RAFT: Realistic Attacks to Fool Text Detectors

James Wang[1*]**, Ran Li**[1*]**, Junfeng Yang**[1]**, Chengzhi Mao**[2]

Columbia University[1], Rutgers University[2]

{jlw2247, rl3424, jy2324}@columbia.edu, cm1838@rutgers.edu

Abstract

Large language models (LLMs) have exhibited remarkable fluency across various tasks. However, their unethical applications, such as disseminating disinformation, have become a growing concern. Although recent works have proposed a number of LLM detection methods, their robustness and reliability remain unclear. In this paper, we present *RAFT*: a grammar error-free black-box attack against existing LLM detectors. In contrast to previous attacks for language models, our method exploits the transferability of LLM embeddings at the word-level while preserving the original text quality. We leverage an auxiliary embedding to greedily select candidate words to perturb against the target detector. Experiments reveal that our attack effectively compromises all detectors in the study across various domains by up to 99%, and are transferable across source models. Manual human evaluation studies show our attacks are realistic and indistinguishable from original human-written text. We also show that examples generated by *RAFT* can be used to train adversarially robust detectors. Our work shows that current LLM detectors are not adversarially robust, underscoring the urgent need for more resilient detection mechanisms.

1 Introduction

Large language models (LLMs) such as ChatGPT (Ouyang et al., 2022), LLaMA (Touvron et al., 2023), and GPT-4 (OpenAI, 2023) have exhibited transformative abilities to generate remarkably fluent and cogent long-form text in response to user queries. However, LLMs have been misused to disseminate disinformation, commit academic dishonesty, and launch targeted spear phishing campaigns against vulnerable populations (Hazell, 2023). To mitigate harm from malicious use, the capability to distinguish machine-generated text and human-written text is paramount.

To defend against these malicious use cases, various methods have been developed to successfully detect machine-generated text such as OpenAI's GPT-2 supervised detector (Solaiman et al., 2019), watermarking (Kirchenbauer et al., 2023), and likelihood-based zero-shot detectors such as DetectGPT (Mitchell et al., 2023). In response, red-teaming methods for attacking machine-generated text detectors were created to identify vulnerabilities. Red-teaming methods are primarily based on paraphrasing or word substitution. Paraphrasing-based attacks such as *DIPPER* fine-tune a generative language model on a large set of manually collected paraphrase pairs (Krishna et al., 2023; Sadasivan et al., 2023). Word substitution-based attacks have leveraged masked language models or auxiliary LLMs to generate replacement candidates (Krishna et al., 2023; Shi et al., 2024). Despite their effectiveness in subverting various detectors, word substitution-based attacks often contain numerous grammatical errors and semantic inconsistencies that are readily discernible upon human evaluation.

In this work, we explore whether machine-generated text can subvert detection with *realistic* perturbations that remain inconspicuous to human readers. A perturbation is considered realistic if it maintains part-of-speech (POS) consistency, minimally increases perplexity, and is indistinguishable from human-written text in manual evaluations.

We present *RAFT*, a zero-shot black-box attack framework to subvert machine-generated text detectors. *RAFT* leverages an auxiliary LLM embedding to optimally select words in machine-generated text for substitution by performing a proxy task. It then employs a black-box LLM to generate replacement candidates, greedily selecting the one that most effectively subverts the target detector. *RAFT* only requires access to an LLM's embedding layer, making it easily deployable and adaptable with the numerous powerful open-source LLMs available (Hugging Face Inc., 2022).

16923

Original GPT3.5	**Red-Teaming (Shi et al.)**	**Ours**
Maj Richard Scott, 40, is accused of driving at speeds of up to 95mph (153km/h) in bad weather before the fatal crash that claimed the lives of two young children. The incident occurred on the A34 motorway near Newbury, Berkshire, last Saturday.	Maj Richard Scott, Two score, is accused of driving at quickens of up to lightning tempo (swift pace) in bad Sunny before the fatal crash that claimed the lives of two young children. The circumstance betided on the path motorway near Newbury, Berkshire, last Saturday.	Maj Richard Scott, 40, is reproached of steering at speeds of up to 95mph (153km/h) under bad weather before the fatal crash that claimed the deaths of two young minors. The mishap occurred near the A34 motorway near Newbury, UK, previous Saturday.
DetectGPT LLM Likelihood: 0.7100 Perplexity: 12.14	DetectGPT LLM Likelihood: 0.0500 Perplexity: 137.72	DetectGPT LLM Likelihood: **0.0400** Perplexity: 43.14

Figure 1: *RAFT* can attack a sample text generated by GPT-3.5-turbo more effectively to subvert detection by DetectGPT than recent red teaming attack efforts (Shi et al., 2024) while preserving language fluency and semantic consistency. By enforcing grammatical consistency in the substituted words through POS correction, *RAFT* achieves significantly lower perplexity than attacks that do not enforce grammar. Qualitative evaluation also highlight *RAFT*'s language fluency and semantic consistency with the original text. Red text represents substituted words with grammatical errors or semantic inconsistencies. Blue text represent error-free substitutions.

Our results show *RAFT* can reduce detection performance by up to 99% while preserving the part-of-speech and semantic consistency of the replaced words. Additionally, we show that the replacement words selected to greedily subvert one target detector can be effectively transferred to attack other detectors, outperforming benchmarked attack methods. Furthermore, we demonstrate that *RAFT*'s outputs can be leveraged to enhance a detector's robustness through adversarial training. Our findings suggest that current detectors are vulnerable to adversarial attacks and highlight the urgency to develop more resilient detection mechanisms. Our code and data is available at https://www.github.com/jameslwang/raft.

2 Related Work

Substitution-based Attacks against NLP targets: While existing gradient-based adversarial attacks are effective in the vision and speech domains (Carlini and Wagner, 2017), attacks in the text domain present unique challenges due to its discrete nature. Attacks in the natural language domain are also constrained by language fluency, semantic consistency, and human prediction consistency. Jin et al. (2020) introduce a black-box word substitution-based attack that fulfills all these criteria by utilizing semantic similarity and POS matching to greedily replace words with synonyms until a successful attack. In this work, we use an LLM instead to generate semantically consistent word replacement candidates and greedily select the POS-consistent word that most effectively attacks the target detector.

