Adversary is all you need
–a novel method for context detection inspired by GAN

Yiming Xiong 120090721
Ye Dou 120090709
Jun Xiao 120090044
Zhizhen Chen 120090823

1 Introduction: Significance and Novelty

The advancing development of artificial intelligence has brought increased convenience to our daily life. However, it has also raised concerns over its negative impacts, such as the growing trend of using GPTs for writing essays among students. While AI offers a simple and efficient solution for writing, it poses a significant threat to academic integrity and could cause safety issues.

To further explore text detection, we replicated related methods, including zero-shot, feature-based, pre-trained language model-based methods. And we found that BERT and RoBERTa have impressive performances among those methods on ChatGPT.

Although BERT and RoBERTa could achieve a relatively good accuracy in essay detection, they did not take into account the characteristics of ChatGPT. However, ChatGPT with Reinforcement Learning from Human Feedback(RLHF) could enhance alignment with user needs through multi-round dialogue, instead of using fine-tuning. This provides a chance to do a multi-round adversary between the detector and ChatGPT since we don't need to further train GPT to get a stronger sample(more similar to human writing). Thus we utilized a training paradigm that is similar to GAN between ChatGPT and the BERT or RoBERTa. We then evaluated our method and proved its efficiency. Finally, we analyzed the improvement using SHAP.

2 Related Work

Currently, many researchers used feature-based methods for text detection [1, 5]. For ChatGPT, [4] utilized the XGBoost algorithm with TF-IDF and hand-crafted feature extraction modes on detection of TODEL writing, to detect whether given texts are machine-generated or human-written, with an overall accuracy of 0.96.

Since the generative models are capable to recognize the texts generated by themselves, zero-shot detection is another widely used text detection method that does not need to fine-tune the model for other downstream tasks [1, 5]. [6] shows that Grover, a news generated-model, has an impressive ability to recognize news generated by AI. Yet [1] states that others have shown Grover has poor generalization ability on other kinds of texts.

The state-of-the-art approaches, fine-tuned mdoels, are the most popular methods for neural detection of machine-generated text at the moment[1, 5]. Fine-tuned models are based around fine-tuning of Pre-trained Language Models(PLMs) like Bidirectional Encoder Representation from Transformers(BERT) and A Robustly Optimized BERT(RoBERTa). Due to the special bidirectional encoding structure, the models could learn more information from the previous and the following context to transform the meaning of current word into a vector. [3] has shown that RoBERTa can have a good performance after double-tuning on the testing domain to detect the GPT-2 generated abstract. And for ChatGPT, [2] utilized Distil-BERT can achieve an accuracy of 0.70 on comments on the restaurants.

3 Methodology

We first implemented three mainstream methods, then we chose PLMs as the baseline detectors we used in our method. Denoted the dataset as $D = \{x_i, y_i\}^{2n}$, where n is the number of human-written texts. We use the human-written texts to get the machine-generated text by asking ChatGPT to rewrite them.

3.1 Current Methods

We first use the XGBoost algorithm in conjunction with TF-IDF and hand-crafted feature extraction modes to identify machine-generated text. Hand-crafted extracted features are based on part of speech and large word percentage. TF-IDF is based on the frequency of words. We first extracted these features from the text. Then we use XGboost, which works as a classifier.

Zero-shot detection is another method that no need to train or fine-tune the model. Using ChatGPT, we input the one paragraph that is written by ChatGPT or a human as the testing data and ask ChatGPT whether this sample is provided by human, and ChatGPT will respond "yes" or "no". We then counted the prediction and output the accuracy.

Finally, we implement BERT and RoBERTa, with an additional fully connected layer as detectors. We perform fine-tuning with the training dataset and cross-entropy loss. Then we evaluate the performance by settling down the parameters and only inputting the tokenized testing data.

