Adversarial Learning for AI Text Detection

Reshma Ramachandra
Georgia Institute of Technology
Atlanta, USA
reshmaram@gatech.edu

Aryan Vats
Georgia Institute of Technology
Atlanta, USA
avats31@gatech.edu

Deeksha Manjunath
Georgia Institute of Technology
Atlanta, USA
deekshamanjunath@gatech.edu

Abstract

Lately, Large Language Models (LLMs) are being explored quite extensively for text generation. However, with the advent of such technologies, it is very important to prevent misinformation. Thus, detection of text generated by Artificial Intelligence (AI) is necessary to distinguish between actual and fake information. This project proposes a model that can identify if the given text was AI-generated or not using adversarial learning. Inspired by the RADAR framework [4], we train two networks, an ensemble paraphraser and a detector, that compete with one another to learn and enhance the efficiency of the detection model. The proposed approach outperforms the RADAR implementation on the Wiki Intro Dataset[1] with an AUROC score of 0.78.

1. Introduction

In recent times, the widespread adoption of LLMs for text generation, exemplified by the popularity of ChatGPT, has led to a paradigm shift in information retrieval. While these models offer transformative benefits, they also pose significant challenges and potential drawbacks. One major concern revolves around the inadvertent generation of misleading or false information, as LLMs may produce content lacking accuracy or context [1, 9, 17]. This issue extends to academia, where the use of AI-generated content may contribute to a decline in research quality. Moreover, the possible misuse of such models raises concerns about academic dishonesty and cheating. Additionally, the challenge faced by fact-checkers in discerning AI-generated content from genuine information further exacerbates the potential for misinformation.

Consequently, there is a pressing need for the development of effective detection mechanisms to identify text generated by Artificial Intelligence (AI), particularly models like ChatGPT. Despite recent efforts to address this problem, existing approaches have neither yielded satisfactory performance for GPT-generated text nor have been generalizable. This project aims to fill this gap by leveraging Adversarial Learning to create a robust model capable of accurately discerning whether a given text was generated by the latest open-source GPT model [8] or not.

Our approach for AI-text detection is inspired by the the RADAR framework [4] which proposes the idea of the usage of Adversarial Learning akin to the idea behind Generative Adversarial Networks (GANs). In this approach, we train two models, an ensemble paraphraser and a detector, that aim to work against each other and attempt to improve their own loss curves and policy that in turn fortifies their learning. The Wiki Intro Dataset [1], consisting of human-written and GPT-generated Wikipedia introductions for a given topic, is used to train an Adversarial Learning-based detector model. About 10% of the data is used to test if the detector model can accurately classify given text as GPT-generated or not. The main objectives include:

- Design a robust AI-text detection model trained using Adversarial Learning that provides accurate results across various text-generation models.
- Implement a novel ensemble paraphraser model to generate paraphrases of the given input text attempting to avoid detection. While the RADAR implementation utilizes the Text-to-Text Transfer Transformer (T5-large) paraphraser, we combine the outputs of the Text-to-Text Transfer Transformer (T5-large) model fine-tuned for paraphrasing [2], the Bidirectional and Auto-Regressive Transformers (BART) model [5] fine-tuned for paraphrasing [3] and the Pegasus [16] model fine-tuned for paraphrasing [4] to enhance the quality of the information fed to the detector model.

[1] https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro

[2] https://huggingface.co/ramsrigouthamg/t5-large-paraphraser-diverse-high-quality

[3] https://huggingface.co/eugenesiow/bart-paraphrase

[4] https://huggingface.co/tuner007/pegasus_paraphrase

- Design an AI-text detector model initialized with the RoBERTa model [6] fine-tuned for AI-text detection [5]. Unlike the original RADAR approach, which used the pre-trained RoBERTa version, this modification is intended to enhance the detector's performance. The detector and the ensemble paraphraser are trained in an adversarial manner, where each model influences the training of the other.

