Evaluation and Enhancement of Detector Robustness with Seven Advanced Text Attack Recipes

CSC3160 Default Project

Yang Ruichun
School of Data Science
Chinese University of Hong Kong, Shenzhen
ruichunyang@cuhk.edu.cn

Abstract

The emergence of advanced large-scaled language generation models, which are capable to produce natural and indistinguishable text, have drawn increasing attentions to the AI-text detectors that prevent malicious use of fake texts. However, the existing language model based detectors, in type of text classification models, are susceptible to adversarial examples, perturbed versions of the original text imperceptible by humans but can fool DL models. There is still lack of studies that explore the ability of AI-text detectors resisting to state-of-the-art text attack recipes. In this project, I trained a BERT-based detector and evaluate its robustness under seven edging black-box text classification attack methods. To enhance the detector's ability against attacks, I further perform adversarial training on the base detector and evaluate its effectiveness through adversarial attacks.

1 Key Information to include

- The project is completed individually.
- Code URL: https://github.com/SLPcourse/CSC3160-118010368-RuichunYang.git

2 Introduction

ChatGPT, a large-scaled, transformer-based language model developed by OpenAI, has revolutionized the field of AI with it capability to generate coherent and nature text on a wide range of topics. However, large language models' production of highly human-like text raises the potential risk of unethical or illegal utilization of fake texts, such as academic fraud and fake news generation. Many detection approaches have been proposed to solve the security issues raised by fake texts, one of the effective method is the black-box deep learning classifier that uses language models (i.e. BERT, RoBERTa) as backbones[1][2]. This type of detectors are developed through supervised fine-tuning of pre-trained language models on a mixture of human-written text and machine-generated text.

However, the vulnerabiltiy of text classification models under adversarial attacks that have imperceptible alterations from the original counterparts but can mislead model's judgement. Currently, the problem of adversarial attack on AI-text detectors is rarely studied. As it is helpful to evaluate or even improve the robustness of detectors by exposing the maliciously crafted adversarial examples, the exploration of detectors' performance under text classification is worth noticing.

In real practice, the attacker is usually inaccessible to detector model's parameters and training data, and the only feedback is a classification label and confidence score, in which case the attacks are classified as black-box. Based on the importance of defending black-box attack, I utilize seven state-of-the-art text attack algorithms (textfooler[3], textbugger[4], deepwordbug[5], bae[6], pwws[7], pruthi[8], checklist[9]) to assess performance of the BERT-based detector under severe attacks. Different from image classification attacks, it is more challenging to deal with text data due to its high

requirements such as similar meanings and grammar correctness for imperceptibility. The adversarial samples should idealy follow strict constrains: (1) human prediction consistency— prediction by humans should remain unchanged, (2) semantic similarity— The written example should reflect a similar meaning to the source, as judged by humans, and (3) language fluency—generated examples should be natural and grammatical. However, it is of high cost to evaluate adversarial samples by humans. For substitution, the adversarial text generated in the experiments follows machine-recognized rules such as maximum word embedding distance, part-of-speech consistency, grammar checker, minimum sentence encoding cosine similarity.[10]

The contributions of this project are summarized as follows:

- I trained a BERT-based ChatGPT-generated text detector that reaches over 99% accuracy on the Human ChatGPT Comparison Corpus (HC3) dataset.
- I performed seven adversarial text attack methods on the detector with TextAttack framework and analyze the effects based on experiment results.
- I applied adversarial training on the detector based on the mixture of original and adversarial data, and evaluate the effectiveness of adversarial training by the robust detectors' performance against fooling of perturbed machine-generated text.

3 Related Work

3.1 Deep Classification Model in Black-Box Method

The access to many large language models (LLM), i.e. ChatGPT, is restricted to API-level, where the detection task falls in the black-box region. The differentiation between LLM-generated texts and human-written texts belongs to a binary classification problem. To construct a well-performed classifier, the black-box method requires to collect data consisting of human-written text and the corresponding machine-generated text from the target large language model. The collected data is mainly used in feature extractions or fine-tuning pretrained language models, such as BERT[11] and RoBERTa[1].