LLM Detector Attack Frameworks: Existing algorithms for detecting machine-generated text can be categorized into three categories: supervised classifiers (Solaiman et al., 2019; Hovy, 2016; Zellers et al., 2019), watermark detectors (Kirchenbauer et al., 2023; Grinbaum and Adomaitis, 2022; Abdelnabi and Fritz, 2021), and zero-shot statistical-based methods (Mitchell et al., 2023; Tian, 2023; Lavergne et al., 2008; Solaiman et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2020). As new detection methods continue to be developed, parallel efforts in red-teaming these detectors have also gained momentum. The primary techniques for attacking them include text paraphrasing and word replacement. Krishna et al. (2023) present *DIPPER*, a paraphrase generation model that can be conditioned on surrounding context to effectively attack state-of-the-art detectors while controlling output diversity and maintaining semantic consistency. Sadasivan et al. (2023) iterate on this method and present a recursive paraphrasing attack that breaks watermarking and retrieval-based detectors with slight degradation in text quality by using a lightweight T5-based paraphraser model. These attack frameworks are more vulnerable to attack since they are reliant on one large fine-tuned LLM model and utilize smaller paraphrasing models that are relatively weaker than the source LLM. Shi et al. (2024) introduce a word replacement method that utilizes an LLM to randomly generate substitution candidates for multiple words, selecting the optimal replacement candidates using an iterative evolutionary search algorithm to minimize detection score. We improve upon this framework by introducing a replaceable proxy scoring model that uses an auxiliary LLM embedding to rank which words in the machine-generated text should be replaced and greedily select LLM-generated candidates that effectively subvert the target detector.
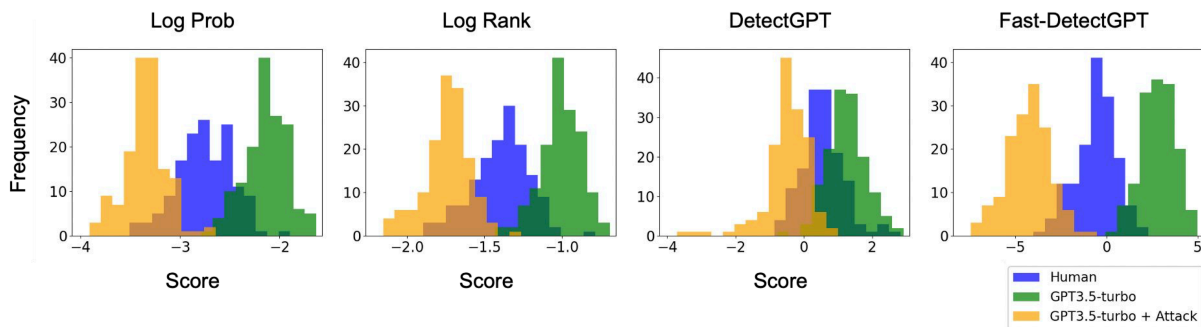
Figure 2: Histograms show the distributions of different detection scores for human-written text, GPT-3.5-Turbo generated text, and *RAFT*-attacked GPT-3.5-Turbo text. The horizontal axis represents the raw output from the detector. The diagrams illustrate that our attack effectively shifts the distribution of generated data towards the negative region, fooling the detectors.

3 Method

3.1 Preliminaries

Setup: Given a text passage $\mathbf{X}$ consisting of N words $[x_1, \ldots, x_N]$, we consider a black-box detector $D(\mathbf{X}) \in [0, 1]$ that predicts whether the input $\mathbf{X}$ is machine-generated or human-written. A higher $D(\mathbf{X})$ score indicates a greater likelihood that $\mathbf{X}$ is machine-generated. We denote τ as the detection threshold, such that $\mathbf{X}$ is classified as machine-generated if $D(\mathbf{X}) \geq \tau$.

Adversarial Attack for LLM Detector: The goal of the attack is to perturb the input passage $\mathbf{X}$ into $\mathbf{X}'$ such that $D(\mathbf{X}')$ incorrectly classifies $\mathbf{X}'$ as human written, while ensuring that $\mathbf{X}'$ remains indistinguishable from human-written text when manually reviewed. To preserve semantic similarity between $\mathbf{X}$ and $\mathbf{X}'$, the number of words to be substituted is constrained to $k\%$ of $\mathbf{X}$. Additionally, to maintain grammatical correctness and fluency, $\mathbf{x}_i$ and $\mathbf{x}'_i$ must have consistent part-of-speech (Brill, 1995). We formulate the attack on the LLM Detector $\mathbf{D}$ as a constrained minimization problem, with the objective to modify $\mathbf{X}$ such that $D(\mathbf{X}') \leq \tau$:

$$\mathbf{X}' = \underset{\mathbf{X}'}{\operatorname{argmin}} D(\mathbf{X}') \quad \text{s.t.} \quad pos(\mathbf{x}'_i) = pos(\mathbf{x}_i),$$
$$\mathbf{x}'_i \in \{\mathbf{x}_i\} \cup s(\mathbf{x}_i, \mathbf{X}, t),$$
$$\sum_i^N \mathbf{1}(x'_i \neq x_i) \leq kN$$

$$(1)$$

where $pos(\mathbf{x}_i)$ returns the part-of-speech label of any word $\mathbf{x}_i$ and $s(\cdot)$ is a word substitution generator that outputs t candidates for $\mathbf{x}_i$ using the surrounding context in $\mathbf{X}$.

3.2 Our Attack

Finding Important Words for Substitution using a Proxy Task Embedding Objective: We capitalize on Freestone and Santu (2024)'s observations that LLMs share similar latent semantic spaces and perform similarly on semantic tasks. To effectively minimize $D(\mathbf{X}')$, we use a white-box LLM M to perform a word-level task $\mathbf{F}$ that generates a score $\mathbf{f}_i$ for each word that acts as a *proxy* signal for selecting words to replace, where M does not necessarily need to be the same source LLM model used to generate $\mathbf{X}$. We choose LLM embedding tasks correlated with identifying words that would alter the statistical properties of the machine-generated text, such as next-token generation and supervised LLM text detection. From $\mathbf{F}$, we choose

$$\mathbf{X}_k = \operatorname{argmax}_{kN} \mathbf{F}(M, \mathbf{X}) \qquad (2)$$

where $\mathbf{X}_k$ is the subset of $\mathbf{k}\%$ words in $\mathbf{X}$ to perturb from.