- Incorporate a Reinforcement Learning (RL) algorithm, specifically Proximal Policy Optimization (PPO) [11], into the adversarial learning process. We compare this approach against a baseline method that solely focuses on training the detector network without employing adversarial learning. This comparative analysis aims to assess the influence of adversarial learning, particularly through PPO, on the task of detecting AI-generated text.

By introducing PPO as part of the adversarial learning, we seek to understand how this reinforcement learning technique enhances the performance of the model compared to the baseline, which relies solely on detector network training. This will provide insights into the effectiveness of adversarial learning strategies in the context of AI-text detection.

If successful, this project will significantly enhance the reliability of online information by delivering a robust AI-text detection model. This would help foster trust in content by accurately distinguishing between AI-generated and human-authored text, preventing the spread of misinformation, improving fact-checking capabilities, and ensuring adaptability across evolving text-generation models. Additionally, the accessibility aspect is crucial—unlike many existing AI-text detection tools that come with associated costs [6] [7], the proposed solution aims to provide a valuable resource freely.

2. Related Works

There have been several attempts at detecting AI-generated text in the past. Pegoraro et al. [10] outlines most of the recent work done in AI text detection and their drawbacks. These approaches range from simple classifiers based on Logistic Regression [2], transformer-based models such as BERT [15] and RoBERTa [2] and several online tools such as ZeroGPT [8] and Perplexity (PPL) [9]. The overarching problems with these approaches include imbalanced datasets with not enough AI-generated text, tested on fake news and fake reviews datasets but not on actual text generated by GPT models and very poor detection performance. Moreover, the online tools that perform better comparitively are usually paid. The Wiki Intro Dataset, that we use, however, contains about 150K human-written and corresponding GPT-generated texts,ensuring relevant training.

Wang et al. [13] utlizes BERT and fine-tuned RoBERTa for detecting fake news generated by ChatGPT. Pre-training in RoBERTa focuses on predicting next few words given current word which helps in understanding the language structure better. Although the overall classification accuracy is about 97%, the precision values for the fake news category is much lower. This is due to the highly imbalanced dataset used with only about 30% of the data generated by AI. Our approach uses equal amounts of human and AI-generated datapoints to ensure stability and avoiding a skew in data.

Mitchell et al. [7] introduces DetectGPT, a novel approach for zero-shot machine-generated text detection using probability curvature. The method relies on analyzing the probability distribution of text generated by a language model and detecting deviations in the distribution's curvature, which often occur in machine-generated text due to its distinct characteristics. However, AI generated text is diverse and continually evolving, and relying solely on probability curvature may not capture all the nuanced characteristics of AI generated content accurately.

Our approach uses adversarial learning to counter the limitations of relying on probability curvature by training the model with intentionally deceptive examples, improving its ability to recognize and adapt to nuanced characteristics in machine-generated text beyond curvature, leading to increased robustness and reduced false positives [4].

Most recently, Hu et al. [4] introduced RADAR (Robust AI-Text Detection via Adversarial Learning), a framework leveraging Adversarial Learning inspired by Generative Adversarial Networks (GANs). RADAR involves training two networks, a paraphraser and a detector, in a mutually adversarial manner. These networks strive to enhance their loss curves and policies by working against each other, thereby strengthening their learning. The paraphraser's role is to generate diverse versions of given text, attempting to deceive the detector into misclassifying the text as human-written. The log probabilities of the detector's predictions serve as rewards for the paraphraser during training. This adversarial approach aims to improve the overall robustness and effectiveness of the RADAR framework.

