NIST Trustworthy and Responsible AI
NIST AI 700-1

2024 NIST GenAI (Pilot Study):

Text-to-Text Evaluation Overview and Results

Hari Iyer
Seungmin Seo
Lukas Diduch
Kay Peterson
George Awad
Yooyoung Lee

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.700-1

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Trustworthy and Responsible AI
NIST AI 700-1

2024 NIST GenAI (Pilot Study):
Text-to-Text Evaluation Overview and Results

Hari Iyer
Seungmin Seo
Lukas Diduch
Kay Peterson
George Awad
Yooyoung Lee

Information Access Division &
Statistical Engineering Division
Information Technology Laboratory

**All work for this publication is done under the GenAI program supported by Information Access Division*

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.700-1

June 2025

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Acknowledgments
The authors would like to acknowledge Mark Przybocki for his invaluable support and guidance, Peter Fontana for his contributions and helpful discussions, and Jesse Dunietz for his detailed and insightful comments. They also thank Jim Golden for providing hardware infrastructure support.
This report is dedicated to Chuck Romine and Mary Theofanos for their exceptional leadership and mentorship at NIST.

Publication History
Approved by the NIST Editorial Review Board on 2025-06-10

How to cite this NIST Technical Series Publication:
Hari Iyer, Seungmin Seo, Lukas Diduch, Kay Peterson, George Awad, Yooyoung Lee (2025) 2024 NIST GenAI (Pilot Study): Text-to-Text Evaluation Overview and Results. (National Institute of Standards and Technology, Gaithersburg, MD), NIST AI 700-1. https://doi.org/10.6028/NIST.AI.700-1

Contact Information
genai-poc@nist.gov
National Institute of Standards and Technology
Attn: Information Technology Laboratory
100 Bureau Drive (Mail Stop 8900) Gaithersburg, MD 20899-8900

Additional Information
Additional information about this publication and other GenAI publications are available at
https://ai-challenges.nist.gov/genai

Abstract

The 2024 NIST Generative AI (GenAI) Pilot Study focuses on evaluating text-to-text (T2T) generation and discrimination tasks to assess the capabilities and limitations of generative AI models and AI detectors. The study aims to measure the effectiveness of AI-generated text in mimicking human writing and the ability of AI-based discriminators to distinguish between human- and AI-generated content. A curated dataset of article groups and associated human- and machine-generated summaries served as the benchmark, with performance assessed using statistical and machine learning-based metrics, including AUC (Area Under the Curve) and Brier scores.

The results indicate that while AI-generated summaries increasingly resemble human writing, detection models remain reasonably effective in distinguishing between them. Performance varies significantly depending on the systems used, but there are some generators that could deceive most discriminators, and there are discriminators that could detect AI-generated content from almost all generators. There is certainly room for improvement for both generator and discriminator systems. We also found that discriminator systems improved over the multiple rounds of testing.

Moving forward, future work will focus on refining evaluation methodologies, expanding multi-modal assessments across text, image, and audio domains, and developing standardized benchmarking protocols. These efforts aim to provide a robust test and evaluation framework for assessing generative AI technologies and AI detector technologies, guiding both researchers and policymakers in understanding their evolving impact.

Keywords

Artificial Intelligence (AI), Generative AI, Discriminative AI, Deepfakes, Large Language Model (LLM), Forensics, Evaluation, Measurement, Provenance, Authenticity, Detection, Accuracy, and Robustness.

Table of Contents

List of Tables

List of Figures

1. Introduction

In recent years, digital content produced by generative artificial intelligence (AI) — such as deepfakes — has experienced unprecedented growth and spread across multiple modalities, spanning images, videos, audio, text, and even code. This surge in generative AI presents both opportunities and challenges. The technologies have facilitated creative expression, enabling artists, designers, and writers to generate visually stunning content as well as fast professional written content. On the other hand, it has raised concerns regarding the authenticity and integrity of media in the digital content world. With the recent advancements in generative AI technology, it is becoming increasingly difficult to distinguish AI-generated content from human-generated content in digital media.

The NIST Generative AI (GenAI) program is an umbrella program that supports evaluations for research and measurement science in Generative AI across different modalities. It provides a platform for testing and evaluation to measure the performance of AI content "generators" and AI content "discriminators" (i.e., detectors). The platform is planned to support multiple modalities and technologies.

For the pilot study, the evaluation helps determine strengths and weaknesses of generative AI systems for the task of summarizing a collection of articles. Generator (G) teams are tested in two ways:

1. Their system's ability to generate content that is indistinguishable from human-generated content. This is typically accomplished by inducing detectors to assign scores to AI-generated content such that the score distribution for AI-generated content is statistically the same as the score distribution for human-generated content. This step is meant to help humans calibrate their trust in AI systems' ability to perform generation tasks that are traditionally performed by humans.
2. Their system's ability to generate content that can defeat skillful human and/or AI discriminators by leading them to claim the content is human-generated. This is typically accomplished by inducing detectors to assign lower detection scores to AI-generated content than to human-generated content. This can be important from a national security and public safety perspective, especially when such tools are used by adversaries.

The pilot study also helps develop insights into how (via what means or cues) and when humans and AI can detect AI-generated content. Discriminator (D) teams are tested on their system's ability to differentiate between AI-generated content and human-generated content.

In April 2024, NIST launched the GenAI Text-to-Text (T2T) pilot study. In the generator task, the objective of Text-to-Text Generators (T2T-G) is to automatically generate high-quality summaries given a statement of information needed ("topic") and a set of source articles to summarize. On the other side, the pilot Text-to-Text Discriminator (T2T-D) task is to detect if a summary was generated using a generative AI system or a human. This evaluation assumes completely AI-generated content, i.e., it ignores cases where humans

use AI tools to co-author content, such as rephrasing, grammar correction, editing, etc. Participants could join the study as part of a generator team, a discriminator team, or both.

This report outlines the task definitions, the evaluation protocol, the data used, the performance metrics, and the evaluation results for both generators and discriminators. The insights gained from this evaluation will help in selecting future research directions and provide recommendations and guidance for a better understanding of generative AI and discriminative AI technologies, as well as the performance gap between them.

2. Related Work

Detection methods for AI-generated content are a constant cat-and-mouse game between the generation and detection communities. When a new detection method is developed, the generation community can quickly respond with a countermeasure. In particular, detectors are often tied to specific generators and may not perform well on unknown ones. However, when a new generation method is developed, the detection community will often respond with new detectors capable of recognizing content created by the new generation method.

The NIST report on synthetic content [17] classified synthetic content detection techniques into three categories: provenance data detection (e.g., extracting watermarks or metadata), automated content detection (e.g., examining for inconsistencies in the content), and human-assisted detection (e.g., consulting domain experts).

In this section, we summarize insights from key references, highlighting current detection tools and the importance of initiatives such as NIST GenAI. To better understand AI-generated content and its detection, we classify research into the following categories:

Tools for detecting AI-generated content: several tools for detecting AI-generated content have been developed to distinguish AI-generated content from human-authored materials. The research points to three primary methods: linguistic analysis, deep learning models, and metadata-based techniques.

- Linguistic and Statistical Analysis: Some detection tools focus on the distinct statistical patterns found in AI-generated text. Kumarage et al. [12], Penn State News [22], Shu et al. [24], Weber-Wulff et al. [29] highlight linguistic markers such as lack of coherence over long passages, repetitive structures, unnatural phrasing, and deviations in lexical richness and syntax. These methods are widely used but face limitations as AI models improve their natural language generation capabilities.
- Deep Learning-Based Detection Models: Deep learning techniques form the backbone of most modern detection efforts. Adobe Content Authenticity Initiative (CAI) [1], DeepMind [3], DetectGPT [4], FakeCatcher [5], GPTZero [7], Harvard & MIT-IBM Watson Lab [8], Hive AI [9], Kirchenbauer et al. [11], Microsoft [16], OpenAI [19, 21], Turnitin AI Detection [26] discuss neural network-based classifiers that analyze text and images, including: OpenAI's GPT-Detector (e.g., trained to distinguish between

AI and human-generated text), MIT's DeepFake-Text (e.g., used transformer-based models to spot unnatural sentence formations), Deepfake detection algorithms for images and videos (e.g., leveraged Convolutional Neural Networks (CNNs) and adversarial training). These models show promise but require continuous updates as AI-generated content becomes more sophisticated.

- Metadata and Provenance-Based Techniques: Baly et al. [2], Turnitin AI Detection [26], Wardle and Derakhshan [28] explore metadata-based solutions, which include: Watermarking AI-generated content, Cryptographic signatures to trace content origins, Provenance tracking for images and videos (Adobe Content Authenticity Initiative). While effective, these solutions require widespread adoption to make a meaningful impact.

Role of Social Media and Information Dissemination: AI-generated information, whether factual or not, spreads rapidly through social media platforms. Research [6, 24, 27, 30] identifies key factors such as algorithmic amplification, echo chambers, and cognitive biases that exacerbate the problem. Detection tools are now integrating real-time monitoring solutions to flag inappropriate content.

Fact-checking and Verification techniques: Despite technological advancements, human-led fact-checking and verification remain crucial. Pennycook and Rand [23], Tan et al. [25], Wardle and Derakhshan [28] highlight hybrid approaches where AI assists human fact-checkers by pre-filtering likely incorrect information candidates, allowing experts to focus on verification and context.

These categories provide a holistic view of AI-related challenges, both positive and negative, the credible information problem, and the countermeasures available.

The National Institute of Standards and Technology (NIST) plays a vital role in evaluating AI-generated content detection. NIST evaluations do not take a position on whether the AI-generated content is factual or not. Our GenAI program focuses on benchmarking detection tools, ensuring reliability and robustness. This initiative addresses key challenges, including: Evaluating model performance across different AI architectures and establishing benchmark test datasets for consistent detection results.

While detection tools continue to improve, challenges remain:

- Adversarial evolution: AI models learn to evade detection, requiring continual updates to detection techniques.
- Scalability: Real-time, large-scale monitoring is resource-intensive.
- Legal & Ethical Considerations: Striking a balance between detection and privacy rights is crucial.

The battle against AI-generated deepfakes (information credibility) is ongoing. Research across multiple disciplines has produced effective detection methods, but as AI evolves, countermeasures may also need to evolve.

The NIST GenAI initiative provides a much-needed evaluation framework for benchmark, ensuring that detection tools remain robust against ever-advancing generative models. Future efforts may integrate AI detection with human oversight, policy intervention, and public awareness campaigns to safeguard information integrity in the digital age.

Just like deepfake detectors, AI generators, and large language models (LLMs) have room for improvement. They should improve at understanding context, giving accurate information, and reducing training bias. Enhancing creativity and clarity of AI-generated content will help in education, research, and content creation. Adding safeguards such as digital watermarks can prevent misuse. As AI-generated content becomes more advanced, making it more transparent, fair, and trustworthy will ensure it benefits society.

3. Evaluation Framework

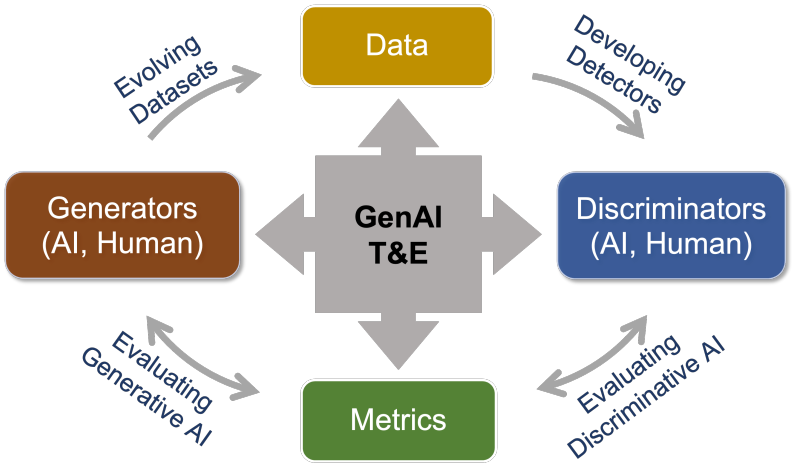

Figure 1. NIST GenAI, a generative adversarial paradigm.

The GenAI evaluation framework includes multiple rounds of evaluations for both generators and discriminators. The concept of the GenAI evaluation paradigm illustrated in Figure 1 is that generators create evolving AI data and provide it to discriminators for evaluation. The discriminators then determine if the output content is AI or human-generated, and provide performance results for each generator's output. Generators, in turn, provide new, presumably improved, content for each discriminator's evaluation. This process continues in a recurring manner, resembling a generative adversarial paradigm. G-participants, as well as D-participants, have opportunity to improve their systems to perform better against opposing systems than in previous rounds.

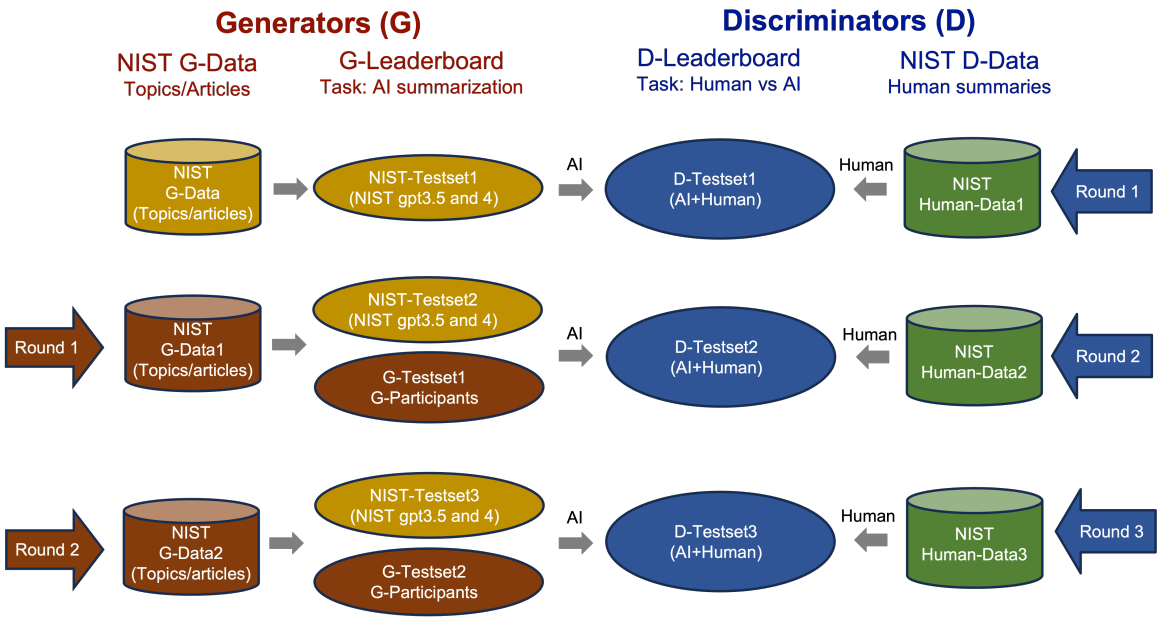

Figure 2. GenAI multiple rounds of evaluation framework.

In the pilot study, as illustrated in Figure 2, generators had two rounds shown in brown, and discriminators had three rounds shown in blue. NIST provided initial source articles, organized by topic, to G-participants, who each submitted their AI-generated summaries to the G-leaderboard. NIST created a testset consisting of "genuine" human reference summaries and generators' AI summaries for discriminator evaluations. D-participants scored every summary in the testset and submitted their system output to NIST. NIST displayed its results on the D-Leaderboard.

In Round-1 discriminator evaluation, the NIST GenAI team used a baseline generator system to create AI-generated summaries that detectors would have to discriminate from the human reference summaries.

Round-1 G-submissions provide AI summaries for Round-2 D-testset and Round-2 G-submissions for Round-3 D-testset. Although the process could continue, we concluded it at Round 3 and report the results in this study. Using relevant metrics described in Section 6, we evaluate the performance of each generator based on the D-submissions. Simultaneously, we assess each discriminator's performance on the G-submissions. At each round, both G and D-participants can see how their G-systems or D-systems are improving.

The submissions from Round-3 were completed on January 27, 2025. The entire collection of submissions, from both G-participants and D-participants over the three rounds, was analyzed. Details and analysis of the datasets and the G- and D-submissions are provided in Sections 7 and 8.

4. Evaluation Infrastructure

This section provides a brief description of the NIST GenAI evaluation management and pipeline (backend and frontend).

4.1. Evaluation Management

4.1.1. Goal and Function within Evaluation

The evaluation management platform provides functionality and key components, supporting a research and development cycle driven by adversarial evaluation. The platform allows for public-facing resource access, public- and private content management, and full evaluation cycle management including registration, submission access, and evaluation result access. Furthermore, conditions and requirements vary across challenge tasks and modalities, thus requiring mechanisms for constant adaptation to evaluation challenges. Lastly, the platform addresses core requirements such as availability, security, and usability, along with other NIST requirements. A high-level breakdown of requirements and features is provided below.

