GenAI Content Detection Task 3: Cross-Domain Machine-Generated Text Detection Challenge

Liam Dugan[1], **Andrew Zhu**[1], **Firoj Alam**[3], **Preslav Nakov**[2]
Marianna Apidianaki[1], **Chris Callison-Burch**[1]
University of Pennsylvania[1] MBZUAI[2] Qatar Computing Research Institute[3]
{ldugan, andrz, marapi, ccb}@seas.upenn.edu
preslav.nakov@mbzuai.ac.ae, fialam@hbku.edu.qa

Abstract

Recently there have been many shared tasks targeting the detection of generated text from Large Language Models (LLMs). However, these shared tasks tend to focus either on cases where text is limited to one particular domain or cases where text can be from many domains, some of which may not be seen during test time. In this shared task, using the newly released RAID benchmark, we aim to answer whether or not models can detect generated text from a large, yet fixed, number of domains and LLMs, all of which are seen during training. Over the course of three months, our task was attempted by 9 teams with 23 detector submissions. We find that multiple participants were able to obtain accuracies of over 99% on machine-generated text from RAID while maintaining a 5% False Positive Rate—suggesting that detectors are able to robustly detect text from many domains and models simultaneously. We discuss potential interpretations of this result and provide directions for future research.

1 Introduction

The detection of AI generated text is an increasingly relevant task in the modern age. Such detection can help combat misinformation (Sharevski et al., 2023), phishing attacks (Bethany et al., 2024), and other fraudulent activities (Weiss, 2019; Lund et al., 2023). This is particularly important given that humans struggle to detect generated text reliably (Dugan et al., 2020, 2023) and that generated text is often more persuasive than human-written text (Spitale et al., 2023).

Recently there has been an increase in large scale shared tasks for AI generated text detectors (Wang et al., 2024; Fivez et al., 2024; Bevendorff et al., 2024). These shared tasks tend to either focus on one particular domain or hold out many domains and LLM generators in the final test set. This prevents us from understanding how well a single detector can detect text from a large fixed set of models and domains. Such a setting is important to understand as it helps us to delineate the boundaries of our detector capabilities. For example, it is clear that detectors trained on a single LLM can accurately detect text from that model (Solaiman et al., 2019) but does this extend to arbitrarily many LLMs? What about arbitrarily many domains?

In particular we attempt to answer the following research questions:

- *(RQ1) Can a single detection model be trained to detect generated text from many different known domains and LLMs accurately?*
- *(RQ2) Can a single detection model be robust to many different known adversarial attacks?*

In order to test these research questions, we conduct this shared task using the newly released RAID benchmark (Dugan et al., 2024). RAID is a dataset of over 10 million documents from 11 generative models, 8 textual domains, 4 decoding strategies, and 11 adversarial attacks. We chose RAID as it is one of the largest benchmarks currently available, and it features variation across decoding strategies and adversarial attacks as well as a large variety of textual domains. Crucially, the test set of RAID does not include any held-out models or domains and has yet to be released publicly. Therefore RAID allows us to answer our research questions most effectively.

We received 23 submissions to the shared task from 9 different teams, and 7 system description papers. The results of the evaluation are publicly available.[1] The two best performing teams (Pangram and Leidos) achieved extremely strong performance without adversarial attacks (99.3%) and with adversarial attacks (97.7%). In this paper we will summarize our general takeaways from this re-

[1] https://raid-bench.xyz/shared-task

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 377–388
January 19, 2025.

Sources in the RAID dataset	
Generative Models ($n = 11$)	GPT2, GPT3, GPT4, Cohere, Cohere Chat, MPT 30B, MPT 30B Chat, Mistral 7B, Mistral 7B Chat, ChatGPT, Llama 70B Chat
Domains ($n = 8$)	Abstracts, Recipes, Books, Reddit, News, Reviews, Poetry, Wiki
Decoding Strategies ($n = 4$)	Greedy (temp=0), Random Sampling (temp=1, top-p=1), Greedy + Repetition Penalty, Sampling + Repetition Penalty
Adversarial Attacks ($n = 11$)	Alternative Spelling, Homoglyph, Article Deletion, Number Swap, Insert Paragraphs, Upper Lower Swap, Paraphrase, Synonym Swap, Zero Width Space, Misspelling, Whitespace Addition

Table 1: The domains, models, decoding strategies and attacks covered by RAID.

Model	Num. Gens	Toks	Self-BLEU	PPL -L7B	PPL -G2X
Human	14971	378.5	7.64	9.09	21.2
GPT 2	59884	384.7	23.9	8.33	8.10
GPT 3	29942	185.6	13.6	3.90	8.12
ChatGPT	29942	329.4	10.3	3.39	9.31
GPT 4	29942	350.8	9.42	5.01	13.4
Cohere	29942	301.9	11.0	5.67	23.7
(+ Chat)	29942	239.0	11.0	4.93	11.6
Mistral	59884	370.2	19.1	7.74	17.9
(+ Chat)	59884	287.7	9.16	4.31	10.3
MPT	59884	379.2	22.1	14.0	66.9
(+ Chat)	59884	219.2	5.39	7.06	56.3
LLaMA	59884	404.4	10.6	3.33	9.76
Total	509k	323.4	13.7	6.61	23.8

Table 2: Statistics for the generations in train and test without adversarial attacks. **PPL-L7B** refers to mean perplexity according to LLaMA 7B and **PPL-G2X** refers to mean perplexity according to GPT 2 XL.

sult and offer future directions for effective benchmarking of generated text classifiers.

2 Related Work

The shared tasks most similar to ours are the SemEval24 Task 8 and GenAI Content Detection Task 1 (Wang et al., 2024, 2025). Both of these tasks include outputs from many generative models and domains in both their train and test set. However, as mentioned in the introduction, these tasks hold out many domains and models in their test set to test the generalization performance of classifiers to new unseen models and domains. In our task we explicitly give all domain and model information to our participants up front and only hold out particular articles within such domains.