[5] https://huggingface.co/roberta-large-openai-detector
[6] https://www.zerogpt.com/pricing
[7] https://copyleaks.com/pricing
[8] https://www.zerogpt.com/
[9] https://www.perplexity.ai/

While RADAR claims to have achieved state-of-the-art performance on the WebText dataset [10], it faced a significant challenge when applied to the Wiki Intro Dataset[11]. In this evaluation, RADAR was found to accurately identify AI-generated text in less than 50% of the randomly sampled 100 test cases. Thus, the proposed approach is designed to address specific areas for improvement identified in the RADAR framework that include,

– RADAR utilizes pre-trained T5-large and RoBERTa-large models. When models are fine-tuned on task-specific data, they become finely attuned to the nuances and patterns relevant to the problem, resulting in better performance. This could enhance the model's contextual awareness, enabling it to make more accurate predictions that are tailored to the task's intricacies [3] [14]. Thus, initializing these models with models fine-tuned to their respective tasks, as implemented in the proposed approach, could help in improving performance.

– RADAR utilizes the T5-large paraphraser to create diverse paraphrased versions of given text in order to improve the detector's ability to handle various text forms which, in turn, enhances the detector's real-world performance. Introducing an ensemble paraphraser that incorporates outputs from multiple paraphrasing models with diverse language architectures, further enriches the input variety for the detector. This approach ensures the detector's robustness across a spectrum of AI-generated text, contributing to improved adaptability and overall effectiveness in diverse scenarios.

3. Approach

We propose an Adversarial Learning approach for detecting AI-generated text as shown in 1. The human-written Wikipedia introductions from the Wiki Intro Dataset constitutes the $\mathcal{H}$ corpus and the corresponding AI-generated text constitutes the $\mathcal{M}$ corpus. Drawing inspiration from the RADAR network [4], we have two LMs,

- Samples x_m from $\mathcal{M}$ are fed into the Ensemble Paraphraser $\mathcal{P}'_\phi$. This is called as an ensemble model since it generates paraphrased texts of the given text by utilizing three LMs: fine-tuned T5-large, BART and Pegasus. The sentences are paraphrased using the same prompt as in RADAR (to maintain consistency), "Enhance word choices to make the sentence sound more like a human". This aims to distort the text in an attempt to circumvent detection by the detector model.
- The human-written text samples x_h, AI-generated text samples x_m and the the paraphrased texts x_p are fed into the Detector $\mathcal{D}'_\phi$ which aims to predict the probabilities of given input text being AI-generated or not. The log of these probabilities are used as rewards in updating the corresponding paraphrasers by following Proximal Policy Optimization (PPO). The average loss of the predictions are fed back to the detector $\mathcal{D}'_\phi$ during backward propagation.
- Finally, while training, 10% of the Wiki Intro dataset is used to test the detector model. The paraphraser and detector models are updated until there is no improvement in the area under the receiver operating characteristic (AUROC) value.

Thus, with the updates, the new reward can be defined as

$$R(x_p, \phi) = \mathcal{D'}_\phi(x_p) \in [0, 1]$$

where x_p represents a set of paraphrases $\{x_p^1, x_p^2, x_p^3\}$. The reward $R(x_p, \phi)$ is thus calculated for each paraphrase in the set. The log probability is then computed as

$$\log P_{\mathcal{P'}_\sigma}(x_p|x_m) = \sum_{i=1}^{N} \log P_{\mathcal{P'}_\sigma}(x_p^i|x_m, x_p^{1:i-1})$$

where the log probabilities for each paraphrase in the set x_p are summed up separately. Here $\mathcal{P}$ represents the corpus formed by the paraphrased texts.

The paraphrasers are rewarded for successfully tricking the detector into thinking their generated text is human-like. So, during the training of each paraphraser (let's call it $\mathcal{P}_i$), we use the log probability of the detector predicting that the paraphrased text is human-written as a reward. For the detector update, the loss is computed by averaging the loss on human-written text $L_\mathcal{H}$, the loss on AI-generated text $L_\mathcal{M}$ and the loss on the three paraphrased AI-texts $L_\mathcal{P}$ generated by the three paraphrasers in the ensemble paraphraser.

In simpler terms, the paraphrasers are rewarded for making their generated text look human-like to the detector, and the detector learns from the combined losses on human-written text, AI-generated text, and paraphrased AI-texts.