Platform Requirements

- Provide a content management system for the evaluation-driven research and development cycle, utilizing an adversarial approach
- Manage and provide role-based access and functions
- Track participant progress
- Manage participant document and file access
- Handle dataset release and related licensing workflow
- Interface with various scoring backends and collect scoring results
- Manage result release and report rendering
- Provide an area for embedding additional micro-services like data visualization

Core Platform Features

- User accounts with roles (participant, liaison, admin, license-liaison, principal-investigator)
- Dynamic content management (front page, notices, public/private content pages)
- Dynamic asset management (public and private assets)
- Dynamic submission management and submission mode configuration
- Score pipeline integration: Each submission is automatically assigned scoring runs that are processed by the scoring backend
- Integration of aggregation pipeline: Scoring backend is capable of summarizing scoring runs to track progress over time or provide a leader board
- Custom reports generation on the backend
- Business logic (BL) object-driven design: BL objects embed functionality that can be selected and chained to support dynamic use cases across various parts of the system
- Creation of platform Objects (e.g., Sites, Submissions, Licenses) that are parameterizable using JSON schema and modifiable using business logic

4.1.2. Evaluation Platform architecture overview

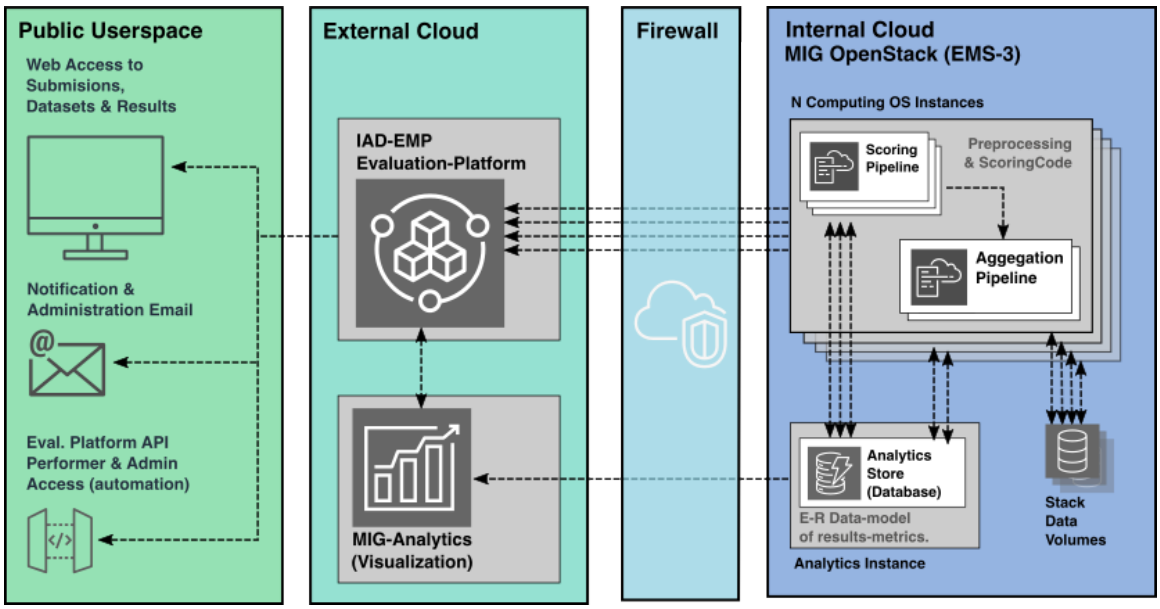

Figure 3. System architecture showing frontend and backend components. The backend components can be scaled dynamically to match demand using a cloud cluster.

The evaluation platform is composed of a public-facing webapp frontend system and an internally operated and fully automated scoring backend system. The public-facing frontend application is run outside of the NIST firewall with no access to internal NIST infrastructure. Additionally, there is an independent instance of an R-shiny data visualization server as well as a PostgreSQL database instance, which are orchestrated to provide live scoring data visualization on the frontend by being updated as soon as scores are computed on the backend. The scoring backend is decoupled and resides within NIST premises to meet security requirements. Due to this system architecture, the scoring constantly polls the front end to keep the global application state synchronized.

4.1.3. Registration and Licensing process

To participate in NIST GenAI evaluation tasks, participants create an account on the web frontend. They need a login.gov account to do so. After signing up, participants need to complete the following workflow steps before being able to obtain the data and upload results:

- Provide participant information (e.g., name, country, affiliation, affiliation type)
- Create a team for licensing purposes
- Register for desired evaluation tracks
- Complete required agreements; the NIST GenAI team reviews these before the next step (see section 4.3)
- Obtain evaluation dataset(s) and generate system output
- Create a system slot on the submission dashboard

4.1.4. Submission Process

Using the obtained evaluation dataset, participants generate system outputs to be evaluated. Submissions can then be uploaded directly through the web platform via the submission dashboard, which presents a matrix of task phases across user systems. On the upload web platform, the participant specifies parameters tuned for each evaluation task if required. Once a submission is uploaded, the system automatically generates a parameterized scoring run, which is subsequently picked up by the scoring backend for processing. Upon completion of scoring, the results are uploaded back to the webapp and made available on the participant submission dashboard.

4.1.5. Analytics Backend

The scoring backend is responsible for processing all system submissions. The system pulls the submission file, scoring run information, and submission metadata, then processes the output using a linear pipeline of processing steps. If the submission scores successfully (i.e., all pipeline steps succeed), the scores will be uploaded to the evaluation platform, where they can be viewed by participants. If a step fails, the participant is provided with a detailed error available on the web app dashboard.

Besides static report cards with each submission, which are available only to participants on their dashboard, the system also provides real-time analytics in the form of a public, interactive leaderboard. The leaderboard application draws data from the scoring database and is hosted on a separate R-Posit Connect instance within the virtual private cloud network of the system. The leaderboard application is implemented in R-Shiny to allow for interactive data exploration and is updated automatically for each successfully scored system output.

4.2. Evaluation Pipeline

4.2.1. Implementation Overview

The backend scoring system is operated behind the NIST firewall. It is utilizing a Cloud Cluster (OpenStack or AWS), which allows for dynamic scaling of instances. Depending on the setup, it consists of one or multiple compute instances that have one-directional network access to the frontend web server. To orchestrate computation across instances we use an in-house developed software called “IndusR”. Within each instance, IndusR allows the creation of a set of independent agents that are orchestrated to process each submission independently. In addition to scaling the number of compute instances, the number of IndusR agents can also be scaled to accommodate increased workload during high-demand periods of evaluation. Each set of agents is executing a linear scoring workflow, which is defined to carry out specific steps for each individual task’s phase, using a YAML configuration file. The agents manage the complete life cycle of system output scoring in an automated fashion by orchestrating tasks using a queuing system. Each submission is processed by executing a set of scoring pipeline steps. Each scoring step is expressed as a user-land script,

which can be implemented and leveraged in any programming language accessible on the instance. The pipeline either proceeds after each successful step or stops on a failed step.

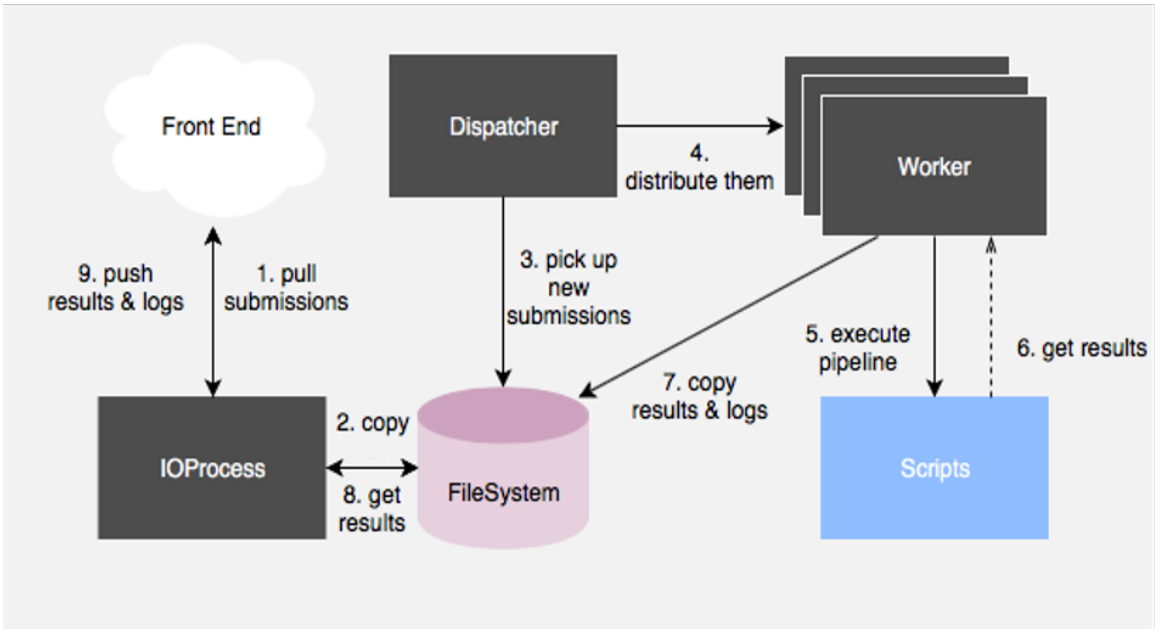

Figure 4. The backend scoring system is a set of independently operating agents. The set of agents, as well as the hosting instances, can be scaled dynamically to match demand.

4.2.2. Scoring Process

Each scoring pipeline is configured individually per phase, but across tasks, the workflow can be generically described as follows:

- Pull scoring run, submission, and metadata from front end
- Setup scoring job and scoring parameters
- Validate submission file existence, syntax, and semantics
- Score system output
- Generate user report
- Ingest scores into scoring database
- Upload results to front end

4.2.3. Text-to-Text Validation and Testing

Each validation script for the Generator and Discriminator tracks ensures the correctness and consistency of system output submissions by verifying structural integrity and format compliance.

Generator Track Validation
The key validation steps for the Generator system outputs include:

- XML format validation: Uses xmllint to check if the submission file is well-formed and conforms to the DTD schema

- Topic ID validation: Ensures that each GeneratorTopicResult has a unique topic attribute and matches predefined topics from the reference .sgml file
- Elapsed time check: Verifies that the elapsedTime attribute contains only numeric values
- Word count enforcement: Ensures that each topic summary:
  - Is not empty
  - Does not exceed the maximum word count limit (250 words)
- Missing topic detection: Checks that all required topics from the reference .sgml file are included in the submission

Discriminator Track Validation
The key validation steps for the Discriminator system outputs include:

- Column integrity check: Ensures that all required columns (DatasetID, TaskID, DiscriminatorID, TaskID, DiscriminatorID, ModelVersion, FileID, ConfidenceScore) are present and correctly formatted.
- TaskID verification: Confirms that all entries belong to the 'detection' task to maintain consistency.
- Model version consistency: Checks that only one unique ModelVersion exists in the submission.
- DatasetID matching: Ensures that DatasetID values in the submission match those in the reference index file.
- FileID coverage: Verifies that all files listed in the index are accounted for in the submission.
- Confidence score range: Ensures that scores are valid floating-point values between 0 and 1.
- Score variability check: Detects cases where all detection scores are identical, which may indicate an issue with model predictions.
- Index column detection: Flags any unintended index columns that may have been included.

Testing Scoring System
To ensure the reliability and robustness of our scoring system, we implemented a structured testing framework that includes the following:

- Unit testing for the scorer: We developed automated unit tests to validate individual components of the scoring software, ensuring correctness in calculations and adherence to expected behavior.
- Test submissions: We generated a diverse set of valid and invalid test submissions to assess the scorer's ability to handle different input scenarios, including edge cases and incorrect formats.
- We performed continuous updates and refinement: Regular update of the testing suite to align with software modifications, ensuring that changes in the scoring system or platform are consistently validated before deployment.

By continuously refining our tests alongside software updates, we maintain evaluation accuracy and catch potential issues early in development.

4.2.4. Text-to-Text Scoring Software

Generator Track Scoring Software

The scoring software for the Generator track is responsible for evaluating the quality, content appropriateness, and overall reliability of AI-generated summaries. It follows a structured pipeline to ensure consistency and fairness in evaluation while integrating necessary preprocessing, validation, and scoring steps. The key components of the scoring pipeline are as follows:

- Parsing System Output Submissions: The software first processes and extracts the necessary information from system submissions. Submissions are expected in a predefined format, and the parser ensures correct structure, extracting relevant fields such as system-generated summaries, metadata, and any additional information provided by participants.
- Post-processing of Summary Text Files: Once parsed, the system output undergoes post-processing to standardize the format and clean the text. This step includes removing extraneous whitespace, normalizing encoding issues, and preparing the summaries for subsequent validation and scoring.
- Sanity Check and Toxicity Filtering: To ensure content appropriateness, each generated summary is analyzed for potential harmful or toxic content. This is done using:
  - Detoxify: A deep learning-based toxicity classifier that identifies harmful or offensive language.
  - Toxin: An additional filtering mechanism to cross-check and mitigate toxicity risks.

  Summaries flagged for toxicity are either filtered out or issued warnings to notify participants of potential concerns.
- Quality Evaluation Against Source Articles: The core scoring function computes the quality of AI-generated summaries by comparing them against the original source articles. The methodology and quality metrics used for this comparison are detailed in Section 6.2. These evaluations provide insights into fluency, informativeness, coherence, and factual consistency.
- Baseline Detector Performance Measurement: To provide a comparative benchmark, the scoring software also computes the performance of the NIST baseline detector on the provided generator. This includes:
  - AUC (Area Under the Curve): Measures the classifier's ability to distinguish between AI-generated and human-authored summaries.
  - Brier Score: Evaluates the accuracy of the baseline detector's probabilistic predictions.

  These scores are displayed on the G-leaderboard to provide participants with a reference benchmark. The details of Generator features and performance metrics are discussed in Section 6.

- Summary Packaging for Discriminator Track: The final stage involves preparing the generated summaries for use in the Discriminator track. The processed summaries are structured, formatted, and packaged before being delivered to Discriminator track participants, ensuring that they are correctly labeled and meet the required specifications.

Discriminator Track Scoring Software

The Discriminator track scoring software evaluates the effectiveness of AI-generated text detection systems by analyzing their detection scores and computing performance metrics. The scoring process integrates multiple inputs and generates statistical evaluations to assess the accuracy and reliability of each submission.

- Input and Data Sources: The scorer processes the following inputs:
  - Summaries categorized into three groups:
    - Human-authored summaries
    - NIST-generated AI summaries
    - AI-generated summaries from Generator track participants (G-participants)
  - A list of detection scores assigned by the Discriminator system for each summary
  - Index and reference files containing ground-truth labels and metadata to validate scoring results.
- Computation of AUC and Brier Scores: For each Discriminator submission, the scorer evaluates detection performance using AUC and Brier Score.
- Sub-score Computation for G-submissions: In addition to overall performance, the scorer calculates sub-scores for each individual Generator submission. This allows a breakdown of how well the Discriminator system identifies AI-generated summaries across different Generator systems. These sub-scores contribute to an aggregate evaluation of Discriminator performance.
- Visualization and Performance Metrics: To provide a comprehensive performance analysis, the scoring software generates:
  - *ROC (Receiver Operating Characteristic) Curve* – A graphical representation of the trade-off between true positive and false positive rates. See [14].
  - *DET (Detection Error Tradeoff) Curve* – A detailed performance visualization focusing on error rates in detection. See [14].

  These curves are displayed on the platform to help participants assess their system's effectiveness in distinguishing human and AI-generated summaries. The details of Discriminator performance metrics are discussed in Section 6.

4.2.5. Text-to-Text Baseline Models

To establish a performance benchmark for both the Generator and Discriminator tracks, we implemented a set of baseline models using widely recognized generative AI systems.

These baselines provide reference outputs for validating the evaluation pipeline as well as evaluating the effectiveness of participating systems.

Generator Baseline Outputs
For text generation, we employed two large-scale language models to provide baseline outputs:

- GPT-3.5 Turbo: A widely used large language model (LLM) [31] optimized for efficiency and speed, serving as a strong benchmark for AI-generated summaries.
- GPT-4: A more advanced model [20] with improved contextual understanding and coherence, offering a higher-quality baseline for comparison.

These models were used to generate summaries based on the same inputs as participant systems, allowing for direct comparison of output quality, fluency, and informativeness.

Discriminator Baseline Outputs
For AI-generated text detection, we selected two baseline models designed to assess whether a given text was human-written or machine-generated:

- RADAR-Vicuna-7B: A fine-tuned Vicuna model [10] designed for text authenticity detection, using transformer-based representations for classification.
- RoBERTa-base-openai-detector: A RoBERTa-based classifier [18] trained specifically for identifying AI-generated content, providing a robust benchmark for evaluating detection performance.

These models were used to score system output, measuring their effectiveness in distinguishing human-authored and AI-generated text.

By establishing these baselines, we ensure that all submitted systems are evaluated with the reference outputs, offering a clear point of comparison for both generative and discriminative performance. The generator baseline output also provides data for evaluating discriminators in Round 1.

4.3. Administrative and Regulatory Requirements

Evaluations conducted at NIST involve several administrative and regulatory requirements to ensure a smooth, secure, and legally compliant process in which the roles and responsibilities of both NIST and participants are clearly defined and understood by all. The requirements necessary in the context of GenAI are briefly described below. The GenAI team has developed internal best practice documentation to facilitate the process for these requirements in the future.

4.3.1. Registration

To participate in a GenAI evaluation, a participant should register and create a profile on the GenAI website (https://ai-challenges.nist.gov/genai). A login.gov email address should be used. During registration, the participant provides their name, country, affiliation, af-

filiation type, then creates a team or joins an existing one, then registers for the desired task(s).

Registration by a non-US participant may be subject to additional vetting steps before approval for participation is given.

4.3.2. Data Usage Agreements

For both the Generator and the Discriminator tracks, NIST provides data that participants process. The rules governing permissible use of this data are outlined in a data usage agreement (DUA) that participants are required to complete and return before participation is granted. The GenAI DUA was developed in coordination with NIST's Office of Chief Counsel (OCC).

4.3.3. Data Transfer Agreement for Generators

The evaluation structure foresees output provided by Generator participants to be used as evaluation material downstream for Discriminator participants. For this reason, a Data Transfer Agreement (DTA) between NIST and the Generator providing data to NIST is being used. Such a process was developed and is completed in coordination with NIST's Technology Partnerships Office (TPO, https://www.nist.gov/tpo). Completion of the DTA by both the Generator participant and NIST is necessary for participation in the Generator track.

4.3.4. Human Subjects Research Determination

The entire protocol for the GenAI pilot study was submitted to NIST's Research Protection Office (RPO, https://www.nist.gov/adlp/research-protections-office) with the title "Generative AI Challenge" and assigned number ITL-2023-0644. It received a determination of exempt human subjects research.

4.3.5. Software Usage Approvals

Any non-NIST software or tools used by NIST to generate evaluation data should be either approved and available at NIST already, or go through an approval process that includes:

- A review of the software's terms of service or license, and a potential requirement to enter into a terms of service addendum that the provider agrees to after negotiation. This process is done in coordination with legal experts at NIST's OCC.
- An IT security approval for use of a Low-Risk Internet Service (LRIS) for the use case of GenAI. This process is done in coordination with NIST's Office of Information Systems Management (OISM, https://www.nist.gov/oism).

5. Tasks

The primary goal of the pilot GenAI evaluations is to understand system behavior for detecting AI-generated versus human-generated content. This includes characteristics of un-

detectable AI-generated content, how human content differs from AI content, and how the conclusions of the task can provide guidance to end users to help differentiate between the two types of content they may encounter regularly. This pilot evaluation does not address the differentiation between "factual" and "non-factual" semantic content; however, this remains a potential topic of interest for future challenge problems.

