A Foundation Model Approach to detect Machine Generated Text

Jonathan Pan
Home Team Science and Technology Agency, Singapore
Jonathan_Pan@htx.gov.sg

Abstract— **Large Language Models with autoregression generative capabilities like ChatGPT have garnered lots of attention from its launch. However, the cyber security community is also wary of the threats that it poses with cybercriminal and cyber security threat related activities. It could generate highly deceptive phishing and social engineering attacks that could evade human detection and render existing phishing or social engineering detection tools useless. Inspired by the approach used to develop Foundation Model that resulted with amazing capabilities from the contemporary model constructs like ChatGPT, our research endeavour demonstrates a model construct developed using Foundation model approach could yield potential as defensive tool to detect GPT generated text. Preliminary evaluation results show promising results.**

Keywords— Foundation Model; Machine Generated Text Detector; Transformer

I. INTRODUCTION

Since the introduction of ChatGPT and other similar autoregression generative models, the cyber security community has responded both positively and cautiously to this Artificial Intelligence (AI) advancement. The cautionary posture stems from the adversarial use of such AI models to generate effective adversarial and malicious attacks on their intended victims. This could be in form of human like crafted text for phishing email or other forms of social engineering attacks [1].

As with past developed and deployed cyber security solutions to detect such adversarial threats, researchers have started working on how to have detectors detect such malicious ingress feeds would be key to protect an individual and organization from such attacks. However, the challenge with such detectors is that these generative models used are constantly advancing towards being more human like where even humans would be challenged to distinguish AI generated text from human generated text [2].

This research work attempts to take on the challenge of detecting generated text using Large Language Models (LLM) like ChatGPT. The approach we took follows Stanford's framework for Foundation Model where a 'model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks' [3]. While our model is significantly smaller compared to popular LLM models like ChatGPT, we applied the framework to develop the 'small' foundation model to detect generated text that produced promising results.

In the next section, we will review related work about Foundation Model and some of prevailing research work and solutions developed thus far to detect AI generative text. This is followed by a coverage of the algorithm that we developed using Foundation Model approach and details of the experimentation setup with its evaluation. This paper concludes with a conclusion and its future research direction.

II. RELATED WORK

In this section, we review the current development with Foundation Model and detecting generated text.

A. *Foundation Model*

Stanford Institute for Human-Centered Artificial Intelligence (HA) coined the term Foundation Model to represent any model that is trained with broad data typically using either unsupervised or self-supervised as a upstream task and later fine-tuned to a narrower task as a downstream task [3]. Stanford argues that while such approach of developing model has been used prior to the use of this term, however they noted the trend towards where the potential of one model trained with large amount of unlabeled data to be used in many different purposes [4]. The notable examples include ChatGPT and BERT. Aside from these natural language processing or text based AI models, there are also other forms of pretrained foundation models trained with computer vision and graph learning task.

B. *Need to Detect Generated Text*

With the advancement of natural language generation AI models, Crothers et. al argues that the quality of machine generated text has improved significantly [5]. With the availability of such models, they could be used for malicious intent to create content to achieve its intended objective with their targets. It is observed that human performance in detecting machine generated text is relatively poor [6]. This is especially so when human evaluators are not trained to detect such generated text.

Even with additional training that included example-based trainings to human detectors, there was little significant improvement with such detections.

With the advancement of generative AI, Crothers et. al modelled the potential risks of adversarial applications of machine generated text.

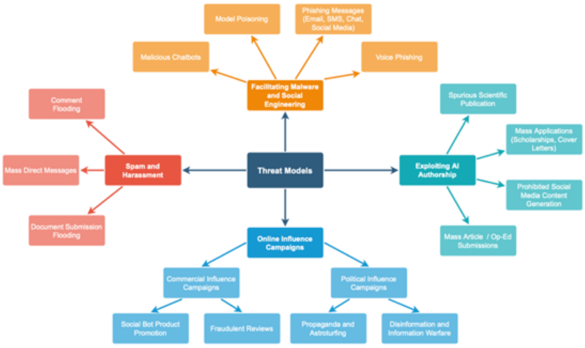
Model Poisoning
Phishing Messages (Email, SMS, Chat, Social Media)
Malicious Chatbots
Voice Phishing
Facilitating Malware and Social Engineering
Comment Flooding
Spurious Scientific Publication
Mass Applications (Scholarships, Cover Letters)
Mass Direct Messages
Spam and Harassment
Threat Models
Exploiting AI Authorship
Prohibited Social Media Content Generation
Document Submission Flooding
Mass Article / Op-Ed Submissions
Online Influence Campaigns
Commercial Influence Campaigns
Political Influence Campaigns
Social Bot Product Promotion
Fraudulent Reviews
Propaganda and Astroturfing
Disinformation and Information Warfare