Deep language models ahve been used to extract in-context text features for downstream classifier training and prediction, as demonstrated by Zellers et al.[12] who proposed a detector based on a linear classifier on top of GROVER model, but with a low interpretability due to their black-box nature. Ippolito et al.[13] fine-tuned the BERT model on a collected dataset of generated-text pairs. They futher compared the accuracy of recognizing LLM-generated text between human raters and the BERT model, showing that humans have significantly lower accuracy than automatic discriminators in identifying neural text. RoBERTa-based detector trained on GPT-2's top-p examples in Solaiman et al. [14] experiments and establishes the edging performance in identifying the web pages generated by the largest GPT-2 model with around 95% accuracy. The result that fine-tuning using the RoBERTa model achieves higher accuracy than fine-tuning a GPT-2 model with equivalent capacity is possibly due to the superior quality of the bidirectional representations inherent in the masked language model. The advantage of these approaches is that few parameters need to be learned from scratch.

3.2 Adversarial Attacks on Text Classification Model

While deep neural network (DNN) models have exhibited state-of-the-art performance in various tasks such as classification, detection and optimization, their weak defense against adversarial samples has been widely recognized. Many existing studies have investigated the security of current DL models and proposed different effective attack strategies, a majority of which focus on the image domain. However, the adversarial attacks on text data is quite more challenging. In the image domain, the perturbation can often be made virtually imperceptible to human perception, causing humans and state-of-the-art models to disagree. However, subtle changes of expression are usually perceptible in the language domain and replacing one word with another could have a major impact on its meaning.[4] Therefore, a set of constraints are required for adversarial text generation, such as maximum word embedding distance, part-of-speech consistency, grammar checker, minimum sentence encoding cosine similarity.[10]

In recent years, there has been a growing number of black-box techniques for generating adversarial text examples. A novel scoring strategy for text data is introduced in the Deepwordbug[5], indicating

the most important token whose modification will change classification probabilities. Deepwordbug performs character-level perturbation on high scoring words. Ren et al.[7] proposed a greedy algorithm called probability weighted word saliency (pwws) to generate adversarial text, which introduces word replacement order determined by both the word saliency and the classification probability. The method has been evaluated on three text models (word-based CNN-2, word-based CNN-3, and Bi-LSTM) to show its advanced transfer-ability. Li et al.[4] developes Textbugger, an efficient adversarial sample generation framework that preserves 99.4% of valid adversaries that can be correctly recognized by human readers. The modification strategies in Textbugger includes insertion, deletion, swap, and substitution of characters as well as word substitution.

4 Approach

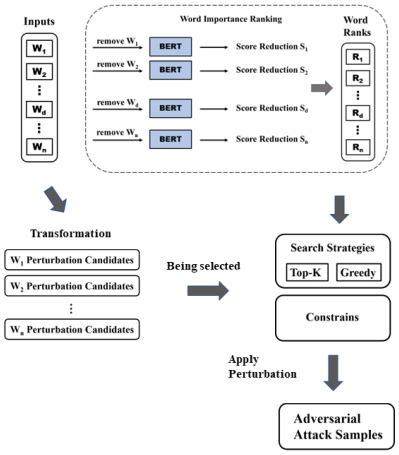

Figure 1: Adversarial Samples Generation

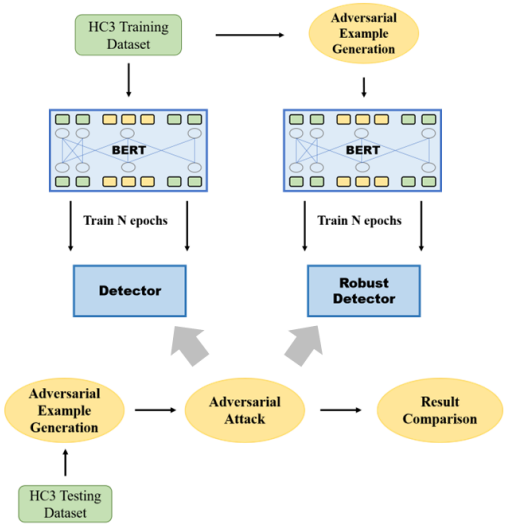

Figure 2: Training of Base Detector and Robust Detector

The adversarial text generation is as following steps: (1) generate candidate perturbations for each word (2) filter out the candidates that violate the constrains (i.e. semantic similarities larger than a threshold) (3) according to the importance ranking, find the least-number perturbation words that cheat the detector's classification. The ranking of words are based on the extent of how their replacement will influence the confidence scores of the target model.

The first BERT detector is finetuned based on origin HC3 train dataset while the robust detector will be trained on the modified HC3 train dataset mixed by original texts and adversarial examples (adversarial training). The HC3 dataset was split into train set, validation set, and test set in the based-detector training. The adversarial data used in robustness evaluation and adversarial training are generated based on the fake text of HC3 dataset. To prevent possible data leakage, the adversarial sets for training, validation, and testing are generated from the data in the based-detector's training set, validation set, and testing set respectively.