Other shared tasks in the past have also evaluated generated text detectors in specific high-risk domains such as academic essays (King et al., 2023; Chowdhury et al., 2025) scientific papers (Kashnitsky et al., 2022) or news articles (Bevendorff et al., 2024). Previous shared tasks have also studied detection in a multilingual context (Shamardina et al., 2022; Sarvazyan et al., 2023; Fivez et al., 2024; Wang et al., 2025; Chowdhury et al., 2025). While such tasks are interesting and valuable, they do not test what we are interested in, namely the ability of single detectors to detect text from many different models and domains.

Finally, the recent Voight-Kampff task (Bevendorff et al., 2024) is particularly noteworthy. In their task they employ a set of builders (who build detectors) and breakers (who create adversarial datasets to fool the detectors). They are the first shared task to explicitly include adversarial constraints into their evaluation—experimenting with homoglyph attacks and different detailed prompt formulations with bullet points and summaries of the original source texts. However, they conduct their task in a pairwise manner, giving each detector two texts, one of which must be human-written, and asking the detector to select the human-written text. While they showed that detectors can do well on this highly adversarial task (96.1 ROC AUC score for the top team), we target the more difficult yet more realistic version of the task, where a single document is given as input.

3 Task Setup

3.1 RAID Benchmark

The RAID benchmark was created by sampling roughly 2000 human-written documents from each of 8 domains. Then, for each human-written document, a machine-written version is generated from each of the 11 LLMs with each of the 4 decoding strategies. Finally, each of the 11 adversarial attacks are applied to all machine-written documents. The test set consists of 200 human-written documents per domain selected from the train set and all generations based on those documents. The documents were then checked to prevent duplication and leakage and to ensure no overlap between train and test data.

In Table 1, we list the domains, models, decoding strategies, and adversarial attacks covered in RAID. In Table 2 we report summary statistics and

		Human	ChatGPT	dav.-003	GPT-4	Cohere	Coh.-C	GPT-2	MPT	MPT-C	Mistral	Mist.-C	Llama2-C
Train	Abstracts	1766	3532	3532	3532	3532	3532	7064	7064	7064	7064	7064	7064
	Books	1781	3562	3562	3562	3562	3562	7124	7124	7124	7124	7124	7124
	News	1780	3560	3560	3560	3560	3560	7120	7120	7120	7120	7120	7120
	Poetry	1771	3542	3542	3542	3542	3542	7084	7084	7084	7084	7084	7084
	Recipes	1772	3544	3544	3544	3544	3544	7088	7088	7088	7088	7088	7088
	Reddit	1779	3558	3558	3558	3558	3558	7116	7116	7116	7116	7116	7116
	Wiki	1779	3558	3558	3558	3558	3558	7116	7116	7116	7116	7116	7116
	Reviews	943	1886	1886	1886	1886	1886	3772	3772	3772	3772	3772	3772
	Total	13371	26742	26742	26742	26742	26742	53484	53484	53484	53484	53484	53484
Test	Abstracts	200	400	400	400	400	400	800	800	800	800	800	800
	Books	200	400	400	400	400	400	800	800	800	800	800	800
	News	200	400	400	400	400	400	800	800	800	800	800	800
	Poetry	200	400	400	400	400	400	800	800	800	800	800	800
	Recipes	200	400	400	400	400	400	800	800	800	800	800	800
	Reddit	200	400	400	400	400	400	800	800	800	800	800	800
	Wiki	200	400	400	400	400	400	800	800	800	800	800	800
	Reviews	200	400	400	400	400	400	800	800	800	800	800	800
	Total	1600	6400	3200	3200	3200	3200	3200	6400	6400	6400	6400	6400

Table 3: Number of documents in the RAID dataset by model and domain. Each human-written document has exactly one machine-written counterpart from each model with each of the decoding strategies listed in Table 1. Due to lack of support for repetition penalty sampling, API-based models have two outputs per human document and open-weight models have four outputs per human document. "-C" in model name indicates the chat fine-tuned version of the model. The adversarial data has an identical distribution but with 12x more documents per cell.

in Table 3 we report the exact distribution of documents from the training set and test set.

3.2 Subtask A: Cross-Domain Detection

For Subtask A, participants were asked to submit detectors that would be robust to all 8 domains in the main RAID dataset without adversarial attacks. In Table 9, we provide a breakdown of the addressed domains, and links to the original data sources for extra training. We provided our teams with these links in order to help them secure as much training data as possible. This is particularly important given that RAID has a roughly 40:1 class imbalance of AI vs. human-written text. We observed that many teams took advantage of this and sampled extra human data from these sources.

3.3 Subtask B: Adversarial Robustness

In Subtask B, the participants had to evaluate their detectors on all data from Subtask A with the addition of 11 adversarial attacks. In Table 10 we list the adversarial attacks applied as well as the source papers that first study them. For this subtask, the participants also had access to the original code[2] used to create the adversarial attacks. This allowed them to adversarially modify any existing piece of data to assist in training.

[2] https://github.com/liamdugan/raid/tree/main/generation/adversarial/attackers

3.4 Baselines

We use the following models as baselines:

- **Openai-RoBERTa-large** (Solaiman et al., 2019): RoBERTa-large model fine-tuned on GPT-2 generations.
- **RADAR** (Hu et al., 2023): Vicuna 7B model trained on adversarial paraphrases
- **Binoculars** (Hans et al., 2024): Current SoTA metric-based detection model. Uses perplexity divided by cross-entropy between Falcon 7B and Falcon 7B Instruct.
- **GLTR** (Gehrmann et al., 2019): Baseline metric-based detection model. Uses GPT-2 small and rank=10 for vocabulary cutoff.

We gave our participants access to the code to run these baselines as well as the outputs for these models on the training set.

4 Metrics

4.1 Performance Metric

Due to the class imbalance in the RAID dataset, typical metrics like Accuracy and AUROC are not appropriate for this task. Thus, in keeping with the original RAID paper, we use domain-adjusted TPR@FPR=5% as our metric. This metric represents how much of the generated text we are able to correctly identify while maintaining a 5% false

positive rate (where a false positive is defined as incorrectly labeling a human-written text as being machine generated).

