Spotlights and Blindspots: Evaluating Machine-Generated Text Detection

Kevin Stowe, Kailash Patil
Pindrop
{kevin.stowe, kpatil}@pindrop.com

Abstract

With the rise of generative language models, machine-generated text detection has become a critical challenge. A wide variety of models is available, but inconsistent datasets, evaluation metrics, and assessment strategies obscure comparisons of model effectiveness. To address this, we evaluate 15 different detection models from six distinct systems, as well as seven trained models, across seven English-language textual test sets and three creative human-written datasets. We provide an empirical analysis of model performance, the influence of training and evaluation data, and the impact of key metrics. We find that no single system excels in all areas and nearly all are effective for certain tasks, and the representation of model performance is critically linked to dataset and metric choices. We find high variance in model ranks based on datasets and metrics, and overall poor performance on novel human-written texts in high-risk domains. Across datasets and metrics, we find that methodological choices that are often assumed or overlooked are essential for clearly and accurately reflecting model performance.

Keywords: machine-generated text detection, deepfake detection, evaluation, metrics

1. Introduction

Recent years have witnessed remarkable advancements in generative systems capable of producing realistic video, audio, and text. While these technologies offer significant benefits, they also introduce serious challenges in verifying the origin of content. Distinguishing between machine-generated and human-written text, in particular, has become increasingly critical. In response, research into machine-generated text detection[1] – identifying and sourcing content produced by generative systems – has grown rapidly.

Machine-generated text detection is essential across domains such as education, media, and security, with use cases including the mitigation of spam, plagiarism, fraud, propaganda, and more (Rosca et al., 2025; Lee et al., 2023; Saravani et al., 2021). Modern large language models (LLMs) pose unique challenges, as even humans have difficulty detecting machine-generated text (Lee et al., 2025; Uchendu et al., 2023). Evaluation practices are inconsistent, with significant variations in datasets, metrics, and methodologies, and the rationale behind these methodological choices is often unclear. While previous research has explored the effects of data and metrics (Pudasaini et al., 2025; Bhattacharjee et al., 2024; Zhang et al., 2023), the scope of this work is limited in scope based on models, datasets, metrics, and depth of analysis.

To address this gap, we examine 15 detection variants across six systems, alongside pretrained transformer- and feature-based models. Our work uniquely takes a deeper look at the dataset and metric attributes that are often overlooked, and in doing so, brings to light essential issues in model performance and development. Our analysis uncovers findings in two main areas:

- **Data**: Model performance varies substantially depending on the evaluation data: across four datasets, F1 scores range from approximately 0 to 0.982, with nearly all models performing well on certain datasets and weak on others (Section 4). With regard to training, fine-tuning on in-domain data yields models that outperform zero-shot and externally trained public models, though for some datasets, out-of-domain training yields better results (Section 4.1). On three novel human-written datasets, all but three variants exhibit an error rate of at least a 15% on one or more datasets, while those with lower error rates also suffered from low recall across all datasets. This highlights potential risks in real-world deployment (Section 4.2).

- **Metrics**: Metrics can be exploited to misrepresent model performance: common metrics like F1 score, area under the receiver operating curve (AUROC), and true positive rate at false positive rate 1% (TPR@FPR 1%) pattern differently depending on the label distribution of the evaluation set, which are frequently imbalanced (Section 5.1). Over eight commonly used metrics, model rankings variances ranges from 0.77 to 15.25 over 15 models, with

[1] As a terminological note, we prefer the term "machine-generated" to "deepfake," as there may not be any intent to deceive. This also mirrors the language of the datasets used.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 4173–4187
11–16 May 2026.

the choice of classification threshold having a significant impact (Section 5.2).

We conclude with an analysis of potential causes of model disparity, including input length, punctuation, repetition, and perplexity. Our analysis shines a light on dataset and metric-related aspects of the task that are often overlooked. We establish the necessity of using multiple, well-motivated metrics and datasets for evaluation, as errors can be obfuscated and model performance misinterpreted depending on evaluation settings.

2. Systems

We assemble a diverse collection of contemporary machine-generated text detection systems, including zero-shot and trained public models, pretrained transformers, and feature-based approaches. Table 1 summarizes the selected model variants, detailing their original evaluation data, metrics, and reported performance. Our goal is to explore impacts of datasets and metrics through rigorous evaluation and analysis as in Wu et al. (2025); we build on their work by including more models, as well as a deeper analysis of performance as it relates to metrics and datasets.

2.1. Public Models

We explore both zero-shot and trained models that are publicly available. We focus on detection models that satisfy three key criteria: (1) contemporary (since 2023), (2) fully open-source (enabling transparency and customization), and (3) free to use (excluding systems dependent on proprietary APIs like OpenAI, which incur additional expenses and requirements). We prioritize transparent, easily implementable systems with minimal operational costs. These systems all report strong performance, with reported F1/AUROC scores over 0.95 on many of their respective datasets.

2.2. Pretrained Transformers

We include in our evaluation fine-tuning of five transformer-based architectures of varying scales and configurations: DistilBERT (Shah et al., 2023), BERT (Devlin et al., 2019), RoBERTa (Zhuang et al., 2021), Longformer (Beltagy et al., 2020), and DeBERTa (He et al., 2021). Notably, prior work has shown RoBERTa and Longformer to be particularly effective for machine-generated detection (Mozilla, 2025; Li et al., 2024; Pu et al., 2023). We fine-tune and evaluate each of these models on a shared dataset, enabling direct comparison with the above approaches.

2.3. Feature-based Models

Feature-based models have demonstrated notable effectiveness in machine-generated text detection, leveraging the distinct linguistic and stylometric patterns characteristic of different LLMs (McGovern et al., 2025; Muñoz-Ortiz et al., 2024). We implement two variations: (1) a feature-based classifier based on the work of McGovern et al. (2025) (MCGOVERN), which uses a combination word, part-of-speech, and character n-grams, trained with a gradient boosting classifier, and (2) a series of linguistic and stylometric features adapted from Almazrouei et al. (2023) (STYLO). This model combines linguistic and stylometric features to train an ensemble of classifiers. [2]

2.4. Other models

We aim for a diverse set of models based on the above criteria, but many other models are available. Some were excluded for targeting specific generative models (e.g., GPT) or datasets (Venkatraman et al., 2024; Koike et al., 2024; Wu et al., 2023; Yang et al., 2023; Gehrmann et al., 2019). Others were built on free models from OpenAI which are no longer available or require restricted model access (Bao et al., 2025; Verma et al., 2024; Mao et al., 2024): we restrict our evaluation to only free and open models. We also exclude the many commercial systems for similar reasons, as well as the inherent opacity of proprietary models.

