Evaluating the Robustness of LLM-based AI Text Detector on the RAID Benchmark

Kishor Kumar Bhaumik
StudentID: 862542862
kbhau001@ucr.edu

ABSTRACT

The proliferation of advanced generative models necessitates the development of robust AI text detectors. However, the performance of these detectors can be significantly undermined by adversarial attacks designed to evade them. This paper investigates the robustness of transformer-based detectors by conducting a two-part empirical study on the RAID benchmark. First, we compare the performance of four standard BERT variants—BERT-Base, Distil-BERT, Tiny-BERT, and ALBERT—when confronted with ten distinct adversarial attack types. Second, we analyze the impact of parameter-efficient fine-tuning by applying QLoRA[2] to a vanilla BERT model, performing a sensitivity analysis on the adapter rank from 4 to 128. Our findings reveal that while no single architecture excels against all attacks, Distil-BERT shows strong overall performance, and certain attacks like homoglyphs present a universal challenge. Furthermore, we demonstrate that the QLoRA rank is a critical hyperparameter for robustness, with a rank of 64 achieving the optimal balance between performance and efficiency in our experiments. This work provides actionable insights into the selection of both model architectures and fine-tuning strategies for building more resilient AI text detectors. Our code is publicly available at https://github.com/Kishor-Bhaumik/Data-Mining235.

ACM Reference Format:
Kishor Kumar Bhaumik. 2025. Evaluating the Robustness of LLM-based AI Text Detector on the RAID Benchmark. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The recent advancements in Large Language Models (LLMs) have led to a paradigm shift in content creation, but they have also introduced significant challenges related to academic integrity, the spread of misinformation, and digital security. In response, a critical area of research has emerged: the detection of machine-generated text. While numerous detectors have been developed, their practical utility is often limited by a crucial vulnerability—their susceptibility to adversarial attacks. Simple, automated modifications such as synonym substitution, character-level manipulation (e.g., homoglyphs), or the insertion of zero-width spaces can cause state-of-the-art detectors to fail, misclassifying AI-generated text as human-written. This highlights a critical gap in detector evaluation: performance on clean, unperturbed data is not a reliable indicator of real-world effectiveness. To address this, benchmarks like RAID (A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors) have been developed, providing a standardized framework for assessing detector robustness against a comprehensive suite of realistic adversarial evasion techniques.

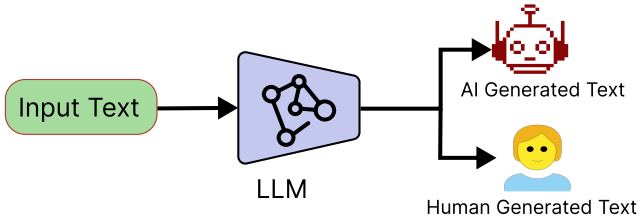

Figure 1: AI generated text detection using LLM.

In this project, we leverage the RAID dataset to investigate two fundamental questions concerning the development of robust detectors:

- Which standard BERT-based architectures offer the best inherent robustness against common adversarial attacks when fine-tuned for AI text detection?
- How does the parameter-efficient fine-tuning technique QLoRA, and specifically its rank hyperparameter, influence the robustness of a BERT-based detector?

To answer these questions, we conduct two sets of experiments. First, we fine-tune four popular BERT variants (BERT-Base, Distil-BERT, Tiny-BERT, and ALBERT) and evaluate their defensive capabilities against ten attack types. Second, we focus on a vanilla BERT model and apply QLoRA, a parameter-efficient fine-tuning method, to analyze how varying the adapter rank (4, 8, 16, 32, 64, 128) affects its resilience. Our contributions include a direct comparative analysis of architectural robustness and a sensitivity analysis of a key QLoRA hyperparameter, offering practical guidance for building detectors that are not only accurate but also resilient.

2 RELATED WORK

Language modeling is a long-standing research topic, dating back to the 1950s with Shannon's application of information theory to human language, where he measured how well simple n-gram language models predict or compress natural language text [9]. Since then, statistical language modeling became fundamental to many natural language understanding systems and generation tasks, ranging from speech recognition, machine translation, to information retrieval [5, 7, 10]. The recent advances on transformer-based large language models (LMs), pretrained on Web-scale text corpora, significantly extended the capabilities of language models (LLMs).