Constraints for Realistic Generation: To perturb words in $\mathbf{X}_k$ while ensuring that $\mathbf{X}$ remains indistinguishable to a human evaluator as machine-generated, we constrain the replacement words such that they must not induce grammatical errors and are semantically consistent with the original text. We use GPT-3.5-Turbo (OpenAI, 2024b) as our word substitution candidate generator by prompting it with the word to replace and its surrounding context using the following prompt:

```
Q: Given some input paragraph,
   we have highlighted a word
   using brackets. List top {t}
   alternative words for it that
   ensure grammar correctness
```

```
            and  semantic  fluency.  Output
            words  only.\n{paragraph}
A:  The  alternative  words  are  1.
     2.  ...
```

Using an LLM for word substitution allows us to conveniently obtain context-compatible candidates in one step, instead of needing to compute candidates using word embeddings followed by an additional model to check context compatibility (Alzantot et al., 2018). After retrieving t replacement candidates from GPT-3.5-Turbo, we filter out words that have inconsistent part-of-speech with the original word by using the NLTK library (Bird et al., 2009) and then select the candidate that minimizes D.

3.3 Implementation Details

We set k to 10% across all experiments to evaluate the effectiveness of our attack with a limited number of changes. We evaluate the effectiveness of using language modeling heads for next-token generation and supervised LLM detection tasks as proxy scoring models to optimally select words to substitute in $\mathbf{X}$. For next-token generation, we use the probability of the next token being X_i from the language modeling head as the proxy objective. Intuitively, replacing tokens with the highest likelihood from the LLM allows us to alter the statistical properties of the machine-generated text most effectively. For LLM detection, we iteratively compute the importance of each word based on the decrease in detection score $D(\mathbf{X})$ by assigning 0 to its corresponding tokens in the detector's attention mask, and ranking the score changes in descending order, where the word that yields a higher absolute change in detector score is considered to be more important for detection.

4 Experiments

4.1 Datasets and Metrics

Datasets: We use three datasets to cover a variety of domains and use cases. We use 200 pairs of human-written and LLM-generated samples from each of the XSum (Narayan et al., 2018) and SQuAD (Rajpurkar et al., 2016) datasets generated by Bao et al. (2024) using GPT-3.5-turbo. Additionally, we use the ArXiV Paper Abstract dataset (Mao et al., 2024) which contains 350 abstracts generated using GPT-3.5-turbo from ICLR conference papers.

Metrics: We use the Area Under the Receiver Operating Characteristic Curve (AUROC) to summarize detection accuracy for our attack framework under various thresholds. We also measure the True Positive Rate at a 5% False Positive Rate (TPR at 5% FPR), as it is imperative in this context for human-written text to not be misclassified as machine-generated text. To measure text quality, we measure the perplexity of the attacked text against GPT-NEO-2.7B (Gao et al., 2021).

4.2 Embeddings for Proxy Scoring Models

We use the language modeling heads of GPT-2 (Radford et al., 2019), OPT-2.7B (Zhang et al., 2022), GPT-NEO-2.7B (Gao et al., 2021), and GPT-J-6B (Wang and Komatsuzaki, 2021) for next-token generation, and the RoBERTa-base and RoBERTa-large supervised GPT-2 detector models (Solaiman et al., 2019) for LLM detection as proxy tasks to rank which words from the original text to substitute. We present results for OPT-2.7B and RoBERTa-large proxy scoring models in Table 1 and the rest in Table A.1.

4.3 Detectors

We evaluate our attack against a variety of target detection methods:

Log Likelihood (Gehrmann et al., 2019) is a classical threshold-based zero-shot method where passages with higher log probability scores are more likely to have been generated by the target LLM.

Log Rank (Solaiman et al., 2019) is a classical threshold-based zero-shot method where passages with above average rank are more likely to have been generated by the target LLM.

DetectGPT (Mitchell et al., 2023) is a state-of-the-art zero-shot detector that leverages the likelihood of generated texts to perform thresholding for detecting machine-generated text.

Fast-DetectGPT (Bao et al., 2024) is a state-of-the-art detector that improves upon DetectGPT by introducing conditional probability curvature to underscore discrepancies in word choices between LLMs and humans to improve detection performance and computational speed.

Ghostbusters (Verma et al., 2024) is a state-of-the-art detector that uses probabilistic outputs from LLMs to construct features to train an optimal detection classifier.

Raidar (Mao et al., 2024) is a state-of-the-art detector that uses prompt rewriting and an output's

Original GPT3.5	Red-Teaming (Shi et al.)	Ours
The incident occurred on March 30, 1981, when John Hinckley Jr. attempted to assassinate President Ronald Reagan. The shooting took place outside the Hilton Hotel in Washington, just moments after President Reagan had delivered a speech.	The incident occured on March tridsat' (Russian), 1981, when John attempted killer Jr. attempted to assassinate chief magistrate Ronald Reagan. The shooting took place outside the accommodations Hotel in Washington, just bat of an eye after President Reagan had delivered a speech.	The event occurred on March 30, 1981, when John Hinckley Jr. endeavored to assassinate President Ronald Reagan. The shooting took place outside a Hilton Hotel in Washington, merely moments after President Reagan had delivered a dialogue.
Log Rank LLM Likelihood: 0.7900 Perplexity: 7.60	Log Rank LLM Likelihood: 0.1200 Perplexity: 253.66	Log Rank LLM Likelihood: **0.0800** Perplexity: 17.14
The ban specifically targets clothing and symbols associated with Islam, such as the hijab and the crescent moon and star. This decision follows similar measures implemented in other parts of the country.	The inhibit specifically targets habit and symbols joined with Islam, such as the hijab and the crescent earth's natural satellite and star. This decision follows similar measures implemented in other parts of the country.	The interdiction specifically targets clothing and symbols associated with Islam, such as the hijab and the crescent moon and insignia. This decision follows similar steps implemented among other parts of the country.
Ghostbuster LLM Likelihood: 0.9627 Perplexity: 16.23	Ghostbuster LLM Likelihood: 0.0352 Perplexity: 161.73	Ghostbuster LLM Likelihood: **0.0262** Perplexity: 39.29
The impact caused both cars to veer off the road and crash into a nearby tree. Tragically, the two children, aged 3 and 5, who were passengers in the other vehicle, were pronounced dead at the scene. The driver of that car, a 32-year-old woman, sustained serious injuries and is currently in critical condition at a local hospital.	The impact caused both coaches to veer off the road and crash into a nearby tree. Tragically, the binary children, past one's prime 3 and quinquevalent, who were journeyers in the other vehicle, were pronounced dead at the scene. The driver of that wheels, a 32-year-time-honored woman, sustained serious lesions and is currently in critical condition at a local hospital.	The impact caused both motorcars to veer off the pathway and crash into a nearby timber. Tragically, the two juveniles, aged 3 and 5, who were passengers in the other vehicle, were pronounced dead at the vicinity. The driver of that car, a 32-year-old woman, endured serious wound and is currently in critical condition in a local hospital.
Fast-DetectGPT LLM Likelihood: 0.9800 Perplexity: 6.31	Fast-DetectGPT LLM Likelihood: **0.0001** Perplexity: 46.43	Fast-DetectGPT LLM Likelihood: **0.0001** Perplexity: 17.25