3. Data

To ensure standardized, robust evaluations, we establish a unified dataset comprising seven test sets derived from four benchmark datasets:

MAGE (aka Deepfake) (Li et al., 2024): This dataset consists of a 447k human and AI-generated text samples covering diverse models and methodologies. We extract three test sets: (1) a class-balanced sample of 10k from the full dataset (MAGE); and (2-3) the two "wilder" test sets, one for out-of-domain texts (MAGE-OOD) and one for out-of-domain paraphrases (MAGE-OOD-PARA).

RAID (Dugan et al., 2024): The RAID corpus comprises 11 million human-written and machine-generated texts with adversarial examples crafted to evade detection systems. We extract a class-balanced subset of 10k samples from the RAID training partition (RAID), as the official test set labels are withheld for leaderboard purposes.

[2] Implementation details for all systems in Appendix B.

Zero-shot Models	Variants	Evaluation Data	Key Performance
Binoculars (Hans et al., 2024)	Falcon	Verma et al. (2024), Lian et al. (2023), custom	TPR-FPR: 0.76 - 0.98, F1: 0.985 - 0.994
Fast-DetectGPT (FDG) (Bao et al., 2024)	gpt-neo gpt-j falcon-7b	Custom	AUROC: 0.9754 - 0.9984
Zippy (Thinkst, 2023)	LZMA Ensemble	Various [a]	AUROC: 0.76 - 0.82
Trained Models	**Variants**	**Evaluation Data**	**Key Performance**
RADAR (Hu et al., 2023)	Base	Custom	AUROC: 0.763 - 0.955
BiScope (Guo et al., 2024a)	Arxiv Yelp Essay Creative	Custom	F1: 0.5456 - 1.0
DeTeCtive (Guo et al., 2024b)	MAGE (Deepfake) M4 TuringBench OUTFOX	MAGE (Deepfake) M4 TuringBench OUTFOX	F1: 0.8260 - 0.9974
Pretrained Transformers[b]	**Model**	**Evaluation Data**	**Key Performance**
DistilBERT	`distilbert-base-cased`	N/A	N/A
BERT	`bert-base-cased`	N/A	N/A
RoBERTa	`roberta-base`	N/A	N/A
Longformer	`longformer-base-4096`	N/A	N/A
DeBERTa	`deberta-v3-base`	N/A	N/A
Feature-Based	**Features**	**Evaluation Data**	**Key Performance**
MCGOVERN	Word, part-of-speech, character n-grams	MAGE (Deepfake) M4 Guo et al. (2023) OUTFOX Ghostbuster	AUROC: 0.943 - 0.996 F1: 0.947 - 0.987
STYLO[c]	Linguistic/stylometric	N/A	N/A

[a] See https://blog.thinkst.com/2023/06/meet-zippy-a-fast-ai-llm-text-detector.html

[b] While previous work has included transformer fine-tuning, there is no consensus on dataset, metrics, or performance.

[c] This system is implemented independent of any specific previous work.

Table 1: Summary of the models and variants evaluated. We report the main datasets and metrics used, along with the range of values for the given system for the key metric(s).

H3C+ Corpus (Su et al., 2024): This corpus extends the H3C corpus (Guo et al., 2023) with semantic-invariant perturbations, shown to be more difficult to detect. We utilize their two English test sets, sampling 10k class-balanced texts each from the English partitions for question answering (H3C+ QA) and semantic invariance (H3C+ SI).

M4GT-Bench (Wang et al., 2024): A multilingual, multidomain corpus covering several generated methods. We take a class-balanced sample of 10k texts from the English partition of their test corpus (M4GT).

These datasets were chosen due to (1) public availability, (2) diversity in models and domains, and (3) common usage. They each contain many models used for generation, different domains, and different styles of text. There are many other high-quality datasets available (Wu et al., 2025; Koike et al., 2024; Verma et al., 2024; Liu et al., 2024; Lian et al., 2023; Uchendu et al., 2021); see Gritsai et al. (2025) for a contemporary overview of datasets and resulting challenges in detection. Our goal is not to comprehensively evaluate models against all datasets, but rather highlight disparities in performance across representative datasets.

We conduct comprehensive evaluations across all seven test sets, including an aggregated analysis of the combined datasets (ALL). Some public models (e.g., DeTeCtive (MAGE)) were trained on

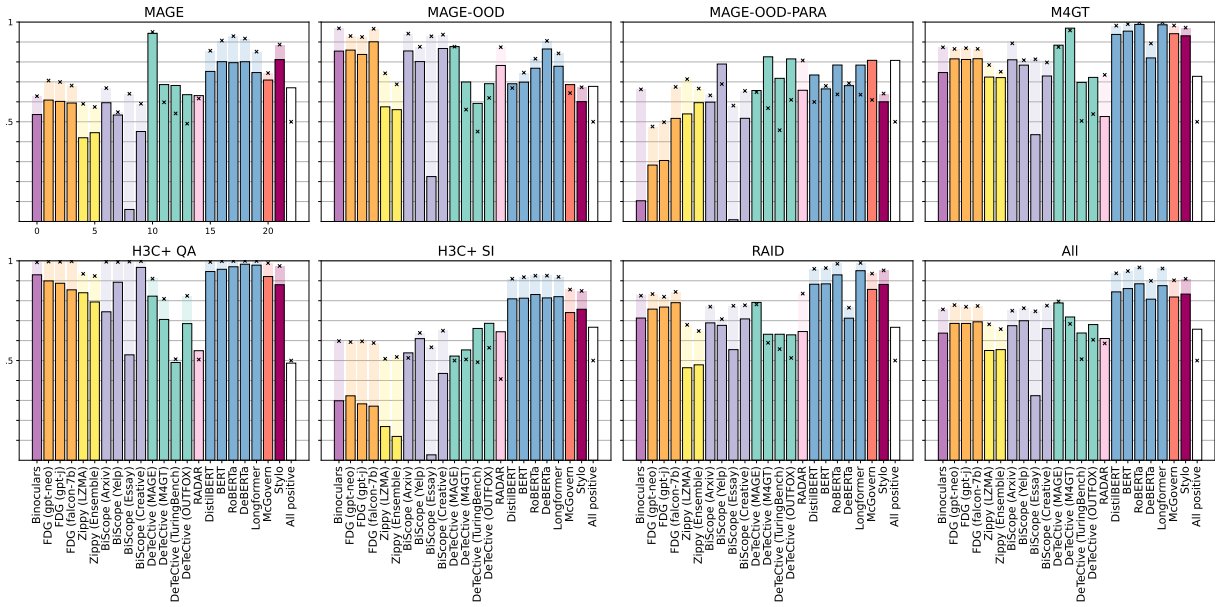

Figure 1: F1 and AUC (marked with ×) for each model on each dataset. For the pretrained transformers, these were trained on in-domain training data from the respective dataset.

other partitions from these resources: before evaluation, we have ensured that there is no overlap between the training data and our evaluation sets. For our training, we create training sets of 10k texts for each dataset by random sampling from the respective training partitions, ensuring no overlap with the evaluation data.