5.1. Text-to-Text Generators (T2T-G)

Teams participating in the Text-to-Text Generators (T2T-G) task were given the following set of instructions for their task:

> Given a topic and a set of about 25 relevant documents as input, create from the documents a brief, well-organized, fluent summary output which answers the need for information expressed in the topic statement. NIST human assessors developed topics of interest. Each assessor created a topic and chose a set of 25 documents relevant to the topic. The testing dataset documents came from a corpus comprising multiple newswire articles from the website https://duc.nist.gov/. G-participants should assume that the target audience of the summary is a supervisory information analyst who needs the summary to inform decision-making.
>
> - All processing of documents and generation of summaries should be automatic.
> - The summary can be no longer than 250 words (whitespace-delimited tokens). Submissions with summaries longer than 250 words will not be accepted by the G-validator.
> - No bonus will be given for creating a shorter summary.
> - No specific formatting other than linear is allowed (i.e., plain text).
>
> There will be about 45 topics in the test data for generator teams. This set of summaries from all generator teams will serve as the testing data for discriminator teams, who will work on detecting whether the written content is human-generated or AI-generated. The summary output will be evaluated by determining how easy or difficult it is to discriminate AI-generated summaries from human-generated summaries, i.e., the goal of generators is to output a summary that is indistinguishable from human-generated summaries.

5.2. Text-to-Text Discriminator (T2T-D)

The Discriminators received testsets consisting of AI-generated and human-generated summaries but did not receive the source articles. Their task was to detect if a target text summary was generated using large language models (LLMs) such as ChatGPT or was written by a human. For each T2T-D trial consisting of a single summary, the T2T-D detection system should render a detection score (a real number between 0 and 1), with higher numbers indicating the target text summary is more likely to have been generated using LLM-based models. The primary metric for measuring detection performance is the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) as described in Section 6.

6. Performance Metrics

6.1. Discriminator Metrics

This section describes the metrics that are used for measuring the Discriminator system's performance.

6.1.1. Receiver Operating Characteristic (ROC)

The receiver operating characteristic (ROC) curve is a graphical visualization of discriminator performance across all detector score thresholds. Macmillan and Creelman [13] provides detailed information about ROC curves for system evaluation. The curve can be drawn by calculating and plotting the true positive rate (TPR) and false positive rate (FPR) at different thresholds. In what follows,

- TP stands for True Positive (those correctly detected as AI-generated),
- FN stands for False Negative (those incorrectly detected as human-generated),
- FP stands for False Positive (those incorrectly detected as AI-generated), and
- TN stands for True Negative (those correctly detected as human-generated).

The vertical axis is the True Positive Rate (TPR), where TPR=TP/(TP+FN), and the horizontal axis is the False Positive Rate (FPR), where FPR=FP/(TN+FP), which is also known as False Acceptance Rate or False Alarm Rate.

6.1.2. Area Under the ROC curve (AUC)

The area under the ROC curve (AUC) is a score metric for the detection system. The AUC score quantifies the overall ability of a system to discriminate between two classes. The AUC value of a system output is a number between 0 and 1. A system no better at identifying true positives than random guessing has an AUC of 0.5. A perfect system (no false positives or negatives) has an AUC of 1. Partial AUC (pAUC) is AUC at a specified False Positive Rate (FPR).

6.1.3. True Positive Rate (TPR) at False Positive Rate (FPR)

Another score metric used for the detection system is True Positive Rate (TPR) rate at a specified False Positive Rate (FPR), abbreviated as TPR@FPR=x. In our evaluations we use TPR@FPR=0.1.

6.1.4. Detection Error Tradeoff (DET) and Equal Error Rate (EER)

The Detection Error Tradeoff (DET) curve is used as one of the graphical performance analysis tools [15]. The horizontal axis is the False Positive Rate (FPR), and the vertical axis is the False Negative Rate (FNR). [14] provides detailed information about DET curves for detection system evaluation. Equal Error Rate (EER) is the point at which the False Positive Rate (FPR) and False Negative Rate (FNR) are equal.

6.1.5. Brier Score

The Brier Score (BS) can be thought of as a cost function that measures how far system predictions are from the true values using ground-truth known data. It measures the mean square error between the predicted probability p_i assigned to the possible outcomes for an event i and the actual outcome o_i:

$$BS = \frac{1}{n}\sum_{i=1}^{n}(p_i - o_i)^2$$

where p_i is the predicted probability of occurrence of the event, and o_i is equal to 1 if the event occurred (target) and 0 if it did not (nontarget). This definition assumes that the prior probability that event i is AI-generated is 0.5 for each i. If this prior probability is known to be λ, the definition remains valid as long as the test examples are chosen "randomly" from the population of examples of which a proportion λ is AI-generated. If the number of AI-generated examples is n_1 and the number of human-generated examples is n_0 in the testset, where n_1 and n_0 are selected by the experimenter, then the definition of BS requires the following modification:

$$BS = \frac{\lambda}{n_1}\sum_{i=1}^{n_1}(p_i - 1)^2 + \frac{1-\lambda}{n_0}\sum_{i=1}^{n_0}(p_i - 0)^2.$$

BS is the sum of two terms. The first term captures the system performance on a set of examples known to be targets (i.e., AI-generated). This term may be labeled 'BrierT' (Brier score from Targets only). The second term captures system performance on a set of examples known to be nontargets (i.e., human-generated). This term may be labeled 'BrierN' (Brier score from nontargets only). Thus,

$$\text{BrierT} = \frac{1}{n_1}\sum_{i=1}^{n_1}(p_i - 1)^2$$

and

$$\text{BrierN} = \frac{1}{n_0}\sum_{i=1}^{n_0}(p_i - 0)^2.$$

Consequently, we have

$$BS = \lambda\ \text{BrierT} + (1-\lambda)\ \text{BrierN}.$$

In practice, λ is seldom known and it is customary to use $\lambda = 1/2$ in the definition of BS. In this case, BS is simply the average of BrierT and BrierN.

The goal for discriminator systems would be to get a high AUC score (closer to 1 is better) and low BrierT scores (closer to zero is better) against all participating generator systems and low BrierN scores for human-generated content.

6.2. Generators Metrics

There are two distinct ways in which the success of generators can be measured.

1. The system's ability to generate content that is indistinguishable from human-generated content. This is typically accomplished by training detectors to assign scores to AI-generated content such that the score distribution for AI-generated content is statistically the same as the score distribution for human-generated content. For this scenario, the system AUC score should be around 0.5.
2. The system's ability to generate content that can mislead human and/or AI discriminators to claim the content is human-generated. This is typically accomplished by training detectors to assign lower detection scores to AI-generated content than to human-generated content. In this scenario, the AUC score for the system will be between 0 and 0.5. Also, BrierT scores would be closer to 1; that is, detection scores would be closer to zero.

Thus, the goal for generator systems could be either (a) to achieve a AUC score close to 0.5 (for human-like performance) or (b) to drive down the discriminator AUC score (below or equal to 0.5 is better) and drive up the discriminator BrierT scores for AI-generated content against all participating discriminator systems (for misleading detector systems into classifying AI generated content as human generated and vice versa).

In some situations, we will also use a metric we call 'BrierMax', which is the maximum of BrierT and BrierN. Since the goal of discriminators is to drive down both BrierT and BrierN, a lower BrierMax score is better for discriminators.

After each round of the GenAI evaluation process, we report AUC and Brier scores for Generators and AUC, EER, Brier, AUC@FPR=0.1, TPR@FPR=0.1, and TNR@FNR=0.1 for Discriminators in our leaderboard. We also have metrics of BrierT, BrierN, and other generator content quality measures, which are described in Section 6. To provide a comprehensive understanding of the system's performance, we used AUC, BrierT, and BrierN scores for following data analysis.

6.2.1. Assessment of the Quality of AI-generated Summaries

As stated in Section 2.1, our main interest is in evaluating the ability of humans and state-of-the-art (SOTA) algorithms to discriminate between AI-generated summaries and human-generated summaries. However, there are many metrics available in the literature that attempt to evaluate the quality of the summary generated by AI or by a human. This evaluation is done by comparing the summaries with the source documents. This evaluation can also be performed by comparing the selected features of AI-generated summaries with the same features of human-generated summaries based on various attributes, including those that arise in the context of natural language processing. The metrics listed below have been proposed in the literature for such assessments. A subset of the following automatic G-metrics will be adopted to evaluate G-participants' data outputs:

- **Syntactic Evaluation**: Automatic analysis of syntactic complexity. https://www.benjamins.com/catalog/ijcl.15.4.02lu
- **BERTScore (Bidirectional Encoder Representations from Transformers Score)**: Similarity score for each token in the candidate sentence with each token in the reference sentence using contextual embeddings. https://arxiv.org/pdf/1904.09675.pdf
- **BLEU (BiLingual Evaluation Understudy)**: A metric based on n-gram overlap designed for machine translation evaluation. This score is based on the idea of 'modified word n-gram counts" when quantifying 'precision" (the number of words in a candidate sentence that appear in a reference sentence). The BLEU score is a geometric average of modified n-gram counts for $n = 1, 2, ..., N$ (N pre-assigned) penalized by what the authors call a "brevity penalty" (BP). https://aclanthology.org/P02-1040.pdf.
- **METEOR (Metric for Evaluation of Translation with Explicit ORdering)**: A machine translation evaluation metric based on a generalized concept of unigram matching between the machine and human reference translations. This uses an alignment process between a reference document and a candidate document and then calculates a "penalized" F-score, where the F-score is a weighted harmonic mean of precision and recall. Unlike BLEU, METEOR uses not only exact matches, but also stemmed matches, synonym matches, and paraphrase matches to improve flexibility. url https://aclanthology.org/W05-0909.pdf.
- **CHRF (Character n-gram F-score)**: A metric based on character n-gram F-score. https://www.statmt.org/wmt17/pdf/WMT70.pdf.
- **SummaQA**: A metric based on Question Answering. https://aclanthology.org/D19-1320.pdf
- **SUPERT (SUmmarization evaluation with Pseudoreferences and bERT)**: A metric based on selected salient sentences from the source documents, using contextualized embeddings and soft token alignment techniques. This metric measures the relevance of a summary in two steps: (i) identifying the salient information in the input documents to build a pseudo reference summary, and (ii) measuring the semantic overlap between the pseudo reference and the summary to be evaluated. https://aclanthology.org/2020.acl-main.124.pdf
- **BLANC (Bacronymic Language model Approach for summary quality estimatioN)**: A measure of the performance boost gained by a pre-trained language model with access to a document summary while carrying out its language understanding task on the document's text. https://arxiv.org/pdf/2002.09836.pdf
- **Misc. statistics (extractiveness, novel n-grams, repetition, length)**: https://aclanthology.org/N18-1065/

We intend to use these metrics to investigate the possibility of building an automatic classifier to discriminate between AI-generated summaries and human-generated summaries. We will investigate the possibility of a fusion metric that is a good discriminator. Discrim-

ination capability will be assessed using ROC curves (or DET curves) constructed from our empirical data.

7. Data and Submissions

7.1. Data

Generators are evaluated on two rounds of submissions, while Discriminators are evaluated on three.

In the first round of submissions from the generators (referred to as G-round-1), the G-participants were provided with 10 topics below selected from a total of 45 topics, along with a set of 25 relevant documents for each topic.

```
G-round-1 topics:
"topic_3130", "topic_3693", "topic_4745", "topic_4803", "topic_4936",
"topic_5089", "topic_5246", "topic_5760", "topic_6609", "topic_9024".
```

In G-round-2, the G-participants were provided with all 45 topics, which included the additional 35 topics that were not provided in the first round. The additional 35 topics are shown below.

```
G-round-2 topics:
"topic_0421", "topic_0923", "topic_1350", "topic_1526", "topic_2220",
"topic_2438", "topic_2840", "topic_3674", "topic_4283", "topic_4322",
"topic_4380", "topic_4600", "topic_4786", "topic_4793", "topic_5007",
"topic_5138", "topic_5567", "topic_5611", "topic_5940", "topic_6728",
"topic_7551", "topic_7793", "topic_7798", "topic_7834", "topic_8208",
"topic_8292", "topic_8723", "topic_8944", "topic_8976", "topic_9102",
"topic_9170", "topic_9182", "topic_9430", "topic_9494", "topic_9856".
```

For evaluating discriminators' system across the three rounds, three testsets of summaries were created which we refer to as testset-1, testset-2, and testset-3.

Testset-1 consisted of AI summaries generated by the NIST GenAI team (NIST baseline generators: NIST-gpt3.5 and NIST-gpt4) on 10 topics out of a total of 45 available topics and used for the first round of discriminator evaluation (D-round-1). The 10 topics for D-round-1 are the same as those from G-round-1.

```
D-round-1 topics:
"topic_3130", "topic_3693", "topic_4745", "topic_4803", "topic_4936",
"topic_5089", "topic_5246", "topic_5760", "topic_6609", "topic_9024".
```

For each topic, 4 variations of AI-summaries were created. NIST-gpt3.5 could only create AI summaries for 6 out of the chosen 10 topics (`topic_3130`, `topic_4745`, `topic_4936`, `topic_5246`, `topic_6609`, `topic_9024`) due to token limitations. Four variations of human summaries (by 4 different human summarizers) per topic were also included for these 10 topics. Altogether, there were 104 summaries in testset-1 (24 AI summaries from NIST-gpt35, 40 AI summaries from NIST-gpt4, and 40 human summaries).

Testset-2 consisted of AI summaries generated by G-participants on the 10 topics considered for testset-1, along with testset-1 itself. Each submission from each G-participant contained only one summary per topic (no variations). The NIST GenAI team considered 25 topics (10 topics from testset-1 and 15 additional topics) for which AI summaries were created with 4 variations for each considered topic. The 15 additional topics considered by NIST-gpt4 were

D-round-2 topics:
"topic_1350", "topic_2840", "topic_3674", "topic_4283", "topic_4322",
"topic_4380", "topic_4793", "topic_5007", "topic_5567", "topic_5611",
"topic_8208", "topic_8292", "topic_8723", "topic_9430", "topic_9494".

NIST-gpt3.5 could only create AI summaries for 9 of these additional 15 topics (shown below) due to a token limitation.

"topic_1350", "topic_3674", "topic_4283", "topic_4380", "topic_5007",
"topic_5567", "topic_5611", "topic_8292", "topic_8723".

Four variations of human summaries (by 4 different human summarizers) per topic were also included in this testset. Altogether, there were 530 summaries in testset-2 (270 AI summaries from external G-participants, 60 AI summaries from NIST-gpt35, 100 AI summaries from NIST-gpt4, and 100 human summaries).

Testset-3 was constructed using the 270 AI summaries from the external G-participants from testset-2 and an additional 945 AI summaries (1 AI summary per topic for each of the 45 topics by each of 21 G-submissions). NIST-gpt35 generated 108 AI summaries (4 variations per topic for 27 topics – due to a token limitation, only 27 out of the 45 topics could be processed by NIST-gpt35). These 27 topics are listed below.

"topic_0421", "topic_0923", "topic_1350", "topic_2220", "topic_3130",
"topic_3674", "topic_4283", "topic_4380", "topic_4600", "topic_4745",
"topic_4936", "topic_5007", "topic_5138", "topic_5246", "topic_5567",
"topic_5611", "topic_5940", "topic_6609", "topic_7551", "topic_7798",
"topic_8292", "topic_8723", "topic_8944", "topic_8976", "topic_9024",
"topic_9102", "topic_9856".

NIST-gpt4 generated 180 AI summaries (4 variations per topic for all 45 topics). A total of 180 human summaries (4 different human summarizers per topic for all 45 topics) were also included in this testset. This resulted in a total of 1683 summaries (1503 AI summaries and 180 Human summaries) in testset-3. The full list of 45 topics is shown below.

D-round-3 topics:
"topic_0421", "topic_0923", "topic_1350", "topic_1526", "topic_2220",
"topic_2438", "topic_2840", "topic_3130", "topic_3674", "topic_3693",
"topic_4283", "topic_4322", "topic_4380", "topic_4600", "topic_4745",
"topic_4786", "topic_4793", "topic_4803", "topic_4936", "topic_5007",
"topic_5089", "topic_5138", "topic_5246", "topic_5567", "topic_5611",
"topic_5760", "topic_5940", "topic_6609", "topic_6728", "topic_7551",
"topic_7793", "topic_7798", "topic_7834", "topic_8208", "topic_8292",

"topic_8723", "topic_8944", "topic_8976", "topic_9024", "topic_9102", "topic_9170", "topic_9182", "topic_9430", "topic_9494", "topic_9856".

7.2. Submissions

A total of 172 entities from 13 different countries initially registered for the evaluation. However, a total of 6 G-teams and 11 D-teams from 14 organizations, which included academia, industry, and government, ultimately participated in the first pilot study of GenAI T2T. All participant information, including their institution and country details, was self-reported.

7.2.1. Generators

Table 1 shows which G-teams participated in G-round-1 and G-round-2. For generators, a total of 48 valid submissions (without duplicates) were received, 27 for G-round-1 and 21 for G-round-2. Figure 5 illustrates the submission count per team for each round for generators.

Table 1. G-teams participation summary. Six teams participated in G-round-1. Of these 6, only 3 teams participated in G-round-2.