Figure 3: Generated texts from LLMs and their respective attacks using Shi et al. (2024)'s query-based word substitution attack and *RAFT* (ours) using the RoBERTa-large proxy scoring model, evaluated against Log Rank, Ghostbuster, and Fast-DetectGPT detectors. *RAFT* demonstrates the greatest reduction in detection likelihood while maintaining grammatical correctness and semantic consistency with the original text. Red text represents substituted words with grammatical errors or semantic inconsistencies. Blue text represent error-free substitutions.

edit distance to gain additional context about the input.

4.4 Baselines

We compare our attack method with *DIPPER* (Krishna et al., 2023), a paraphrase generation model using settings of 20 lexical diversity and 60 order diversity. This corresponds to about 20% lexical modification – the minimum modification we can set on this method. We also compare with Shi et al. (2024)'s query-based word substitution attack and limit the number of substituted words to be at most 10% to match our substitution frequency.

4.5 Results

Tables 1 and 2 demonstrate that our attack effectively compromises all tested detectors while causing only a modest change in perplexity from the original machine-generated text. Using next-token generation with OPT-2.7B and LLM detection with RoBERTa-large as proxy scoring models for *RAFT* achieved lower AUROC across all datasets and target detectors when compared to the original text, and in most cases, lower than both *DIPPER* and Shi et al. (2024)'s query-based word substitution. Although *DIPPER* preserves the text quality better in terms of perplexity, its AUROC is significantly higher than *RAFT* and Shi et al. (2024)'s attack. The TPR at 5% FPR was 0 for almost all *RAFT*

attacked text, which we present in Table A.2. For more insight, we present the ROC curve for our experiments in Figure 6. Between Shi et al. (2024)'s query-based attack and our method, *RAFT* consistently yields lower perplexity scores across all scenarios. Raidar stands out as the most robust detector against attacks, likely due to the unique edit distance of rewriting used in the approach. Qualitative results shown in Figures 1 and 3 highlight our method's semantic consistency and language fluency. Additionally, cosine similarity calculations between the original and perturbed texts shown in Table 3 using state-of-the-art LLMs Mistral-7B-v0.3 (Jiang et al., 2023) and Llama-3-8B (Dubey et al., 2024) highlight their strong semantic similarity. We also show in Figure 2 that our attack effectively alters the distribution of detection likelihood scores, diverging from the distribution associated with the machine-generated text, thereby subverting detection.

4.6 Human Evaluation

To validate that *RAFT* preserves text quality, we conducted a crowd-sourced human evaluation using Amazon Mechanical Turk (MTurk). We selected the first 100 pairs of human-written and GPT-3.5-Turbo-generated texts from the XSum, SQuAD, and Abstract datasets. After applying *RAFT* to the LLM-generated text, three MTurk workers evalu-

Table 1: *RAFT* attack results. We evaluate *RAFT* performance against 6 target detectors using GPT-3.5-Turbo generated text from 3 datasets, measuring the detector's performance before and after attack using the AUROC metric. **Bolded** AUROC results indicate best attack performance. These results show the superiority of our attack compared to benchmarked methods.

Metric	Log Probability	Log Rank	Ghostbuster	DetectGPT	Fast-DetectGPT	Raidar	Average
XSum / Unattacked	0.9577	0.9584	0.6637	0.7853	0.9903	0.7667	0.8537
Dipper	0.7981	0.8080	0.7196	0.4693	0.9610	**0.4667**	0.7038
Query-based Substitution	0.0481	0.0739	0.0980	**0.0384**	0.2308	0.6000	0.1815
OPT-2.7B (Ours)	**0.0035**	**0.0069**	0.0826	0.1273	**0.0006**	0.7000	0.1535
RoBERTa-large (Ours)	0.0346	0.0568	**0.0004**	0.0704	0.0371	0.6000	**0.1324**
SQuAD / Unattacked	0.9027	0.9075	0.7659	0.7916	0.9800	0.7833	0.8552
Dipper	0.7929	0.8067	0.7959	0.5916	0.9492	**0.5333**	0.7449
Query-based Substitution	0.1542	0.1852	0.2032	0.1408	0.3624	0.8333	0.3132
OPT-2.7B (Ours)	**0.0496**	**0.0659**	0.0851	0.1539	**0.0131**	0.8333	0.2002
RoBERTa-large (Ours)	0.0942	0.1199	**0.0166**	**0.1262**	0.1039	0.7167	**0.1963**
Abstract / Unattacked	0.6329	0.6502	0.8455	0.1538	0.9148	0.7667	0.6607
Dipper	0.5029	0.5370	0.8826	0.1049	0.9441	0.6833	0.6091
Query-based Substitution	0.0234	0.0364	0.3142	0.0046	0.2976	0.7167	0.2322
OPT-2.7B (Ours)	0.0945	0.1249	0.0841	0.3131	**0.0399**	0.7667	0.2372
RoBERTa-large (Ours)	**0.0162**	**0.0336**	**0.0374**	**0.0044**	0.1481	**0.6500**	**0.1666**

Table 2: Perplexity of text after different attacks measured by GPT-NEO-2.7B. *RAFT* attacked texts were optimized against Fast-DetectGPT detector. Lower perplexity indicates better text quality. The results show that our attack is able to maintain text quality while subverting detection.

Dataset	Unattacked	Dipper	Query-based Substitution	OPT-2.7B (Ours)	RoBERTa-large (Ours)
XSum	8.4804	11.3649	28.0979	17.6181	22.4542
SQUAD	9.7947	11.9064	30.0879	19.6190	25.1480
Abstract	12.9136	15.2685	36.6523	26.8810	31.6123

Table 3: Cosine similarity, evaluated across multiple LLM embeddings between the original texts and those perturbed by *RAFT* using RoBERTa-base as the proxy scoring model and Fast-DetectGPT as the target detector, indicates that the texts maintain semantic similarity.