4.2 Threshold Search

In order to measure TPR@FPR=5% we need to find a binary classification threshold that results in a 5% false positive rate on the human data for each detector and domain. The search algorithm we use for this purpose is the same algorithm that is described in the RAID paper. We start at the threshold corresponding to the mean score of human data (50% FPR), and approach the desired false positive rate by iteratively incrementing or decrementing the threshold. If we overshoot the target FPR, we divide our increment in half and flip the sign. We continue to do this process until the false positive rate is within $\epsilon = 0.0005$ of the desired false positive rate or until 50 iterations are reached. If 50 iterations are reached without convergence, then we select the threshold corresponding to the FPR that is closest to the target while still being less than the target.

4.3 Robustness Metric

In addition to this performance metric, we also calculate the standard deviation of TPR@FPR=5%. This measures how robust each model is across each domain of comparison. For subtask A, this metric will be measured across domains and for subtask B, this metric will be measured across adversarial attacks.

5 Submissions

We received 23 submissions to the shared task from 9 different teams, and system description papers from 7 of 9 teams. In this section, we describe the systems submitted by each team in detail. In Section 6 we will discuss aggregate results for the teams and in Section 7 we will discuss the broader trends across our participant submissions.

Team LuxVeri [Lx] (Mobin and Islam, 2025): This team fine-tuned both RoBERTa-base and RoBERTa-base-openai-detector on a subset of the RAID training data for 3 epochs, using a learning rate of 2e-5, a batch size of 4, and the AdamW optimizer. These trained models were then used to compute weights for ensembling based on an inverse perplexity weighting technique, which was applied to the ensemble for the adversarial task. For the non-adversarial task, they only used RoBERTa-base with the same hyperparameters.

Team Random [Ra] (Agrahari et al., 2025): This team's contributions include a pipeline that integrates XLM-RoBERTa embeddings for enhanced text representation, domain adaptation using a Domain-Adversarial Neural Network (DANN) (Ganin et al., 2016) to minimize domain-specific biases and improve generalization across diverse text domains, and adversarial robustness through incorporating adversarial attack classification to detect and mitigate manipulative techniques.

Team USTC-BUPT [Us] : This team fine-tuned RoBERTa-Large via focal loss (Lin et al., 2017) on the RAID training set by adding four samples for each human sample using synonym replacement. They then down-sampled the AI-written texts from a 10:1 ratio to a 2:1 ratio of AI to human text to form the training set. In the hyperparameters for focal loss, they set alpha to 0.65 and gamma to 2.5. They used a learning rate of 1e-6 and AdamW as the optimizer. In addition to this, they also had a secondary submission where they simply fine-tuned RoBERTa-base on a subset of the RAID training set with learning rate of 5e-5.

Team ALERT [Al] (Kandula et al., 2025): This team's approach uses robust authorship style representations to distinguish between human-authored and machine-generated text (MGT) across various domains. Their authorship attribution (AA) systems are trained with contrastive learning. They employ an ensemble-based AA system (ALERT v1.1) that integrates stylistic embeddings from two complementary subsystems: One system that focuses on cross-genre robustness with hard positive and negative mining strategies using Linq-Embed-Mistral (Kim et al., 2024) as backbone architecture, and a second system with Semantic, Lexical, Clustering based hard positive and negative mining which uses Qwen2-1.5B.

Team Leidos [Le] (Edikala et al., 2025): This team trained four Distil-RoBERTa-Base detectors to evaluate both binary and multi-class classification, exploring the effects of class weighting to address dataset imbalance, in order to distinguish human-written from machine-generated text. The "Leidos Detector v1.0.1" is a Binary Classifier without Class Weighting (BC): Human vs. Machine, trained without applying class weights. The "Leidos Detector v1.0.3" is a Binary Classifier with

Subtask A: Performance Across Domains (Official Results)									
	News	**Wiki**	**Reddit**	**Books**	**Abs.**	**Reviews**	**Poetry**	**Recipes**	**Total** (σ)
[Le] Leidos v1.0.3	**99.9**	99.8	98.3	99.4	99.9	98.6	99.3	**100.0**	**99.4** (0.6)
[Pa] Pangram	99.7	99.1	98.5	99.5	99.3	**99.6**	98.8	99.9	99.3 (**0.4**)
[Le] Leidos v1.0.2	**99.9**	**99.9**	**99.4**	99.5	99.9	95.9	**99.6**	**100.0**	99.3 (1.2)
[Le] Leidos v1.0.4	**99.9**	99.7	99.0	99.3	**100.0**	96.5	99.4	**100.0**	99.2 (1.1)
[Le] Leidos v1.0.1	**99.9**	99.8	98.6	99.4	99.9	96.2	99.4	**100.0**	99.1 (1.2)
[Us] R-L Focal Loss	99.0	97.8	96.1	98.1	99.8	97.0	97.0	99.9	98.1 (1.3)
[Al] ALERT v1.1	99.7	95.4	75.7	**99.9**	99.9	87.2	78.3	98.3	91.8 (9.4)
[Cn] DistilBERT-NITS	89.9	87.7	90.0	93.5	90.9	85.9	90.0	96.0	90.5 (2.9)
[Al] ALERT v1.2	99.5	91.3	87.2	99.2	99.9	89.9	64.9	82.8	89.3 (11.0)
[Lx] R-B & R-Oai	87.5	90.2	62.4	89.5	99.2	83.7	73.5	75.1	82.6 (10.9)
[Lx] R-Oai & BERT	91.8	87.3	75.1	87.1	97.0	86.0	76.3	59.4	82.5 (11.1)
[Lx] Fine-tuned R-B	87.5	89.7	61.7	89.6	98.8	82.5	66.3	74.6	81.3 (11.9)
[Ba] Binoculars	80.7	76.5	81.3	82.8	76.0	78.0	80.1	76.4	79.0 (2.4)
[Mo] MOSAIC-4	79.5	67.6	78.2	79.8	77.1	81.4	63.7	75.8	75.2 (5.9)
[Mo] MOSAIC-5	79.0	65.8	76.7	79.8	76.5	77.2	64.8	75.1	74.5 (5.4)
[Lx] Radar & R-L	91.6	73.7	76.3	78.1	74.2	58.7	45.7	73.5	71.5 (12.8)
[Ba] RADAR	87.4	77.3	73.6	78.1	67.5	6.3	46.0	88.7	65.6 (25.7)
[Ba] GLTR	67.7	63.6	63.2	71.9	60.1	65.0	18.2	67.9	59.7 (16.0)
[80] L3-60 Zero-shot	63.6	36.5	61.5	65.4	55.3	68.9	51.5	53.9	57.1 (9.6)
[80] M3-60 Zero-shot	58.1	58.1	65.8	63.3	44.1	67.1	53.2	50.5	56.5 (7.4)
[Ba] openai-roberta-large	67.8	59.4	60.0	52.5	64.8	52.8	23.3	65.1	55.7 (13.3)
[Cn] Adv.-submission-3	27.1	26.1	52.8	57.1	30.1	48.6	38.0	94.0	46.7 (21.1)
[Cn] Adv.-New-Detector	14.0	16.2	40.4	39.2	34.7	29.4	17.8	91.0	35.3 (23.2)
[Us] Roberta_dataaug.	4.6	3.6	40.5	7.3	83.1	3.1	5.1	98.8	30.8 (36.8)
[Cn] Adv._Data_Detector	10.1	17.5	27.9	24.8	27.7	28.7	13.5	88.0	29.8 (23.0)
[Lx] Radar R-B CGPT-R	20.0	16.0	4.8	2.5	51.1	62.1	4.4	32.9	24.2 (21.1)
[Ra] Adv. CDMGTD	4.2	3.4	2.1	2.1	6.8	2.9	1.7	2.4	3.2 (1.6)
Average Performance	70.7	66.6	68.4	71.8	74.6	67.7	58.1	78.5	69.5 (5.7)