G-Team	G-round-1	G-round-2
0782f	Yes	No
0dea0	Yes	No
6fc49	Yes	Yes
804fe	Yes	Yes
87a8c	Yes	Yes
aa872	Yes	No

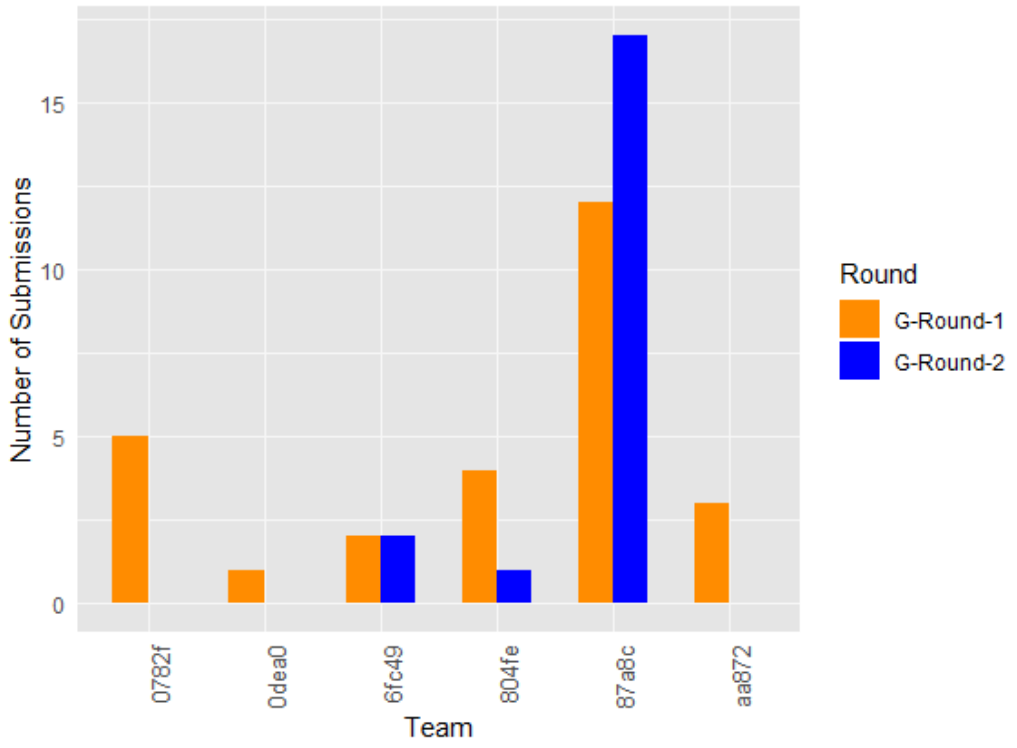

Figure 5. GenAI T2T G-submission count per team

In G-round-1, there were six G-participants identified using the labels ("0782f", "0dea0", "6fc49", "804fe", "87a8c", "aa872"). In addition, there were two NIST baseline generators: `NIST-gpt35` and `NIST-gpt4`. The G-participants list and their submission IDs are given in Table 10 in Appendix-A.

In G-round-1, the G-participants were given 10 topics (the same topics as in Discriminator round-1). Not counting the NIST baseline submissions, there were a total of 27 submissions, each submission consisting of one AI summary for each of the 10 G-Round-1 topics. Thus, a total of 10 summaries were presented as part of each submission by each G-participant. In addition, NIST-gpt3.5 and NIST-gpt4 each submitted 4 summaries per topic. These two G-baselines considered 25 topics (10 from Discriminator round-1 and 15 new topics), of which GPT3.5 could only generate summaries for 15 topics due to token limitations.

The topics given to GPT4 are shown below.

```
Testset-1 topics
 "topic_3130", "topic_3693", "topic_4745", "topic_4803", "topic_4936",
 "topic_5089", "topic_5246", "topic_5760", "topic_6609", "topic_9024"

Testset-2 topics
 "topic_1350", "topic_2840", "topic_3674", "topic_4283", "topic_4322",
```

"topic_4380", "topic_4793", "topic_5007", "topic_5567", "topic_5611",
"topic_8208", "topic_8292", "topic_8723", "topic_9430", "topic_9494"

GPT3.5 used only the following subset of 15 (out of the 25) topics.

Testset-1 topics
"topic_3130", "topic_4745", "topic_4936", "topic_5246", "topic_6609",
"topic_9024".

Testset-2 topics
"topic_1350", "topic_3674", "topic_4283", "topic_4380", "topic_5007",
"topic_5567", "topic_5611", "topic_8292", "topic_8723".

In G-round-2, there were 3 G-participants and two NIST baseline Generators (NIST-gpt3.5 and NIST-gpt4). There were a total of 21 G-participant submissions. In addition, NIST-gpt4 submitted 180 AI summaries (45 topics with 4 summaries per topic) and NIST-gpt35 submitted 108 AI summaries (4 summaries for each of 27 of the 45 topics; token limits prevented NIST-gpt3.5 from producing summaries for the remaining 18 topics). See Table 12 in Appendix-A for the details.

7.2.2. Discriminators

Table 2 shows which D-teams participated in D-round-1, D-round-2 and D-round-3.

Table 2. D-teams participation summary. 11 teams participated in D-round-1. Of these 11, only 8 teams participated in D-round-2. Of these 8, only 6 participated in D-round-3. In addition to these, NIST made a submission using a baseline, but this is not shown in this table, but it is shown in Figure 6

D-Team	D-round-1	D-round-2	D-round-3
0dea0	Yes	No	No
18126	Yes	Yes	Yes
29d48	Yes	Yes	Yes
6655b	Yes	Yes	Yes
6fc49	Yes	Yes	No
804fe	Yes	Yes	Yes
87a8c	Yes	Yes	Yes
993ad	Yes	No	No
9de37	Yes	Yes	No
b3cd9	Yes	Yes	Yes
d718e	Yes	No	No

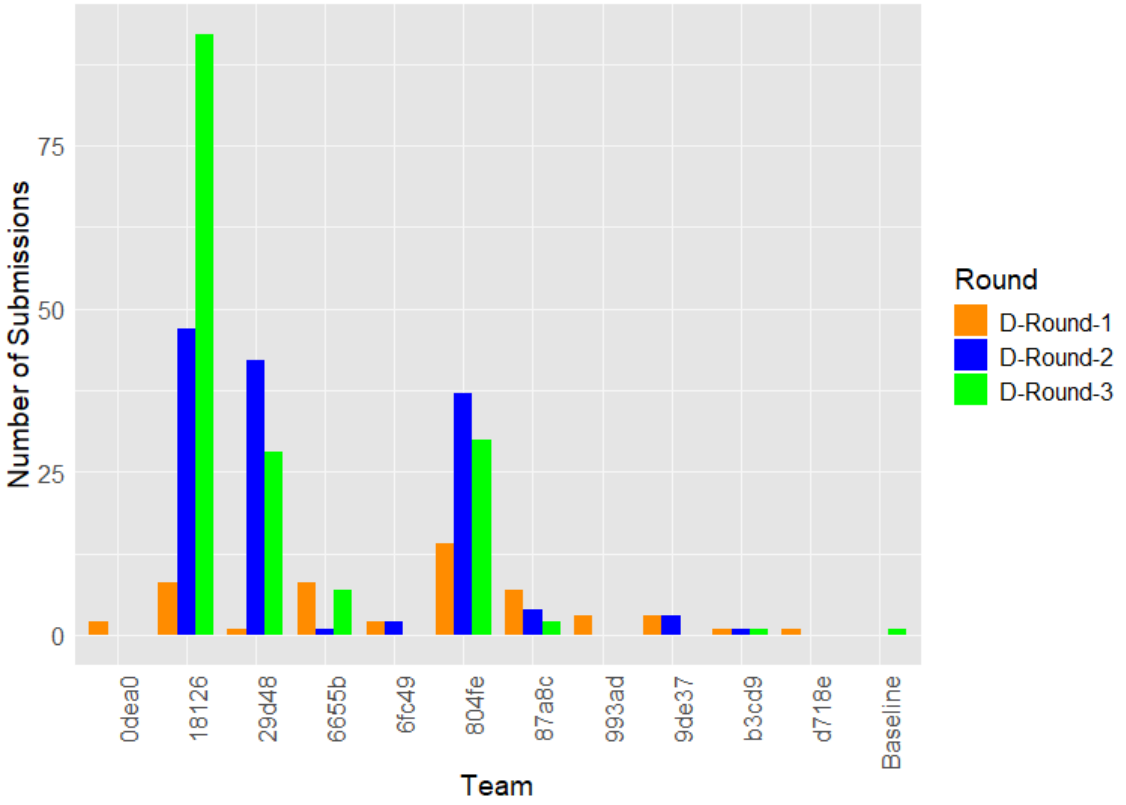

Figure 6. GenAI T2T D-submission count per team

For discriminators, a total of 348 valid submissions (without duplicates) were received, with 50 for the D-round-1, 137 for the D-round-2, and 161 for the D-round-3 (note: one of these 161 submissions is from a NIST baseline algorithm). Figure 6 illustrates the submission count per team for each round for discriminators, respectively.

In D-round-1, the Testset-1 data was used to evaluate the D-participants. Each of the 11 D-participants submitted a variable number of evaluations of the testset. In D-round-2, the total number of AI summaries used is 430 (10 from each of the 27 G-submissions + 100 from GPT4 + 60 from GPT3.5). A total number of human summaries used in this round is 100 (4 human summaries for each of 25 topics). Thus, each D-participant evaluated 530 summaries (430 AI + 100 Human). There were a total of 137 D-submissions from 8 D-participants. The list of participants and their submission IDs is given in Table 11 in Appendix-A. In D-round-3, there were 7 discriminator teams (including the NIST baseline submission) and a total of 161 submissions. Table 13 in Appendix-A gives a list of the Discriminator IDs and their respective submission IDs.

8. Data Analyses and Results

We measure the performance of generators (G) and discriminators (D) using multiple metrics (AUC, BrierT, BrierN). Keep in mind that the D-participants have already trained their systems on their own training data and do not know the ground truth in any of the test sets. An AUC value of 1 indicates that the detection scores for targets (AI-generated) are all higher than any of the detection scores for non-targets (human-generated). However, the AUC metric does not reflect whether or not the detector is properly calibrated. In particular, even if all the detection scores are close to 1, or all the detection scores are close to zero, as long as the detection scores for targets are greater than the detection scores for non-targets, the AUC value will still be 1. However, for the detection scores to be useful for a general user of the detector, it would be advisable to calibrate the system such that the detection scores are as close to one as possible for targets and as close to zero as possible for nontargets. To what extent these goals are met can be quantified by the BrierT scores (for targets) and the BrierN scores (for nontargets). Hence, in addition to reporting the AUC metric, we also report BrierN and BrierT scores to help participants assess how well calibrated their systems are.

Our data analysis is primarily driven by the following main questions. Other questions of secondary interest will be discussed in a future publication.

(1) Which G-submissions perform well against all or most of the D-submissions on each of the metrics (AUC, BrierT, BrierN)?

(2) Which D-submissions perform well against all or most of the G-submissions on each of the metrics (AUC, BrierT, BrierN)?

8.1. Results for Test Set-1 (D-Round 1)

To understand how successful gpt3.5 and gpt4 are in generating summaries that are indistinguishable from human summaries, we calculated AUC values, BrierT scores, and BrierN scores across all 50 D-submissions. The distributions of these metric values are illustrated in Figure 7.

The red points correspond to AUC values for gpt3.5 and the blue points are AUC values for gpt4, both computed across all 50 detector submissions. The AUC values ranged from about 0.4 to 1 for both discriminators. The orange and lavender points are for gpt3.5 and gpt4, respectively, showing the distribution of Brier scores (BrierT) for AI-generated summaries (which we refer to as 'targets'). Recall, D-submissions with BrierT scores close to zero are desired. However, we see some submissions resulting in a BrierT score as high as 1 (the maximum possible value). This means the discriminator was very confident that the summary was human-generated (detector scores close to 0) even though it was generated by AI.

The dark green points show the BrierN scores across all 50 detector submissions for human-generated summaries, which we refer to as "nontargets." There is only one plot for BrierN since this score does not depend on who generated the AI summaries; it is calculated based

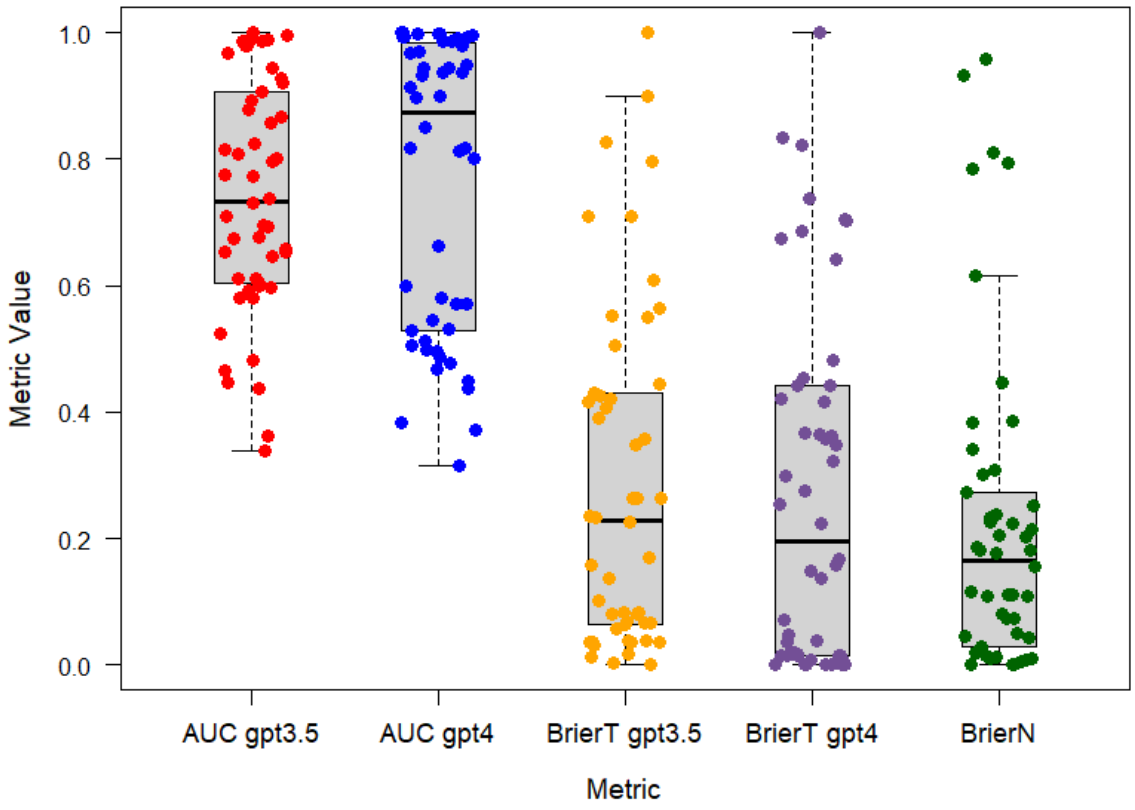

Figure 7. Box plot with superimposed jittered points: D-round-1 comparison of the NIST-gpt3.5 and NIST-gpt4 generators across all discriminators for AUC (targets+nontargets), BrierT (targets), and BrierN (non-targets) scores

on human summaries only. Discriminator performance is good when BrierN score is close to zero. However, we see that there are D-submissions with BrierN scores close to 1. This means the discriminator was very confident (detector scores close to 1) that the summary was generated by AI even though it was human-generated.

The distributions of detector scores (pooled across all D-submissions) for summaries generated by NIST-gpt3.5 and NIST-gpt4 were not statistically different (P-value = 0.1124). Likewise, BrierT scores were also not statistically different at a significance level of 0.05. See Table 3.

Table 3. Kolmogorov-Smirnov test results comparing NIST-gpt3.5 and NIST-gpt4.

Metric	p-value
AUC	0.1124
BrierT	0.06779

A generator may be considered "successful" against discriminators if the AUC values from all, or almost all, discriminators are less than or equal to 0.5. If the AUC value is 0.5, it means that the discriminator is unable to tell the difference between AI-generated and human-generated content, and thus randomly guessing. If the AUC value is less than 0.5, it means that more AI-generated content has detection scores closer to zero than human-generated content, and thus confuses AI-generated content with human-generated content and vice versa. If the AUC for a detector is close to 1, but scores for AI-generated content are close to zero. i.e., highly confident the content is human-generated, then the BrierT scores will be large, indicating that the detector needs to be calibrated.

Figure 8 shows a plot of the AUC values against BrierT scores. The red points are from NIST-gpt3.5, and the blue points are from NIST-gpt4. Each red point represents an evaluation of the AI content produced by NIST-gpt3.5 by a discriminator. Likewise, each blue point corresponds to an evaluation of the AI content from NIST-gpt4 by a discriminator. Successful generators will have low AUC values or high BrierT values against any discriminator. Here we see that several of the blue points and the red points meet this description. See the unshaded region. Thus, generally speaking, gpt3.5 and gpt4 appear to have reasonable success in making many detectors believe the content generated by them are

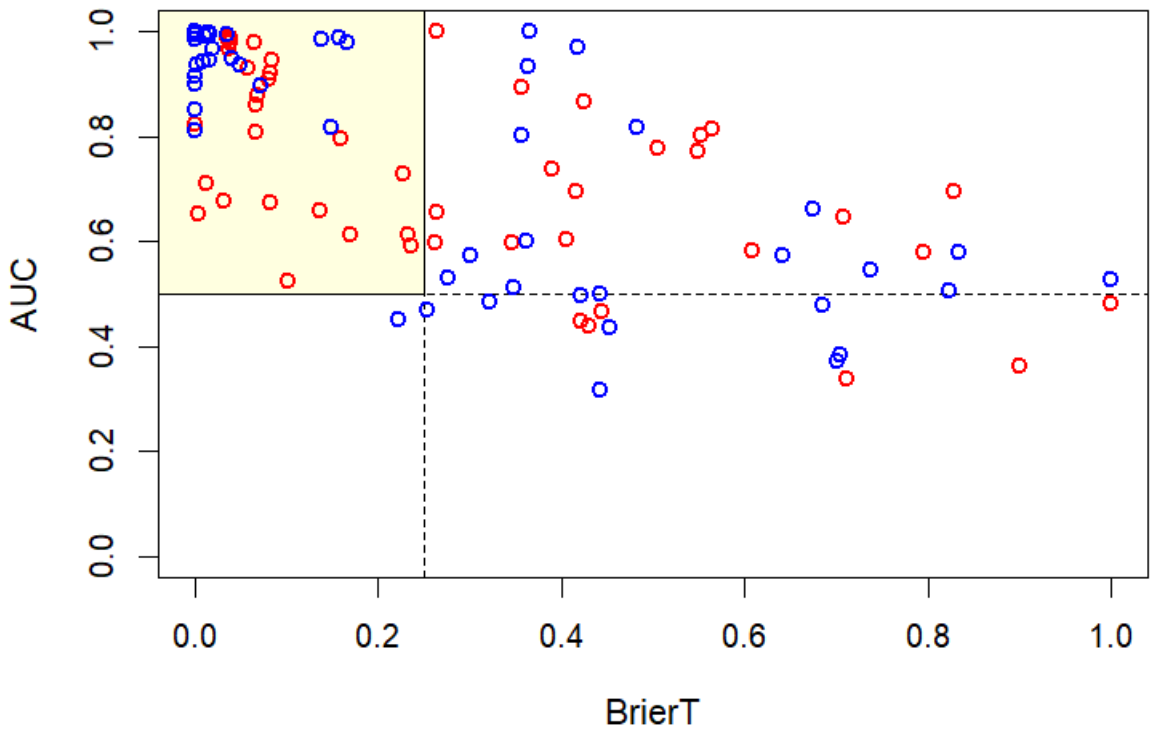

Figure 8. AUC versus BrierT scores for all D-submissions on the NIST-gpt3.5 (red) and NIST-gpt4 (blue) generator data. The yellow shaded region in the top-left corresponds to D-submissions with AUC $\geq$ 0.5 and BrierT score $\leq$ 0.25. The points in the unshaded region correspond to detector submissions that either have low AUC values or have high BrierT scores.