Embedding Model	XSum	SQuAD	Abstract
RoBERTa-large	0.9999	0.9999	0.9999
Llama-3-8B	0.9747	0.9759	0.9841
Mistral-7B-v0.3	0.9761	0.9735	0.9847

ated each pair of original human-written and *RAFT*-modified texts, indicating their preference for one of them or expressing no preference. *RAFT*'s perturbations were deemed indistinguishable from human-written text if two or more annotators either preferred the perturbed text or were indifferent. To ensure English proficiency, we included a screening question using a text comparison task sourced from the ETS TOEFL website. Out of valid 396 responses, 185 preferred the human-written texts, 182 were indifferent, and 29 responses were excluded for rating both texts as low quality. A two-tailed binomial test yielded a p-value of 0.917 at

$\alpha < 0.05$, supporting the null hypothesis that the two texts are indistinguishable. The Fleiss' kappa was 0.774, indicating strong agreement among annotators.

5 Discussion

5.1 Effect of Scoring Model

We perform ablation studies to evaluate the isolated effectiveness of the proxy scoring model (ranking) and the greedy selection of generated POS-consistent replacement words aimed at subverting detection (optimization). For brevity, we refer to these two methods as "ranking" and "optimization", respectively. As shown in Table 5, the study is conducted under four settings: neither ranking nor optimization, ranking only, optimization only, and both ranking and optimization. The results indicate that ranking is about as effective as optimization, significantly reducing AUROC when applied, supporting the idea that LLM embeddings are transferable. However, the effects of ranking and optimization are not necessarily additive.

Table 4: AUROC of *RAFT*-attacked text, using Word2Vec embedding model trained on the Google News corpus for word replacement candidate generation instead of GPT-3.5-turbo on the XSum and Abstract datasets, suggests that using a classic word embedding in place of an LLM also yields effective results.

Proxy Model/Detector	Log Probability	Log Rank	Ghostbuster	Fast-DetectGPT
XSum/Unattacked	0.9577	0.9584	0.6637	0.9903
OPT-2.7B (Ours)	0.0052	0.0144	0.0408	0.0034
RoBERTa-large (Ours)	0.0016	0.0064	0.0000	0.0698
Abstract/Unattacked	0.6329	0.6502	0.8455	0.9148
OPT-2.7B (Ours)	0.1041	0.1577	0.0873	0.0711
RoBERTa-large (Ours)	0.0021	0.0040	0.0075	0.0346

Table 5: Effect of scoring models. To show the isolated effectiveness of the proxy scoring model, we attack the Ghostbuster and Fast-DetectGPT detectors using OPT-2.7B next-token generation on the XSum dataset across four different configurations. Here, "ranking" refers to the proxy scoring model, and "optimization" refers to the greedy selection of generated POS-consistent words against the target detector. The results indicate both techniques are effective, but their combined effect is not necessarily additive. **Bolded** AUROC results denote best attack performance.

Setting	Neither	Optimization Only	Ranking Only	Ranking + Optimization
Ghostbuster	0.3341	0.1001	**0.0981**	0.1000
Fast-DetectGPT	0.7510	0.0026	0.0030	**0.0006**
Raidar	0.7667	0.6333	0.6667	**0.6000**

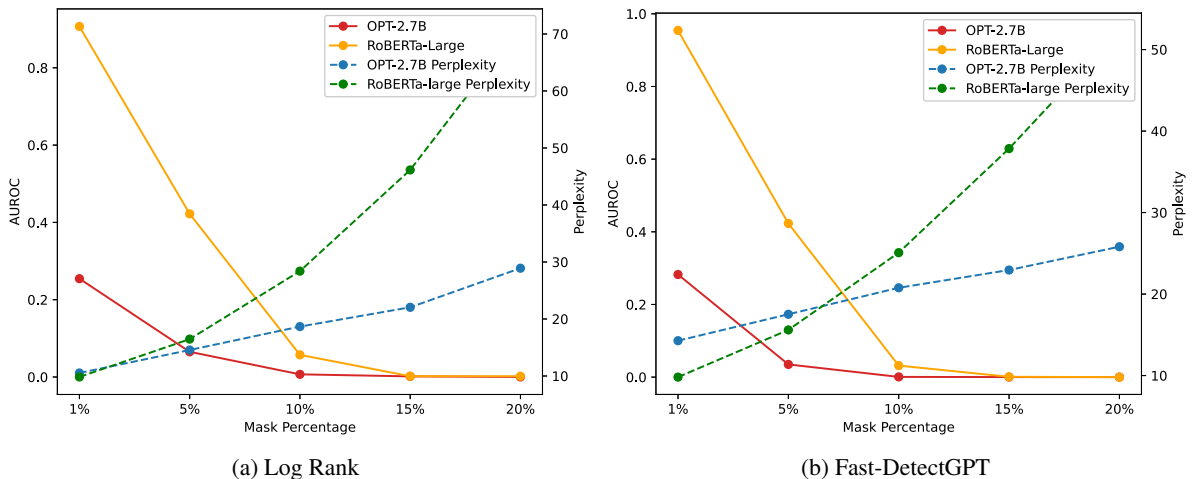(a) Log Rank

(b) Fast-DetectGPT

Figure 4: Study on the impact of different mask percentages. We use OPT-2.7B and RoBERTa-large as proxy scoring models to attack Log Rank and Fast-DetectGPT detectors on the XSum dataset at 1%, 5%, 10%, 15%, and 20% masking rates while measuring detection performance and text quality in terms of AUROC and perplexity. The AUROC approaches 0 at around 10% with a moderate increase in perplexity. Masking percentages beyond 15% degrade text quality across both detectors.

5.2 Impact of Word Replacement Generation Method

We evaluate the effectiveness of replacing GPT-3.5-Turbo with a traditional Word2Vec embedding model (Mikolov et al., 2013a). Specifically, we use the Word2Vec model trained on Google News corpus, which contains 1 billion words (Mikolov et al., 2013b), to locally retrieve $t = 10$ POS-consistent synonyms as word replacement candidates. We show in Table 4 that using a word embedding model instead of an LLM also produces effective results.

5.3 Impact of Masking Percentage

We evaluate the performance and text quality of *RAFT* across various masking percentages. Figure 4 shows that the AUROC stabilizes around 0 when the masking percentage reaches 10%, accompanied by a moderate increase in perplexity. Masking percentages exceeding 15% are unnecessary and lead to a significant degradation in text quality.