Table 4: TPR at FPR=5% for detectors across different domains on the RAID test set along with their standard deviation (σ). Baselines are given the [Ba] tag. "Abs." is shorthand for Abstracts. Team rankings are determined based on the highest performing submission for each team (see Table 7).

Class Weighting (BW) which is similar to the BC model but trained with class weights to address class imbalance. The "Leidos Detector v1.0.4" is a multi-class classifier without class Weighting (MC): A multi-class classifier that predicts which generator model produced the text or if it was human-written, trained without class weights. Finally the "Leidos Detector v1.0.2" is a Multi-class Classifier with Class Weighting (MW) which is the same as the MC model but trained with class weights to mitigate class imbalance.

Team MOSAIC [Mo] (Dubois et al., 2025): This team submitted a completely unsupervised approach which uses a mixture of models to score the texts. Their ensemble method is grounded in fundamental information-theoretic principles from universal compression in order to optimally combine the strengths of multiple LLMs for machine-generated text detection. The four models used are Tower-7b, Tower-13b, Llama-2-7b and Llama-2-7b-chat.

Team CNLP-NITS-PP [Cn] (Teja et al., 2025): This team submitted a model that first classifies whether the text has been adversarially attacked or not. If an attack is detected, the text undergoes preprocessing to mitigate the attack, after which the preprocessed text proceeds to the model for MGT detection. Finally, their detector works by fine-tuning a DistilBERT model to extract semantic features.

Team 1-800-SHARED-TASKS [80] (Kadiyala, 2024; Kadiyala et al., 2024): This team submitted a model that uses a token classifier for detecting change in authorship within the same text. The data from the current benchmark would be both unseen domain and unseen generators' data. Inference was done without any pre-processing against adversarial methods, nor were those methods present in the training data. The final generated score was based on the proportion of tokens classified as AI generated among the ones within the supported context length.

Subtask B: Performance Across Adversarial Attacks (Official Results)												
	AS	**AD**	**HG**	**IP**	**NS**	**PP**	**MS**	**SY**	**UL**	**WS**	**ZW**	**Total** (σ)
[Le] Leidos v1.0.2	99.2	99.0	**97.3**	98.7	99.2	92.3	98.8	98.6	98.9	99.0	92.7	**97.7** (**2.5**)
[Pa] Pangram	99.2	98.7	91.9	**99.3**	99.2	91.6	99.0	96.2	**99.3**	**99.3**	99.3	**97.7** (2.9)
[Le] Leidos v1.0.4	99.1	99.0	94.7	98.7	99.2	94.8	98.8	98.6	98.9	98.8	90.9	97.6 (2.6)
[Le] Leidos v1.0.3	**99.3**	**99.3**	93.6	98.7	**99.4**	**96.3**	**99.2**	**99.1**	99.2	99.2	84.2	97.2 (4.4)
[Le] Leidos v1.0.1	99.0	99.0	86.1	98.1	99.1	94.8	98.9	98.8	98.8	98.5	78.8	95.7 (6.4)
[Us] R-L Focal Loss	97.9	98.2	84.5	93.6	98.1	84.0	97.8	97.4	97.9	97.9	67.1	92.7 (9.5)
[Al] ALERT v1.1	91.8	92.1	68.5	89.7	91.8	57.7	91.0	87.3	91.3	91.2	46.8	82.6 (15.5)
[Lx] Fine-tuned R-B	80.5	78.1	90.4	79.8	79.8	77.9	77.9	74.4	75.0	66.2	**100.0**	80.1 (8.4)
[Al] ALERT v1.2	89.9	89.0	61.9	84.1	88.6	57.1	88.6	84.1	87.2	85.6	40.2	78.8 (16.0)
[Lx] R-B & R-Oai	81.7	79.4	41.7	81.2	81.1	78.1	79.3	75.8	76.1	68.0	86.9	76.0 (11.6)
[Lx] R-Oai & BERT	81.6	79.4	20.9	81.7	82.2	75.8	79.6	77.6	76.7	77.1	83.7	74.9 (17.0)
[Ba] Binoculars	78.2	74.3	37.7	71.7	77.1	80.3	78.0	43.5	73.8	70.1	99.1	71.3 (16.2)
[Ba] Radar	70.8	67.9	59.3	73.7	71.0	67.3	69.5	67.5	70.4	66.1	82.2	69.6 (5.3)
[Mo] MOSAIC-5	72.2	69.5	90.2	73.3	69.7	70.3	71.7	22.7	66.5	67.0	85.5	69.4 (16.3)
[Mo] MOSAIC-4	72.9	70.8	86.6	74.5	71.3	71.9	72.5	28.5	68.6	67.5	71.4	69.3 (13.6)
[Lx] Radar & R-L	70.3	61.2	21.2	73.0	69.9	73.0	63.9	74.9	55.7	60.2	91.3	65.5 (16.6)
[Ba] GLTR	61.2	52.1	24.3	61.4	59.9	47.2	59.8	31.2	48.1	45.8	97.2	53.5 (18.1)
[80] L3-60 Zero-shot	56.6	50.5	3.0	57.4	56.3	50.6	55.6	53.5	57.1	61.9	57.1	51.4 (15.4)
[Ba] openai-roberta-L	52.4	33.2	21.3	55.1	51.7	72.9	39.5	79.4	19.3	40.1	99.9	51.3 (23.6)
[80] M3-60 Zero-shot	55.6	48.6	3.6	56.7	52.2	37.7	53.7	40.2	56.5	59.7	56.5	48.1 (15.4)
[Cn] Adv.-sub.-3	46.7	45.1	20.8	46.7	46.5	18.0	46.8	41.6	46.7	46.7	46.7	41.6 (10.4)
[Cn] Adv.-New-Det.	35.3	35.2	18.9	35.3	35.4	11.9	35.4	31.6	35.3	35.3	35.3	31.7 (7.7)
[Us] Roberta_dataaug.	30.8	31.6	16.4	31.8	30.8	26.8	30.4	30.1	30.8	29.5	11.6	27.6 (6.5)
[Cn] Adv._Data_Det.	29.7	29.4	18.5	29.8	29.6	8.5	29.8	26.9	29.8	29.8	29.8	26.8 (6.5)
[Lx] Radar R-B C-R	22.3	15.2	0.4	4.9	22.0	34.9	18.1	30.0	6.6	4.3	11.0	16.2 (10.6)
[Ra] Adv. CDMGTD	3.2	3.0	24.8	3.2	3.2	3.5	3.2	3.2	3.2	3.2	20.8	6.5 (7.6)
Average Performance	68.4	65.3	49.2	67.4	67.9	60.6	66.8	61.3	64.1	64.2	67.9	64.3 (5.3)