Table 4. Performance metrics for selected subset of G-submission/D-submission pairs where the D-submissions have difficulty reliably recognizing AI-generated content.

G-Participant	D-Submission	D-Participant	AUC	BrierT
NIST-gpt4	13	18126	0.3713	0.7017
NIST-gpt3.5	14	18126	0.3625	0.9003
NIST-gpt3.5	15	6655b	0.4818	1.0000
NIST-gpt3.5	78	87a8c	0.3396	0.7105
NIST-gpt4	78	87a8c	0.3831	0.7043
NIST-gpt4	52	993ad	0.4778	0.6859

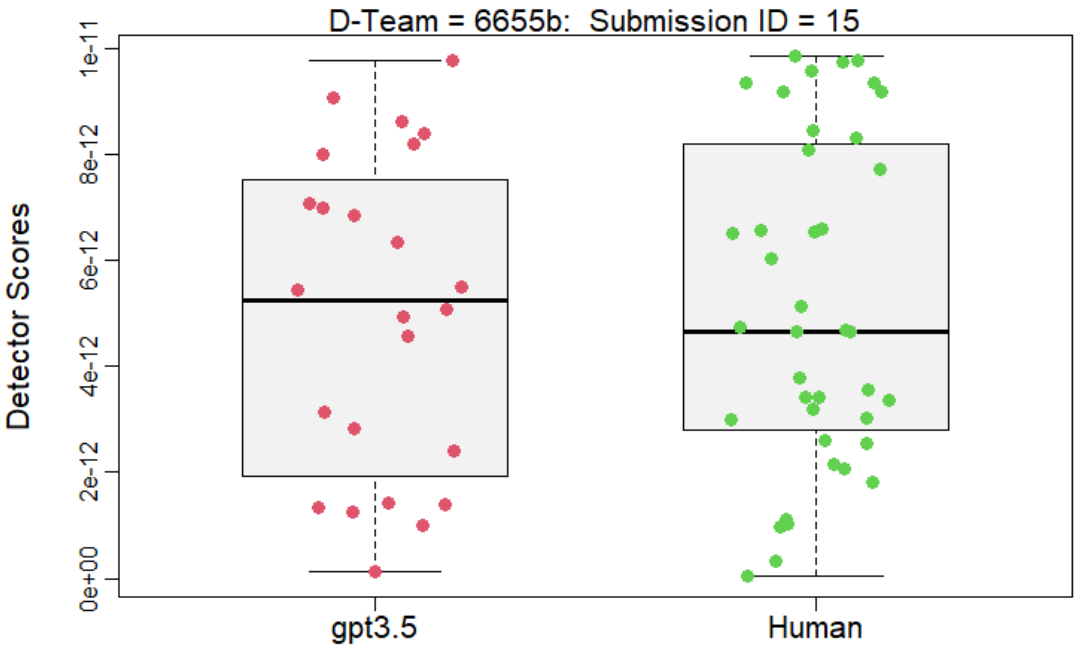

Figure 9. Box plots of detection scores assigned by D-round-1 D-submission (ID: 15) to NIST-gpt3.5 generated summaries and to human-generated summaries. All scores are either zero or very close to zero.

It may be surprising to see a BrierT score of 1 (NIST-gpt3.5/D-Submission 15), but an examination of the submitted scores for both AI-generated and human summaries, we see that the AI-generated summaries receive a detector score of zero or very close to zero (the discriminator thinks the AI-generated content is human-generated), resulting in a BrierT score of 1. See Figure 9.

Figure 10 shows a plot of $\sqrt{BrierN}$ versus $\sqrt{BrierT}$. The submissions with AUC scores equal to or greater than 0.9 are depicted using solid filled red circles (gpt3.5) or solid filled blue circles (gpt4). Open circles have AUC less than 0.9. The size of the circles is proportional to the AUC values. The yellow shaded region represents discriminator submissions that have AUC values greater than or equal to 0.9 and also have BrierT and BrierN scores less than or equal to 0.25 (0.5 in the square root scale).

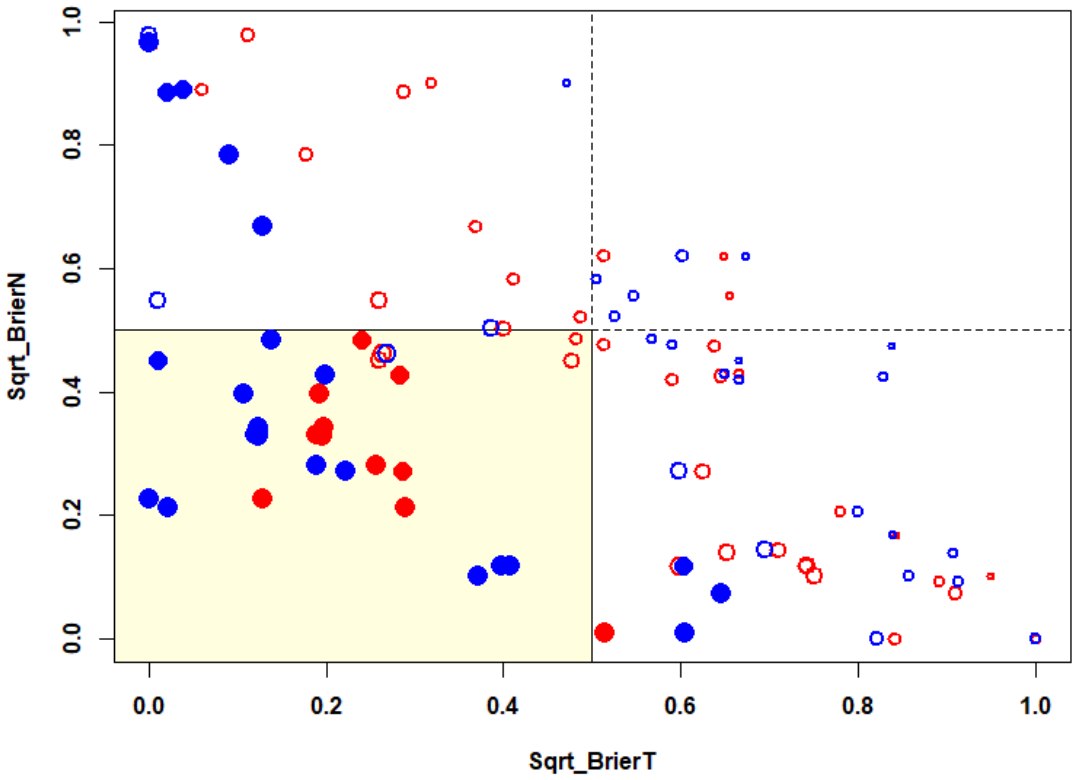

Figure 10. BrierN versus BrierT plotted using a square root scale. Red points correspond to discriminators evaluation of gpt3.5 generated summaries. Blue points correspond to discriminators evaluation of gpt4 summaries. Solid filled circles correspond to discriminator AUC value of 0.9 or higher. Open circles correspond to AUC values of less than 0.9. Smaller the circle the lower is the AUC value. Solid filled circles in the yellow shaded region may be regarded as 'better' discriminators since they have lower BrierT and BrierN scores and a higher than 0.9 AUC.

Performance metrics (AUC, BrierT, and BrierN) for those D-submissions with high AUC values that also appear to be well calibrated (low BrierT and BrierN scores) when evaluated with the reference NIST-gpt3.5 are given in Table 5 and against NIST-gpt4 in Table 6.

Table 5. Performance metrics for selected D-submissions against NIST-gpt3.5

G-Submission	D-Submission	AUC	BrierT	BrierN	D-participant
gpt3.5	12	0.996	0.017	0.051	6655b
gpt3.5	84	0.989	0.036	0.110	804fe
gpt3.5	18	0.988	0.038	0.108	804fe
gpt3.5	86	0.987	0.037	0.110	804fe
gpt3.5	87	0.985	0.037	0.110	804fe
gpt3.5	83	0.979	0.065	0.080	804fe
gpt3.5	85	0.978	0.039	0.117	804fe
gpt3.5	11	0.967	0.037	0.157	804fe
gpt3.5	8	0.944	0.084	0.046	0dea0
gpt3.5	10	0.929	0.057	0.234	804fe
gpt3.5	34	0.921	0.082	0.074	b3cd9
gpt3.5	9	0.907	0.080	0.183	804fe

Table 6. Performance metrics for selected D-submissions against NIST-gpt4

G-Submission	D-Submission	AUC	BrierT	BrierN	D-Participant
gpt4	12	1.000	0.000	0.051	6655b
gpt4	84	0.997	0.014	0.110	804fe
gpt4	86	0.997	0.015	0.110	804fe
gpt4	87	0.997	0.015	0.110	804fe
gpt4	18	0.996	0.015	0.108	804fe
gpt4	85	0.993	0.015	0.117	804fe
gpt4	83	0.993	0.036	0.080	804fe
gpt4	11	0.992	0.011	0.157	804fe
gpt4	25	0.987	0.158	0.014	9de37
gpt4	37	0.986	0.138	0.011	9de37
gpt4	8	0.985	0.000	0.046	0dea0
gpt4	36	0.979	0.166	0.014	9de37
gpt4	10	0.968	0.019	0.234	804fe
gpt4	9	0.948	0.040	0.183	804fe
gpt4	34	0.938	0.049	0.074	b3cd9
gpt4	21	0.900	0.000	0.204	18126

Figure 11 shows BrierN scores for all D-submissions. A well-performing, well-calibrated discriminator would give low scores to human-generated content; hence its BrierN score would be close to zero. It is reasonable to consider submissions with BrierN score above 0.25 to be poorly calibrated. There are a number of submissions that are in the poorly calibrated range. It is possible for a poorly calibrated discriminator to have an AUC value close to 1 or even equal to 1. In such cases, interpreting the "detector score" as a probability would be highly misleading and cause many false positives if, for instance, one uses a threshold of 0.5 for the classification.

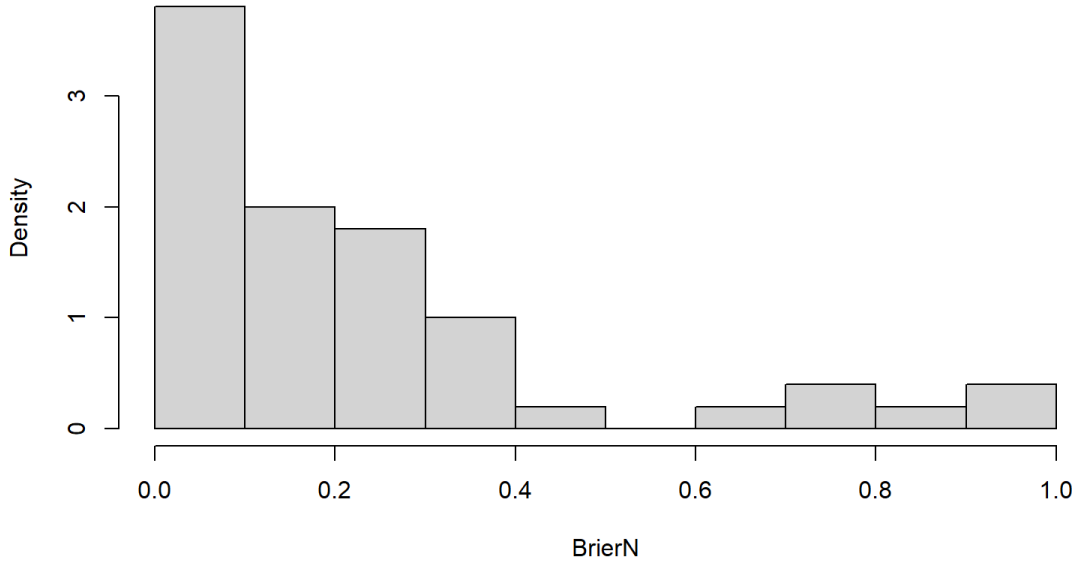

Figure 11. Density Histogram of BrierN scores across all D-submissions

8.2. Results for Test Set-2 (D-Round 2)

To understand how indistinguishable summaries created by the generators are from human summaries, we calculated AUC values and BrierT scores for each G-submission, across all 137 D-submissions.

Figure 12 shows a heatmap of the AUC scores corresponding to each Generator-Discriminator pair, with rows labeled by Discriminator submissions and columns labeled by Generator submissions. Yellow color in a cell indicates that the discriminator is able to discriminate very well against the corresponding generator. Thus, rows that show mostly yellow correspond to discriminator submissions that are performing well (on this metric) against most generators. Conversely, purple color (also dark green and dark blue to a lesser extent) indicates cells where the generator is winning well against the corresponding discriminator. We see that there are some generators that win against most discriminators on the AUC metric.

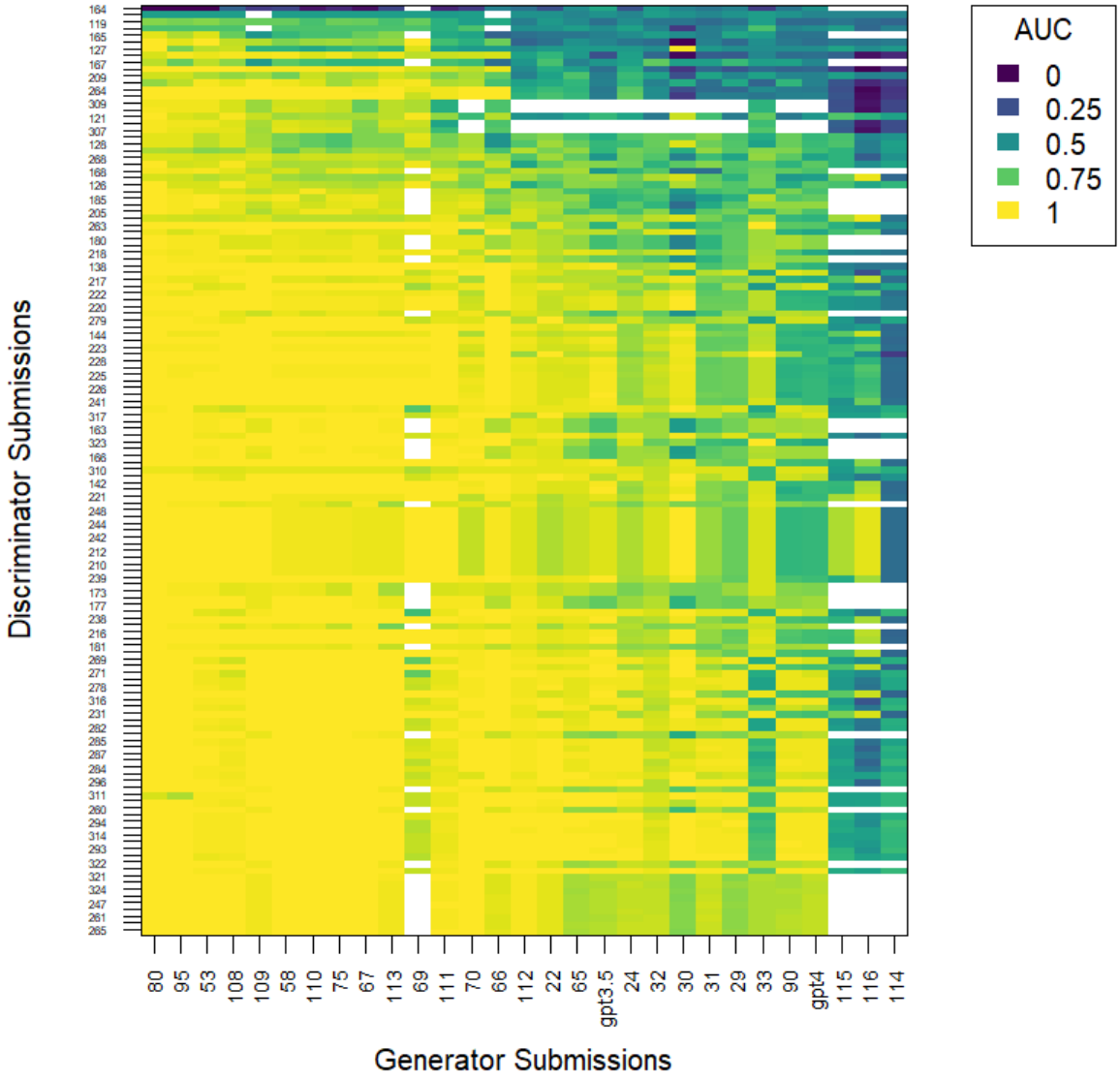

Figure 12. Heatmap of AUC scores corresponding to each Generator and Discriminator pair. Rows are labeled by Discriminator team submissions and columns by Generator team submissions. Yellow cells identify Generator-Discriminator submission pairs where the D-submission wins against the G-submission on the AUC metric. Rows that are mostly yellow indicate D-submissions that perform well against most G-submissions. Columns that are mostly purple, blue, or green indicate generator submissions that confused most of the discriminators. Cells that are colored white correspond to cases where the generator submission and the discriminator submission are both from the same team and hence were not considered in this heatmap.

Figure 13 shows a heatmap of BrierT scores for all Generator-Discriminator combinations. Here a generator does well against a discriminator if the cell color is green, light green, or

yellow (cells with BrierT scores ≥ 0.25). We see that only a few generators do well against most of the discriminators on this metric.