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

For example, OpenAI's ChatGPT and GPT-4 can be used not only for natural language processing, but also as general task solvers to power Microsoft's Co-Pilot systems, for instance, can follow human instructions of complex new tasks performing multi-step reasoning when needed. LLMs are thus becoming the basic building block for the development of general-purpose AI agents or artificial general intelligence (AGI).

As the field of LLMs is moving fast, with new findings, models and techniques being published in a matter of months or weeks [3, 5, 6, 8, 10]. AI researchers and practitioners often find it challenging to figure out the best recipes to build LLM-powered AI systems for their tasks. This paper gives a timely survey of the recent advances on LLMs. We hope this survey will prove a valuable and accessible resource for students, researchers and developers. LLMs are large-scale, pre-trained, statistical language models based on neural networks. The recent success of LLMs is an accumulation of decades of research and development of language models, which can be categorized into four waves that have different starting points and velocity: statistical language models, neural language models, pre-trained language models and LLMs.

Statistical language models (SLMs) view text as a sequence of words, and estimate the probability of text as the product of their word probabilities. The dominating form of SLMs are Markov chain models known as the n-gram models, which compute the probability of a word conditioned on its immediate proceeding $n-1$ words. Since word probabilities are estimated using word and n-gram counts collected from text corpora, the model needs to deal with data sparsity (i.e., assigning zero probabilities to unseen words or n-grams) by using smoothing, where some probability mass of the model is reserved for unseen n-grams [1]. N-gram models are widely used in many NLP systems. However, these models are incomplete in that they cannot fully capture the diversity and variability of natural language due to data sparsity.

3 METHODOLOGY

To systematically evaluate the robustness of AI text detectors, we designed a reproducible experimental pipeline using PyTorch Lightning. Our methodology is structured into three main stages: data preparation, model fine-tuning, and evaluation.

4 DATASET DESCRIPTION

For this project, we utilize the **RAID** benchmark [4], a comprehensive dataset designed for evaluating the robustness of AI-generated text detectors. The dataset contains a mix of human-written and AI-generated texts, with the latter subjected to a variety of adversarial attacks. A preliminary analysis of the dataset's characteristics reveals several key distributions that inform our experimental design. Figure 2 illustrates the distribution of character lengths for the clean, non-attacked human-written and AI-generated (GPT-2) texts used in our training set. Both distributions are heavily right-skewed, indicating that a majority of documents are relatively short. However, the human-written texts exhibit a significantly longer and heavier tail, with some documents extending to nearly 70,000 characters, whereas the AI-generated texts are more concentrated at shorter lengths. This disparity suggests that document length itself could serve as a weak predictive feature, making it crucial for our models to learn more nuanced linguistic patterns.

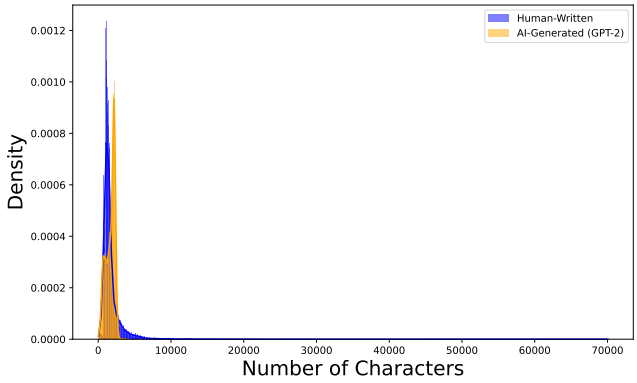

Figure 2: Density plot showing the distribution of character lengths for human-written vs. AI-generated (GPT-2) texts in the training set.

To ensure a balanced evaluation of model robustness, we analyzed the distribution of adversarial attacks within the test data, as shown in Figure 3. The bar plot confirms that the RAID dataset is exceptionally well-balanced across the eleven attack types and the 'none' (no attack) category. Each category contains a nearly identical number of articles (approximately 4,800), which allows for a fair and direct comparison of a detector's performance against each specific evasion technique without bias from varying sample sizes.

