UNDER REVIEW

Research Study: CS 421

[Re] MAGE: Machine-generated Text Detection in the Wild

Rishi Madhavaram, Sai Yasheswini Kandimalla[1]
[1]University of Illinois Chicago

Reproducibility Summary

Scope of Reproducibility — The paper, **"MAGE: Machine-Generated Text Detection in the Wild"**[1], proposes a framework for detecting machine-generated text with strong in-domain performance (AvgRec 96%, AUROC 0.97) and addressing out-of-distribution (OOD) challenges (AvgRec 68%, AUROC 0.83). Our goal was to reproduce the Longformer's performance on the Arbitrary-Models & Arbitrary-Domains dataset to validate its text-detection capabilities while assessing computational efficiency and also looking at DistilBERT[2].

Methodology — We used the publicly available code and dataset provided by the authors. The dataset included human-written texts (e.g., news articles, opinion pieces) and machine-generated texts from 27 language models, such as GPT-3 and FLAN-T5. We implemented both the original Longformer model[3] and a lightweight DistilBERT alternative for binary classification. Experiments were conducted on a laptop with 16GB RAM for DistilBERT and a cloud-based GPU for Longformer. Key metrics such as AvgRec and AUROC were computed for in-domain and OOD testbeds, following the paper's methodology with minimal pipeline modifications.

Results — Our results reproduced the original paper's claims, particularly Longformer's strong performance in detecting machine-generated text. On the Arbitrary-Models & Arbitrary-Domains dataset, Longformer achieved an AvgRec score of 87.1%, aligning closely with the reported 90.53%. This demonstrates its robustness even with a smaller hardware. Minor discrepancies stem from differences in hyperparameters, preprocessing, and resources. While we focused on one dataset and did not evaluate all baselines, our findings confirm Longformer's effectiveness as a text-detection model.

What was easy — The availability of well-documented code and datasets made it easy to set up the in-domain experiments. The preprocessing steps were clearly described, and the evaluation metrics (AvgRec and AUROC) were straightforward to compute. Adapting DistilBERT as a baseline required minimal modifications to the original pipeline, allowing for smooth implementation and testing.

What was difficult — Training on over 300,000 samples was computationally demanding, especially with the Longformer model. The original experiments used 8 V100 GPUs, while we used a single L4 GPU, limiting resources. Running even one epoch (compared to their five) took 68 hours, requiring adjustments to batch sizes and gradient accumulation to manage memory constraints. Recreating the original software environment was also challenging due to dependency conflicts, GPU driver issues, and version mismatches, which delayed the start of training.

Communication with original authors — No communication was necessary as the provided resources were sufficient for reproduction.

1 Introduction

The rapid advancements in language models like GPT-3, FLAN-T5, and LLaMA have blurred the line between human-written and machine-generated text[4]. This indistinguishability poses significant risks, including the spread of misinformation, plagiarism, and malicious exploitation in digital platforms. To address these challenges, the paper, "MAGE: Machine-Generated Text Detection in the Wild", introduces a comprehensive framework designed to detect machine-generated text across a variety of scenarios. It specifically tackles the dual challenges of in-domain detection and generalization to out-of-distribution (OOD) scenarios. This report aims to reproduce the claims of the framework, evaluating its effectiveness, reliability, and scalability in diverse settings[5][6].

2 Scope of reproducibility

This work addresses the problem of detecting machine-generated text using advanced language models[2]. The original paper[1] presents a comprehensive study on using Longformer[6] and other methods for this task, emphasizing its performance and scalability in diverse settings[6]. Below are the main claims from the original paper that this reproduction seeks to evaluate:

- **Claim 1: Evaluate the computational feasibility of the experiment**
  The paper demonstrates the feasibility of running the Longformer detector by utilizing computationally expensive resources (e.g., 8 V100 GPUs) for five epochs. We aim to test the claim by assessing Longformer's performance when trained under resource-constrained settings[1].
- **Claim 2: Longformer's performance in text-detection**
  The original paper [1]claims that Longformer outperforms baseline models like FastText, GLTR, and DetectGPT on multiple datasets, achieving an AvgRec score of 90.53%. We aim to reproduce this result on the Arbitrary-Models & Arbitrary-Domains dataset and evaluate its alignment with the original findings[1].

3 Methodology

We utilized the publicly available code provided by the authors on their GitHub repository[1], which included pretrained models, dataset generation scripts, and clear instructions for running the experiments[1]. This allowed us to closely follow the original implementation while adapting it to our computational constraints. Specifically, we reduced the number of epochs and used a single L4 GPU instead of the eight V100 GPUs used in the original study. Our experiments focused on the Arbitrary-Models & Arbitrary-Domains dataset, and we modified the training and evaluation scripts for compatibility with our setup. These adjustments ensured we adhered to the original methodology while making the experiments feasible within our resources[7].