Figure 1. Threat Model with machine generated text [5]

Coupled with the lack of human abilities to detect such content, there is an urgent need to develop and deploy technical solutions to detect such deceptive malicious content. Even with such detection solutions, Crothers et. al further argues that there are significant challenges posed to such solutions like class imbalance and continued advancement of generative model that could affect the detection accuracy performance which in turn could cause larger potential harms. Our work attempts to build a detector based on the way such generative AI models are developed, more specifically using Foundation Model framework.

III. Model

Our model composes of three layers of Transformer encoder [7] and a single layer of fully connected classical neural network layer with a Softmax activation function to perform the classification task of recognizing generated text vs human text. The model, though small, is trained as a Foundation model using an upstream task using self-supervised learning approach. The model is then fine-tuned for this specific task. The model ingests textual data using character encoding that removes the need for any preprocessing of training or test data from the LLM or human generated text.

A. *Upstream Task*

This stage of model development involves the training of a newly instantiated Transformer model with three layers of encoders capped with a single layer of fully connected neural network layer. The training data used is human generated textual data. The character encoding is used to convert textual data into vectorized arrays.

As self-supervised learning is used to first train the model for feature learning, the pretext task is to compare the original textual data against augmentation of these data. We applied natural language augmentation using the NLPAug tool [8] where y_i is either 0 or 1 to represent human or generated text and $\mathcal{L}$ is the loss function for this binominal classification task.

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{nK} \sum_{i=1}^{n} \sum_{k=1}^{K} \mathcal{L}\left(f(x_i;\theta), y_i^k\right) + \lambda\Omega(\theta) \qquad (1)$$

B. *Downstream Task*

This stage entails the fine tuning of the model to distinguish between computer generated text from human generated text. We applied transfer learning to realign the model's prior upstream training (in recognizing augmented text from non-augmented ones) to this task. We froze the Transformer and allowed only the fully connected layer to be finetuned.

IV. Methodology and Analysis

For our experiment, we used open-sourced datasets and trained our model on that dataset. We evaluated the model's performance using Accuracy and F1 measurements.

A. *Dataset*

We used the dataset published on Hugging Face [10] that contained Wikipedia introductions curated by the community and GPT (Curie) generated introductions on 150 thousand topics. The following are two samples of the introduction texts from Wikipedia and GPT respectively.

Wikipedia	GPT Generated
"In combat sports such as boxing, an orthodox stance is one in which the boxer places their left foot farther in front of the right foot, thus having their weaker side closer to the opponent. Because it favors the stronger, dominant side—often the right side, see laterality—the orthodox stance is the most common stance in boxing and MMA. It is mostly used by right-handed boxers. Many boxing champions have fought in an orthodox stance. [More text follows]"	"In combat sports such as boxing, an orthodox stance is a standing position with the feet slightly wider than shoulder-width, the weight shifted towards the ball of the foot, and the hands held close to the body. The orthodox stance is considered to be one of the most effective stances in boxing. It allows a boxer to cover more ground with their footwork, and also keep their opponent at a distance by preventing them from landing clean punches."
"In mathematics, an almost periodic function is, loosely speaking, a function of a real number that is periodic to within any desired level of accuracy, given	"In mathematics, an almost periodic function is, roughly speaking, a function whose behavior appears to be periodic, but for which there is

suitably long, well-distributed "almost-periods". The concept was first studied by Harald Bohr and later generalized by Vyacheslav Stepanov, Hermann Weyl and Abram Samoilovitch Besicovitch, amongst others. [More text follows]”	not a precise mathematical description. Such functions can be difficult to analyze and predict, and their properties are not always well-understood."