4. Experiments

We evaluate each system on each of the eight datasets described above. For comparison include a trivial baseline system (All positive) that assigns every sample a score of 1 (machine-generated).[3] For our initial evaluation, we use F1 score (threshold = 0.5) and AUROC; we discuss more on metrics in Section 5. Results are shown in Figure 1.

We start by assessing some key trends in our evaluation. First, performance varies substantially across models, with no system dominating uniformly. Even architecturally similar models exhibit irregular performance patterns.

Among public models, DeTeCtive (MAGE) achieves the highest aggregate F1/AUROC scores, though Binoculars, BiScope, Fast-DetectGPT, RADAR, and Zippy perform competitively on specific test sets. In several cases, the "All positive" baseline outperforms many systems, particularly with regard to F1 score (e.g., on the MAGE-OOD dataset). As some datasets are heavily weighted towards machine-generated content, this baseline is deceptively strong, and underscores the limitations of F1 score as a metric (Section 5).

Models exhibit significant variability in performance on their best and worst datasets as well, as detailed in Table 2. Binoculars has the highest minimum AUROC (0.60): *every* model explored performs near chance ($\leq$0.60) on at least one of the evaluation sets, yet these models all perform strongly on others. Excepting the TuringBench DeTeCtive variant, all models scored $\geq$0.82 AUROC on their best dataset. This underscores the importance of diverse evaluation data: models can easily be over- or under-represented best on their performance on different datasets.

Pretrained models outperform publicly available systems in this setting, in which they are trained on in-domain data. The transformer variants all exhibit similar performance. DeBERTa exhibits higher variance, while the other four models are consistent. Feature-based models, both the base n-grams of MCGOVERN and the more complicated stylometric features, are notably strong, achieving performance comparable to transformers in many cases. We assess the ability of models trained in a cross-domain setting in Section 4.1.

Performance disparities are stark: while most models excel on the H3C+ QA dataset, the H3C+ SI and MAGE-OOD-PARA datasets prove challenging across all models. This is critical when comparing and evaluating models: the choice of evaluation dataset is highly predictive of model performance.

[3]Notation varies among the datasets about which class is machine-generated; we ensure that all datasets and models are normalized to have 1 as the machine-generated class.

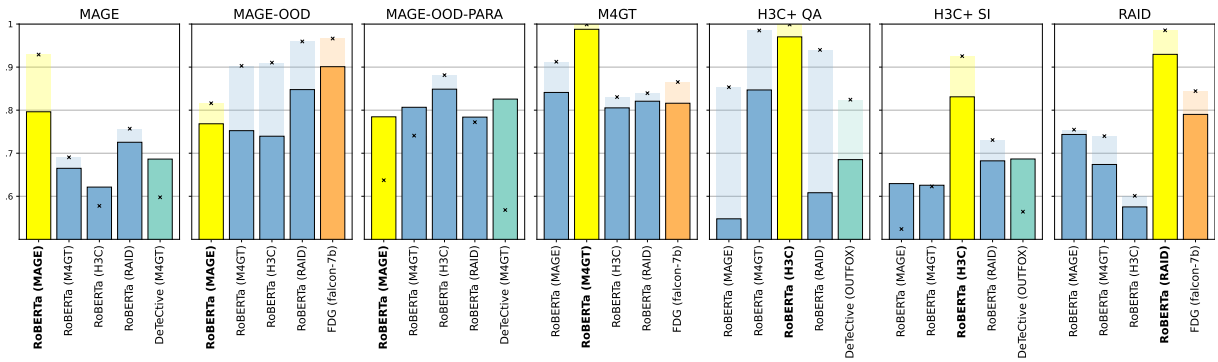

Figure 2: F1 and AUROC scores for cross-trained models. Bold models are those trained on the in-domain dataset. Each plot reflects a dataset, with the best-performing public model on that dataset included for reference.

Model	Low	Dataset	High	Dataset	Mean
Binoculars	0.60	MAGE-OOD-PARA	0.99	H3C+ QA	0.79
FDG (gpt-neo)	0.48	MAGE-OOD-PARA	1.00	H3C+ QA	0.77
FDG (gpt-j)	0.50	H3C+ SI	1.00	H3C+ QA	0.77
FDG (falcon-7b)	0.59	H3C+ SI	1.00	MAGE-OOD	0.80
Zippy (LZMA)	0.51	H3C+ SI	0.93	H3C+ QA	0.71
Zippy (Ensemble)	0.52	H3C+ SI	0.92	H3C+ QA	0.68
BiScope (Arxiv)	0.51	H3C+ SI	1.00	MAGE-OOD	0.77
BiScope (Yelp)	0.55	MAGE	0.99	H3C+ QA	0.75
BiScope (Essay)	0.57	MAGE-OOD-PARA	1.00	RAID	0.76
BiScope (Creative)	0.59	H3C+ SI	1.00	H3C+ QA	0.77
DeTeCtive (MAGE)	0.50	H3C+ SI	0.95	MAGE	0.79
DeTeCtive (M4GT)	0.51	H3C+ SI	0.96	M4GT	0.66
DeTeCtive (TuringBench)	0.45	H3C+ QA	0.56	MAGE-OOD-PARA	0.50
DeTeCtive (OUTFOX)	0.49	RAID	0.82	MAGE-OOD-PARA	0.59
RADAR	0.41	M4GT	0.87	MAGE-OOD	0.68

Table 2: Lowest and highest reported AUROC scores for the best performing variant of each model across the seven unique datasets.

4.1. Cross-Training

In order to evaluate the robustness of these trained models to new datasets, we conduct cross-training analysis, where models are trained on one dataset and evaluated on others. We train four models using 10k samples from each of the above datasets (MAGE, M4GT, H3C+, and RAID), and evaluate on the previous defined test sets. MAGE-trained models treat the three MAGE-based sets as in-domain; H3C+-trained models treat the two H3C+-based sets as in-domain; similarly for RAID and M4GT; the evaluation datasets which don't match the training data are considered out-of-domain. Results are shown in Figure 2.

As expected, in-domain performance is substantially better than out-of-domain for most cases, with a mean AUROC score over all datasets 0.06 above the best out-of-domain model and better on five of the seven datasets. Models capture dataset-specific generation patterns but struggle to generalize to adversarial or out-of-domain examples. This is especially evident in the RAID dataset, which contains more specific adversarial attacks.