3.2 Our Methods

As shown in Fig.1, we first utilized human-written samples to generate GPT-written samples by asking ChatGPT to rewrite the human-written paragraph in its own words. Then we used the human-written and the GPT-generated samples together as the training dataset to first fine-tune BERT or RoBERTa, with a fully connected layer as a classifier. After the first round of fine-tuning, we further utilized the prediction of the fine-tuned model on the ChatGPT-generated text in the training dataset. If the fine-tuned model correctly recognizes the ChatGPT-generated text in the training dataset, we would ask ChatGPT that its generation is not quite similar to the corresponding human-written text, and thus let ChatGPT generate a more similar text compared to the corresponding human-written text. The newly generated text would be fed to the first-round fine-tuned detector to see whether the new text can deceive the detector. If not, we asked ChatGPT to regenerate a more similar text than before. For each original correctly detected GPT-generated text, we repeated this process at most three times, which means if three newly generated texts still cannot deceive the detector, we would skip to the next data. If the GPT-generated text successfully deceives the detector within

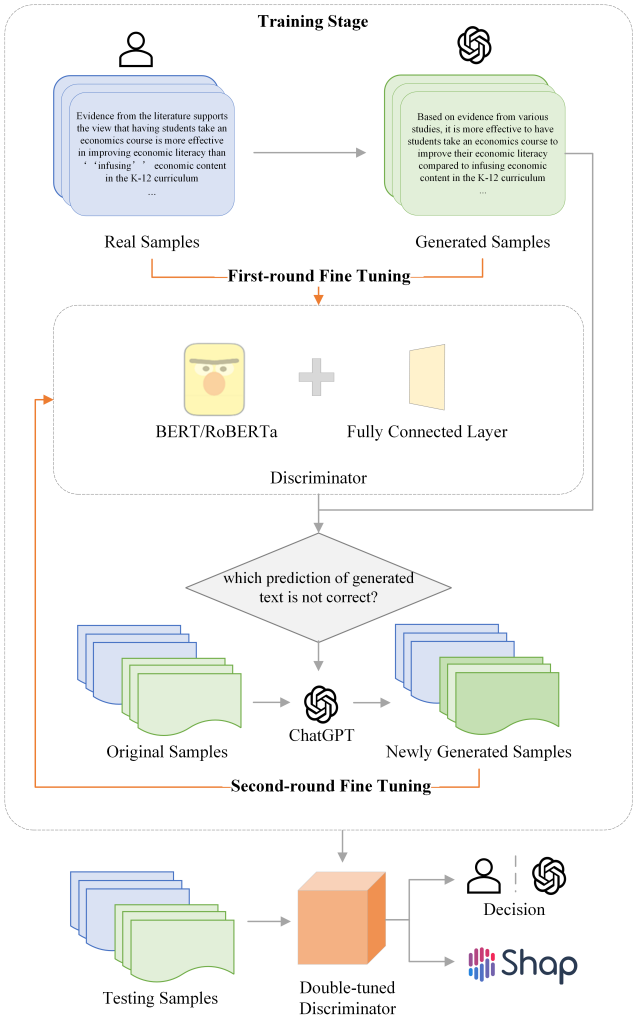

Figure 1: The training paradigm for our method. We check whether the first-round fine-tuned detector could correctly predict the label of ChatGPT-generated text. If yes, we will ask ChatGPT to rewrite a paragraph that is much similar to the corresponding human-written paragraph in order to deceive the detector. If it can successfully deceive the detector, we will replace the original data by the newly generated one. And utilizing the modified data to further fine-tune the detector.

3 rounds, we would replace the original paragraph with the newly generated data. After regenerating all originally correctly detected texts, we then used the changing data to fine-tune the detector again. The double-tuned detector could learn more from its mistake.

Then we used the double-tuned detector to test both on In-Domain(ID) and Out-Of-Domain(OOD) data, for its accuracy and generalization ability, and used Shapley Additive Explanations(SHAP) to figure out the features detector focus on after one-round and second-round tuning.

4 Experiment

4.1 Evaluation Metrics

In this paper, we aim to explore a more accurate way based on adversary to detect whether the given paragraphs of essays are generated by humans or by the ChatGPT, and how much percentage of one essay is written by humans. We use the following evaluation metrics to present the experiment results, accuracy, precision, recall, and text percentage, which is the percentage of human-written paragraphs in an essay. The closer the text percentage is to the true one, the better performance.

$$Percentage_{true} = \frac{1}{n} I(label_i = 1)$$
$$Percentage_{pred} = \frac{1}{n} I(pred_i = 1) \quad (1)$$

For all the symbols above, true labels and predicted labels can take value 0 or 1, where 0 represents ChatGPT-generated text and 1 represents Human-generated text.