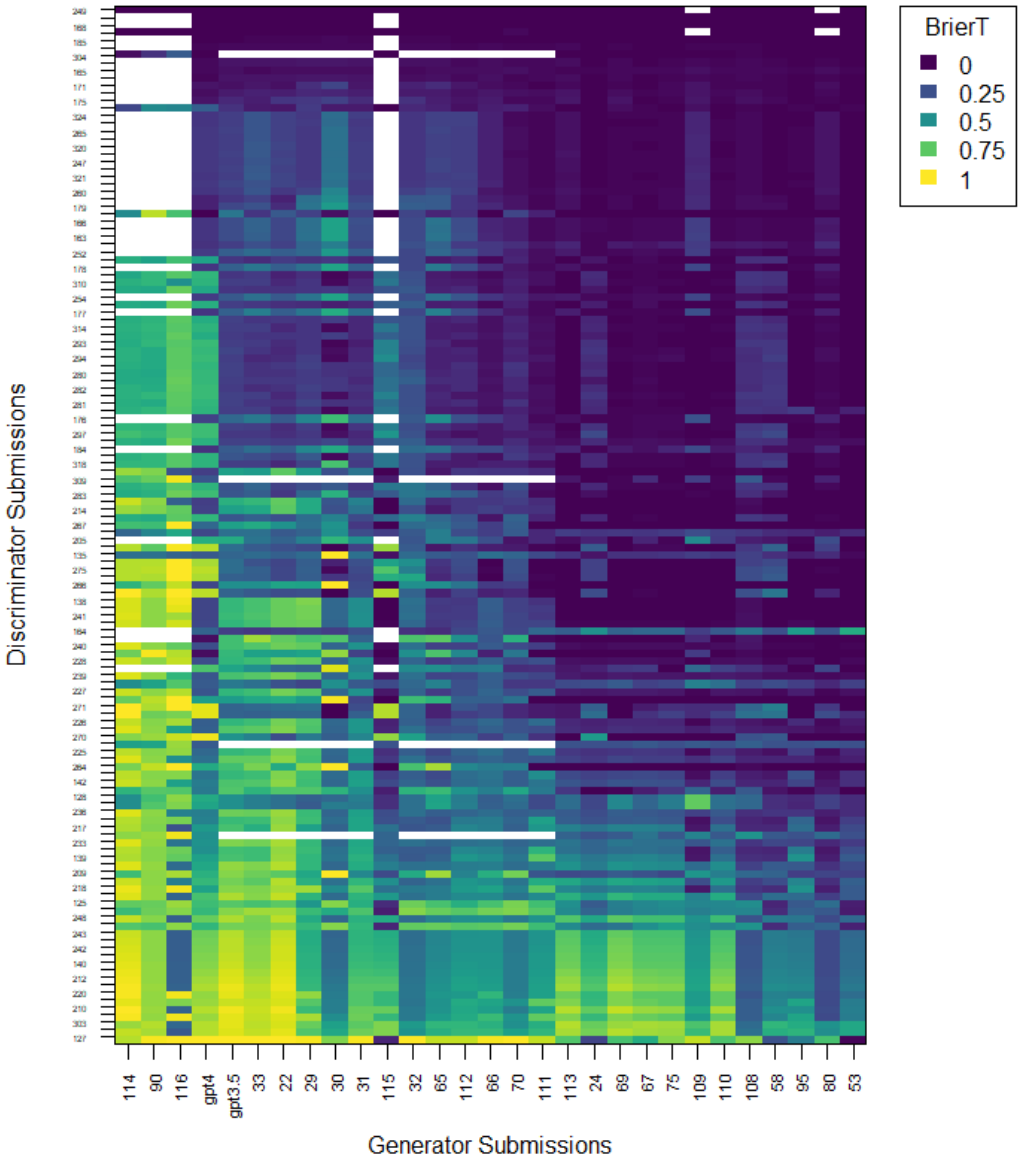

Figure 13. Heatmap of BrierT scores corresponding to each Generator and Discriminator pair. Rows are labeled by Discriminator team submissions and columns by Generator team submissions. Green, light green, or yellow cells identify Generator-Discriminator submission pairs where the D-submissions give a low score to G-submissions, in the range of scores for human-generated content. Rows that are mostly purple indicate D-submissions that perform well against most G-submissions. Columns that are mostly yellow, light green, or green correspond to generators that perform well against most discriminators. White cells are those generator/discriminator pairs involving the same team for both tasks and are not considered in performance evaluations here.

Figure 14 shows a bar-plot of BrierN scores for Discriminators. Here purple and blue bars indicate that the discriminator performs very well in terms of recognizing Human-generated content.

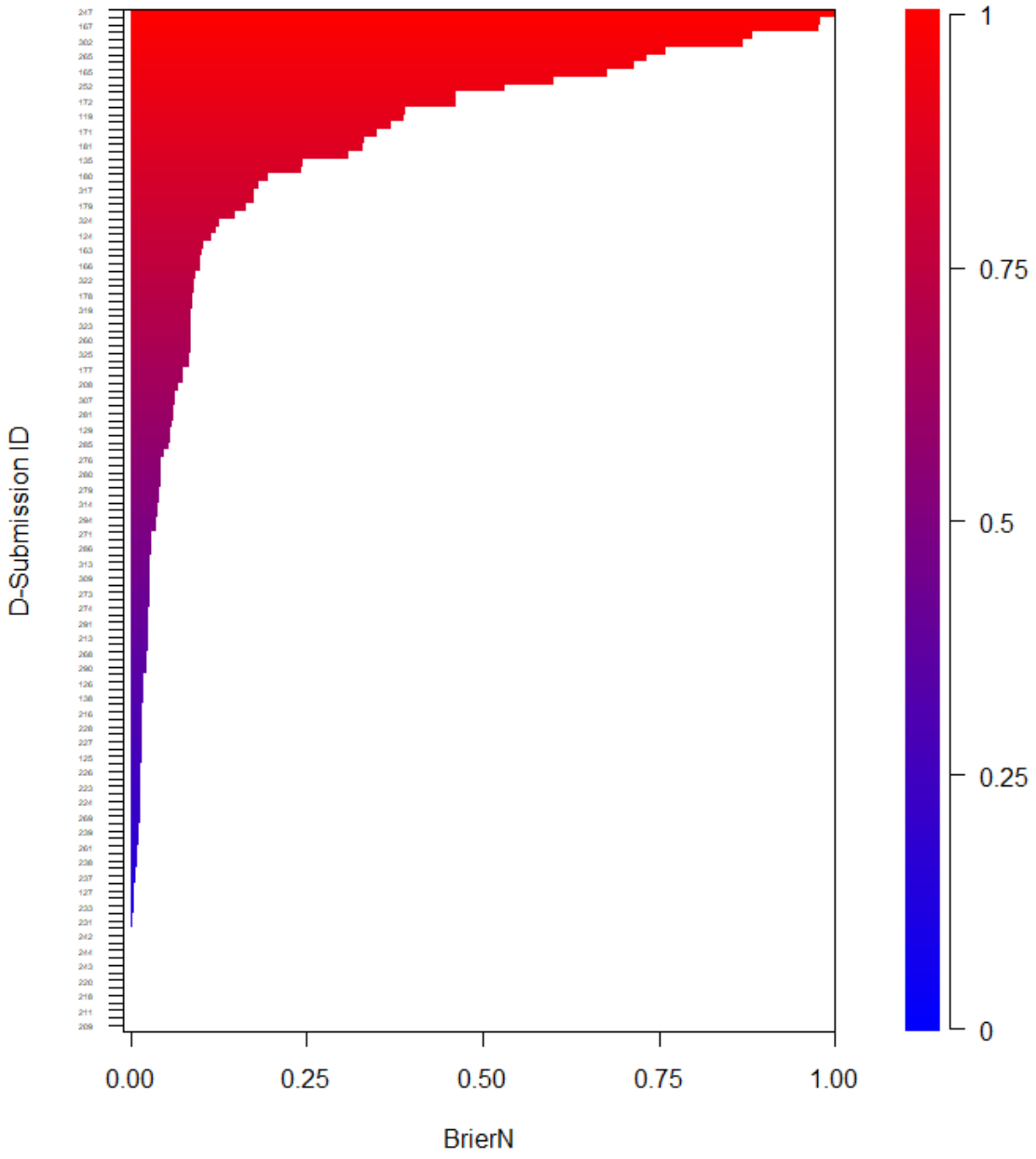

Figure 14. Bar-plot of BrierN scores for discriminators. Shades of red identify Discriminator submissions that give a high detection score (closer to 1 than closer to zero) to human-generated content. Shades of blue are more desirable for discriminators.

8.3. Results for Test Set-3 (D-Round 3)

Generator submissions that do well against Discriminator submissions will have a low AUC score (less than or equal to 0.5) or a high BrierT score (greater than or equal to 0.25) or both. Table 7 displays the 27 submission pairs for which AUC values are below or equal to 0.5 and BrierT scores above or equal to 0.99. Note that G-submission IDs 53 and 95 are from G-team labeled 804fe, and G-submission ID 110 is from G-team labeled 87a8c.

Table 7. G-submissions that perform well against the indicated D-submissions. For each G-submission, the AUC value for the corresponding D-submission is less than or equal to 0.5 and BrierT scores are greater than or equal to 0.99.

G-Participant	G-Submission ID	D-Participant	D-Submission ID	AUC	BrierT
804fe	95	6655b	439	0.5000	0.9996
6fc49	22	18126	397	0.4972	1.0000
804fe	95	18126	397	0.4972	1.0000
87a8c	111	18126	397	0.4972	1.0000
87a8c	113	18126	397	0.4972	1.0000
804fe	95	18126	412	0.4856	0.9987
87a8c	188	29d48	346	0.4709	0.9993
Baseline2	gpt4	18126	412	0.4637	0.9973
804fe	80	6655b	439	0.4406	0.9994
804fe	162	18126	412	0.4311	0.9984
aa872	114	29d48	346	0.4144	0.9995
87a8c	67	804fe	417	0.4006	0.9940
0dea0	58	18126	412	0.3994	0.9993
0782f	32	29d48	346	0.3872	0.9995
87a8c	188	6655b	439	0.3778	0.9992
87a8c	187	6655b	439	0.3556	0.9993
87a8c	190	6655b	439	0.3333	0.9993
804fe	53	6655b	435	0.3139	1.0000
804fe	53	6655b	436	0.3139	1.0000
804fe	53	6655b	437	0.3139	1.0000
804fe	53	6655b	438	0.3139	1.0000
87a8c	110	6655b	435	0.3139	1.0000
87a8c	110	6655b	436	0.3139	1.0000
87a8c	110	6655b	437	0.3139	1.0000
87a8c	110	6655b	438	0.3139	1.0000
87a8c	189	6655b	439	0.2222	0.9996
804fe	95	18126	474	0.0083	0.9999

High performance D-submissions should have a high AUC value and, if they are calibrated reasonably well, should have low BrierT and BrierN scores. The G-submissionID/D-submissionID pairs that have AUC above or equal to 0.995 and BrierT and BrierN scores below 0.0125 are listed in Table 8.

Table 8. D-submissions that perform well against the indicated G-submissions. For each G-submission, the AUC value for the corresponding D-submission is close to 1 and BrierT and BrierN scores are close to 0.

DTeam	D.subID	GTeam	G.subID	AUC	BrierT	BrierN
29d48	186	6fc49	358	1.0000	0.0000	0.0047
Baseline1	186	6fc49	327	1.0000	0.0000	0.0003
29d48	186	6fc49	359	1.0000	0.0000	0.0062
804fe	114	aa872	417	1.0000	0.0000	0.0094
29d48	186	6fc49	371	1.0000	0.0001	0.0077
29d48	186	6fc49	372	1.0000	0.0001	0.0112
29d48	186	6fc49	356	1.0000	0.0001	0.0062
29d48	192	87a8c	358	1.0000	0.0089	0.0047
18126	193	87a8c	397	0.9972	0.0000	0.0056
18126	195	87a8c	397	0.9972	0.0000	0.0056
18126	196	87a8c	397	0.9972	0.0000	0.0056
18126	198	87a8c	397	0.9972	0.0000	0.0056
18126	200	87a8c	397	0.9972	0.0000	0.0056
18126	201	87a8c	397	0.9972	0.0000	0.0056
18126	306	87a8c	397	0.9972	0.0000	0.0056
b3cd9	24	804fe	409	0.9972	0.0001	0.0062
b3cd9	31	0782f	409	0.9972	0.0001	0.0062
b3cd9	32	0782f	409	0.9972	0.0001	0.0062
b3cd9	65	87a8c	409	0.9972	0.0001	0.0062
b3cd9	90	6fc49	409	0.9972	0.0001	0.0062
b3cd9	114	aa872	409	0.9972	0.0001	0.0062
b3cd9	115	aa872	409	0.9972	0.0001	0.0062
b3cd9	116	aa872	409	0.9972	0.0001	0.0062
b3cd9	162	804fe	409	0.9972	0.0001	0.0062
b3cd9	191	87a8c	409	0.9972	0.0001	0.0062
b3cd9	193	87a8c	409	0.9972	0.0001	0.0062
b3cd9	195	87a8c	409	0.9972	0.0001	0.0062
b3cd9	198	87a8c	409	0.9972	0.0001	0.0062
b3cd9	199	87a8c	409	0.9972	0.0001	0.0062
b3cd9	200	87a8c	409	0.9972	0.0001	0.0062
b3cd9	306	87a8c	409	0.9972	0.0001	0.0062
b3cd9	312	87a8c	409	0.9972	0.0001	0.0062
b3cd9	gpt4	baseline2	409	0.9972	0.0001	0.0062
b3cd9	201	87a8c	409	0.9972	0.0002	0.0062
b3cd9	206	6fc49	409	0.9972	0.0004	0.0062
b3cd9	196	87a8c	409	0.9972	0.0011	0.0062
b3cd9	192	87a8c	409	0.9970	0.0108	0.0062
b3cd9	189	87a8c	409	0.9970	0.0113	0.0062
b3cd9	30	0782f	409	0.9969	0.0001	0.0062
b3cd9	29	0782f	409	0.9969	0.0006	0.0062
b3cd9	33	0782f	409	0.9969	0.0006	0.0062

Figure 15 shows a heatmap of the AUC scores corresponding to each Generator-Discriminator pair. Yellow color in a cell indicates that the discriminator is winning (on this metric) against the corresponding generator. Thus rows that show mostly yellow correspond to discriminators that are winning against most generators. Conversely, purple color (also dark green and dark blue) indicates cells where the generator is winning well against the corresponding discriminator. We see that there are some generators that win against most discriminators and a few discriminators that win against most generators.

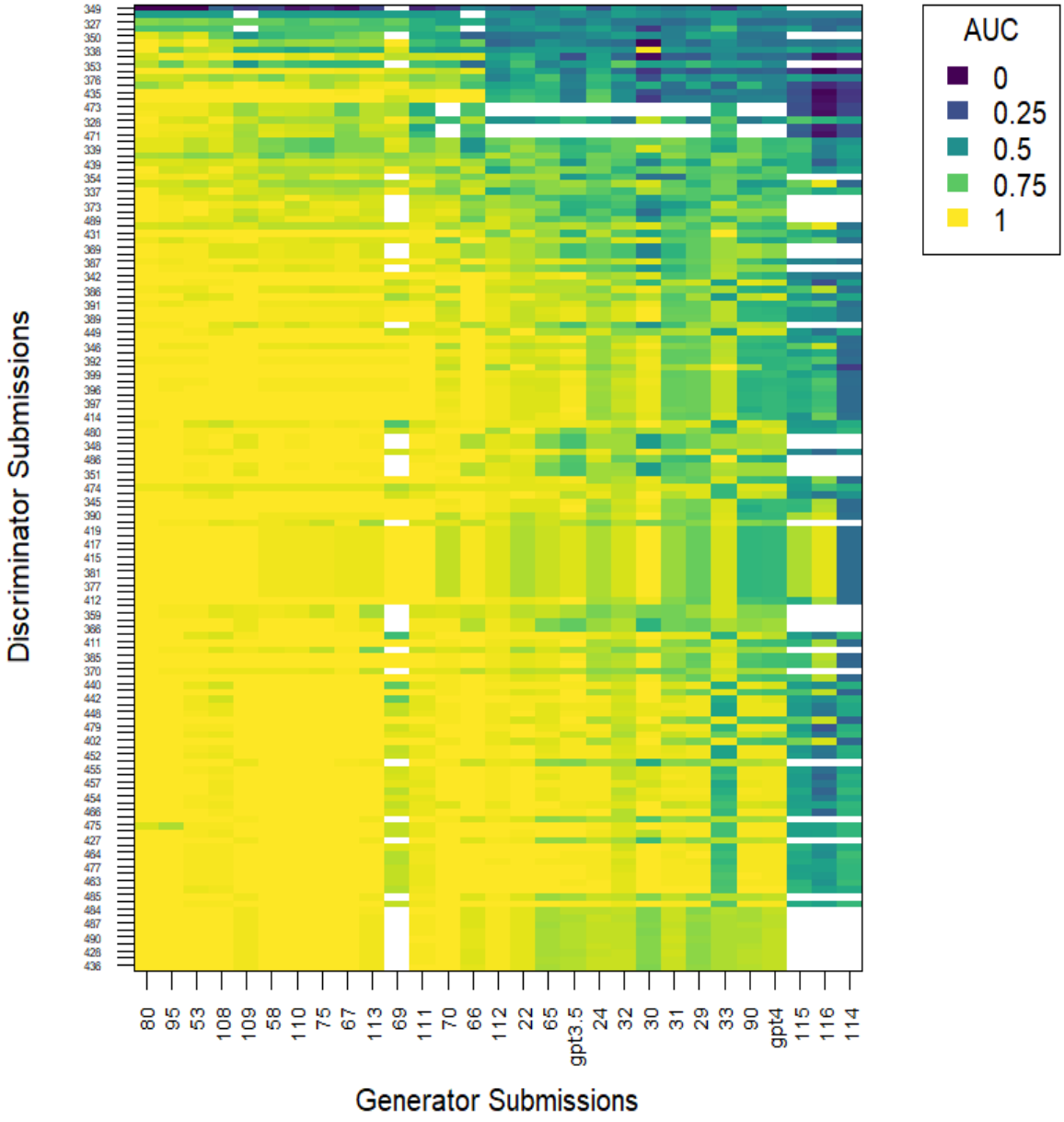

Figure 15. Heatmap of AUC scores corresponding to each Generator and Discriminator pair. Rows are labeled by Discriminator team submissions and columns by Generator team submissions. Yellow cells identify Generator-Discriminator submission pairs where the D-submission wins against the G-submission. Rows that are mostly yellow indicate D-submissions that perform well against most G-submissions.