Table 1. Samples of the Introduction Texts from Wikipedia and GPT

We analyzed the profile of the dataset and noticed that the GPT generated introduction is generally shorter than those from Wikipedia (Table 2). This is lead to incorrect generalization of the model. Hence, we introduced a constant length constraint to the text to be analysed. For our experiment, we set the constraint length to 700 characters.

Text Length	Wikipedia	GPT generated
Average	1201	784
Standard Deviation	271	350
Maximum	4042	2034
Minimum	425	35

Table 2. Characteristics of the dataset

B. *Evaluation Metrics*

As the dataset used has binary classification labels (Wikipedia or Generated) and contains balanced classes of data, we evaluated the model using Accuracy measurement (Equation 1). TP (True Positive) represents the number of correctly classification of generated text. TN (True Negative) represents the number of correctly classification of Wikipedia text. FP (False Positive) is the number incorrect classification of Wikipedia text as generated text. FN (False Negative) is the number of incorrect classifications of generated text as Wikipedia text.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

Additionally, we computed the Precision, Recall and F1 score for this classification to assess the model’s general inference inclination if any. The Precision measurement provides the extent of Type I error while Recall measurements provides the extent of Type II error. We used *F1 score* to measure the harmonic mean of *precision* and *recall.*

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{5}$$

C. *Model Preparation and Evaluation*

As mentioned in the previous section, we first trained the model in the upstream stage using self-supervised approach with only the Wikipedia introduction text with NLP augmentation applied. We then fine-tuned the model by using transfer learning to the segment of the dataset with labels exposed.

We evaluated our model with four tests. The first test is to evaluate the performance of the model against the designated test dataset (that is 20% of the entire dataset) of the model with upstream training and the model with both upstream and downstream. The objective of this test is to assess the improvement gained from the downstream training. The second test compares the test results of the test dataset being extended from 20% of the dataset to 80% of the dataset. This test was used to evaluate the model’s performance of detecting other GPT generated text within the dataset that the model has not been trained on and infer preliminary the generalization of the model. The third test is to assess the model’s performance when it is trained using Foundation Model approach (that is to use self-supervised learning approach) and classical model development approach (that is to use supervised learning approach). The final evaluation compares the results of our model with another that was recently evaluated due to the recency of GPT advancement.

D. *Results and Analysis*

From our experiment test, we observed the model improved measurably after the model was fine-tuned for the downstream task of classifying the source of the text (Wikipedia or generated) in comparison to its initial upstream training of having only the human and augmented text to train (Chart 1).

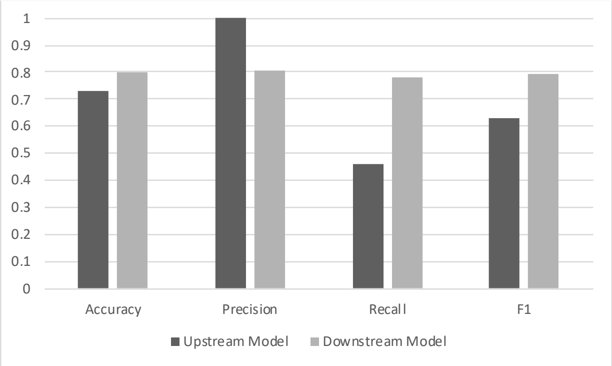

Chart 1. Evaluation performance comparison between Upstream and Downstream Trained Model

Additionally, the model performed consistently well when the rest of the unseen generated text were given to the model for further evaluation when the model had ‘sight’ to only 20% of the dataset with generated text (Chart 2). We opined that our model could generalize well with only relatively small set of labels.