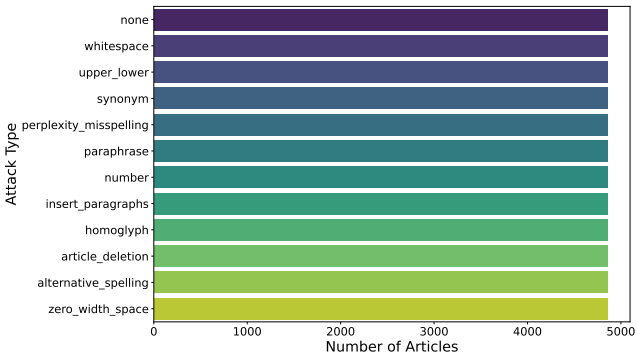

Figure 3: Distribution of articles across different adversarial attack types in the RAID test set, demonstrating a well-balanced dataset.

Finally, we examined the impact of these attacks on the character length of the documents, visualized in Figure 4. The box plots show that the distribution of character lengths remains remarkably consistent across all attack categories. Most adversarial manipulations, such as 'whitespace' or 'synonym' substitution, are subtle and do not significantly alter the macroscopic feature of text length. This uniformity implies that detectors cannot simply rely on document

Attack Type	BERT-Base	Distil-BERT	Tiny-BERT	ALBERT
synonym	**52.89**	50.01	37.67	42.42
article deletion	**53.12**	51.61	38.29	41.42
whitespace	35.12	**52.87**	38.31	40.11
homoglyphs	52.89	**63.17**	33.12	33.32
zero width space	52.58	**52.87**	38.31	0.02
perplexity misspelling	**52.89**	52.88	38.31	40.11
numbers	52.89	**53.62**	38.51	41.76
upper lower	**52.89**	52.87	38.31	40.11
alternative spelling	**48.69**	20.33	38.31	40.22
insert paragraphs	**52.89**	52.87	38.31	40.13

(a) Macro F1 score of different types of BERTs variants where they were trained on non-attacked human and AI generated texts.

Attack Type	rank-4	rank-8	rank-16	rank-32	rank-64	rank-128
synonym	43.45	49.84	50.29	48.61	**51.87**	40.14
article deletion	45.74	51.02	50.57	50.37	**53.37**	45.29
whitespace	45.73	49.71	50.56	49.03	**53.34**	46.62
homoglyphs	31.68	52.94	41.65	33.89	**77.15**	41.67
zero width space	45.73	49.7	50.56	49.03	**53.34**	46.62
perplexity misspelling	45.72	49.71	50.56	49.03	**53.34**	46.63
numbers	46.09	49.87	51.91	50.41	**54.83**	49.58
upper lower	45.73	49.71	50.56	49.03	**53.34**	46.62
alternative spelling	45.73	51.11	50.57	49.08	**53.36**	46.68
insert paragraphs	45.73	49.72	50.56	49.03	**53.34**	46.62

(b) Macro F1 score of Vanilla BERT model where it was trained on non-attacked human and AI generated texts. Rank specifies the LoRA rank used in the QLoRA.

length to identify perturbed text and must instead learn to recognize the underlying structural and semantic artifacts introduced by the attacks.

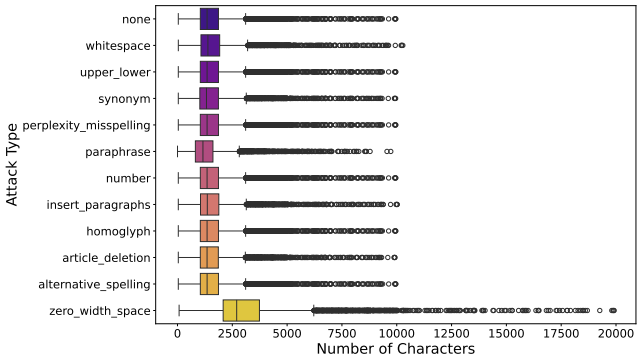

Figure 4: Box plots showing the distribution of character counts for documents subjected to each attack type.

4.1 Dataset Preprocessing

Our study is based on the **RAID (A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors)** dataset, a large-scale benchmark designed specifically for testing detector robustness. The task is a binary classification problem to distinguish between human-written text (label 0) and AI-generated text (label 1).