	Biology	Law	Psychology	OOD Data
number of essays	18	32	17	83
number of training paragraphs	373	390	324	-
number of validating paragraphs	142	122	99	-
number of testing paragraphs	140	194	99	2814

Table 1: Detailed Distribution of Datasets

4.2 Datasets

Due to the lack of standard datasets, we manually extracted and processed in-domain(ID) datasets containing essays from the flieds of biology, law and psychology, and an out-of-domain(OOD) dataset that contains essays not in those domains such as chemical and politics. To make sure those essays are indeed written by humans, we only extracted the essays that published before 2008 from the website *springer.com*. After collecting the human-generated essays, we spilt it into paragraphs, and asked ChatGPT to rewrite those paragraphs in its own words for collecting ChatGPT-generated paragraphs. Here we provide a table (Table 1) explaining the detailed information of our datasets.

Specially, the training datasets should have balanced human-generated and ChatGPT generated samples, in order to prevent bias towards one class and improve model performance.

4.3 Parameters and Libraries

When implementing our novel two-round training model, we defined the following hyper-parameters and libraries used. For every training dataset, we tuned the hyper-parameters for it and build different models. We implemented our two-round training models based on adversary using BERT and RoBERTa in *Python*. We used the Transformer library of *Hugging Face*, and "bert-base-uncased" and "roberta-base" for implementing BERT and RoBERTa respectively.

5 Results

5.1 Result 1: Reproduction of Related Work Methods

We replicated the related work methods and tested our own in-domain(ID) datasets on those models. Here, we present the results using the format *Accuracy[Precision, Recall]* in Table 2. We can see from Table 2 that BERT and RoBERTa have impressive accuracy(over 70%) for every dataset on detecting whether they are written by humans or ChatGPT. Although XGBoost algorithm with TF-IDF and hand-crafted feature extraction modes shows well on TOEFL writing detection, they have a relatively bad performance(roughly 50%) in essay detection, which means the feature they cultivated is not profound enough. Also, from the result of *Chat2Chat*, we can see that ChatGPT is not confident in recognizing the text generated by itself.

	TF-IDF	Hand-crafted	Chat2Chat	BERT	RoBERTa
Biology	0.54[0.53,0.60]	0.49[0.49,0.56]	0.513[0.484,0.65]	0.721[0.897,0.642]	0.736[0.811,0.614]
Law	0.64[0.61,0.78]	0.65[0.67,0.59]	0.485[0.491,0.804]	0.722[0.978,0.454]	0.773[0.965,0.567]
Psychology	0.56[0.55,0.57]	0.58[0.56,0.65]	0.525[0.521,0.74]	0.707[0.714,0.700]	0.970[0.978,0.960]

Table 2: Results of Related Work Methods on ID Datasets

		Biology	Law	Psychology	Total
BERT	ID	0.700[0.967,0.414]	0.722[0.978,0.454]	0.707[0.714,0.700]	0.804[0.946,0.645]
	OOD	0.596[0.969,0.199]	0.718[0.981,0.444]	0.748[0.798,0.665]	0.842[0.964,0.709]
Double-tuned BERT	ID	0.707[0.892,0.471]	**0.763[0.947,0.557]**	**0.788[0.784,0.800]**	**0.852[0.953,0.742]**
	OOD	**0.706[0.956,0.432]**	**0.746[0.970,0.507]**	**0.773[0.838,0.667]**	**0.862[0.964,0.751]**
RoBERTa	ID	0.736[0.811,0.614]	0.773[0.965,0.567]	0.970[0.978,0.960]	0.848[0.963,0.724]
	OOD	0.609[0.830,0.274]	0.772[0.967,0.564]	0.966[0.950,0.983]	0.882[0.969,0.790]
Double-tuned RoBERTa	ID	0.736[0.789,0.643]	**0.804[0.954,0.639]**	**1.000[1.000,1.000]**	**0.871[0.965,0.770]**
	OOD	**0.666[0.813,0.432]**	0.772[0.968,0.563]	0.973[0.950,1.000]	0.886[0.966,0.801]

Table 3: Comparison between the double-tuned detectors and baseline method

5.2 Result 2: Our Method

We implemented double tuning on both BERT and RoBERTa, as shown in table 3. At the paragraph level, the increment in accuracy that is larger than two percent is bolded. It shows that there's an impressive increment of accuracy on both the ID and OOD accuracy for double-tuned BERT in most of the fields. And there's also an obvious increment of double-tuned RoBERTa on ID accuracy. At the passage level, we show the BERT with total data testing on OOD data as an example. As shown in figure 2, the BERT after second-round tuning has a higher frequency of gap between 0 to 0.1, which indicates a better performance at the passage level.