PPO is a superior choice for incorporating rewards in training paraphrasers in adversarial learning. Compared to TRPO, PPO excels in scalability, robustness, and data efficiency. TRPO, while reliable, faces limitations in scalability and architectural compatibility for adversarial tasks [11]. PPO's clipped probability ratios provide a pessimistic estimate, ensuring a lower bound on policy performance. The alternating sampling and optimization

[10] https://huggingface.co/datasets/Skylion007/openwebtext
[11] https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro

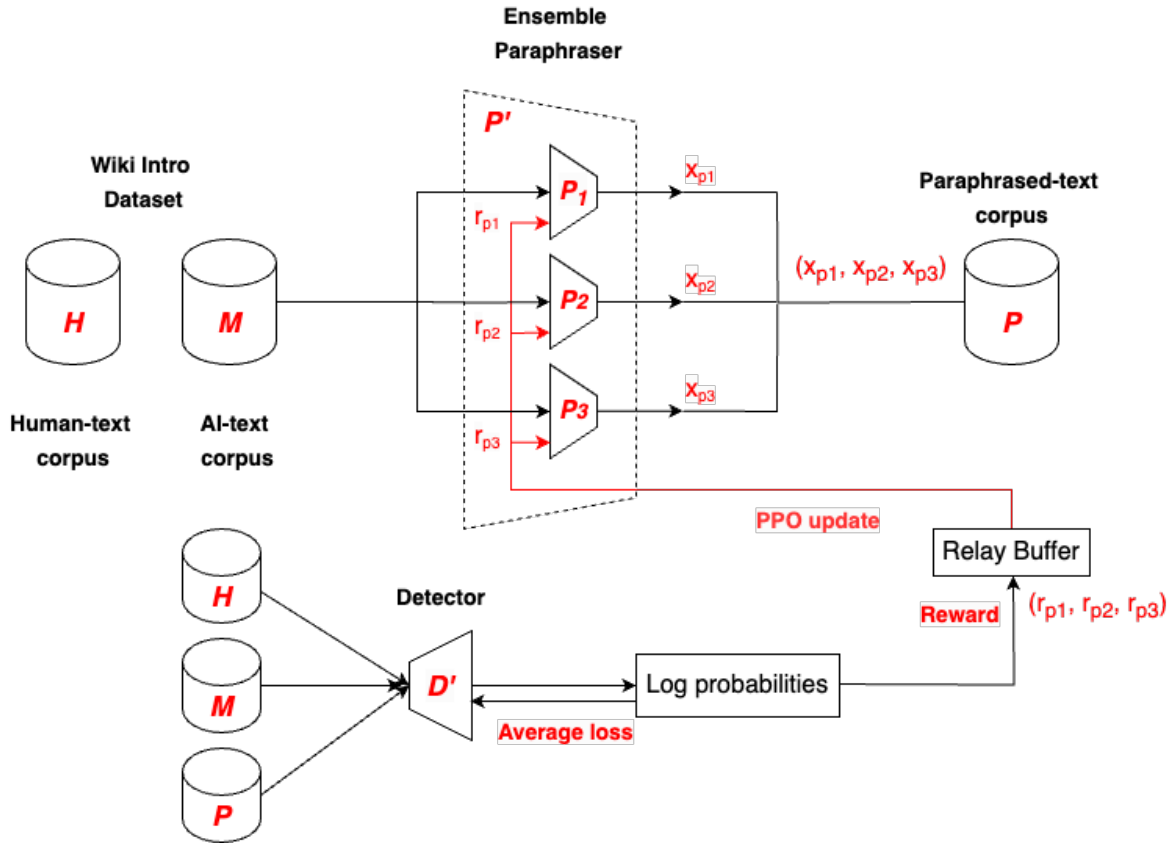

Figure 1. An overview of the proposed approach.

contribute to robustness, and its emphasis on first-order optimization streamlines implementation. Further, RADAR's implementation [4] of adversarial learning that utilizes PPO outperforms other existing AI-text detection methods which do not use PPO.