5 Experiments

5.1 Data

The dataset used in this project is the Human ChatGPT Comparison Corpus (HC3), which contains tens of thousands of comparison responses from both human experts and ChatGPT, with questions ranging from open-domain, financial, medical, legal, and psychological areas.

5.2 Evaluation method

The detection of machine-generated text is a binary classification problem, the discrimination evaluation of which during the classification training can be defined based on confusion matrix. The evaluate metrics of detector performance in this project are accuracy, precision, recall, and F-1 score.

In the text attack experiments, two evaluation metrics are used to evaluate the effects and efficiency of different methods:

- **Accuracy reduction** Accuracy reduction equals that model prediction accuracy on unmodified machine-generated text (original acc.) minus the prediction accuracy under adversarial perturbation on input text. The extent of accuracy reduction shows the model' s degree of effects from adversarial attacks.
- **Average perturbed words** The average percentage of perturbed words in adversarial texts that succeed in cheating the detector. A large average portion of perturbed words show model's strong resistance to attacks.

5.3 Experimental details

- **Target model and training hyper-parameters** The training and evaluation were performed on a BERT-based detector. The configurations training for base detector and robust detectors are set to 4 epochs(the model usually converges at the 3rd epoch), 8 batch size, 2e-5 learning rate and Adam optimizer.
- **Adversarial samples** In robustness evaluation, each recipe generate 200 adversarial samples limited by Colab GPU resources. For adversarial training, 1000 adversarial AI-samples generated per recipe are mixed with original AI-text and real text.[1]
- **Platform** I trained and attacked the target detector on a Colab server machine with GPU.

5.4 Results

Table 1 shows the original model prediction accuracy, prediction accuracy under attacks, and average perturbed words in total text inputs in detecting adversarial samples of machine-generated texts (Machine) and human-written texts (Human) respectively.

Table 1: Detecting fake text under attacks

	Origin accuracy		Attack accuracy		Avg. perturbed words	
recipes	Machine	Human	Machine	Human	Machine	Human
textfooler	100%	100%	1.00%	93.00%	6.91%	13.72%
textbugger	100%	100%	3.00%	98.00%	25.81%	10.91%
bae	100%	100%	30.00%	90.00%	5.66%	17.45%
pwws	100%	100%	0.00%	100.00%	4.89%	\
deepwordbug	100%	100%	10.00%	100.00%	4.31%	\
pruthi	100%	100%	94.29%	100.00%	1.47%	\
checklist	100%	100%	95.00%	100.00%	15.19%	\

Due to the limitation of Colab GPU computing resources, only short text with under 100 tokens were selected to generate adversarial text. Table 2 shows the metrics of base detector and robust detectors (with adversarial training) under four adversarial attacks.

Table 3 demonstrates the adversarial training performance. The test set used for evaluation in this table is generated by the four attack methods targeting at the based-detector, while the test set in table 2 is generated with the robust detectors as target.

[1]As Rodriguez et al.[15] showed that a few hundred labeled in-domain genuine and synthetic texts are sufficient for good performance.

Table 2: Attack to Base Detector and Robust Detector

	Origin accuracy		Attack accuracy		Avg. perturbed words	
recipes	base detector	robust detector	base detector	robust detector	base detector	robust detector
textfooler	99.50%	99.50%	0.50%	81.91%	11.83%	29.94%
textbugger	99.50%	99.50%	3.00%	43.00%	52.55%	61.81%
pwws	99.50%	99.50%	1.00%	53.00%	9.11%	26.18%
deepwordbug	99.50%	99.50%	0.50%	44.00%	7.76%	26.08%

Table 3: Testing of Robust Training

	fake	precision	recall	f1
textfooler	0	0.9897	0.9600	0.9746
	1	0.9791	0.9947	0.9868
textbugger	0	1.0000	0.9650	0.9822
	1	0.9804	1.0000	0.9901
pwws	0	1.0000	0.9650	0.9822
	1	0.9817	1.0000	0.9908
deepwordbug	0	0.9948	0.9550	0.9745
	1	0.9762	0.9973	0.9866

6 Analysis

6.1 Attack results analysis

The most likely reason for the salient difference between these two types of classification is the difference of importance distribution when the detector encodes fake texts and real texts. Since the attack recipes follow the word order strategy that ranks the words in a single text by their scores in the target model, where the larger change in prediction probability when the word is modified, the higher score the word will gain. In the BERT-based detector's classification task, the important words (high-score words) could be in a small portion, making the model decision concentrated on several words. Therefore, the attacks can easily find crucial targets to modify and interfere the confidence score in prediction. This idea is supported by the average percentage of perturbed words of textfooler and bae in table 1, as the percentages have significant growth in successful attacks on real text compared to those on fake text (6.91% to 13.72% and 5.66% to 17.45%).