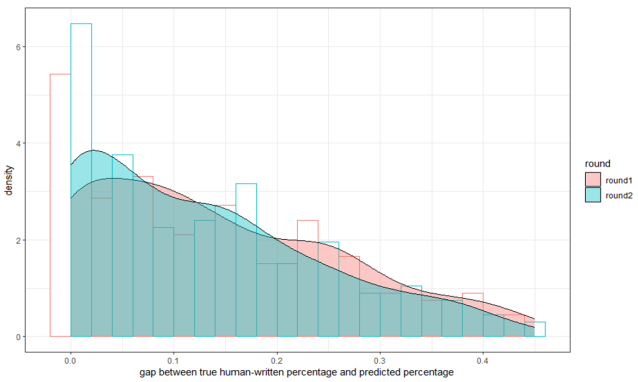

Figure 2: The distribution of the gap between true human-written percentage and the predicted percentage. The X-axis indicates the gap, and the Y-axis indicates the density. Higher density close to 0 indicates a better performance.

Although our method shows an increment of in-domain accuracy and out-of-domain generalization ability in general, there are still two points that need further discussion.

Table 3 shows that the precision is much larger than recall after the first-round tuning. For example, the first-round tuned BERT in the Biology field has a precision of 0.969 but the recall is only 0.199 on OOD data, which means the detector has a prediction bias. It tends to predict human-written paragraphs as GPT-generated, even though our training dataset has the same number of human-written and GPT-generated samples. We think this is because, after first-round fine-tuning, the detector learns some obvious yet not profound features. Although these features are very common in GPT-generated text, they are also frequently used in human writing. This could be eased by double-tuning. Since the double-tuned BERT has a precision of 0.956, similar to 0.969, but a large increment in the recall, from 0.199 to 0.432. Because double-tuning allows the detector to learn more profound features to differentiate these two classes. We further verify our thoughts in the explanation with SHAP, which will be illustrated in section 5.3.

Besides, the generalization ability of double-tuned RoBERTa is not obviously improved, as shown in Table 3. The increment of generalization ability is relatively small in Law, Psychology, and Total. One reason is that the one-round fine-tuning has already given an impressive generalization ability. For example, the first-round tuned RoBERTa on Psychology has an OOD accuracy of 0.966, which means there's not enough space for us to get a large enough increment from second-round tuning. Another reason is that relatively high accuracy, such as in Law and Total, means it's hard to change the prediction in three-round-perturbation, thus leading to a smaller increment in generalization ability. To further verify our thoughts on the reasons, we tested the correlation between the percentage of changing data and the improved percentage of accuracy. Shown in figure 3, there's a moderate correlation between the percentage of changing data regenerated by CatGPT after first-round tuning and the improved percentage of OOD accuracy. We found that RoBERTa has a relatively small changing percentage, and thus means a smaller increment in the percentage of OOD accuracy, which verified our thoughts.

5.3 Result 3: SHAP Evaluation

To further explore what profound features learned by double-tuned detector, we use local shap to interpret the behavior of BERT on law data. In the SHAP text diagram, each row represents an instance, and each cell represents the SHAP value of that feature in that instance. The SHAP score may be positive or negative, indicating the positive or negative impact of the feature on the results of the sample. In general, light-colored cells indicate that the feature contributes less to the result, and dark-colored cells indicate that the feature contributes

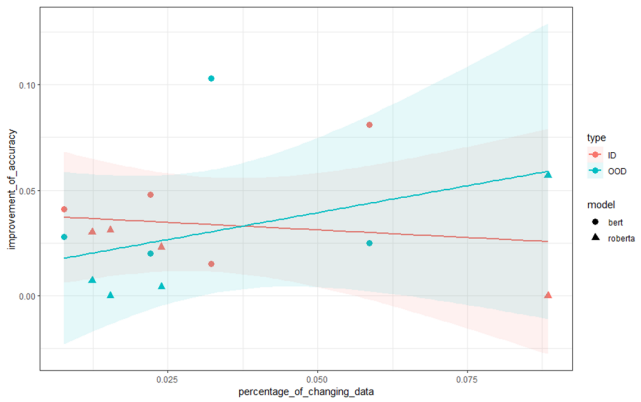

Figure 3: The correlation between the percentage of changing data and improved percentage of accuracy. On ID data there is only a weak correlation around -0.16, while on OOD data there is a moderate correlation around 0.41.

more to the result.