Table 5: TPR at FPR=5% for detectors across different adversarial attacks along with their standard deviation (σ). Baselines are given the [Ba] tag. Abbreviations are: AS: Alternative Spelling, AD: Article Deletion, HG: Homoglyph, IP: Insert Paragraphs, NS: Number Swap, PP: Paraphrase, MS: Misspelling, SY: Synonym Swap, UL: Upper Lower Swap, WS: Whitespace Addition, ZW: Zero-Width Space Addition. Team rankings determined by the highest performing submission (see Table 8).

Team Pangram [Pa] (Emi et al., 2025): This team pretrained an autoregressive LLM-based detector on a wide variety of datasets, domains, languages, prompt schemes, and LLMs used to generate the AI portion of the dataset. They aggressively employed several augmentation strategies and preprocessing strategies to improve robustness. They then mined the RAID train set for the AI examples with the largest error based on the original classifier, mixed those examples and their human-written counterparts back into the training set, and retrained the detector until convergence.

6 Results

6.1 Subtask A: Cross-Domain Detection

In Table 4, we report the official results for Subtask A. Overall, we see that a large fraction of our teams beat our provided baselines and established strong new results. The winning team, Leidos, achieved 99.4% TPR across all 8 domains—a substantial improvement over the prior state of the art result on RAID. This suggests that it is possible to build classifiers that are robust across a finite set of domains and models.

In terms of comparisons between domains, the most difficult domain to classify across all submissions was Poetry (58.1%), and the easiest were Recipes (78.5%) and Abstracts (74.6%). Poems are inherently difficult to identify for machines as they rely on rare and unusual word choice, and are likely to be surprising even to well trained classifiers. On the other hand, both the Recipes and Abstracts domains were unexpectedly easy for the detectors.

6.2 Subtask B: Adversarial Robustness

In Table 5, we report the official results for subtask B. We observe similarly high performance, with the two winning teams (Leidos and Pangram) getting a surprisingly high 97.7% TPR on the dataset. Once again, we see many teams beat our strong baselines and feel that these results validate our intuition that classifiers, when given a finite set of adversarial

attacks to defend against, can do quite well.

In addition, while many attacks from the original RAID paper are effective when the defender is not prepared for them (e.g., Whitespace Insertion, Article Deletion), such attacks seem to be relatively easy to defend against. One interesting finding is that the most difficult attacks to defend against were the Homoglyph attack (49.2%), the Paraphrase attack (60.6%), and the Synonym attack (61.3%). Despite having access to the code that generated the Homoglyph attack, it is still difficult to create a model that is robust to this attack without any text preprocessing or normalization. In addition, while many attacks are easy to defend against with good knowledge of the attacker, it seems that both the Paraphrase and Synonym attacks remain difficult to deal with even with perfect knowledge. This is because these attacks are not easily solved with simple preprocessing techniques, and because they require models to learn alternative distributions that are often fairly distinct from generic generated text.

7 Broader Trends

Across all submissions to the shared task we find the following broader trends:

Trend 1: Text Preprocessing The first clear trend we saw was the use of text preprocessing. Team [Cn] and [Ra] both trained classifiers to detect particular adversarial attacks in the dataset in order to apply preprocessing, and team [Pa] simply ran all incoming text to the detector through text normalization. Looking at the results, these methods seemed to be largely effective at neutralizing many of the simpler adversarial attacks such as Zero-Width Space, Upper-Lower Swap, Insert Paragraphs, Whitespace Insertion and Alternative Spelling. However, attacks such as Paraphrasing, Synonym Swap, Article Deletion, and Misspelling seem to be more difficult to preprocess away. Such attacks, which either create or destroy vital information in the text, can be seen as fundamentally different and more difficult to defend against.

Trend 2: Hard Positive and Negative Sampling The second clear trend we found was the active sampling for hard examples in the dataset. Of the top four teams, three reported using this particular technique. Team [Us] used focal loss, which concentrates learning on hard misclassified examples; Team [Al] used pairs of difficult examples to train their contrastive learning objective, and Team [Pa] looked specifically for examples where their classifier had large error and incorporated them into their existing pre-training dataset. While it is difficult to isolate the effect of any one particular method or technique in a study like this, it is clear that sampling particularly hard examples is a promising direction for building robust classifiers.