The baseline approach as shown in 2, exclusively focuses on training the detector network, which is initialized with the pre-trained RoBERTa model [6]. Additionally, it incorporates a single paraphraser model, specifically the pre-trained T5-large model, to augment the information presented to the detector. Here pre-trained is used to demonstrate the impact of using a fine-tuned paraphrasing model over a normal pre-trained paraphraser model. Notably, this baseline approach does not involve any adversarial learning. The purpose of this baseline approach is to assess the impact of introducing adversarial learning and ensemble paraphrasing on the task of AI-text detection. By contrasting this baseline with the proposed adversarial learning approach, we aim to understand whether the performance of the detector improves when trained concurrently with an ensemble paraphraser and if adversarial learning contributes to enhanced detection capabilities.

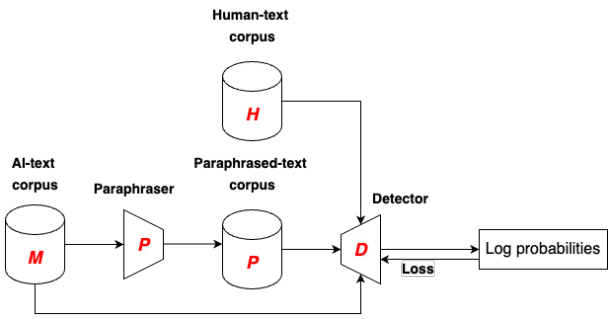

Figure 2. An overview of the baseline approach.

Moreover, recent work has shown that adversarial learning is effective for training models to detect AI-generated text because it enables the model to learn from both genuine and adversarial examples, enhancing its robustness [4]. This approach exposes the model to diverse inputs, improving its ability to discriminate between real and AI-generated text.

The authors of RADAR [4] provide a detailed analyis of

RADAR's performance and its comparison with previous state-of-the-art approaches for AI-text detection [7, 12] using the the area under the receiver operating characteristic (AUROC) metric. This score is a way to measure how well a text detector can distinguish between AI-generated and human-generated text. By varying the threshold used to classify text, we can see how well the detector balances correctly identifying AI-generated text (true positives) and incorrectly flagging human-generated text (false positives). A higher AUROC score indicates a better-performing detector, while a score of 0.5 means the detector performs no better than random guessing. Thus, the AUROC metric will be used to evaluate the proposed approach as well to maintain consistency with the original RADAR implementation and for easy comparison with RADAR's performance on the validation dataset.

Since RADAR was not trained originally on the Wiki Intro dataset, we get its predictions for the validation dataset by utilizing the pre-trained model [12] recently made publicly available.

4. Data

The Wiki Intro Dataset [13] is used to train the model. It comprises of 150k topics, with a varied distribution of machine-generated text, created by GPT (Curie) model and human-written text. These samples cover various domains, collected from Wikipedia. A prompt was used to generate the GPT response using the title of the Wikipedia page and the first seven words from the introduction paragraph. Prompt used for generating text:

200 word Wikipedia-style introduction on '{title}'
{starter_text}

where {title} is the title for the Wikipedia page, and {starter_text} is the first seven words of the Wikipedia introduction.

The dataset also has useful metadata such as title length, wiki intro length, generated intro length, etc. With the aforementioned huge collection of text data, especially with equal amounts of data belonging to both the human-written and AI-generated classes, the Wiki Intro Dataset seems promising to effectively train models for AI text detection. Table 1 details the configuration of the GPT model that was used to generate the AI-generated text in the Wiki Intro Dataset.

5. Experimental Setup

To understand how our proposed model performs, we implement a simple baseline that trains the detector without any updates to the paraphrasers. This is done to clearly outline the effect Adversarial Learning has on the detector's performance. We describe the models, and their training specifications below.