- **Training Data:** To build a foundational detector, we created a clean training set by combining samples from the `train` split of RAID. This set consists exclusively of non-attacked, human-written texts and non-attacked texts generated by the GPT-2 model. This ensures the models learn the fundamental differences between human and machine writing styles without the influence of adversarial perturbations.
- **Testing Data:** To evaluate robustness, we constructed multiple distinct test sets from the `extra` split of the dataset. Each test set comprises human-written text that has been subjected to one of eleven specific adversarial attacks: synonym, article_deletion, whitespace, homoglyph, zero_width_space, perplexity_misspelling, numbers, upper_lower, alternative_spelling, and insert_paragraphs. This setup allows us to isolate and measure the impact of each attack on the detector's performance.
- **Tokenization:** All texts were tokenized using the AutoTokenizer corresponding to the pre-trained model being used. We applied a maximum sequence length of 512 tokens, truncating longer texts. The data was then organized into a DatasetDict for efficient handling during training and testing.

4.2 Model Architectures and Fine-Tuning

Our investigation follows two experimental tracks to assess robustness from both an architectural and a parameter-efficient fine-tuning perspective, as encouraged by the project guidelines.

- Experiment1:ComparisonofBERTVariants: We selected four widely-used BERT variants from the Hugging Face library: bert-base-uncased, distilbert-base-uncased, prajjwal1/bert-tiny, and albert-base-v2. Each model was fully fine-tuned on our prepared training data for the sequence classification task. This allows for a direct comparison of their inherent architectural resilience to the adversarial attacks in our test sets.
- **Experiment 2: QLoRA Sensitivity Analysis:** We employed Quantized Low-Rank Adaptation (QLoRA) to efficiently fine-tune the `bert-base-uncased` model. QLoRA enables training by quantizing the base model to 4-bits and then inserting small, trainable Low-Rank Adaptation (LoRA) adapters into the attention layers (`query`, `value`). Only these adapter weights are updated during training, significantly reducing memory and computational requirements. To analyze the impact of adapter capacity on robustness, we conducted a sensitivity analysis on the LoRA **rank `r`** hyperparameter by training separate models for each rank in the set {4, 8, 16, 32, 64, 128}.

4.3 Experimental Setup

All experiments were conducted using a consistent set of hyperparameters to ensure fair comparisons. The models were trained for 5 epochs with a batch size of 128. We used the AdamW optimizer with a learning rate of 5e-5 and a weight decay of 0.1. A linear learning rate scheduler with 500 warmup steps was applied to stabilize training. The entire training and evaluation pipeline was managed using PyTorch Lightning, with `bfloat16` precision enabled for QLoRA experiments to further optimize efficiency.

4.4 Evaluation

The primary metric for our evaluation is the **Macro F1 Score**, which is well-suited for classification tasks as it balances precision and recall. After a model was trained on the clean training set, it was evaluated separately against each of the adversarial test sets. This process yields a distinct F1 score for each attack type, allowing for a granular analysis of a model's specific vulnerabilities and strengths.

5 RESULTS AND ANALYSIS

Our experiments produced two sets of results, presented in Table 1a and Table 1b. These tables allow us to analyze the robustness of AI text detectors from the perspective of both model architecture and fine-tuning strategy. Our primary evaluation metric is the Macro F1 Score, which effectively balances precision and recall for our binary classification task.

5.1 Architectural Robustness of BERT Variants

Our first experiment compares the performance of four fully fine-tuned BERT variants against eleven adversarial attacks. The results, shown in Table 1a, reveal several key insights.

- **Overall Performance:** Among the models tested, **DistilBERT** emerges as the most robust overall performer, achieving the highest F1 scores on seven of the eleven attack types. This suggests that its knowledge distillation process, which aims to preserve the performance of a larger model in a smaller footprint, may also confer a degree of resilience against common adversarial perturbations. In contrast, **TinyBERT**, the smallest model, consistently yields the lowest scores, indicating that its reduced parameter count compromises its ability to generalize against perturbed inputs.
- **Model-Specific Vulnerabilities:** The results highlight significant model-specific weaknesses. Most notably, **ALBERT** suffers a complete performance collapse against the zero_width_space attack, with an F1 score of just 0.02. This suggests a critical failure in its tokenizer or embedding layer to handle these invisible characters. Similarly, while DistilBERT is a strong performer, it is exceptionally vulnerable to the alternative_spelling attack, where its F1 score plummets to 20.33, the lowest of any model against that specific attack.
- **Most Effective Attacks:** The **homoglyph** attack proves to be a formidable challenge for most architectures, significantly degrading the performance of BERT-Base (31.68) and ALBERT (33.32). This attack, which replaces characters with visually identical Unicode characters, effectively bypasses tokenizers that are not trained to handle such variations.