Table 6: Effect of applying our grammar constraint. We use OPT-2.7B and RoBERTa-large as scoring models to attack the Fast-DetecGPT detector on the XSum dataset, comparing performance with and without the part-of-speech constraint on the generated replacement words. The results show a marginal decrease in attack performance but a significant improvement in perplexity when POS constraints are enforced.

| | No POS Correction | | POS Correction | |
	AUROC	Perplexity	AUROC	Perplexity
OPT-2.7B	0.0000	28.32	0.0006	17.53
RoBERTa-large	0.0062	31.36	0.0471	25.08

Original GPT3.5

The ban was backed by local authorities in Urumqi, state media reported. The move is the latest in a campaign against Islamic clothing and symbols, as the authorities strive to promote secularism and maintain social stability in the region.

DetectGPT LLM Likelihood: 0.69
Perplexity: 17.03

Ours + No POS Correction

The ban injunction backed by local authorities preserved Urumqi, state publication indicated. The move is the latest in a campaign against Islamic clothing and symbols, as the authorities strive laboring promote secularism sustain maintain social stability in the region.

DetectGPT LLM Likelihood: 0.10
Perplexity: 157.65

Ours + POS Correction

The ban was backed by local authorities within Urumqi, state media announced. The move represents the latest in a campaign against Islamic clothing and symbols, since the authorities strive to promote secularism and maintain social stability inside the region.

DetectGPT LLM Likelihood: 0.03
Perplexity: 36.70

Figure 5: Comparison on the effects of POS tagging. On the left is unmodified text generated by GPT3.5-turbo; in the middle is *RAFT* attacked text but without POS consistency constraints; and on the right is *RAFT* attacked text with POS consistency. This example illustrates that POS tagging significantly enhances text quality both qualitatively and quantitatively as measured by perplexity, without compromising detection performance.

5.4 Impact of the Source Generation Model

We study the effectiveness of *RAFT* under different source generation models. We evaluate its effectiveness on text generated using GPT-3.5-Turbo, Llama-3-70B (Touvron et al., 2023), and Mixtral-8x7B-Instruct (Jiang et al., 2024), which represent a set of LLMs of varying size, architecture, and trained corpora. We utilize the same generation parameters as those employed by (Bao et al., 2024) for producing the GPT-3.5-turbo generated XSum and SQuAD datasets for Llama-3-70B and Mixtral-8x7B-Instruct. We show in Figure 7 that *RAFT* remains highly effective when next-token generation is used as a proxy task with OPT-2.7B, GPT-NEO-2.7B, GPT-J-6B embeddings model for subverting detection against Log Rank, RoBERTa-large, and Fast-DetectGPT detectors.

5.5 Transferrability of *RAFT* on other Detectors

We study the transferability of *RAFT*-attacked text across various detectors. We evaluate the attacked text generated by using OPT-2.7B next-token generation and RoBERTa-large LLM detection proxy scoring tasks optimized against LogRank and DetectGPT detectors on GhostBuster and Fast-DetectGPT. The results, presented in Table A.3, show that the AUROC only decreases slightly, suggesting that our attack is highly transferable.

5.6 *RAFT* for Adversarial Training

We present evidence that our attack not only effectively subverts detectors but can also enhance their robustness through adversarial training. As shown in Table 7, after the Raidar detector undergoes adversarial training on *RAFT*-attacked text, it consistently demonstrates a significant increase in detection performance under attack compared to the performance decrease observed before retraining. For the Abstract dataset, the AUROC for both attacked and non-attacked text samples increases, indicating that *RAFT* can enhance the robustness of existing detectors through adversarial training. We present this as an important direction for future research.

6 Conclusion

We introduce *RAFT*, an adversarial attack framework for subverting machine-generated text detectors by leveraging auxiliary LLM embeddings. Our method effectively identifies optimal words to perturb using a proxy LLM embedding and perturbs them such that the original text remains semantically consistent, grammar error-free, and reads fluently. Experimental results and manual annotation exercises show that our method successfully compromises various LLM detection methods while maintaining text quality and semantic consistency, highlighting the need for robust LLM content detectors. We also demonstrate that the outputs from *RAFT* can be used to enhance the resilience of existing detectors through adversarial training.
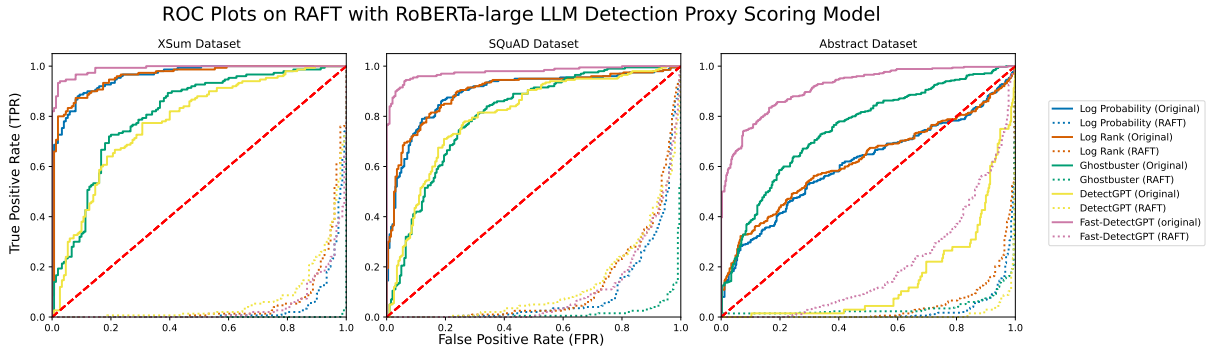
16930