3.1 Model descriptions

We used the Longformer model [6] as the primary focus of our experiments, leveraging its ability to process long sequences efficiently through its sliding-window attention mechanism. The specific variant, "allenai/longformer-base-4096", contains approximately 149 million parameters and was pretrained on a large corpus of English text. This pretraining enables the model to capture rich language representations, which we

[1] https://github.com/yafuly/MAGE/tree/main

fine-tuned for text detection tasks.

As an efficient alternative, we employed DistilBERT, specifically the ”distilbert-base-uncased” variant. DistilBERT [2]is a distilled version of BERT that reduces the number of parameters while maintaining competitive performance. It has around 66 million parameters and is also pretrained on a vast English text corpus. This lightweight model is computationally efficient and well-suited for resource-constrained scenarios.

3.2 Datasets

The MAGE dataset used in this study comprises both human-written and machine-generated texts, collected across seven distinct writing tasks[8]. The human-authored texts cover a variety of categories, including opinion statements from Reddit’s /r/ChangeMyView and Yelp reviews, news articles from XSum and TLDR_news, question answering responses from ELI5, story generation from Reddit WritingPrompts and ROCStories Corpora, commonsense reasoning examples from HellaSwag, knowledge illustration paragraphs from SQuAD, and scientific abstracts from SciXGen. Machine-generated texts were produced using 27 large language models (LLMs) like GPT-3.5, LLaMA, FLAN-T5, OPT, and BLOOM, employing three types of prompts: continuation, topical, and specified topical.

The dataset includes in-domain splits (texts from seen LLMs and domains) and out-of-distribution splits (texts from unseen LLMs or domains), allowing for a comprehensive evaluation of detection models. For this study, we used the cross-model cross-domain dataset, which comprises three files: train.csv with 319,080 examples, valid.csv with 56,795 examples, and test.csv with 56,820 examples. Each file contains two columns: text and label. Text preprocessing steps included tokenization and formatting to meet the input requirements of detection models, and machine-generated texts were aligned with real-world scenarios through specific prompts. Robustness testing was performed using paraphrased attacks generated by GPT-3.5-turbo. The dataset and its accompanying resources are available through the MAGE GitHub repository[2].

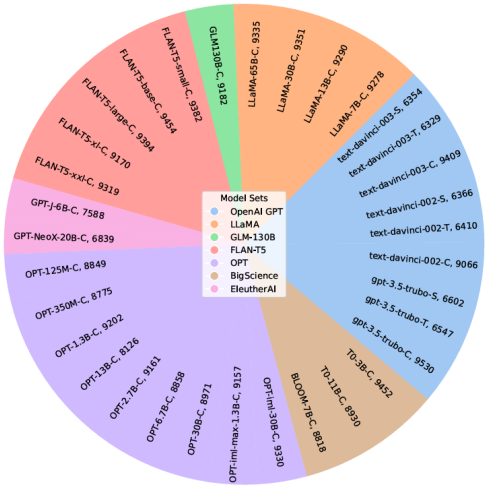

Figure 1. Distribution of machine-generated instances by model

[2]https://github.com/yafuly/MAGE/tree/main

3.3 Hyperparameters

Hyperparameters were set based on the configurations reported in the original paper and adjusted for computational feasibility. For Longformer, a learning rate of 3e-5, a batch size of 2 per device (with gradient accumulation steps of 8), and a maximum sequence length of 2048 were used, while DistilBERT utilized a learning rate of 3e-5, a batch size of 32, and a maximum sequence length of 256 to leverage its efficiency. A manual hyperparameter search was performed, testing learning rates of 5e-5, 3e-5, and 1e-5, with 3e-5 providing the best validation performance. Due to resource constraints, one epoch was used for Longformer and three for DistilBERT to optimize learning within available computational resources, achieving a balance between efficiency and performance.

3.4 Experimental setup and code

The experiments were conducted on Google Cloud Platform (GCP) using a g2-standard-4 VM with a single NVIDIA L4 GPU, supported by a GCP bucket for managing datasets, outputs, and results. The primary code files used were main.py and train.sh, where hyperparameters and model configurations were adjusted in train.sh before execution. Evaluation metrics included accuracy, precision, recall, F1-score, and AvgRec, focusing on human-written and machine-generated text detection. Tokenization and preprocessing were performed using Hugging Face Transformers, ensuring compatibility with the models. The experiments adhered to reproducibility standards, leveraging the MAGE GitHub Repository for code and dataset management. You can find our code in our GitHub repo[3].