Trend 3: Diversity of Approach The third and final trend we feel is relevant to mention is the diversity of approaches we witnessed. We saw submissions involving unsupervised methods [Mo], ensemble methods [Lx], token-level models [80], contrastive learning models [Al], and multi-class classification methods [Le]. Each of the submissions differed drastically from the others not only in terms of performance, but also in adversarial and cross-domain robustness properties. For example, we see that the unsupervised approaches [Mo] exhibited lower overall performance but higher cross-domain robustness than ensemble methods [Lx]. This suggests that there is a large potential for novel modeling work to be done in detection.

8 Discussion

In real-world scenarios, API-based detectors can expect a majority of their text to come from a relatively small set of LLMs and domains. The most common frontier models (Gemini Pro, Claude 3.5, GPT-4o) will mostly likely be used to generate text in a set of high-risk domains (academic essays, news articles, scientific papers) with potentially some paraphrasing or synonym changes applied to help avoid detection. In such cases, it is desirable for a detector to focus on optimizing performance as much as possible on this fixed set of models and domains while not caring as much about performance on other less common models. Our results show that detectors can exhibit very strong performance in such constrained settings.

Future work should seek to further target this particular setting and replicate these findings on a much more diverse corpus. Such a study would entail collecting many generations from a core set of models in high risk domains and conduct a heavy adversarial attack circuit. Since prompt-based attacks are likely to be common (e.g. "write in a way that is not detectable") these should be a major focus. In addition, testing a large set of "humanizing" paraphrase models is also desirable. It is unclear whether or not our results will extend to models that have significant prompt and paraphrase

Domain: Abstracts, **Title**: EdgeFlow: Achieving Practical Interactive Segmentation with Edge-Guided Flow	
ChatGPT	EdgeFlow is a novel approach to interactive image segmentation that combines edge detection and flow-based methods to achieve practical and efficient results. The proposed method utilizes an edge-guided flow algorithm to guide the segmentation process, allowing users to interactively refine the segmentation boundaries. The algorithm incorporates both local and global information to accurately capture [...]
Human	High-quality training data play a key role in image segmentation tasks. Usually, pixel-level annotations are expensive, laborious and time-consuming for the large volume of training data. To reduce labelling cost and improve segmentation quality, interactive segmentation methods have been proposed, [...]
Domain: Recipes, **Title**: Olive Spirals	
ChatGPT	Instructions: 1. Preheat your oven to 400°F (200°C) and line a baking sheet with parchment paper. 2. In a bowl, combine the black olives, green olives, sun-dried tomatoes, feta cheese, basil, parsley, olive oil, salt, and pepper. Mix well to combine all the ingredients. 3. Roll out the puff pastry sheet on a lightly floured surface into a rectangle shape, about 1/4 inch thick. 4. Spread the olive mixture evenly over the puff pastry, leaving a small border around the edges. [...]
Human	Mix yeast, sugar& 125ml warm water in a bowl. Cover and set aside in a warm place for 10 minutes, or until frothy. Sift the flour and salt into a bowl and make a well in the centre. Add frothy yeast, oil and 250ml of warm water. Mix to a soft dough and gather into a ball. Turn out on a floured surface and knead for 10 minutes until smooth. Cover loosely with greased plastic wrap and set aside for 1 hour until [...]

Table 6: Comparison of human-written text to chatgpt-written text in the Abstracts and Recipes domain from the RAID dataset. We can see that the human-written abstract has periodic newline characters and the generated text does not. In addition we see that the ChatGPT-written recipe has numbered lists of instructions while the human-written text is written in paragraph form. Artifacts such as these may trivialize the detection task.

variations applied.

Finally, it is worth noting that detectors doing well in constrained settings does not imply generalization to unseen models. Detectors still suffer from poor generalization across unseen models and domains as discussed in Dugan et al. (2024). However, this discrepancy between in-distribution and out-of-distribution performance is something that is worth highlighting as a potential source for future investigation.

Limitations

Test Data Leakage The first thing to note is the potential for test data leakage. Since the human-written documents are sampled from publicly available datasets and we give our participants the links to such datasets, there is a chance that participants models have seen the human-written documents in our test data and have overfit to them, allowing for high performance. In our eyes this is unlikely, as most of the linked datasets have a large amount of documents and the percentage of documents included in test data from those is vanishingly small. In addition, such leakage would only be problematic when searching for thresholds. Since our metric is TPR, we only measure the accuracy of the classifier in identifying machine-written text, all of which is hidden and has never been released publicly. Nonetheless, this is an important caveat to include.

Confounding Factors On manual investigation of the RAID dataset we found instances of confounding factors in the data that would potentially make detection easier (Table 6). For example, all human-written recipes were written without numbered lists of steps whereas all generated recipes included numbered lists of steps. To investigate the effects of these confounds we manually cleaned the data to remove the most egregious examples and trained a RoBERTa-base model on both the original and cleaned RAID. We saw a drop from 92.67 to 89.67 after cleaning the data—a small but significant performance difference. We are continuing to investigate this and do not yet have enough evidence to conclude whether or not this is the source of the high performance. We hope that future studies can help to shed more light on this issue not only in RAID but in other common benchmark datasets as well.

Acknowledgments

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005. The views and conclusions

contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Shifali Agrahari, Prabhat Mishra, and Sujit Kumar. 2025. Team Random at GenAI Detection Task 3: A Hybrid Approach to Cross-Domain Detection of Machine-Generated Text with Adversarial Attack Mitigation. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. 2024. Large Language Model Lateral Spear Phishing: A Comparative Study in Large-Scale Organizational Settings. *Preprint*, arXiv:2401.09727.

Janek Bevendorff, Matti Wiegmann, Jussi Karlgren, Luise Dürlich, Evangelia Gogoulou, Aarne Talman, Efstathios Stamatatos, Martin Potthast, and Benno Stein. 2024. Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2024. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings. CEUR-WS.org.

Meghana Moorthy Bhat and Srinivasan Parthasarathy. 2020. How Effectively Can Machines Defend Against Machine-Generated Fake News? An Empirical Study. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 48–53, Online. Association for Computational Linguistics.

Shuyang Cai and Wanyun Cui. 2023. Evade chatgpt detectors via a single space. *Preprint*, arXiv:2307.02599.