Figure 16 shows a heatmap of BrierT scores for all Generator-Discriminator combinations. Here a generator does well against a discriminator if the cell color is green, light green, or yellow (in increasing performance). We see that only a few generators do well against most of the discriminators on this metric.

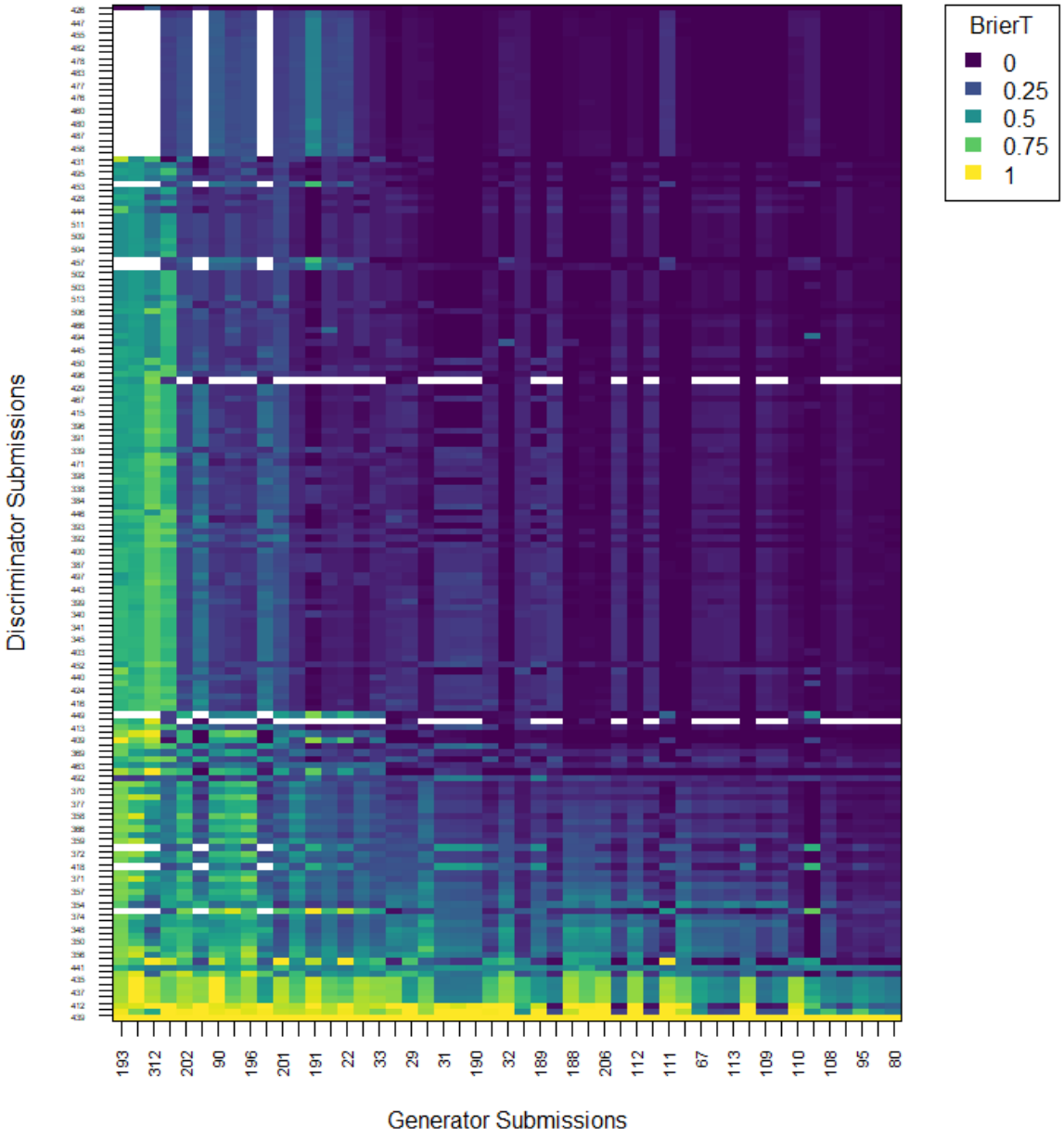

Figure 16. Heatmap of BrierT scores corresponding to each Generator and Discriminator pair. Rows are labeled by Discriminator team submissions and columns by Generator team submissions. Green, light green, or yellow cells identify Generator-Discriminator submission pairs where the G-submission wins against the D-submission. Rows that are mostly green, light green, or yellow indicate D-submissions that perform well against most G-submissions.

Figure 17 shows a bar-plot of BrierN scores for Discriminators. Here purple and blue bars indicate that the discriminator performs very well in terms of recognizing Human gener-

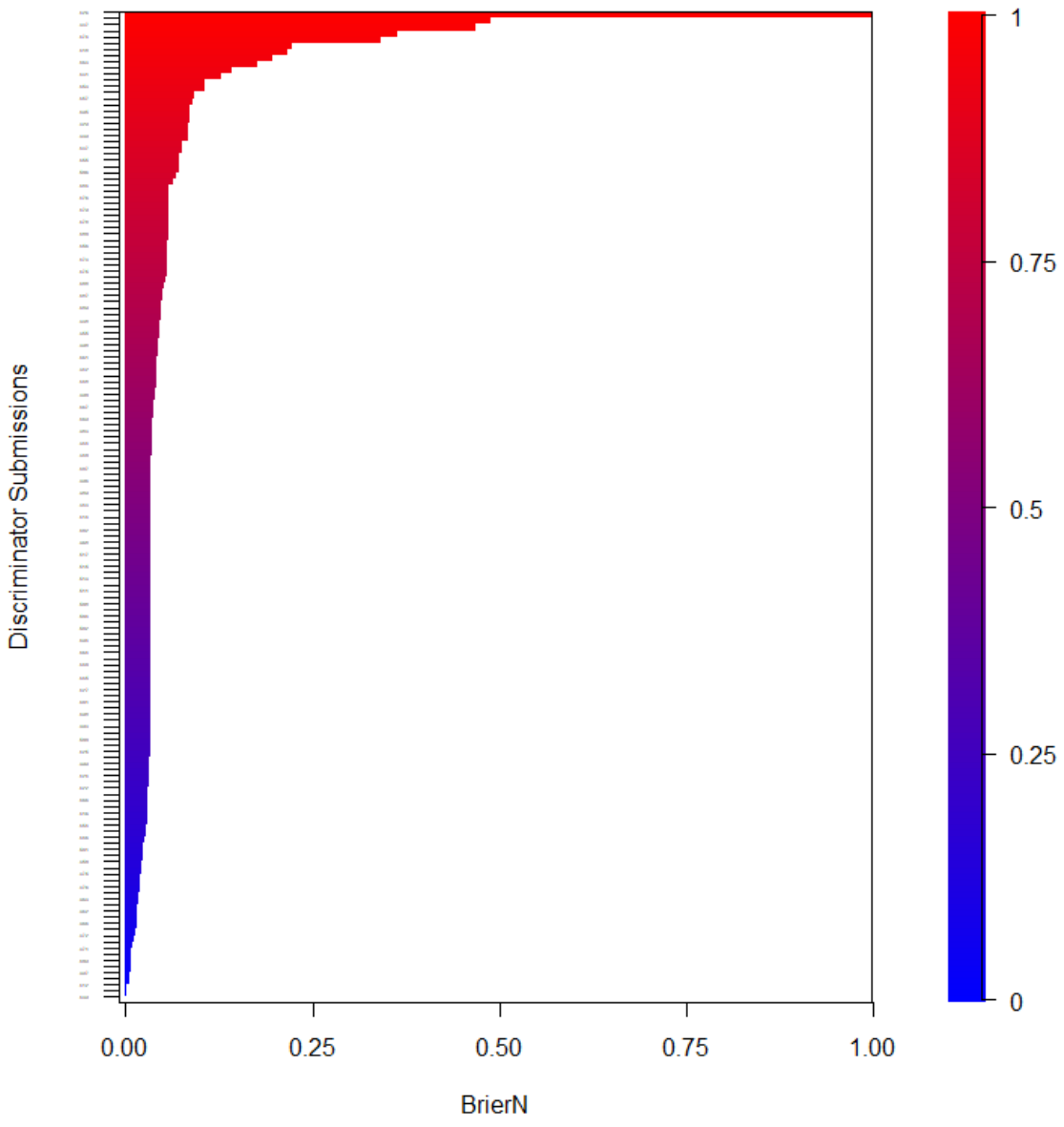

Figure 17. Bar-plot of BrierN scores for discriminators. Shades of red identify Discriminator submissions that give a high detection score (closer to 1 than closer to zero) to human-generated content. Shades of blue are more desirable for discriminators.

8.4. Evolution of Discriminators

A natural question one might ask is whether or not the discriminators improved over the different rounds. One could make this assessment by tracking how AUC scores changed from D-round-1 to D-round-2 to D-round-3.

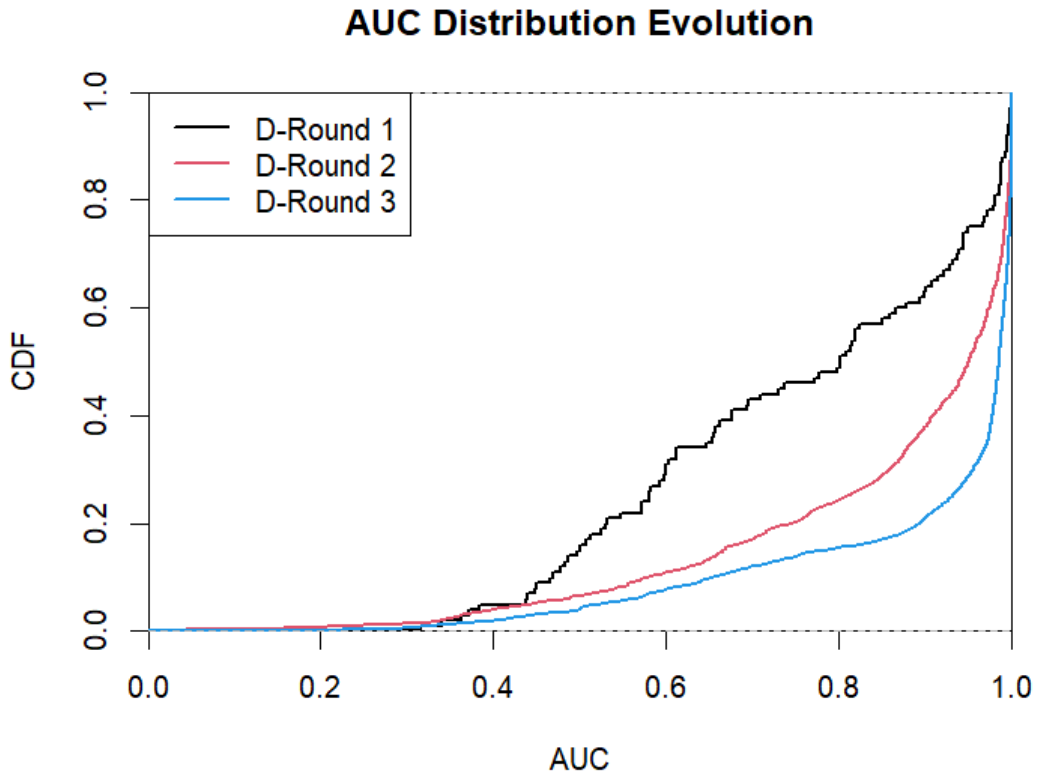

Figure 18. Empirical CDFs of AUC scores for the pool of D-submissions from D-round-1, D-round-2 and D-round-3.

Figure 18 shows the cumulative distribution functions (CDFs) for AUC scores from all D-submissions in round 1 (black), round 2 (red) and round 3 (blue). We see that, generally speaking, there were a greater proportion of AUC scores closer to one for D-round-3 submissions, followed by D-round-2 submissions and then by D-round-1 submissions. Although these rounds used different testsets, we see this pattern as an indication that D-systems generally improved over time, resulting in better discrimination in each round than in earlier rounds. One could track this improvement, or lack thereof, at an individual submission level or even by individual topic levels. However, these more detailed analyses are deferred to a subsequent publication.

9. Strengths, Limitations, and Challenges

9.1. Strengths

Comprehensive Evaluation Framework: This study establishes a robust framework for evaluating generative AI and discriminator models, incorporating multiple rounds of testing and diverse performance metrics such as AUC, BrierT, BrierN, and ROC curves.

Adversarial Testing Approach: The study's iterative process, where generators and discriminators can continuously improve by adapting to each other's outputs, mirrors real-world adversarial scenarios, ensuring that the evaluations remain relevant and challenging.

Diverse Data Sources: The inclusion of multiple topics and multiple human-generated summaries contributes towards a broad testing environment.

Clear Performance Metrics: The study effectively utilizes various statistical measures, such performance heatmaps, the Kolmogorov-Smirnov test for comparing distributions, and Area under the ROC curve (AUC) to assess system accuracy and robustness. The separation of BrierT and BrierN scores enables detailed analysis of model misclassification behavior.

Standardization and Benchmarking: By establishing clear guidelines and standardized evaluation metrics, the study contributes to the broader AI research community by providing a reproducible framework for future assessments of generative AI systems.

9.2. Limitations

Limited Scope of Text Domains: The study focuses on summarization tasks, which, while crucial, may not fully capture the broader landscape of AI-generated content detection across different textual modalities, such as conversational AI, creative writing, or technical documentation.

Assumption of Strict AI-Vs.-Human Authorship Distinction: While the study assumes clear distinctions between AI-generated and human-generated text, the increasing sophistication of AI models may challenge the binary classification framework. Future iterations could explore more nuanced categories, such as hybrid human-AI co-authored texts.

Scalability Constraints: The evaluation process involves extensive human and computational resources, potentially limiting scalability to large-scale assessments. The ability to generalize findings beyond the tested dataset remains uncertain.

Potential Bias in Discriminator Models: Some discriminator models may be biased towards specific linguistic patterns present in human-generated summaries, potentially skewing evaluation results. Ensuring diverse and unbiased training data for discriminators is critical for accurate assessment.

9.3. Challenges

Evolving AI Generation Techniques: Generative models are rapidly improving, often outpacing discriminator capabilities. The study acknowledges this cat-and-mouse dynamic, but maintaining up-to-date benchmarks and evaluation criteria remains a continuous challenge.

Broader Societal Considerations: The study provides a technical evaluation but does not address the societal implications of AI-generated content detection, nor does it consider policy or regulatory perspectives.

Computational Resource Requirements: The extensive evaluation process, particularly the large-scale comparisons of generators and discriminators, requires significant computational resources. Developing efficient evaluation methodologies that balance accuracy and feasibility will be crucial moving forward.

10. Conclusion

The NIST GenAI Text-to-Text Pilot Study presents a well-structured and rigorous evaluation framework for generative AI detection. While it offers valuable insights into the strengths and weaknesses of current-generation models, challenges such as evolving AI capabilities and scalability constraints should be addressed in future research. Expanding the scope of text domains, incorporating hybrid human-AI content, and refining evaluation techniques will further enhance the evaluation's impact in advancing AI-generated content detection methodologies.

11. Recommendations for Future Work

Expanding the Scope of Text Domains: Future research should move beyond summarization tasks to evaluate AI-generated content across diverse domains such as creative writing, conversational AI, technical documentation, and legal texts. This will provide a more comprehensive understanding of generative AI capabilities and detection challenges.

Developing More Robust Discriminators: Given the rapid advancements in generative AI, discriminator models should also evolve. Future work should focus on hybrid approaches combining statistical, linguistic, and deep-learning-based methods for more accurate detection of AI-generated text.

Investigating Human-AI Collaboration: Rather than viewing AI-generated and human-generated texts as distinct categories, future studies should explore hybrid models where AI assists human writers. Understanding how co-authored content differs from purely AI-generated or human-generated text is crucial.

Enhancing Scalability and Efficiency: Large-scale evaluations require extensive computational resources. Future research should explore optimized evaluation methodologies, including active learning and semi-supervised approaches, to improve efficiency while maintaining accuracy.

Addressing Bias in Detection Models: Discriminator models can exhibit skewed performance across different classes of inputs, which can impact their effectiveness. Future studies should investigate methods for mitigating bias, such as diverse training datasets and fairness-aware machine learning techniques.

Developing Benchmark Datasets for Multimodal Content: Since synthetic media spans multiple modalities (text, images, audio, and video), future work should focus on creating benchmark datasets that evaluate AI-generated content holistically across different media formats. Currently, the GenAI team is focused on measuring and evaluating the capabilities

and limitations of Generative AI technologies, without taking socio-technical concerns into account.

Advancing Explainability in Detection Models: It is worth exploring when and for what purposes explanations may be useful to users; for cases where they are, what kinds of explanations would be helpful; and what XAI techniques might be able to provide those kinds of explanations. Future research should consider developing explainable AI (XAI) techniques that provide insights into why a particular text was classified as AI-generated or human-written.

Establishing Continuous Benchmarking and Competitions: Regular benchmarking efforts and competitions, similar to the NIST GenAI Pilot study, should be conducted to track progress in AI generation and detection, fostering collaboration across academia, industry, and policy organizations.

References

[1] Adobe Content Authenticity Initiative (CAI) (2023). Content authenticity initiative.

[2] Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., and Nakov, P. (2020). What was written vs. who read it: News media profiling using text analysis and social media context. In *Findings of EMNLP 2020*.

[3] DeepMind, G. (2023). Synthid: Ai-generated image detection.

[4] DetectGPT (2023). Zero-shot ai text detection via probability curvature. *arXiv Preprint*.

[5] FakeCatcher (2022). Deepfake video detection.

[6] Ferrara, E. (2020). Manipulating social media: Machine learning-based automation in the age of disinformation. *Annual Review of Statistics and Its Application*, 7:273–297.

[7] GPTZero (2023). A classifier for ai-generated text detection.

[8] Harvard & MIT-IBM Watson Lab (2019). Gltr – giant language model test room.

[9] Hive AI (2023). Ai content detector api for enterprises.

[10] Hu, X., Chen, P.-Y., and Ho, T.-Y. (2023). RADAR: Robust AI-text detection via adversarial learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023)*. Accessed: March 2025.

[11] Kirchenbauer, J., Geiping, J., Wen, B., Katz, N., Goldstein, T., and Jermyn, J. (2023). A watermark for large language models. *arXiv Preprint*.

[12] Kumarage, T., Agrawal, G., Sheth, P., Karim, R., Deb, R. P., and Liu, H. (2024). A survey of ai-generated text forensic systems: Detection, attribution, and characterization. *arXiv preprint arXiv:2403.01152*.