Figure 6: ROC curves for machine-generated XSum dataset under *RAFT* attack, using RoBERTa-large as the proxy scoring model, compared across multiple detectors. The ROC curves for both attacked and unattacked text provide a more comprehensive evaluation of *RAFT*'s robustness in subverting text detectors than single metrics.
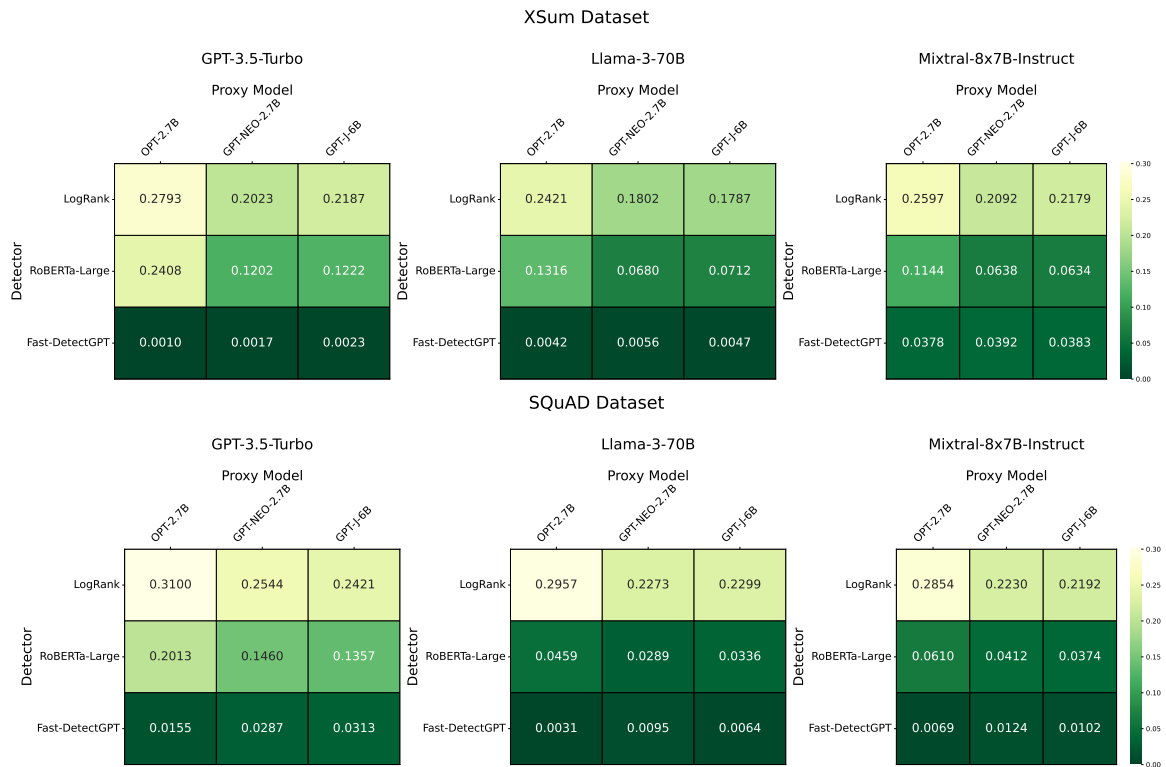

Figure 7: AUROC of using *RAFT* under various source generation models and proxy scoring models against Log Rank, RoBERTa-large, and Fast-DetectGPT detectors on the XSum and SQuAD dataset. The results demonstrate *RAFT* remains robust under various source generation models and proxy scoring model pairs.

Table 7: Adversarial training results. We train the Raidar detector on texts with and without word swapping, denoted as Training Method, and evaluated its performance on samples with (Attack) and without (Clean) word swapping. The result shows the detector becomes more robust under adversarial training. **Bolded** AUROC results denote highest-performing detector.

Dataset	XSum		SQuAD		Abstract	
Training Method	Normal	Adversarial	Normal	Adversarial	Normal	Adversarial
Clean AUROC	**0.8000**	0.7500	**0.6833**	0.6667	0.6500	**0.6833**
Attack AUROC	0.6000	**0.7333**	0.7167	**0.8000**	0.6500	**0.7167**

7 Limitations

While we demonstrate *RAFT*'s effectiveness in compromising various LLM detectors, there are several limitations to note:

Scalability of Human Evaluations: While our manual human evaluation study demonstrated that *RAFT*'s perturbations are realistic and are not necessarily less preferred from the original text, larger-scale human evaluations are necessary to validate the quality and realism of the perturbed texts robustly. Furthermore, we did not extensively explore the demographic and linguistic backgrounds of the human evaluators, which may induce bias in our study.

Computational & Cost Overhead: The runtime performance of *RAFT* is shown in Table A.5. Generating substitution candidates using GPT-3.5-Turbo or using a word embedding for each selected candidate replacement word introduces significant computation and cost overhead. This may limit the practicality of this attack in real-time or in budget-constrained environments. Developing more efficient prompting strategies for effective word-level substitutions would be essential for practical use.

Fixed Perturbation Rate: We fixed the perturbation rate at 10% across all experiments, which is less than the rate set in Shi et al. (2024) and Krishna et al. (2023). While this provides a consistent and strong benchmark, it does not account for scenarios where a smaller perturbation rate may be more effective. Exploring adaptive perturbation strategies based on text complexity and detection sensitivity may yield a more efficient and effective attack.

Limited Detector Evaluation: *RAFT* was tested against various types of LLM detectors. However, as new detection methods emerge, we must continuously evaluate our attack's robustness on novel approaches.

8 Ethics Statement

While our paper presents a method to subvert detection of machine-generated text by LLM detectors, it is imperative to acknowledge that LLMs are predominantly utilized in good faith and have a wide variety of benefits to society, such as improving one's work and efficiency. By scrutinizing LLM detectors through red-teaming, we highlight current vulnerabilities in these systems and urgently advocate for the development of more resilient mechanisms. While we introduce how examples generated by *RAFT* can be utilized for adversarial training, future work should emphasize the development of robust defense mechanisms.

9 Acknowledgements

References

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 121–140. IEEE.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2890–2896. Association for Computational Linguistics.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Comput. Linguistics*, 21(4):543–565.

Nicholas Carlini and David A. Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Matthew Freestone and Shubhra Kanti Karmaker Santu. 2024. Word embeddings revisited: Do llms offer something new? *CoRR*, abs/2402.11094.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Alexei Grinbaum and Laurynas Adomaitis. 2022. The ethical need for watermarks in machine-generated language. *CoRR*, abs/2209.03118.

Julian Hazell. 2023. Large language models can be used to effectively scale spear phishing campaigns. *CoRR*, abs/2305.06972.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews - an adversarial study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Hugging Face Inc. 2022. Transformers: State-of-the-art natural language processing. https://huggingface.co.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1808–1822. Association for Computational Linguistics.

Mojan Javaheripi and Sébastien Bubeck. 2023. Phi-2: The surprising power of small language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Thomas Lavergne, Tanguy Urvoy, and Franccois Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the ECAI'08 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse, Patras, Greece, July 22, 2008*, volume 377 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural*

Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 3111–3119.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024a. Hello, gpt-4o. Accessed: 2024-09-30.