[12] https://huggingface.co/TrustSafeAI/RADAR-Vicuna-7B
[13] https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro

Table 1. GPT Model Configuration Parameters

Parameter	Value
model	"text-curie-001"
temperature	0.7
max_tokens	300
top_p	1
frequency_penalty	0.4
presence_penalty	0.1

5.1. Model 1: Baseline - Probabilistic Loss for Detector Training

The baseline methodology as shown in 2 relies on a straightforward deep learning approach. The paraphraser model is frozen, generating paraphrased inputs as data augmentation for the model to train with. The detector trains on this data, attempting to predict whether the text is AI-generated or not. The detector model trains on Mean-Squared Error loss(MSE), and the training backpropagates through the Value-Head, across the entire LM.

These hyperparameters were selected through a combination of data-exploration and the observation of the loss-curves. Without the adversarial training, the model took much longer to converge to minima. The use of a more aggressive learning rate allowed the model to converge quicker, with the batch size chosen to avoid resource issues on GCP. The rest of the parameters were mainly kept the same as default, with the exception of the lambda value which was used upon reference from the RADAR paper.

Table 2. Baseline Model Training Parameters

Hyperparameter	Value
Epochs	4500
Batch Size	200
Learning Rate	1e-4
Epsilon	0.2
Gamma	1.0
Lambda Value	0.1
Optimizer	Adam

5.2. Model 2: Detector-Ensemble Paraphraser with PPO

Here, the detector model, a fine-tuned implementation of the RoBERTa architecture, goes through the training loop to distinguish between AI-generated and human-written

text. Simultaneously, an ensemble of paraphraser models is employed. These paraphrasers rephrase AI-generated text, aiming to evade detection by the aforementioned detector model. The paraphrasers undergo training where the output from the detector serves as a reward signal via the Proximal Policy Optimization (PPO) framework. This incentivizes the paraphrasers to generate text that the detector is less likely to classify as AI-generated.

The reward function utilized the probability that the paraphrased text is human-generated, whilst the detector used MSE Loss for training.

With adversarial learning the model converged much quicker, requiring half as many epochs. A less aggressive learning rate sufficed in achieving a quicker minima convergence, avoiding overfitting.

Table 3 summarizes key hyperparameters used in the detector model training and Table 4 details the parameters related to the PPO update of the paraphrasers.

Table 3. Detector Model Training Parameters

Hyperparameter	**Value**
Epochs	2700
Batch Size	200
Learning Rate	1e-5
Epsilon	0.2
Gamma	1.0
Lambda Value	0.1
Optimizer	Adam

Most of the PPO hyperparameters were kept as default. The following parameters ensured that the training mechanism was not too strict in punishing bad performance, which led to the model producing sentences that did not make sense.

Table 4. Paraphraser Update Parameters

Hyperparameter	**Value**
Steps per epoch	20000
Learning Rate	1e-5
Adaptive KL Control	True
Initial KL Coef	0.2
Gamma	1
Lambda	0.95
Batch Size	2

To optimize training and avoid running into resource issues, we utilized the Google Cloud Platform(GCP) to avail GPU's to train our model. We used a 'g2-standard-24' Machine Type, using two 'Nvidia L4 GPU's as accelerators, with a Disk Space of 100GB.

For evaluating the outputs, we utilized another machine so as to avoid running into resource issues. We used a 'g2-standard-16' Machine Type supported with one 'Nvidia L4 GPU' accelerator, with a Disk Space of 100GB.

These machines were necessary since a normal 'Nvidia T4' GPU alongside the regular Colaboratory setup was found to be insufficient to conduct effective training. Alongside other issues, to quicken the pre-processing and the training loop, we were able to effectively utilize the credits provided for the project to yield sufficient training.

We used PyTorch for training, validation and saving our model checkpoints.