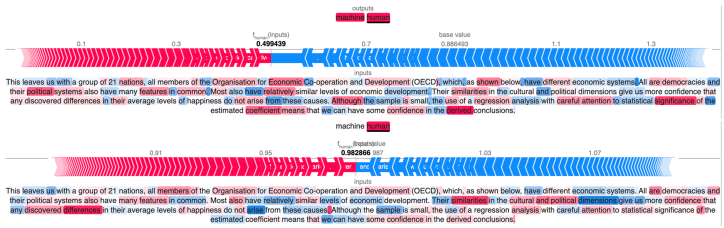

Figure 4: The SHAP evaluation result of sample which is uncorrectly predicted after first round tuning and correctly predicted after double tuning

Here in our samples, the red tokens represent the feature that is regarded as machines' while the blue ones represent the humans. Figure 4 shows the shap evaluation result regarding one sample which is incorrectly predicted in the first round, but correctly predicted after double-tuning. We can infer from the result that the main focus of the detector in the first round are some prepositions such as "the" ,"to", and "in" etc, as well as some punctuations such as ',' and '.'. While after double-tuning, the focus change to some logic and explicit words like 'difference' and 'similarities', these words can represent the logic in the sample. In addition, Figure 5 shows two

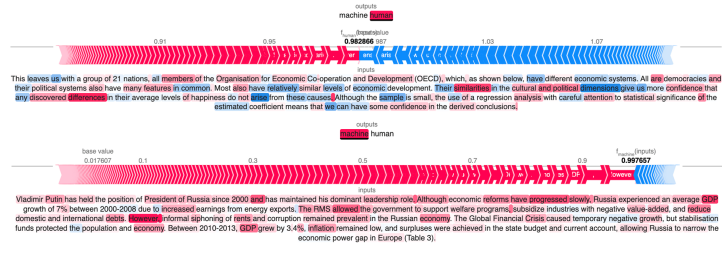

Figure 5: The SHAP evaluation result of two samples correctly predicted as human and as machine after double-tuning

samples which are correctly predicted as human and as machine after double-tuning. We can see from Fig.5 that the detector focus on some subjective words like 'largely' and 'sharply' to predict the sample as human. Contrastly, the detector will focus on objective words like 'however' to predict the sample as machine generated.

Thus, we can conclude that after one-round tuning, detector focus more on prepositions and punctuation. The detector believes these features are shown in machine-written paragraph. Yet these features could also frequently appear in human-written paragraph, which lead to a low recall and the detector predict many human samples as machine-generated sample. After double-tuning, this phenomena is alleviated. The double-tuned detector believes that machines tend to use "however" and "hence" to express logical relationships, while humans tend to express their logic using more implicit way, such as "similarity" and "differ" and some subjective words like "largely" and "sharply". These features are more profound than prepositions and punctuation, and also effective. It gives an explanation why the recall is highly improved after double-tuning while the precision remains the same as the one-round tuning.

6 Conclusion

In conclusion, our study applied double-tuning on AI essay detection task based on adversary and achieved a certain increment in accuracy compared to baseline methods, and finally we interpreted our model using local SHAP.

In this study, two main limitations have been identified. The first limitation is the small size of the datasets, since we only generate three rounds per data. As a result, the changed data is not enough and bad prompt exists, which may restrict the performance of the model. The second limitation is the irregularity of the data, which is partly due to the lack of open standard data.

Despite the limitations of this study, there are still several avenues for future research. Firstly,we can enlarge the dataset to increase the representativeness of the findings. Secondly, apply the current methodology to a third round testing and furthermore, expanding the scope of the research to include other tasks, such as report writing.

References

[1] Evan Crothers, Nathalie Japkowicz, and Herna Viktor. *Machine Generated Text: A Comprehensive Survey of Threat Models and Detection Methods*. 2023. arXiv: 2210.07321 [cs.CL].

[2] Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. *ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-generated Text*. 2023. arXiv: 2301.13852 [cs.CL].

[3] Juan Rodriguez et al. "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*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 1213–1233. DOI: 10.18653/v1/2022.naacl-main.88. URL: https://aclanthology.org/2022.naacl-main.88.

[4] Rexhep Shijaku and Ercan Canhasi. *ChatGPT Generated Text Detection*. Jan. 2023. DOI: 10.13140/RG.2.2.21317.52960.

[5] Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. *The Science of Detecting LLM-Generated Texts*. 2023. arXiv: 2303.07205 [cs.CL].

[6] Rowan Zellers et al. *Defending Against Neural Fake News*. 2020. arXiv: 1905.12616 [cs.CL].