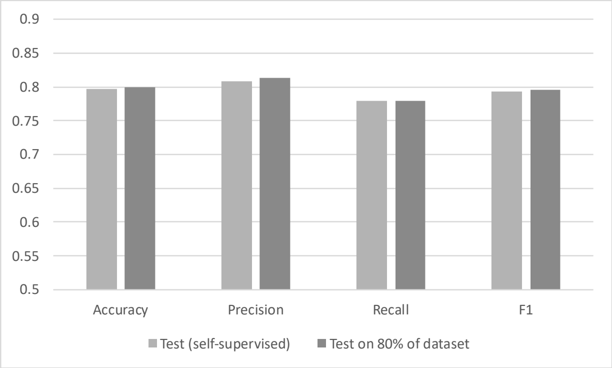

Chart 2. Performance analysis of model against 20% and 80% of dataset

We further compared the performance of the model when we trained it using classical supervised learning approach where the model was provided labeled data for its primary training with 80% of the dataset and 20% for model validation cum testing. We noted that the model performed poorly as compared to the self-supervised approach (Chart 3). Hence, we infer that the textual augmentation to our model construct led to better and consistent performance. This highlights potentially a new approach to train such model construct.

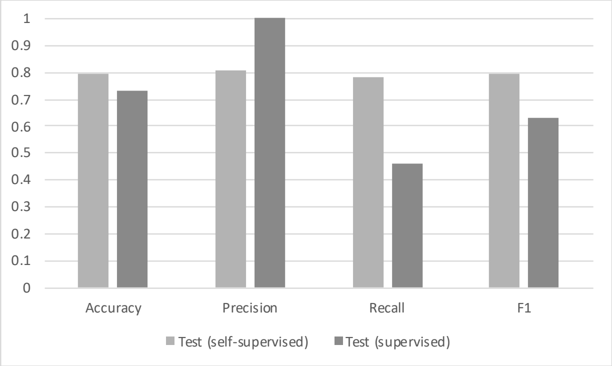

Chart 3. Performance analysis of self-supervised vs supervised

We finally compared our model's evaluation results against other model constructs. It should be noted that the other research work with their model constructs by Soni and Wade [9] used a different dataset from ours however still with human and GPT generated texts.

While the other model performed better in the Accuracy measurement with their Distill-BERT model [9], our model had consistent results with both Accuracy and F1 measurements (Table 3).

ML Model	Accuracy	F1 Score
SentTrans. +XGB	0.50	0.60
Distill-BERT	0.90	0.33
Ours	**0.80**	**0.79**

Table 3. Evaluation performance comparison [9]

V. Conclusion and Future Directions

Using the Foundation Model framework approach, we first trained our model (small one) using only human generated text (from Wikipedia) with self-supervised learning as part of the upstream task. For the downstream task of classifying GPT generated text, we fine-tuned the model using Transfer Learning approach on a small set of generated and Wikipedia text. Our model performed consistently well to classify generated text.

While the model shows promising results from our preliminary work, we will need to further improve the model or improvise new model construct to deal with the continued advancement of growing Large Language Models to generate human like text so that we can deal with the adversarial threats when such AI advancements are used for malicious intent.

References

[1] A. Irei, and A. Krishnan, "5 ChatGPT security risks in the enterprise", TechTarget, https://www.techtarget.com/searchsecurity/tip/ChatGPT-security-risks-in-the-enterprise, Apr. 2023.

[2] R. Rogers, "How to Detect AI-Generated Text, According to Researchers", Wired, https://www.wired.com/story/how-to-spot-generative-ai-text-chatgpt/, Feb. 2023.

[3] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill et al., "On the opportunities and risks of foundation models", arXiv:2108.07258 [cs.LG], 2021.

[4] M. Murphy, "What are foundation models?", IBM Research Blog, https://research.ibm.com/blog/what-are-foundation-models, May 2022.

[5] E. Crothers, N. Japkowicz and H. Viktor, "Machine Generated Text: A Comprehensive Survey of Threat Models and Detection Methods", arXiv, arXiv:2210.07321, Feb. 2023.

[6] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan and N. A. Smith, "All That's Human Is Not Gold: Evaluating Human Evaluation of Generated Text", arXiv, arXiv:2107.00061, Jul. 2021.

[7] A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998–6008.

[8] E. Ma, NLP Augmentation, https://github.com/makcedward/nlpaug, 2019.

[9] M. Soni and V. Wade, "Comparing Abstractive Summaries Generated by ChatGPT to Real Summaries Through Blinded Reviewers and Text Classification Algorithms, arXiv, arXiv:2303.17650, Mar. 2023.

[10] A Bhat, "GPT-wiki-intro", https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro, 2023.