This comparative analysis suggests that there is no single "best" architecture; rather, there is a trade-off between model size, efficiency, and robustness. While Distil-BERT provides a strong baseline, its specific vulnerabilities underscore the importance of testing against a diverse range of attacks.

5.2 Sensitivity Analysis of QLoRA Rank

Our second experiment investigates the impact of the QLoRA rank hyperparameter on the robustness of a fine-tuned vanilla BERT model. The results, presented in Table 1b, demonstrate that the choice of rank is not merely a matter of computational efficiency but is critical to the final performance of the detector.

- **Existence of an Optimal Rank:** A clear trend emerges where performance generally increases with rank, but only up to an optimal point. Across all eleven attack categories, a QLoRA rank of **r=64** achieves the highest Macro F1 score. This suggests that r=64 provides the LoRA adapter with sufficient capacity to learn robust features against adversarial noise without overfitting.
- **Diminishing Returns and Overfitting:** Increasing the rank beyond this optimal point leads to a noticeable degradation in performance. The model with a rank of **r=128** performs worse than the r=64 model on every single attack type. This indicates that a higher-rank adapter, while having more trainable parameters, may begin to overfit to the specific nuances of the clean training data, thereby losing its ability to generalize to the perturbed, out-of-distribution data present in the test sets.
- **Performance at Low Ranks:** The models with low ranks (e.g., r=4 and r=8) provide a reasonable baseline but are clearly outperformed by higher-rank models. The F1 score on the challenging `homoglyph` attack, for instance, jumps from 31.68 at r=4 to a remarkable 77.15 at r=64. This shows that while QLoRA is parameter-efficient, a certain minimum capacity (rank) is necessary to build a truly robust detector.

This sensitivity analysis confirms that the QLoRA rank is a crucial hyperparameter that directly influences model robustness. The results strongly suggest that selecting a rank is a task of finding a "sweet spot" that balances adapter capacity and generalization, rather than simply maximizing the number of trainable parameters. For this specific task and model, a rank of 64 represents this optimal point.

6 CONCLUSION

In this project, we successfully conducted an evaluation of the robustness of AI text detectors, addressing the challenge posed by adversarial attacks as outlined in the RAID benchmark. Our experiments, designed to fulfill the project's requirements for thorough comparison and sensitivity analysis, yield two primary contributions. First, our comparative analysis of four BERT variants reveals that architectural choice is a critical factor in detector resilience. We found that while no single model excelled against all attack types, Distil-BERT provided a strong performance baseline, underscoring the need for diverse testing. Second, our sensitivity analysis of QLoRA fine-tuning demonstrated that parameter-efficient methods are highly effective, but performance is critically dependent on the choice of hyperparameters like the adapter rank. We identified an optimal rank of 64 for our task, beyond which performance degraded, highlighting a clear trade-off between model capacity and generalization against adversarial perturbations. Ultimately, this work demonstrates that building robust AI text detectors requires a holistic approach that considers both the underlying model architecture and the specifics of the fine-tuning strategy.

7 GENAI DISCLOSURE STATEMENT

We acknowledge the use of Claude AI for editorial assistance in improving grammar and correcting writing errors. All technical implementation, including code development and dataset processing, was performed without generative AI support.

REFERENCES

[1] Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language* 13, 4 (1999), 359–394.

[2] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems* 36 (2023), 10088–10115.

[3] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[4] Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 12463–12492. https://aclanthology.org/2024.acl-long.674

[5] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints* (2023).

[6] Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403* (2022).

[7] Frederick Jelinek. 1998. *Statistical methods for speech recognition*. MIT press.

[8] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys* 55, 9 (2023), 1–35.

[9] Claude E Shannon. 1951. Prediction and entropy of printed English. *Bell system technical journal* 30, 1 (1951), 50–64.

[10] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2024. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics* (2024), 1–65.