6. Results and Inference

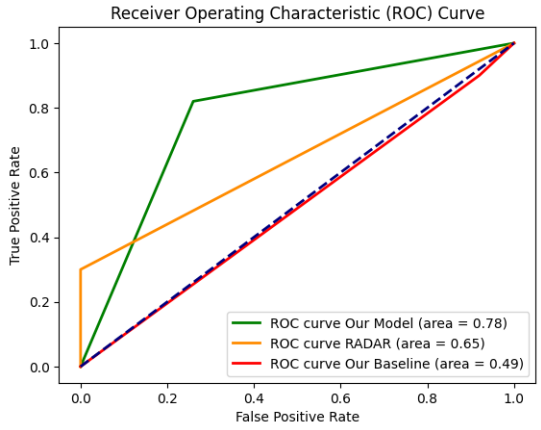

Figure 3. AUROC curves (Our Model refers to the proposed approach)

We plot the AUROC curve, as shown in Fig. 3, for both the baseline implementation and the proposed approach to compare the performances with the RADAR's outputs for the validation set. All three models (RADAR, baseline, proposed approach) were validated on 15,000 datapoints from the Wiki Intro dataset, with the probabilities and true labels used to plot the AUROC curve.

We can clearly see the effects of Adversarial Learning in making the detector robust against different data sources. When just trained on the detector's prediction probabilities, the Baseline model was unable to perform better than normal randomization, with an AUROC score of 0.49. RADAR and the proposed approach both performed better, with RADAR struggling with being able to yield effective True Positive rate. The proposed approach performed significantly better on the Wiki Intro dataset, yielding an AUROC score of 0.78.

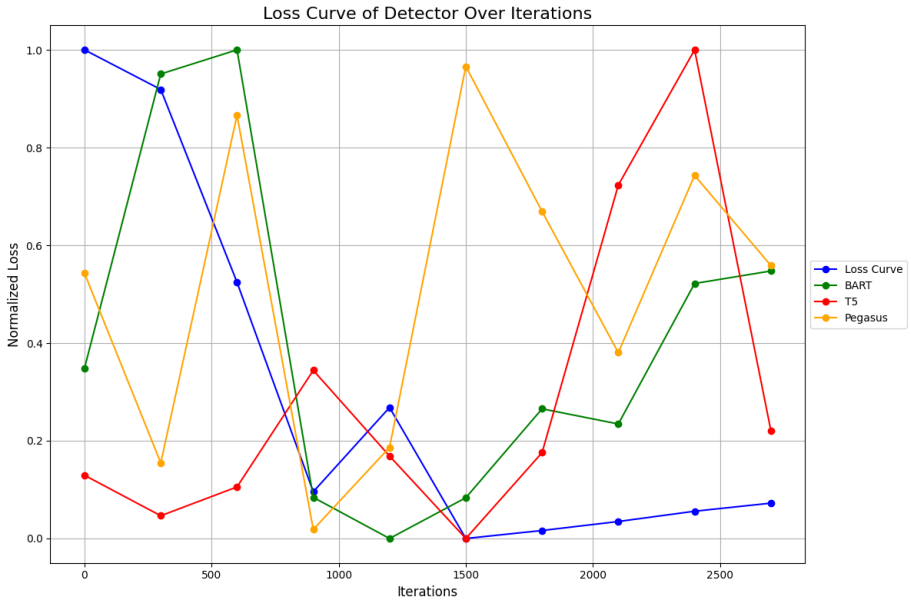

Figure 4. Loss curves of the detector and the paraphrasers

The loss curve appropriately displays the adversarial nature of training, with the paraphrasers and the detector working against each other. The paraphrasers are rewarded for thwarting the detector whilst the detector has to not only thwart the paraphrasers but also maintain a good prediction on human-generated text and normal AI-generated text. At the start of training at around 100-200 epochs, the detector starts out at higher loss while the paraphrasers are relatively lower. During training, the detector starts learning to predict the AI-generated text as well as the paraphrased text, while the losses for the paraphrasers start increasing simultaneously. As the detector starts doing better, we see that while the paraphraser losses climb, the detector loss starts to decrease steeply. Around epoch 600, we see a sharp curve in both directions, the detector having low loss due to performing well, consequently the paraphrasers suffering high losses due to being unable to thwart the detector successfully.