However, we find in some cases out-of-domain performance can exceed in-domain: for the MAGE-OOD and MAGE-OOD-PARA sets, models trained on other datasets perform much better than even those trained on MAGE data, indicating that potential challenges can be mitigated by incorporating other external datasets into training. Out-of-domain models even outperform the best public models on both MAGE and M4GT.

4.2. Novel Human-Written Texts

False positives in machine-generated text detection carry substantial risk for harm, include educational impacts and unfair censorship (Wu et al., 2025). To further explore model performance, we evaluate models on datasets comprising exclusively human-authored texts:

Stories in the Wild: (August et al., 2020): This dataset contains 1,630 samples of human-written creative fiction.

PERSUADE: (Crossley et al., 2024): This corpus comprises 25k persuasive essays by secondary students (ages 11-17); we evaluate on a 5k random subset.

Huang: (Huang et al., 2020): Derived from this multi-lingual dataset built for hate speech analysis on X (formerly Twitter), we aggregate 5k user profiles from the English partition of the dataset by combining all of the users' posts into a single text. This dataset is particularly challenging, as many of the texts are quite short (mean length of 26 words) and contain informal language and punctuation.

We selected these datasets to reflect high-stakes scenarios. August et al. (2020) and Huang et al. (2020) focus on creative works in different domains, both of which could potentially be subject to censorship and/or unfair rejection if detected as machine-generated. Crossley et al. (2024) reflects student work, where the use of generative tools for authorship is often disallowed, and students are susceptible to negative consequences and punitive measures if their work is incorrectly identified as being generated (or assisted) by automated tools. Additionally, these datasets are useful in their disparate domains and styles: each contains very different styles, formatting, length, and other attributes which are potentially difficult for machine-generated text detction systems.

We report accuracy (equal to the true negative rate) on each of these three datasets in Table 3, as well as the mean. These are contrasted with recall scores on the data from Section 4: ideal models should maximize both metrics, with high accuracy of human-written texts (minimal false positives) while maintaining high recall on machine-generated data (minimal false negatives).[4]

On these datasets we again observe high performance variability across models. Models that perform well on this task tend to have worse recall on the original data: they simply make fewer positive predictions. No system simultaneously exceeds 0.8 accuracy and 0.8 recall.

Binoculars, Zippy, BiScope and FDG variants maintain strong accuracy, but suffer from poor recall. DeTeCtive variants and RADAR have higher recall but low accuracy. All other models fall below 85% accuracy on at least one dataset. Some models perform well only on certain datasets: FDG (falcon-7b) is highly accurate on the Huang dataset but worse on the others, while Zippy is strong on both Stories and PERSUADE datasets but classifies more than half of the Huang dataset as machine-generated. Trained models prioritize recall the expense of accuracy, and thus perform poorly, although we again see impacts of training data: models trained on the H3C+ corpus tend to have much better balance.

System	Stories	PERSUADE	Huang	Mean Acc.	Recall (ALL)
Binoculars	0.90	1.00	0.97	0.96	0.50
BiScope-Arxiv	0.86	0.99	0.82	0.89	0.53
BiScope-Yelp	0.85	1.00	0.98	0.94	0.61
BiScope-Creative	1.00	1.00	1.00	1.00	0.41
BiScope-Essay	1.00	1.00	1.00	1.00	0.26
DeTeCtive (MAGE)	0.58	0.88	0.32	0.59	0.83
DeTeCtive (M4GT)	0.45	0.93	0.63	0.67	0.85
DeTeCtive (OUTFOX)	0.70	0.92	0.33	0.65	0.93
DeTeCtive (TuringBench)	0.06	0.03	0.00	0.03	0.90
FDG (gpt-neo)	0.85	0.85	0.83	0.84	0.61
FDG (gpt-j)	0.86	0.79	0.79	0.82	0.63
FDG (falcon-7b)	0.75	0.83	0.96	0.85	0.63
RADAR	0.18	0.68	0.00	0.29	0.66
Zippy (LZMA)	0.95	0.34	0.96	0.75	0.43
Zippy (Ensemble)	0.91	0.20	0.96	0.69	0.46
RoBERTa (H3C+)	0.53	0.69	1.00	0.74	0.83
RoBERTa (M4GT)	0.74	0.93	0.50	0.72	0.82
RoBERTa (MAGE)	0.37	0.54	0.03	0.31	0.91
RoBERTa (RAID)	0.32	0.04	0.15	0.17	0.91
STYLO (H3C+)	0.84	0.86	1.00	0.90	0.52
STYLO (M4GT)	0.12	0.48	0.19	0.26	0.88
STYLO (MAGE)	0.48	0.35	0.56	0.46	0.77
STYLO (RAID)	0.35	0.70	0.02	0.36	0.80
MCGOVERN (H3C+)	0.21	0.15	1.00	0.45	0.85
MCGOVERN (M4GT)	0.11	0.23	0.00	0.11	0.96
MCGOVERN (MAGE)	0.04	0.01	0.00	0.02	0.98
MCGOVERN (RAID)	0.35	0.38	0.00	0.25	0.86

Table 3: Accuracy on novel human datasets, along the recall scores from the original evaluation (Section 4)

Our experiments reveal a significant vulnerability: models struggle with unseen creative human-written texts. Standard evaluations using imbalanced datasets (heavily skewed toward machine-generated content) likely inflate perceived performance, masking this weakness. Most systems either pose substantial risk when applied to out-of-domain human writing or struggle with recall, requiring careful deployment consideration.

Models have highly variable performance based on the evaluation data used, which could highlight or obscure aspects of performance. We now turn to metrics, which can have similar effects.

5. Metrics

Our dataset analysis reveals divergences between F1 and AUROC metrics: while Binoculars, BiScope, and Fast-DetectGPT achieve strong AUROC scores despite comparatively low F1 scores, De-

[4]We use RoBERTa as an exemplar for fine-tuned models; the others had similar performance.