Shammur Absar Chowdhury, Hind Al-Merekhi, Mucahid Kutlu, Kaan Efe Keleş, Fatema Ahmad, Tasnim Mohiuddin, Georgios Mikros, and Firoj Alam. 2025. GenAI content detection task 2: AI vs. human – academic essay authenticity challenge. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Matthieu Dubois, François Yvon, and Pablo Piantanida. 2025. MOSAIC at GenAI Content Detection Task 3: Cross-Domain Machine Generated Text Detection. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

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.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 189–196, Online. Association for Computational Linguistics.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or Fake Text? Investigating Human Ability to Detect Boundaries between Human-Written and Machine-Generated text. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Abishek R Edikala, Gregorios A Katsios, Noelie V Creaghe, and Ning Yu. 2025. Leidos at GenAI Content Detection Task 3: A Weight-Balanced Transformer Approach for AI Generated Text Detection Across Domains. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

Bradley Emi, Elyas Masrour, and Max Spero. 2025. Pangram at GenAI Content Detection Task 3: An Active Learning Approach to Machine Generated Text Generation. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

Pieter Fivez, Walter Daelemans, Tim Van de Cruys, Yury Kashnitsky, Savvas Chamezopoulos, Hadi Mohammadi, Anastasia Giachanou, Ayoub Bagheri, Wessel Poelman, Juraj Vladika, Esther Ploeger, Johannes Bjerva, Florian Matthes, and Hans van Halteren. 2024. The clin33 shared task on the detection of text generated by large language models. *Computational Linguistics in the Netherlands Journal*, 13:233–259.

Rinaldo Gagiano, Maria Myung-Hee Kim, Xiuzhen Zhang, and Jennifer Biggs. 2021. Robustness Analysis of Grover for Machine-Generated News Detection. In *Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association*, pages 119–127, Online. Australasian Language Technology Association.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. *Preprint*, arXiv:1801.04354.

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.

Jesus Guerrero, Gongbo Liang, and Izzat Alsmadi. 2022. A Mutation-based Text Generation for Adversarial Machine Learning Applications. *Preprint*, arXiv:2212.11808.

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. *Preprint*, arXiv:2401.12070.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-Text Detection via Adversarial Learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS 2023, Red Hook, NY, USA. Curran Associates Inc.

Ram Mohan Rao Kadiyala. 2024. RKadiyala at SemEval-2024 Task 8: Black-Box Word-Level Text Boundary Detection in Partially Machine Generated Texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 511–519, Mexico City, Mexico. Association for Computational Linguistics.

Ram Mohan Rao Kadiyala, Siddartha Pullakhandam, Kanwal Mehreen, Ashay Srivastava, Subhasya TippaReddy, Arvind Reddy Bobbili, Drishti Sharma, Suraj Chandrashekhar, Modabbir Adeeb, and Srinadh Vura. 2024. mmgtd-corpus (v1).

Hemanth Kandula, Chak Fai Li, Haoling Qiu, Damianos Karakos, Hieu Man Duc Trong, Thien Huu Nguyen, and Brian Ulicny. 2025. BBN-U.Oregon's ALERT system at GenAI Content Detection Task 3: Robust Authorship Style Representations for Cross-Domain Machine-Generated Text Detection. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

Yury Kashnitsky, Drahomira Herrmannova, Anita de Waard, George Tsatsaronis, Catriona Catriona Fennell, and Cyril Labbe. 2022. Overview of the DAGPap22 shared task on detecting automatically generated scientific papers. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 210–213, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, Jy yong Sohn, and Chanyeol Choi. 2024. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog.

Jules King, Perpetual Baffour, Scott Crossley, Ryan Holbrook, and Maggie Demkin. 2023. Llm - detect ai generated text. `https://kaggle.com/competitions/llm-detect-ai-generated-text`. Kaggle.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. How you prompt matters! Even task-oriented constraints in instructions affect LLM-generated text detection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14384–14395, Miami, Florida, USA. Association for Computational Linguistics.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Preprint*, arXiv:2303.13408.

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.

Gongbo Liang, Jesus Guerrero, and Izzat Alsmadi. 2023a. Mutation-based adversarial attacks on neural text detectors. *Preprint*, arXiv:2302.05794.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023b. GPT detectors are biased against non-native english writers. *Patterns*, 4(7):100779.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017. Focal Loss for Dense Object Detection . In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Los Alamitos, CA, USA. IEEE Computer Society.

Brady D Lund, Ting Wang, Nishith Reddy Mannuru, Bing Nie, Somipam Shimray, and Ziang Wang. 2023. ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, 74(5):570–581.

Md Kamrujjaman Mobin and Md Saiful Islam. 2025. LuxVeri at GenAI Content Detection Task 3: Cross-Domain Detection of AI-Generated Text Using Inverse Perplexity-Weighted Ensemble of Fine-Tuned Transformer Models. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

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, Los Alamitos, CA, USA. IEEE Computer Society.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can AI-Generated Text be Reliably Detected? *Preprint*, arXiv:2303.11156.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. *Preprint*, arXiv:2309.11285.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the RuATD Shared Task 2022 on Artificial Text Detection in Russian. In *Computational Linguistics and Intellectual Technologies*. RSUH.

Filipo Sharevski, Jennifer Vander Loop, Peter Jachim, Amy Devine, and Emma Pieroni. 2023. Talking Abortion (Mis) information with ChatGPT on TikTok. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 594–608. IEEE.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. 2023. AI model GPT-3 (dis)informs us better than humans. *Science Advances*, 9(26):eadh1850.

L D M S Sai Teja, Annepaka Yadagiri, M Srikar Vardhan, and Partha Pakray. 2025. CNLP-NITS-PP at Task 3: Cross-Domain Machine-Generated Text Detection Using DistilBERT Techniques. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. 2024. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2057–2079, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Artem Shelmanov, Jonibek Mansurov, Akim Tsvigun, Vladislav Mikhailov, Rui Xing, Zhuohan Xie, Jiahui Geng, Giovanni Puccetti, Ekaterina Artemova, Jinyan Su, Minh Ngoc Ta, Mervat Abassy, Kareem Elozeiri, Saad El Dine Ahmed, Maiya Goloburda, Tarek Mahmoud, Raj Vardhan Tomar, Alexander Aziz, Nurkhan Laiyk, Osama Mohammed Afzal, Ryuto Koike, Masahiro Kaneko, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2025. GenAI content detection task 1: English and multilingual machine-generated text detection: AI vs. human. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Max Weiss. 2019. Deepfake Bot Submissions to Federal Public Comment Websites Cannot Be Distinguished from Human Submissions. *Technology Science*, 2019121801.