We also see the effects of the paraphrasers doing well, during the 1200 epoch mark where all paraphrasers score low losses and beat the detector, which has a higher loss. While the increment of loss in paraphrasers was pronounced, the detector loss spike is much more subdued. This is due to the fact that the detector trains on the average loss of performance on the AI generated text, paraphrased text and the human generated text. The detector reaches the plateau of it's learning curve near the 2000 epoch mark, whereas the paraphrasers continue to fluctuate in their loss curve. We can attribute the fluctuation of the loss curves of the paraphrasers to the nature of RL training, with the paraphrasers possibly suffering from catastrophic forgetting around the 2300 epoch mark. Due to their inability to thwart the detector, the paraphraser losses remain significantly higher, while the detector loss has reached it's minima.

Overall, due to the nature of the dataset, the approach generalized well to data of different domains. Factual introductions of people, and scientific explanation of phenomena were both a part of the data and in general the proposed approach generalized well to data of different domains. However the data did not include special symbols (mathematics, code snippets) and therefore may not generalize to alphanumeric data that lie outside the realm of pure English language.

7. Conclusion and Future Work

In this research, we delved into AI text detection using adversarial learning, leveraging the RADAR framework as our guiding research. Our approach involved training a detector model alongside an ensemble of paraphrasers, to discern between human and AI-generated text. From experimentation shown above, the proposed approach outperforms the baseline implementation which did not involve any adversarial learning and the RADAR approach. This indicates the impact of reinforcement learning and ensemble paraphrasing on improving the detector's performance. The insights shed light on the effectiveness of adversarial approaches in tackling AI-generated text detection. This work holds potential for combating misinformation and enhancing reliability on online information.

Future work can include the utilization of a different RL framework during paraphraser training. A better framework for improving paraphraser outputs would in turn make the detector more robust to unique ways of avoiding AI-text detection. Furthermore, the realm of AI-text detection is not limited to introductions and paragraphs. Poetry, code snippets, mathematical proofs are also susceptible to AI-generation. Accommodation of these text sources through more extensive data would help future models be robust to different types of text-generation.

8. Team Contributions

Team Member	Contributions
Aryan Vats	Project ideation, Baseline and Proposed approach implementation, analysis, and report.
Deeksha Manjunath	Project Ideation, Preprocessing, Baseline implementation and analysis, and report.
Reshma Ramachandra	Project ideation, Baseline and RADAR implementation, experimentation, and report.

References

[1] Canyu Chen and Kai Shu. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*, 2023. 1

[2] 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. 2

[3] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018. 3

[4] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 2023. 1, 2, 3, 4

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 1

[6] 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. *CoRR*, abs/1907.11692, 2019. 2, 4

[7] 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, 5

[8] OpenAI. Gpt-3.5: A large-scale pretrained language model. https://www.openai.com/gpt-3.5/, 2023. Accessed: January 9, 2024. 1

[9] Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. On the risk of misinformation pollution with large language models. *arXiv preprint arXiv:2305.13661*, 2023. 1

[10] Alessandro Pegoraro, Kavita Kumari, Hossein Fereidooni, and Ahmad-Reza Sadeghi. To chatgpt, or not to chatgpt: That is the question! *arXiv preprint arXiv:2304.01487*, 2023. 2

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2, 3

[12] 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. 5

[13] Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. Implementing bert and fine-tuned roberta to detect ai generated news by chatgpt. *arXiv preprint arXiv:2306.07401*, 2023. 2

[14] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*, 2019. 3

[15] 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. 2

[16] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019. 1

[17] Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2023. 1