Model	Threshold 0.5					Threshold by EER					Threshold invariant			
	Precision	Recall	F1	Accuracy	AvgRec	Precision	Recall	F1	Accuracy	AvgRec	AUROC	TPR@FPR 1%	TPR@FPR .01%	Variance
Binoculars	2.57	11.00	7.86	5.71	4.86	11.71	1.00	7.14	10.57	11.71	4.00	5.29	4.57	11.77
FDG (gpt-neo)	6.43	8.71	6.14	5.00	5.29	5.57	11.29	7.00	4.71	4.71	5.43	3.00	5.57	3.92
FDG (gpt-j)	7.86	8.29	7.14	6.29	6.57	6.71	10.71	7.71	6.00	6.00	5.29	3.29	6.14	2.79
FDG (falcon-7b)	5.14	7.86	6.29	4.71	3.86	3.86	10.43	7.00	4.57	3.57	2.14	2.71	3.57	4.89
Zippy (LZMA)	6.71	13.00	11.57	8.86	8.43	12.71	2.00	8.14	11.57	12.71	7.71	8.29	9.43	8.70
Zippy (Ensemble)	9.14	11.71	12.00	10.00	9.86	13.71	3.00	9.14	12.57	13.71	9.00	9.43	10.29	7.08
BiScope (Arxiv)	8.71	6.71	7.00	7.71	7.29	7.00	9.43	7.71	7.29	6.29	6.86	7.14	6.14	0.77
BiScope (Yelp)	10.43	4.43	6.43	6.86	7.29	7.86	7.43	7.14	5.86	6.14	8.43	9.43	7.57	2.20
BiScope (Essay)	4.00	14.71	14.57	12.14	11.57	3.57	15.00	14.43	10.29	9.29	7.43	6.71	7.14	15.25
BiScope (Creative)	5.86	9.00	7.00	6.29	5.71	4.43	11.43	7.57	5.43	4.43	4.71	6.43	5.71	3.58
DeTeCtive (MAGE)	7.14	4.29	3.86	4.71	4.29	5.43	7.14	4.43	4.71	3.57	10.00	10.43	7.29	4.79
DeTeCtive (M4GT)	10.43	4.00	5.86	8.29	9.43	7.71	7.00	6.00	6.43	7.29	12.00	11.71	10.57	5.58
DeTeCtive (TuringBench)	13.86	3.71	8.43	12.86	14.29	12.14	6.71	9.43	12.00	12.57	14.00	13.57	14.43	10.24
DeTeCtive (OUTFOX)	13.14	3.29	7.57	10.86	11.86	10.57	6.14	7.86	9.14	9.86	12.71	12.29	12.43	7.98
RADAR	8.57	9.29	8.29	9.71	9.43	7.00	11.29	9.29	8.86	8.14	10.29	10.29	9.14	1.10

Table 4: Mean rank (1-15) of each public model over the seven datasets for each metric.

TeCtive models show an inverse relationship. We examine this relationship between metrics further with regard to class imbalance and model ranks.

5.1. Class Imbalance

To assess metric sensitivity to class imbalance, we adjust the MAGE dataset to contain arbitrary percentages of machine-generated data. We find that F1 scores increase logarithmically as the percentage of machine-generated content increases. F1 scores can thus be artificially inflated by predicting a high number of positive samples for a dataset that is skewed towards the positive class. Threshold independent metrics (AUROC, TPR) tend to remain stable regardless of the distribution of classes (see Appendix C).

This partially explains the differences in model performance: models vary in relative strength when evaluated by different metrics. For a further comparative analysis, we study the impact of metrics on model ranks.

5.2. Ranking based on metrics

Model ranks likewise depend on the chosen metrics. We calculate the rank of each system (1 being the best, 15 the worst) across all seven test sets using a series of metrics. We evaluate precision, recall, F1 score, accuracy, and average recall. These metrics are binary and are calculate using a classification threshold.

We experiment with two thresholds: first, we use 0.5 which is considered the default. For some models, however, this threshold disproportionately favors precision or recall, as the models give very high or very low scores to most samples. As a comparison, we identified a model-specific threshold designed to optimize performance ("Threshold by EER").

This threshold is calculated by first separating a sample of 1000 instances from each of the test corpora. We run each model on these samples, then identify the threshold that optimizes equal error rate (EER). This threshold is then used to classify the remainder of the test data; we report results on this subsample of the test data.

We also report results on threshold-independent metrics. We use model scores to calculate AUROC, TPR@FPR 1%, and TPR@FPR 0.01% (See Appendix 5 for definitions of each metric). Table 4 shows each model's mean rank (across models) and variance (across metrics).

The choice of metric can critically influences model ranks among competitors. At the 0.5 threshold, Binoculars and some BiScope models achieve high precision but low recall, DeTeCtive models exhibit high recall but underperform in AUROC and TPR@FPR scores, and FDT models have comparatively low F1 scores compared to other metrics. RADAR and BiScope (Arxiv) are stable but comparatively weak across metrics. Threshold adjustment dramatically alters model ranks: at an optimized threshold, Binoculars swaps from being high-precision to high-recall while FDG and Zippy models score much worse in recall. This highlights the necessity of diverse metrics: our understanding

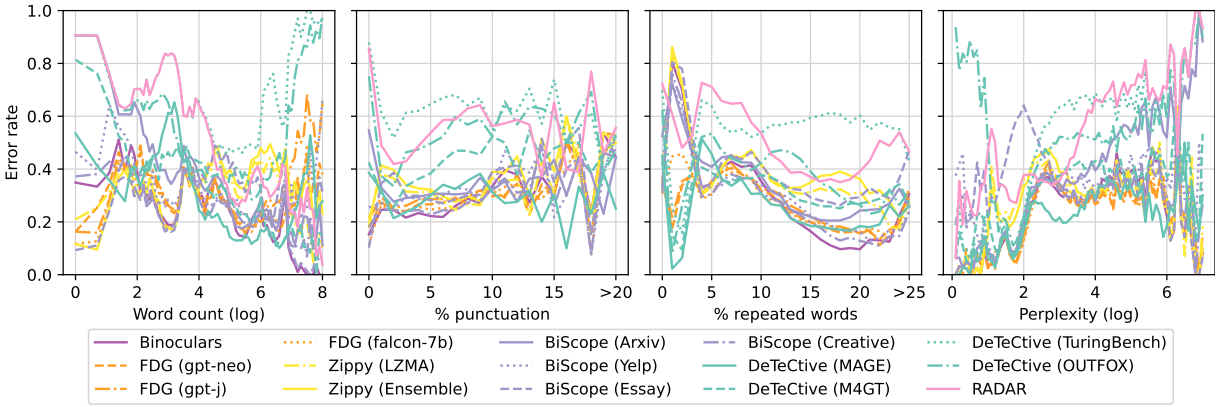

Figure 3: Model error as a function of word count (log), punctuation (%), repeated words (%), and perplexity (log).

of system performance can vary drastically based on these small changes in metric calculations.

These findings suggest two critical factors should be incorporated into metric choice: first, justified evaluation metric choices are essential. Most systems emphasize peak performance without explicitly justifying their metric choices,[5] but each metric captures distinct aspects that must be interpreted in context rather than in isolation. For instance, in student essay evaluation, where machine-generated content may be rare but false positives carry severe consequences, metrics emphasizing low false positive rates (e.g., TPR@FPR) are essential. In spam detection where synthetic content is prevalent and false positives are less risky, metrics like F1 may suffice. We find AUROC to be valuable in most scenarios, as it allows for better understanding of performance across multiple thresholds, which gives the end user better control.