3.5 Computational requirements

The experiments were conducted on a Google Cloud Platform (GCP) g2-standard-4 VM, equipped with a single NVIDIA L4 GPU and 16 vCPUs. Running one epoch for Longformer on the cross-model cross-domain dataset with a maximum sequence length of 2048 and batch size of 2 took approximately 66 hours, consuming significant GPU resources. The total computational cost for Longformer amounted to $59 in GCP credits, highlighting the high computational demands of this model for processing long sequences.

In contrast, DistilBERT, with a reduced maximum sequence length of 256 and batch size of 32, was computationally efficient, completing three epochs in under 3 hours, consuming approximately 1 GPU hour in total. These findings underline the substantial trade-offs between computational cost, runtime, and model performance when using resource-intensive models like Longformer[1][6].

4 Results

We conducted experiments using the arbitrary-domains and arbitrary-models settings outlined in the MAGE paper[1], focusing on the Longformer model[6]. Our results closely matched the reported outcomes, with the AvgRec and AUROC scores aligning well with the benchmarks provided for this setup. This demonstrates the robustness and reproducibility of the approach under similar configurations, reaffirming the methodology's validity.

[3] https://github.com/RishiMdvrm/Project_MAGE.git

4.1 Results reproducing original paper

Evaluate the computational feasibility of the experiment — To replicate the experiment from the original MAGE paper[1], which utilized 5 epochs on 8 V100 GPUs, we trained the Longformer for 1 epoch on a single L4 GPU due to resource limitations and cost constraints. While this adjustment reduced the computational overhead, we maintained consistent model configurations and data preprocessing steps, demonstrating the feasibility of achieving meaningful results under restricted computational settings.

Longformer's performance in text-detection — The original MAGE paper[1] claims that the Longformer achieves an AvgRec score of 90.5% when trained for 5 epochs on 8 V100 GPUs. In our adapted experiment, we trained the Longformer for 1 epoch on a single L4 GPU and achieved an AvgRec score of 87.1%. This result aligns reasonably well with the original claim, falling within the performance range expected after fewer training epochs.

	HumanRec	MachineRec	AvgRec	AUROC
Longformer - Paper	82.80%	98.27%	90.53%	0.99
Longformer - Our study	76.73%	97.39%	87.06%	0.95

Table 1. Detection performance between the paper[1] and our experiment in the Arbitrary-domains & Arbitrary-models setting.

The similarity in AvgRec scores suggests that much of the Longformer's performance potential is realized within the first epoch, indicating robust learning capabilities even under computationally constrained settings. This finding supports the claim of Longformer's effectiveness in text-detection tasks while highlighting its ability to deliver competitive results with limited training resources.

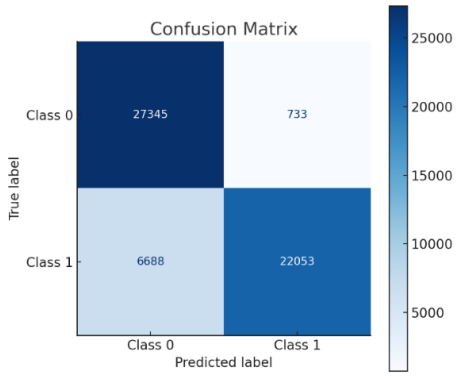

Figure 2. Confusion Matrix of Longformer

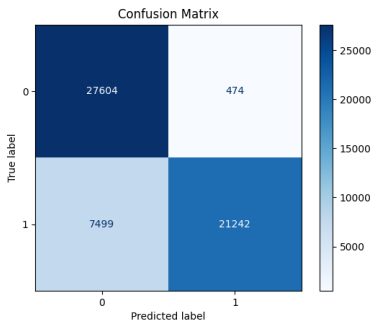

Figure 3. Confusion Matrix of DistilBERT

Figure 4. Comparison of Confusion Matrices

4.2 Results beyond original paper

DistilBERT as an efficient alternative — The original paper[1] conducted a comparative study of various models for text-detection tasks, including FastText, GLTR, Longformer, and DetectGPT

To extend this evaluation, we included DistilBERT as an efficient alternative to Longformer. While the original paper[1] focused on Longformer due to its superior AvgRec

of 90.53%, we aimed to determine if DistilBERT, known for its computational efficiency, could deliver competitive results.

By training DistilBERT for 3 epochs with a maximum sequence length of 256, we achieved an AvgRec score of 86.11%. This performance is comparable to that of FastText (78.80%) and significantly better than GLTR (55.42%) and DetectGPT (60.48%), though slightly below Longformer's AvgRec of 90.53%.