OpenAI. 2024b. Models. https://platform.openai.com/docs/models/gpt-3-5-turbo.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *CoRR*, abs/2303.11156.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2024. Red teaming language model detectors with language models. *Trans. Assoc. Comput. Linguistics*, 12:174–189.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

E. Tian. 2023. Gptzero: An ai text detector.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9051–9062.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

A Appendix

A.1 Tables

Table A.1: Our attack results using additional proxy score models demonstrate *RAFT* is effective against various target detectors, scoring similarly to results shown in Table 1. GPT-2, GPT-NEO-2.7B, and GPT-6B use next token generation and RoBERTa-base uses LLM detection as proxy scoring model tasks. Metric reported is AUROC.

Dataset / Method	Log Probability	Log Rank	Ghostbuster	DetectGPT	Fast-DetectGPT	Average
XSum / Unattacked	0.9577	0.9584	0.6637	0.9903	0.8925	0.8925
GPT-2	0.0046	0.0196	0.0453	0.0211	0.0182	0.0217
GPT-NEO-2.7B	0.0052	0.0144	0.0408	0.0120	0.0160	0.0177
GPT-6B	0.0034	0.0156	0.0426	0.0202	0.0160	0.0196
RoBERTa-base	0.0372	0.0584	0.0003	0.0621	0.0390	0.0394
SQuAD / Unattacked	0.9027	0.9075	0.7659	0.9800	0.8890	0.8890
GPT-2	0.0595	0.0959	0.0862	0.0574	0.0695	0.0737
GPT-NEO-2.7B	0.0532	0.0839	0.0831	0.0518	0.0617	0.0667
GPT-J-6B	0.0524	0.0883	0.0667	0.0508	0.0600	0.0636
RoBERTa-base	0.0999	0.1262	0.0175	0.1433	0.1068	0.0988
Abstract / Unattacked	0.6329	0.6502	0.8455	0.9148	0.7609	0.7609
GPT-2	0.1466	0.1960	0.0912	0.1885	0.1353	0.1515
GPT-NEO-2.7B	0.1041	0.1577	0.0873	0.1491	0.1050	0.1206
GPT-J-6B	0.1066	0.1624	0.0794	0.1515	0.1075	0.1215
RoBERTa-base	0.0296	0.0426	0.0388	0.0079	0.1994	0.0637

Table A.2: Performance of *RAFT* attack measured by TPR at 5% FPR. Our results show that *RAFT* significantly lowers the TPR at 5% FPR to nearly 0 across all detectors and datasets, highlighting the robustness of our approach.

Metric	Log Probability	Log Rank	Ghostbuster	DetectGPT	Fast-DetectGPT
XSum / Unattacked	0.7800	0.8067	0.2200	0.1667	0.9400
OPT-2.7B (Ours)	0.0000	0.0000	0.0000	0.0000	0.0000
RoBERTa-base (Ours)	0.0000	0.0000	0.0000	0.0000	0.0000
RoBERTa-large (Ours)	0.0000	0.0000	0.0000	0.0000	0.0000
SQuAD / Unattacked	0.5750	0.6050	0.1650	0.1533	0.9150
OPT-2.7B (Ours)	0.0000	0.0000	0.0000	0.0000	0.0150
RoBERTa-base (Ours)	0.0000	0.0000	0.0000	0.0000	0.0000
RoBERTa-large (Ours)	0.0000	0.0000	0.0000	0.0000	0.0000
Abstract / Unattacked	0.2086	0.2257	0.2314	0.0000	0.6600
OPT-2.7B (Ours)	0.0000	0.0000	0.0200	0.0000	0.0229
RoBERTa-base (Ours)	0.0000	0.0000	0.0171	0.0000	0.0000
RoBERTa-large (Ours)	0.0000	0.0000	0.0143	0.0000	0.0000

Table A.3: Transferability of *RAFT* attacked text. We evaluate *RAFT* perturbed text, using OPT-2.7B and RoBERTa-large proxy scoring models against LogRank and DetectGPT detectors, on LogRank, GhostBuster, DetectGPT, and Fast-DetectGPT detectors. AUROC metrics show only a slight decrease, suggesting our attack is highly transferable.

RAFT-optimized Detector	Log Rank			DetectGPT		
RAFT Proxy Score Model / Transfer Detector	GhostBuster	DetectGPT	Fast-DetectGPT	Log Rank	GhostBuster	Fast-DetectGPT
OPT-2.7B	0.1082	0.1411	0.0022	0.0235	0.1264	0.0059
RoBERTa-large	0.0578	0.0498	0.1541	0.2247	0.1116	0.2927

Table A.4: Evaluation of *RAFT* by using higher-performing LLMs, based on MMLU benchmark score (Hendrycks et al., 2021), for next-token generation as proxy scoring model on the XSum dataset. GPT-4o (OpenAI, 2024a) is used for word replacement candidate generation instead of GPT-3.5-Turbo. The results illustrate that *RAFT* is highly effective on more recent models.

Proxy Scoring Model	AUROC	TPR at 5% FPR
XSum / Unattacked	0.9903	0.9400
Llama-3-8B	0.0485	0.0000
Mistral-7B-v0.3	0.2071	0.0000
Phi-2-2.7B (Javaheripi and Bubeck, 2023)	0.1873	0.0000

A.2 Human Evaluation Task Details

The workers were paid $0.05 USD for each example. The annotation time for each example varies, but the estimated wage rate is $9/hour, which is higher than the US minimum wage ($7.25/hour).

MTurk Task Prompt:

```
Text 1: ${Text 1}
Text 2: ${Text 2}

Options:
•  Text 1 is better
•  Text 2 is better
•  No Preference
•  Both texts are equally bad
```

Note that ${Text 1} and ${Text 2} are shuffled between the original human-written text and *RAFT* perturbed text to avoid selection bias.

A.3 *RAFT* Runtime Performance

Table A.5: We execute *RAFT* on a Linux compute cluster equipped with 188 GB of RAM and an NVIDIA A100 GPU with 40 GB of memory. Using RoBERTa-base as the proxy scoring model and Fast-DetectGPT as the target detector, both loaded on the GPU, we run *RAFT* on the XSum dataset. Word replacement candidates are generated using GPT-3.5-Turbo.

No. Samples	Masking Rate (k%)	Avg. No. Words / Sample	Avg. No. Words Replaced / Sample	Avg. Runtime (s) / Sample	Avg. Runtime (s) / Word Replaced
150	10%	181	18	21.64	1.20