Second, a diverse set of metrics need to be used to balance potential performance differences stemming from the above findings. A more principled approach to metric selection, tailored to deployment contexts, is essential to ensure appropriate deployment of machine-generated text detection.

6. Analysis

To investigate performance differences, we explore four key textual attributes commonly used for machine-generated text detection: length (in words), punctuation, repetition, and perplexity under the `facebook/opt-1.3b` (Zhang et al., 2022), defined in Appendix D. We plot each models' error rates against each of these attributes (Figure 3) with the goal of identifying trends in these attributes that may contribute to disparity in model performance.

Model behavior patterns in different ways for each attribute. For word count, models exhibit highly variable performance on low-word count texts, but typically improve as the text length increases. However, certain of the DeTeCtive variants (TuringBench, OUTFOX) again struggle with longer texts, particularly as texts become longer than 400 words. Model performance is not strongly tied to punctuation, except in cases where it is very rare ($< 1\%$): in this case, performance varies greatly, though this reflects only 0.7% of samples.

Perhaps most informative is repetition: model families exhibit substantially different error patterns based on repetition. DeTeCtive and FDG models show minimal error with low repetition, with more frequent errors on more repetitive texts, while most others exhibit the opposite behavior. Perplexity is also quite informative: most models achieve strong performance on low-perplexity texts, excepting DeTeCtive (OUTFOX). Some models then degrade substantially when perplexity increases (RADAR, BiScope (Arxiv), DeTeCtive (TuringBench/OUTFOX)), while the others maintain stable performance.

These observations demonstrate that performance variations can be accounted for by some of these attributes: systems must accommodate a wide variety of possible lengths, repetitions, and perplexities. Models efficacy varies substantially across these attributes, and resulting implementations should account for these textual characteristics and how they may impact performance.

7. Conclusions

In this work we demonstrate that the choice of datasets and metrics critically influences our assessment of machine-generated text detection sys-

[5]With the notable exceptions being Bao et al. (2024), who explicitly advocate for AUROC's threshold-agnostic benefits, and Hans et al. (2024), who prioritize TPR@FPR to minimize false positives in high-stakes scenarios.

tem capabilities. Model performance varies substantially across different evaluation frameworks, with each system under specific metric and data conditions. This underscores the necessity of context-aware deployment: no single solution performs optimally across all use cases.

Motivated by our analysis, we recommend system description include: (a) a discussion of the impact of dataset composition and class imbalance on reported performance, (b) rigorous definitions and motivations for all metrics used, and (c) a discussion of potential harms from metric-task misalignment based on intended use-cases (e.g., false positives in academic settings).

We also suggest practitioners use datasets that are diverse as possible in terms of length, punctuation, repetition, and perplexity, as well as more general attributes like domain, style, and authorship. As different models excel at different types of data, more diverse evaluation data is required for accurate comparison. With regard to metrics, practitioners should focus on metrics that cover intended uses, again using different metrics that accurately cover system strengths and weaknesses. These metrics should be carefully considered and motivated particularly with regard to class imbalance in the data, and practitioners should discuss the impact of imbalance on their results along with reported metrics.

For both datasets and metrics, a standardized set of benchmarks would be a useful tool for those developing models. For datasets, we've seen attempts in this direction (eg. (Dugan et al., 2024; Wang et al., 2024), etc), but these benchmarks are static, and thus won't reflect the rapid pace of advancement in generative models. For metrics, we find that most work eschews any discussion or motivation of metrics and this needs to be addressed. Depending on the intended application of a system and the datasets involved, the relevant metrics may be different, and thus comparison becomes difficult. Overall, it is key that both datasets and metrics are diverse and clearly motivated.

While current systems demonstrate tremendous potential, their practical utility hinges on addressing these complexities. Machine-generated text detection remains challenging, with the continuous emergence of new generative models and adversarial attacks demanding corresponding advances in detection methods. The effective and fair deployment of these methods on diverse, real-world scenarios depends on rigorous assessment, and this requires crucial awareness of the impacts of datasets and metrics.

8. Ethical Considerations

The primary ethical consideration surrounding this work is the ethical application of machine-generated text detection models. We aim to avoid making claims about the general usage of these models, and whether it is appropriate, but understand there are ethical implications for employing automated systems for decision making that may negatively impact stakeholders. Our work focuses on potential pitfalls in the evaluation of machine-generated text detection systems. These systems are currently in use in areas such as automated grading where they have the potential for substantial harms, particularly for false positives, in which a student may be punished unfairly. To this end our work is vital: better understanding of the risks of these models is necessary in order to make informed decisions about the deployment of such systems.

9. Limitations

While our evaluation spans a diverse range of models, it is not exhaustive—many systems and benchmarks fall outside our analysis. We demonstrate that our core findings (evaluation variance on different datasets, importance of metrics, and relatively weak performance on human-written texts) hold across a representative sample of prevalent models available for practitioners, but these conclusions are necessarily bounded by the scope of our study. This limitation is particularly salient given the rapid evolution of both generative LLMs and detection methods: as new models and datasets emerge, the detection landscape will continue to shift.

We are similarly limited in language (we focus only on English) and datasets (we select a sample of available datasets). While we are optimistic that our results are likely to hold for other datasets and languages, further verification would be necessary.

Rather than providing a definitive assessment of all available systems, languages, and datasets, our work highlights the critical need for rigorous evaluation methodologies, a principle we validate through our selected models and datasets, and one we argue extends to future research in this evolving field.

10. References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste

Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Tal August, Maarten Sap, Elizabeth Clark, Katharina Reinecke, and Noah A. Smith. 2020. Exploring the effect of author and reader identity in online story writing: the STORIESINTHEWILD corpus. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 46–54, Online. Association for Computational Linguistics.

Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. 2025. Glimpse: Enabling white-box methods to use proprietary models for zero-shot llm-generated text detection.

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.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection.

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

Scott A. Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Meg Benner, and Ulrich Boser. 2024. A large-scale corpus for assessing written argumentation: Persuade 2.0. *Assessing Writing*, 61:100865.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

German Gritsai, Anastasia Voznyuk, Andrey Grabovoy, and Yury Chekhovich. 2025. Are ai detectors good enough? a survey on quality of datasets with machine-generated texts.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. 2024a. Biscope: Ai-generated text detection by checking memorization of preceding tokens. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 104065–104090.

Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024b. Detective: Detecting ai-generated text via multi-level contrastive learning.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 15077–15095. Curran Associates, Inc.

Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J. Paul. 2020. Multilingual Twitter corpus and baselines for evaluating demographic

bias in hate speech recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1440–1448, Marseille, France. European Language Resources Association.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21258–21266.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3637–3647, New York, NY, USA. Association for Computing Machinery.