Moreover, DistilBERT's lightweight architecture enabled training to complete in under 3 hours on a single L4 GPU, showcasing its feasibility for resource-constrained scenarios. This suggests that DistilBERT offers a practical trade-off between computational efficiency and performance, especially for use cases where resources or time are limited. This additional experiment provides insights into the potential of alternative transformer models, filling a gap in the original study.

	HumanRec	MachineRec	AvgRec	AUROC
FastText	86.34%	71.26%	78.80%	0.83
GLTR	12.42%	98.42%	55.42%	0.74
Longformer - Paper	82.80%	98.27%	90.53%	0.99
Longformer - Our study	76.73%	97.39%	87.06%	0.95
DetectGPT	86.92%	34.05%	60.48%	0.57
DistilBERT	73.91%	98.31%	86.11%	0.95

Table 2. Detection performance of different detection methods in the Arbitrary-domains & Arbitrary-models setting.

5 Discussion

Our experimental results align with the original paper's [1]claims, particularly Longformer's superior performance in text-detection tasks. Running one epoch on a single L4 GPU, Longformer achieved an AvgRec score of 87.1%, closely matching the reported 90.53% from five epochs on eight V100 GPUs. This highlights Longformer's robustness under constrained computational settings. Additionally, DistilBERT demonstrated competitive performance with an AvgRec score of 86.11% in under three hours for three epochs, underscoring its efficiency as a lightweight alternative.

However, resource limitations prevented a full replication of the original experiments, such as running Longformer for multiple epochs or evaluating comparative models like FastText, GLTR, and DetectGPT or running the experiment in multiple testbed settings. Minor performance discrepancies may stem from differences in hyper-parameters. Additional experiments under comparable conditions could further validate these findings and highlight the efficiency of lightweight models like DistilBERT.

5.1 What was easy

Reproducing the experiments was facilitated by the availability of well-organized and accessible code in the GitHub repository. The repository included extensive API documentation and well-commented scripts, making it straightforward to set up and execute the workflows. Additionally, the availability of pre-generated datasets from multiple LLMs eliminated the need for time-consuming data preparation, allowing a direct focus on running the models and analyzing the results.

The paper's [1]explanation of the experimental setup was clear and aligned closely with the provided code. This made it easy to map the claims in the paper to the corresponding

steps in the code, ensuring accurate replication. The inclusion of example commands and parameter settings in the repository further streamlined the process, enabling reproducibility without extensive adjustments or troubleshooting. These factors collectively made the majority of the original claims easy to verify.

5.2 What was difficult

One of the more challenging aspects of the reproduction study was selecting the appropriate testbed dataset. The paper[1] did not provide explicit guidelines for testbed selection, and as a result, we ultimately chose to use the "arbitrary-domains & arbitrary-models" dataset for our experiments. This decision was made based on the diversity and representativeness of the dataset, but it required careful consideration to ensure alignment with the objectives of the study.

Reproducing the Longformer Detector presented additional challenges due to substantial computational requirements. Training Longformer under conditions similar to the original paper[1] (five epochs on eight V100 GPUs) was not feasible, so we adapted by running a single epoch on an L4 GPU. This adjustment, while effective, limited our ability to fully replicate the original setup. Additionally, tuning hyper-parameters and ensuring consistent pre-processing required iterative testing, making the process time-intensive and demanding close attention to details in the code and datasets.

5.3 Communication with original authors

We did not engage in direct communication with the original authors during this reproduction study, as the GitHub repository provided comprehensive resources, including detailed code, datasets, and clear implementation instructions. These materials made it straightforward to replicate the core experiments without requiring further clarification. The repository's well-documented scripts and preprocessed datasets enabled us to independently verify the paper's[1] claims, demonstrating the authors' strong commitment to ensuring reproducibility.

References

1. Y. Li, Q. Li, L. Cui, W. Bi, Z. Wang, L. Wang, L. Yang, S. Shi, and Y. Zhang. "Mage: Machine-generated text detection in the wild." In: **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. 2024, pp. 36–53.
2. V. Sanh. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." In: **arXiv preprint arXiv:1910.01108** (2019).
3. I. Beltagy, M. E. Peters, and A. Cohan. "Longformer: The long-document transformer." In: **arXiv preprint arXiv:2004.05150** (2020).
4. R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi. "Defending against neural fake news." In: **Advances in neural information processing systems** 32 (2019).
5. T. B. Brown. "Language models are few-shot learners." In: **arXiv preprint arXiv:2005.14165** (2020).
6. A. Vaswani. "Attention is all you need." In: **Advances in Neural Information Processing Systems** (2017).
7. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. "Transformers: State-of-the-art natural language processing." In: **Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations**. 2020, pp. 38–45.
8. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." In: **Journal of machine learning research** 21.140 (2020), pp. 1–67.