The reason of pruthi's poor effect is possibly its small amount of perturbation. As claim in Pruthi et al.[8]'s paper, their algorithm only perform 1-character or 2-character attacks. We can also see in the average perturbed word of 1.47%, meaning that only one or two words are modified in the input texts which are up to 100 tokens in length. The paper also finds that BERT (a word-piece model) performs the best (over 90%) followed by word+char models, word-only models and then char-only models under their attacks.

The failure of checklist in most cases may be due to its transformation strategy: contract, extend, and substitute name entities (notice that Noun could be Adj. + Noun), as the main target of checklist is sentiment analysis, only raises small disturbance when BERT-detector classifying fake texts.

6.2 Adversarial training result analysis

Table 2 and 3 demonstrate the evaluation results on two different test set: the one in table 2 is generated using the robust detectors after adversarial training as the target model while the robust detectors are tested by adversarial samples using the original base detector as the target model in table 3. Remember that the target model serves as the word importance scoring tool in adversarial modification. Therefore, we can speculate from table 3 that through a directed training on the base detector, the adjustment of neural network parameters greatly reduces importance in decision-making of original high-scoring words, making most adversarial samples targeted at the base detector fail the attacks. Moreover, the adversarial training has minimal impact on the model recognition of undisturbed text, as the accuracy still stays around 99%. (More than half of the data in test set is unmodified text.)

Besides, the detectors after adversarial training perform significantly increased robustness against text attacks using themselves as target model, as the prediction accuracy under attacks and the average percentage of perturbed words in successful attacks tremendously grow (table 2). The large portions of words required to perturb greatly reduce the malicious attacks' efficiency, specially when the text size grows into a larger scale, showing robust detector's strong resistance to perturbations. When the perturbed portion is limited to 15% or lower, the successful attack rate may likely decrease in a huge step. The results are probably because the distribution of importance for decision-making becomes more diverse, and the word scores are more evenly distributed in the detectors after adversarial training, making it very difficult to fool the model by modifying small portion of words.

6.3 Transfer ability

Table 4: Performance of detector trained on Textfooler generating samples

	textfooler	textbugger	pwws	deepwordbug
Success rate	84%	51%	61%	53%
Avg. perturbed words	30.09%	75.63%	26.45%	23.79%

An interesting finding is that the adversarial training on adversarial samples generated by a specific recipe also enhance detector's robustness against other types of attacks. Table 4 shows sample results of evaluating detector trained on textfooler generating data on 100 AI-text samples modified by four attack recipes. The effects are likely due to the decentralization of importance (or weights) in model prediction among words during adversarial training. Therefore, it becomes difficult for all recipes to find small portion of crucial targets to disturb the model's output.

7 Conclusion

The robustness of machine-generated text against malicious attack is worth highlighting as text generating AI models are increasingly used by common people after the appearance of ChatGPT. As the existing detectors have shown great performance in detecting fake texts, they are vulnerable to adversarial attack techniques. In this project, I have trained a BERT-based detector on the HC3 and evaluated its robustness under attacks of seven state-of-the-art blackbox text attack recipes for untargeted classification. Moreover, I analyze the reasons of their superior or inferior performance, associating with specific algorithms.

To enhance robustness of the detector, I further apply adversarial training on the base detector, which significantly increases the model's prediction accuracy and average portions of perturbed words required for successful disturbance under the attack. Furthermore, I found that adversarial training with data generated by specific attack recipe also reduce model's vulnerability under other types of attacks, probably due to those attack recipes using similar core strategy to rank word orders for modification.

However, due to the limited GPU resources of Google Colab platform, the generation of adversarial data for training is set to 1000 samples per method. Though the training resources are limited, the adversarial training still contributes to a big step in robustness improvement. If more software resources are available in the future work, we can train the detector with larger scale of data to further improve its security. An other idea for follow-up work is to repeat the procedures of adversarial sample generation targeted at the new version of detector in each iteration and applying adversarial training, in which way the detector can iteratively optimize its weakness under text attack.

References

[1] 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.

[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] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

[4] J Li, S Ji, T Du, B Li, and T Wang. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*, 2019.

[5] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE Computer Society, 2018.

[6] Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, 2020.

[7] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097, 2019.

[8] Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, 2019.

[9] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020.

[10] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, 2020.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

[13] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, 2020.

[14] 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.

[15] Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 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, 2022.