Jooyoung Lee, Xiaochen Zhu, Georgi Karadzhov, Tom Stafford, Andreas Vlachos, and Dongwon Lee. 2025. Collaborative evaluation of deepfake text with deliberation-enhancing dialogue systems.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. 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)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2024. On the detectability of chatgpt content: Benchmarking, methodology, and evaluation through the lens of academic writing. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 2236–2250, New York, NY, USA. Association for Computing Machinery.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting.

Hope Elizabeth McGovern, Rickard Stureborg, Yoshi Suhara, and Dimitris Alikaniotis. 2025. Your large language models are leaving fingerprints. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 85–95, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Mozilla. 2025. Apollodft.

Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2024. Contrasting linguistic patterns in human and LLM-generated news text. *Artificial Intelligence Review*, 57(10):265.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1613–1630.

Shushanta Pudasaini, Luis Miralles, David Lillis, and Marisa Llorens Salvador. 2025. Benchmarking AI text detection: Assessing detectors against new datasets, evasion tactics, and enhanced LLMs. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 68–77, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Cosmina-Mihaela Rosca, Adrian Stancu, and Emilian Marian Iovanovici. 2025. The new paradigm of deepfake detection at the text level. *Applied Sciences*, 15(5).

Sina Mahdipour Saravani, Indrajit Ray, and Indrakshi Ray. 2021. Automated identification of social media bots using deepfake text detection. In *Information Systems Security*, pages 111–123, Cham. Springer International Publishing.

Aditya Shah, Prateek Ranka, Urmi Dedhia, Shruti Prasad, Siddhi Muni, and Kiran Bhowmick. 2023. Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features. *International Journal of Advanced Computer Science and Applications*, 14(10).

Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2024. Hc3 plus: A semantic-invariant human chatgpt comparison corpus.

Thinkst. 2023. Meet zippy: A fast ai/llm text detector. Accessed 2025.

Adaku Uchendu, Jooyoung Lee, Hua Shen, Thai Le, Ting-Hao 'Kenneth' Huang, and Dongwon Lee. 2023. Does human collaboration enhance the accuracy of identifying llm-generated deepfake texts?

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2024. GPT-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 103–115, Mexico City, Mexico. Association for Computational Linguistics.

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.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2025. Detectrl: Benchmarking llm-generated text detection in real-world scenarios.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. LLMDet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133, Singapore. Association for Computational Linguistics.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona 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.

Yi-Fan Zhang, Zhang Zhang, Liang Wang, Tieniu Tan, and Rong Jin. 2023. Assaying on the robustness of zero-shot machine-generated text detectors.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A. Metrics

We here define the metrics used for evaluation. We use the `numpy` and `scikit-learn` packages to calculate these metric.

- **Precision**: the ratio of true positive predictions (TP) to the total number of positive predictions (true and false): $\frac{TP}{TP+FP}$
- **Recall**: the ratio of true positive predictions to the total number of positive samples (true positives and false negatives)in the dataset: $\frac{TP}{TP+FN}$
- **F1 Score**: the harmonic mean mean of precision and recall: $\frac{2*prec*rec}{prec+recall}$
- **Accuracy**: the ratio of correctly classified samples to the full dataset: $\frac{TP+TN}{TP+TN+FP+FN}$
- **Average Recall**: the mean of recall for the positive class and the negative class.

- **AUROC**: area under the receiver operator characteristic (ROC) curve. This curve is defined as

  $(FPR(x), TPR(x)), x \in [-\infty, +\infty]$

  For each value of x, we plot the false positive and true positive rate; for most models x is a probability such that $x \in [0, 1]$. An AUROC of 1 means the model is perfect; .5 means the model performs at chance.

- **TPR @ FPR 1% / .01%**: True positive rate at a specific false positive rate. This is defined as the best possible percentage of true positives such that the false positive rate is below the given threshold.

Precision, recall, F1 score, accuracy, and average recall are binary classification based metrics: we use the model labels where provided, and classify based on thresholds where not provided. We use a default threshold of 0.5. In Section 5, we also experiment using the threshold that optimizes equal error rate (EER), calculated by finding the point where false positive rate (FPR) and false negative rate (FNR) are closest.

B. System Descriptions

B.1. Binoculars (Hans et al., 2024)

The Binoculars system uses two pre-trained LLMs. The system calculates the perplexity of the text in question using an *observer* and a *performer*, and computes a metric called cross-perplexity, which is a strong signal for machine-generated text. Binoculars requires two LLMs but no training, is free and open-source, and they report excellent performance on a variety of domains, as well as on shortened texts.

We use the implementation provided at `https://github.com/ahans30/Binoculars`.

B.2. BiScope (Guo et al., 2024a)

The BiScope model is based on the assumption that losses between a token and the preceding as well as following tokens can be indicative of AI generated text. They propose a bi-directional method, and train a classifier based on these statistics. BiScope is trained on four different corpora: two "short" text types (Yelp and Arxiv), and two "longer" text types (Creative and Essay): we evaluate these four variants. We do not evaluate the Code subset, as this does not align with our testing scenario.

We use the implementation provided at `https://github.com/MarkGHX/BiScope`: they do not provide an explicit "best" model for each domain, so we train each of our four variants using all the provided data from the respective domains.

B.3. Fast-DetectGPT (Bao et al., 2024)

Fast-DetectGPT (FDG) is an upgrade over the previously established DetectGPT (Mitchell et al., 2023). They calculate a *conditional probability curvature* of a given text, following the assumption that perturbing a machine generated text will lead to a lower conditional probability. Like Binoculars, FDG requires two LLMs: a sampling model and a scoring model, and it shows strong performance on a variety of domains. We evaluate three settings using different models for scoring: `gpt-neo-2.7b` for speed, `gpt-j-6b` as a slower but more accurate model[6], and `falcon-7b` for maximal accuracy.

We use the implementation provided at `https://github.com/baoguangsheng/fast-detect-gpt`.

B.4. RADAR (Hu et al., 2023)

The RADAR system uses adversarial training of a detector and paraphraser. The paraphraser attempts to fool the detector, and joint learning allows their model to adapt to new paraphrasing techniques.

We use the implementation provided at https://github.com/IBM/RADAR.

B.5. DeTeCtivE (Guo et al., 2024b)

The DeTeCtive model relies on a contrastive learning framework, calculating distances between samples which allows for encoding distinctive features based on the author. These encodings are then used at inference time, and the text in question is compared to relevant vectors using K-Nearest Neighbor (KNN) classification. DeTeCtive has four relevant training sets: MAGE (Deepfake), M4, TuringBench and OUTFOX: we evaluate each of these variants.

We use the implementation provided at `https://github.com/heyongxin233/DeTeCtive`.