Max Wolff. 2020. Attacking Neural Text Detectors. *Preprint*, arXiv:2002.11768.

A Team Rankings

Teams are ranked by their highest performing submission. In Table 7 we report the official team ranking for Subtask A and in Table 8 we report the official ranking for Subtask B

B Extra RAID Details

In Table 9 we list the 8 domains present in RAID along with clickable links to the human-written sources from which the data was sampled. These links were given to the participants to assist in curating extra data for training.

In Table 10 we list the 11 adversarial attacks applied to the RAID data along with the relative percentage of attack surface used for the attack and the papers each attack originally came from. We provided participants with the code for these attacks to allow them to train on arbitrarily many examples of each attack at varying attack strengths.

C Recommendations for Future Evaluations

In this section, we will outline recommendations for future robustness studies and shared tasks. These recommendations come not only from our experiences with conducting this shared task, but also from discussions we had with participants about potential areas for future improvements.

Create a common preprocessing script. This should be done with the explicit goal of removing any and all potential confounding factors that

Team Ranking (Subtask A)		
	Best Submission	**Result**
[Le] Leidos	Leidos v1.0.3	**99.4** (0.6)
[Pa] Pangram	Pangram	99.3 (**0.4**)
[Us] USTC	R-L Focal Loss	98.1 (1.3)
[Al] ALERT	ALERT v1.1	91.8 (9.4)
[Cn] CNLP	DistilBERT-NITS	90.5 (2.9)
[Lx] LuxVeri	R-B & R-Oai	82.6 (10.9)
[Ba] **Baseline**	Binoculars	79.0 (2.4)
[Mo] MOSAIC	MOSAIC-4	75.2 (5.9)
[80] 1-800	L3-60 Zero-shot	57.1 (9.6)
[Ra] Random	Adv. CDMGTD	3.2 (1.6)

Table 7: Team ranking for Subtask A ranked by their best submission. Metric used for result is TPR at FPR=5% along with their standard deviation (σ). See Table 4 for full results.

Team Ranking (Subtask B)		
	Best Submission	**Result**
[Le] Leidos	Leidos v1.0.2	**97.7** (**2.5**)
[Pa] Pangram	Pangram	**97.7** (2.9)
[Us] USTC	R-L Focal Loss	92.7 (9.5)
[Al] ALERT	ALERT v1.1	82.6 (15.5)
[Lx] LuxVeri	Fine-tuned R-B	80.1 (8.4)
[Ba] **Baseline**	Binoculars	71.3 (16.2)
[Mo] MOSAIC	MOSAIC-5	69.4 (16.3)
[80] 1-800	L3-60 Zero-shot	51.4 (15.4)
[Cn] CNLP	Adv.-sub.-3	41.6 (10.4)
[Ra] Random	Adv. CDMGTD	6.5 (7.6)

Table 8: Team ranking for Subtask B ranked by their best submission. Metric used for result is TPR at FPR=5% along with their standard deviation (σ). See Table 5 for full results.

Domain	**Source**	**Description**
Abstracts	arxiv.org	ArXiv Abstracts
Recipes	allrecipes.com	Ingredients + Recipe
Books	wikipedia.org	Plot Summaries
Reddit	reddit.com	Reddit Posts
News	bbc.com/news	News Articles
Reviews	imbd.com	Movie Reviews
Poetry	poemhunter.com	Poems (Any Style)
Wiki	wikipedia.org	Article Introductions

Table 9: All domains in the RAID dataset alongside a description of where they are from. Clickable source links go directly to the source dataset from which the human samples were taken.

Attack	θ	**Source**
Alternative Spelling	100%	(Liang et al., 2023b)
Article Deletion	50%	(Liang et al., 2023a; Guerrero et al., 2022)
Homoglyph	100%	(Wolff, 2020; Gagiano et al., 2021)
Insert Paragraphs	50%	(Bhat and Parthasarathy, 2020)
Number Swap	50%	(Bhat and Parthasarathy, 2020)
Paraphrase	100%	(Krishna et al., 2023; Sadasivan et al., 2023)
Misspelling	20%	(Liang et al., 2023a; Gagiano et al., 2021; Gao et al., 2018)
Synonym	50%	(Pu et al., 2023)
Upper Lower	5%	(Gagiano et al., 2021)
Whitespace	20%	(Cai and Cui, 2023; Gagiano et al., 2021)
Zero-Width Space	100%	(Guerrero et al., 2022)

Table 10: The adversarial attacks used in the project. θ represents the manually determined fraction of available attacks carried out. We determine this fraction through manual review.

do not have to do with the text itself. We suggest making the preprocessing script public so that participants can apply it to existing pre-training data and any other data they have found on the web. This script should include text and character normalization and should standardize whitespace and capitalization rules. Another recommendation would be to restrict the length of text to be identical across all models and domains. A good starting point for such a script would be the punctuation normalizer from the Moses toolkit[3] as this is what was used for the MAGE dataset (Li et al., 2024).

Include more variations across prompts and paraphrasers. Prompts have been shown to wildly alter the stylistic components of generative model outputs even when only given task-oriented constraints—fooling detectors in the process (Koike et al., 2024). In particular, experiments that test robustness to many different prompting strategies, paraphrase models, and synonym replacement methods are likely to give a strong sense of how well detectors will hold up in real-world settings. Strategies such as prefix-based prompting, length-conditioned generation, explicitly adversarial prompting, and others are all valid strategies to incorporate into future work.

Include more human-written text. The imbalanced nature of the RAID dataset required participants to use data augmentation or up-sampling techniques that served only to degrade the quality of the data. Future work should seek to provide participants with a much larger corpus of human written texts from diverse domains.

[3] https://pypi.org/project/mosestokenizer/