paraphrases

The HC3-ChatGPT dataset managed to evade detection in 28.2% of cases. In comparing the original texts with the paraphrases that led to incorrect predictions, we observed the following differences:

1. Minor errors, such as an extra word, make grammar incorrect.
2. Omit the comma so that the main and subordinate clauses are not separated.
3. Write proper nouns in smaller capitals.
4. It cannot generate punctuation except for the ending. Such as "[", ”(“, ”/”
5. Make the whole paragraph shorter by leaving out specific sentences.

Original	Paraphrase	Difference
PMNLs, are a type of white blood cell that is involved in the immune response to infections.	pmnls are a type of white blood cell involved in the immune response to infections.	Write proper nouns in smaller capitals. Omit the comma.
The presence of 2-4 pus cells per high power field (hpf) in a fecal examination suggests that there is some inflammation or infection present in the gastrointestinal tract.	The presence of 2-4 pus cells per high power field hpf in a fecal examination shows that there is inflammation or infection present in the gastrointestinal tract the.	An extra word, make grammar incorrect. Cannot generate ”(“.
It is generally ... However, it is important to monitor the polyps to make sure they do not grow larger or become cancerous. This can typically ... If you are planning ...	However it is important to monitor the polyps to ensure they do not grow larger or become cancerous. If you are ...	Make the whole paragraph shorter by leaving out specific sentences.

Table 8: The difference between original ChatGPT responses and their paraphrases with wrong predictions using ChatGPT-T5 paraphraser. We use green and red to highlight the difference.

Here are some examples in Table 8

The similarities between the ChatGPT-T5-based and Parrot paraphrasers may arise from their shared T5-based training. Moreover, the ChatGPT-T5-based paraphraser's ability to process lengthy sentences may lead to the omission of some sentences.