B.6. Zippy (Thinkst, 2023)

Unlike the above, which rely on heavily on pretrained LLMs to calculate metrics, Zippy is compression based, relying on a variety of compression ratios to indirectly measure perplexity of a given text. Zippy starts by 'seeding' a compression with machine-generated text, and then assesses the difference in compression from just this seed to the addition of the text in question. Zippy reports strong performance compared to LLM-based models, and is independent from any models, making it an enticing option.

[6] Following their work, we use `gpt-j-6b` as the reference model and `gpt-neo-2.7b` as the scoring model. For more, see Bao et al. (2024).

We use the implementation provided at https://github.com/thinkst/zippy.

B.7. Transformer models

We use the huggingface platform for model training and evaluation (Wolf et al., 2020). We start with a brief hyperparameter tuning to optimize learning rate and batch size; we find that a learning rate of $2e-5$ and a batch size of eight is effective accross model types. We train each model for 10 epochs, with weight decay of 0.01, then keep the best performing model on the training data by F1 score for prediction. We note that this may bias models towards performance on this particular metric: in practice we find fine-tuning these models using F1 score as the target yields consistent performance across all metrics.

B.8. McGovern Model

We follow the method described by McGovern et al. (2025). We use the scikit-learn package Pedregosa et al. (2011) to do feature extraction, generating n-grams ($n \in 2,3,4$) for words, part-of-speech tags, and characters. We use nltk's word and part-of-speech taggers (Bird et al., 2009). For model training, we use the GradientBoostingClassifier provided by scikit-learn. We perform a grid search over estimators, learning rate, and depth, selecting 100 estimators, a learning rate of 0.01, and a max depth of 3.

B.9. Stylo Model

Features for the STYLO model are given below, along with additional notes for clarity. The text was processed using SpaCy (Honnibal et al., 2020); unless otherwise specified, all syntactic and semantic information was extracted from the SpaCy parse. We calculate each feature as a raw count and also an average based on sentence length. We use univariate feature selection to select the 100 best features for model training. We then utilize ensemble classification via scikit-learn: we incorporate the GaussianNB, AdaBoostClassifier, LGBMClassifier, CatBoostClassifier, and RandomForestClassifier models into scikit-learn's VotingClassifier. We use default model parameters, and run a paraeter search of weightings, which yields even weighting except for the LGBMClassifier which receives double weight.

B.10. Architecture

For model training, inference, and evaluation we use Amazon AWS EC2 instances. For CPU evaluation, we use an r5.2xlarge instances, with 64 GB of memory and 16 CPUs. For GPU testing we use the g6e.xlarge instance type. This instance type has an NVIDIA L40S Tensor Core GPU with 48 GB of GPU memory, allowing us to experiment with models that have larger GPU memory requirements (notably Binoculars and the FDG systems require significant GPU memory).

C. Scores on imbalanced datasets

Our test datasets are built to be balanced, but most systems described are evaluated on custom or previously used datasets that skew heavily towards LLM-generated classes. To investigate this bias, we systematically introduce artificial class imbalance into the MAGE dataset. We select a sample of 5k items with a specific percentage n being from the LLM-generated class. We evaluate these sample from $n = 1$ where 1% of the data is from the LLM-generated class to $n = 100$ where the entire dataset is from the LLM-generated class. Figure 4 compares F1, AUROC, and TPR@FPR 1% metrics using this artificial imbalancing.

Across all models, F1 scores exhibit logarithmic growth as the proportion of positive (LLM-generated) samples increases, while AUROC and TPR@FPR metrics remain stable regardless of class imbalance. This is due to the nature of the metrics: F1 is a threshold based metric and is inherently dependent on the number of true positives, false positives, and false negatives. With very few true samples, the score is dominated by false positives, while with very few false samples, recall can be inflated by classifying every sample as 1. AUROC and TPR metrics apply at all thresholds and are not affected by class imbalance: the key factors are ratios, and thus scaling the number of samples doesn't affect their values.

This helps explain the strong performance reported in system descriptions evaluated with F1 score, as the datasets evaluated tend to skew heavily towards LLM-generated texts (the original MAGE training set is 70.8% LLM-generated; the RAID training set is 97.1% LLM-generated). These data imbalances are motivated by the desire to capture a wide variety of LLM-based generations: datasets are created with many LLM outputs generated from a single human text.

TPR@FPR shows significantly higher variance than AUROC when the positive class dominates, suggesting reduced reliability for skewed datasets. This instability cautions against over-reliance on TPR@FPR in LLM-heavy evaluation scenarios.

D. Attributes Used for Analysis

We evaluate four attributes as potential contributors to model performance. These are defined as

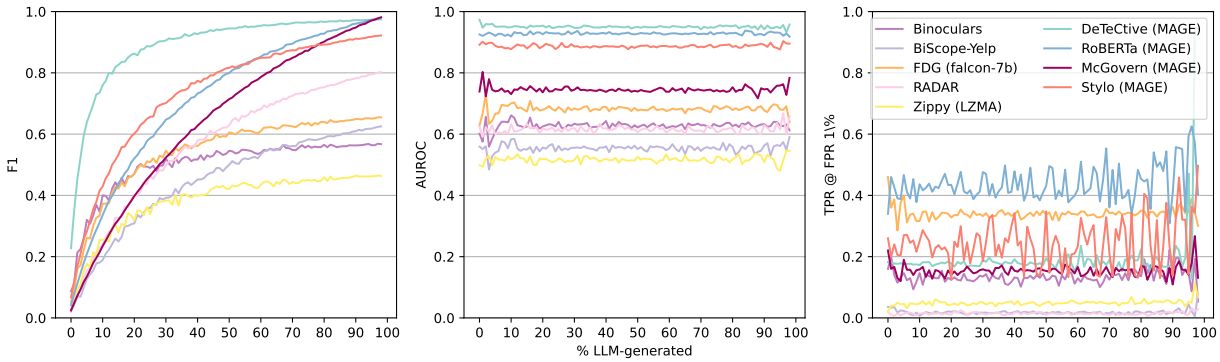

Figure 4: F1, AUROC, and TPR@FPR 1% performance for imbalanced datasets. The x-axis reflects the percentage of samples from the positive (LLM-generated) class.

follows:

- **Length**: we simply use the number of words of the text, defined by splitting the text on whitespace.
- **Punctuation**: the percentage of the text that is punctuation, defined using Python's `string.punctuation`.
- **Repetition**: the percentage of word types in the text that occur more than once.
- **Perplexity**: defined as the exponentiation of the average log-likelihood under a language model for each text ($PPL(x) = e^{-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i|x_{<i})}$). We use the `facebook/opt-1.3b` model to calculate log likelihoods (Zhang et al., 2022).