[13] Macmillan, N. A. and Creelman, C. D. (2005). *Detection Theory: A User's Guide*. Lawrence Erlbaum Associates, Mahwah, NJ, 2nd edition.

[14] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997a). The DET curve in assessment of detection task performance. Technical report, DTIC Document.

[15] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997b). The DET curve in assessment of detection task performance. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech 1997)*, pages 1895–1898.

[16] Microsoft (2023). Deepfake detection by microsoft.

[17] NIST AI 100-4 (2024). Reducing Risks Posed by Synthetic Content: An Overview of Technical Approaches to Digital Content Transparency. Technical Report NIST AI NIST AI 100-4, National Institute of Standards and Technology, Gaithersburg, MD.

[18] OpenAI (2022). Roberta-base openai detector. Accessed: March 2025.

[19] OpenAI (2023a). Ai text classifier. [Discontinued].

[20] OpenAI (2023b). Gpt-4 technical report. *arXiv preprint*, arXiv:2303.08774.

[21] OpenAI (2024). Cryptographic watermarking (in development). OpenAI's watermarking research.

[22] Penn State News (2024). Q&A: The increasing difficulty of detecting AI- versus human-generated text. https://www.psu.edu/news/information-sciences-and-technology/story/qa-increasing-difficulty-detecting-ai-versus-human. Accessed: 2025-02-24.

[23] Pennycook, G. and Rand, D. G. (2019). Fighting misinformation on social media using "accuracy prompts". *Nature Human Behaviour*, 3(5):484–488.

[24] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188.

[25] Tan, R., Plummer, B. A., and Saenko, K. (2020). Detecting cross-modal inconsistency to defend against neural fake news. *arXiv preprint arXiv:2009.07698*.

[26] Turnitin AI Detection (2023). Ai writing detection for academic integrity.

[27] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

[28] Wardle, C. and Derakhshan, H. (2017). Information disorder: Toward an interdisciplinary framework for research and policy making. Technical report, Council of Europe.

[29] Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., Foltýnek, T., Guerrero-Dib, J., Popoola, O., Šigut, P., and Waddington, L. (2023). Testing of detection tools for ai-generated text. *International Journal for Educational Integrity*, 19(1):1–39.

[30] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., and Yu, P. S. (2021). Rumor detection on social media with graph neural networks. In *Proceedings of the Web Conference 2021*, pages 798–808.

[31] Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., Cui, Y., Zhou, Z., Gong, C., Shen, Y., Zhou, J., Chen, S., Gui, T., Zhang, Q., and Huang, X. (2023). A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

A. Supplementary Information

Each of the G and D-participants submitted a variable number of evaluation submissions.

Table 9. D-participants in D-round-1, their submission IDs, and total number of submissions per participant.

D-Participant	D-Submission IDs	Submission Count
0dea0	7, 8	2
18126	13, 14, 19, 21, 44, 47, 49, 68	8
29d48	4	1
6655b	12, 15, 59, 73, 99, 101, 102, 103	8
6fc49	26, 98	2
804fe	1, 2, 5, 6, 9, 10, 11, 18, 82, 83, 84, 85, 86, 87	14
87a8c	76, 77, 78, 92, 93, 96, 104	7
993ad	46, 50, 52	3
9de37	25, 36, 37	3
b3cd9	34	1
d718e	40	1

Table 10. G-participants in G-round-1, their submission IDs, and number of submissions. There were two NIST baseline G-participants: `NIST-gpt35` and `NIST-gpt4`. Not counting the NIST baselines, there were a total of 27 submissions, each submission consisting of one AI summary for each of the 10 topics. Thus, a total of 10 summaries were presented as part of each submission by each external G-participant.

G-participant	G-submission IDs	Submission Count
0782f	29, 30, 31, 32, 33	5
0dea0	58	1
6fc49	22, 90	2
804fe	24, 53, 80, 95	4
87a8c	65, 66, 67, 69, 70, 75, 108, 109, 110, 111, 112, 113	12
aa872	114, 115, 116	3
NIST	gpt3.5	1
NIST	gpt4	1

Table 11. D-participants in D-round-2, submission IDs, and number of submissions. There were a total of 137 D-submissions from 8 D-participants.

D-Participant	D-Submission IDs	Submission Count
18126	119, 127, 128, 129, 135, 207, 208, 252, 254, 262, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 289, 290, 291, 292, 293, 294, 295, 307, 309, 311, 313, 314, 315, 316	47
29d48	138, 139, 140, 142, 144, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 296, 297	42
6655b	124	1
6fc49	205, 247	2
804fe	163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 175, 176, 177, 178, 179, 180, 181, 183, 184, 185, 248, 249, 257, 258, 259, 260, 263, 317, 318, 319, 320, 321, 322, 323, 324, 325	37
87a8c	302, 303, 304, 310	4
9de37	121, 125, 126	3
b3cd9	261	1

Table 12. G-participants in G-round-2, submission IDs, and number of submissions. There were seven external G-participants and two NIST baseline Generators (NIST-gpt35 and NIST-gpt4). There were a total of 47 external participant submissions.

G-participant	G-submission IDs	Submission Count
0782f	29, 30, 31, 32, 33	5
0dea0	58	1
6fc49	186, 206, 22, 90	4
804fe	162, 24, 53, 80, 95	5
87a8c	108, 109, 110, 111, 112, 113, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 198, 199, 200, 201, 202, 306, 312, 65, 66, 67, 69, 70, 75	29
aa872	114, 115, 116	3
baseline1	gpt3.5	1
baseline2	gpt4	1

Table 13. D-participants in D-round-3, submission IDs, and number of submissions. In this round, there were 7 discriminator teams and a total of 161 submissions. One of these submissions is from NIST GenAI team using an in-house baseline algorithm.

D-Participant	D-Submission IDs	Submission Count
18126	328, 338, 339, 340, 341, 342, 343, 344, 345, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 396, 397, 398, 399, 400, 401, 402, 403, 407, 410, 411, 412, 413, 414, 415, 416, 422, 423, 424, 425, 426, 427, 428, 429, 440, 442, 443, 444, 445, 446, 448, 450, 451, 452, 454, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 474, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 508, 509, 511, 512, 513, 514, 515, 516, 517	92
29d48	346, 348, 349, 350, 351, 353, 354, 355, 356, 357, 358, 359, 362, 363, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 379, 381	28
6655b	431, 435, 436, 437, 438, 439, 441	7
804fe	329, 417, 418, 419, 421, 447, 449, 453, 455, 456, 457, 458, 460, 473, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490	30
87a8c	335, 337	2
b3cd9	409	1
Baseline1	327	1

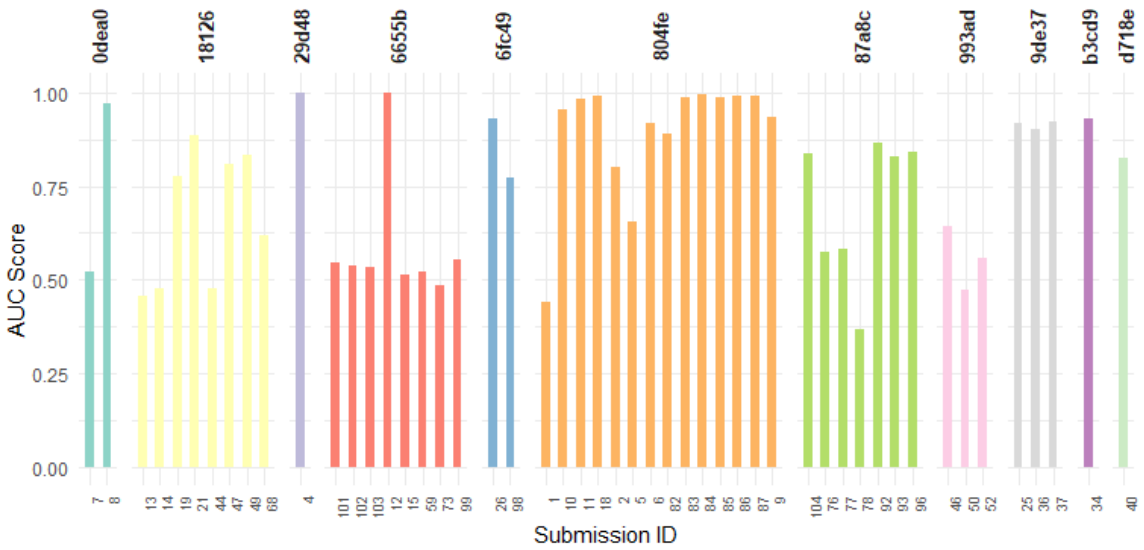

Figure 19. AUC for the pool of D-submissions from D-round-1.

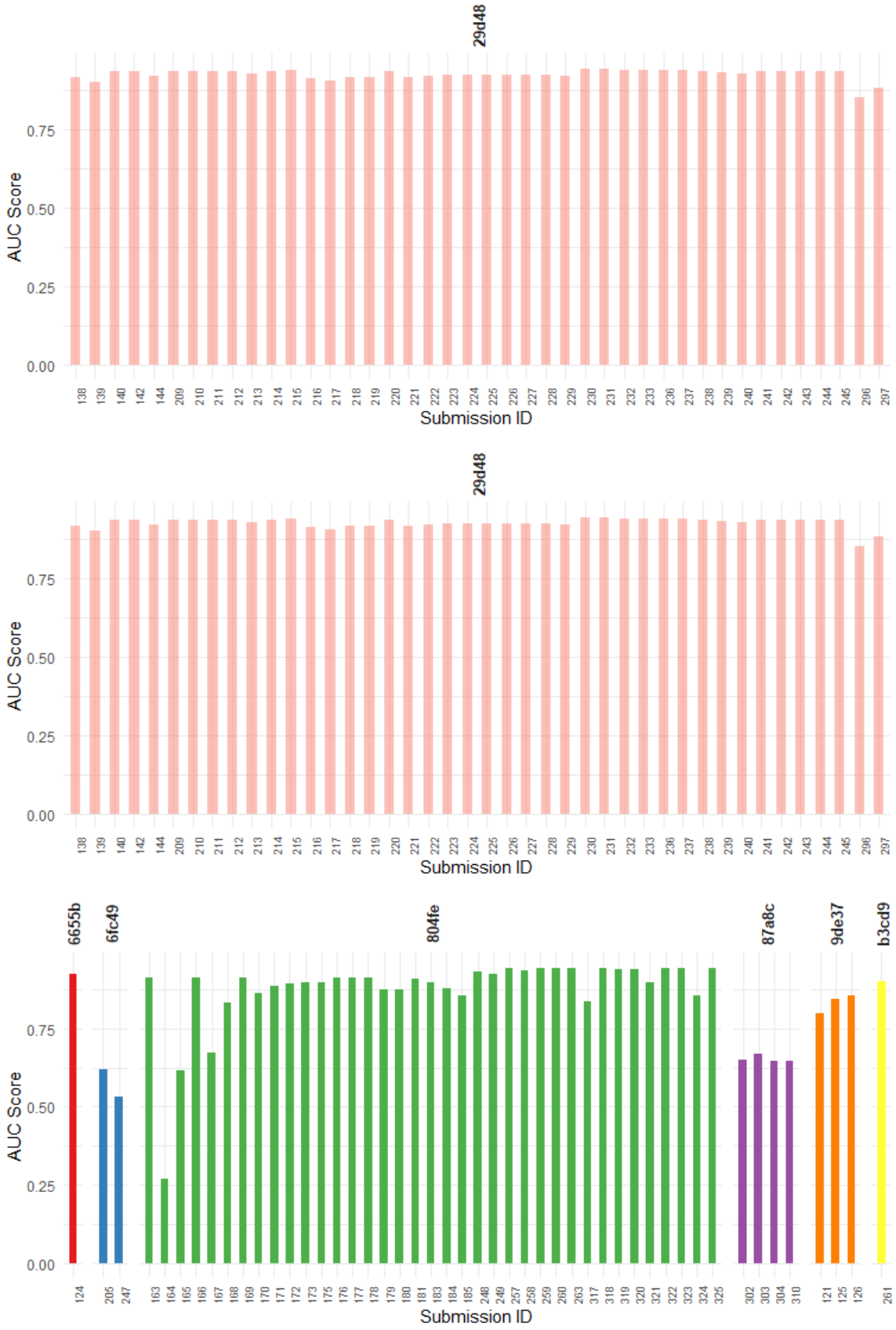

Figure 20. AUC for the pool of D-submissions from D-round-2.

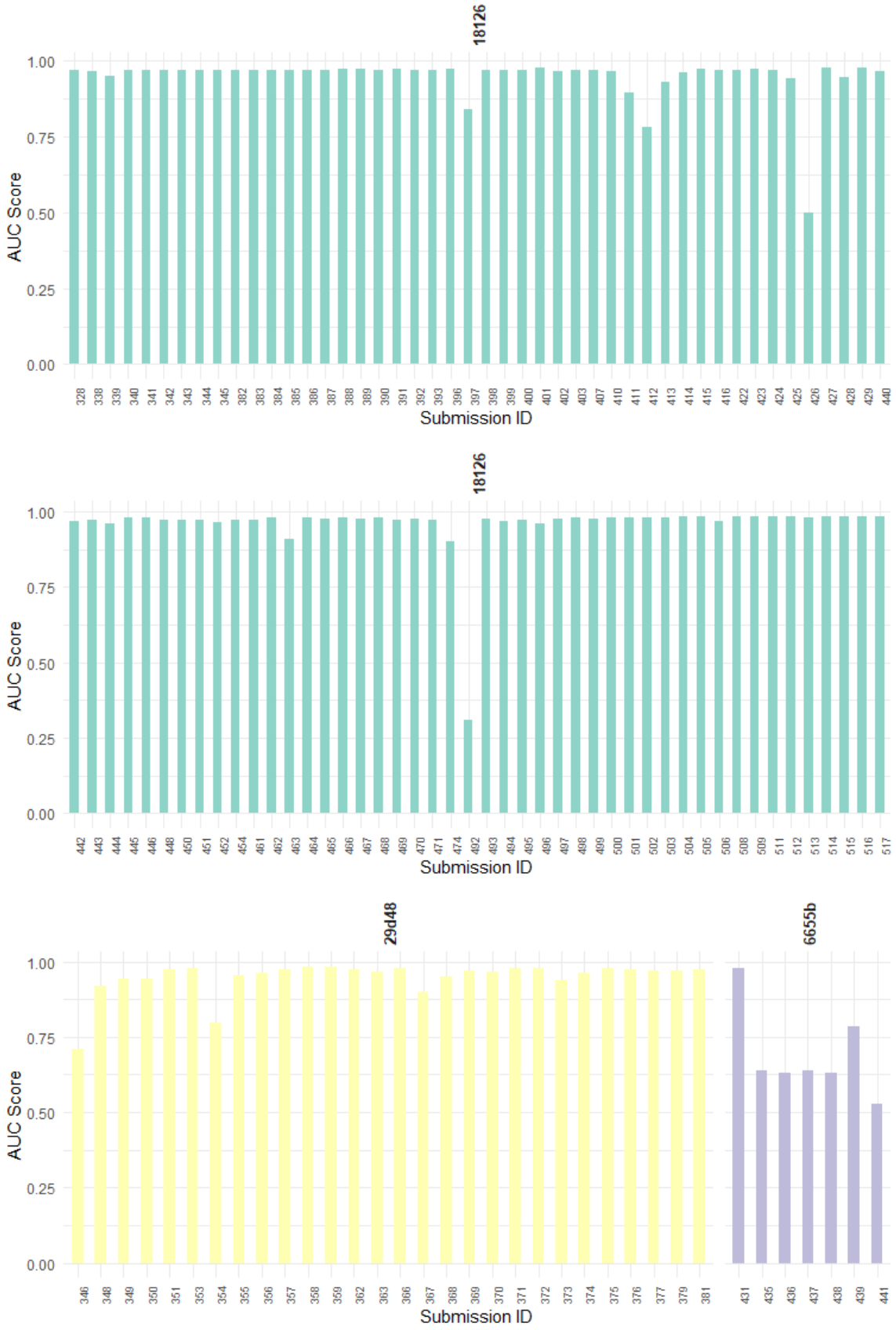

Figure 21. AUC for the pool of D-submissions from D-round-3.

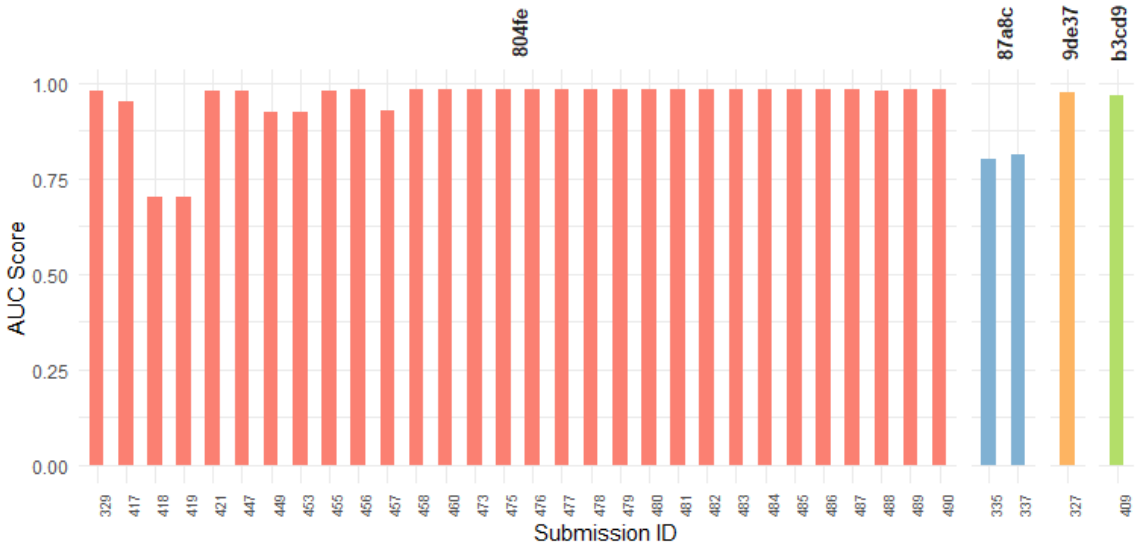

Figure 21. (continued). AUC for the pool of D-submissions from D-round-3.