2026 NIST GenAI Text Challenge Evaluation Plan

NIST GenAI Team

February 24, 2026

Abstract

The NIST Generative AI (GenAI) Text Challenge is an evaluation program designed to probe the capabilities and limitations of generative AI systems in producing text, from three complementary perspectives: Generator, Prompter, and Discriminator. The challenge will focus on two core aspects: **indistinguishability** from human-written content and the **believability** of generated narratives. In the **Generator task**, participants are to build systems that produce text that is indistinguishable from human-authored passages. In the **Prompter task**, participants craft prompts that elicit both highly accurate narratives and misleading, but persuasive, narratives from a generator. In the **Discriminator task**, participants develop detectors that not only help decide whether a passage is AI-generated or human-written, but also predict how believable the content will be to a lay audience. Participation is open to all individuals and organizations that agree to follow the established rules and procedures.

Contents

DISCLAIMER

Certain commercial equipment, instruments, software, or materials are identified in this document to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor necessarily the best available for the purpose. The descriptions and views contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NIST or the U.S. Government. This project has been reviewed by the Research Protections Office (RPO) under the reference number ITL-2023-0644.

Revision History

2025-09-01	First version.
2026-02-23	Second version (updated Appendix A, Appendix B, and Appendix C).

1 Introduction

In recent years, the quality of digital content generated by artificial intelligence (AI) has advanced considerably across various modalities, including image, video, audio, and text. This surge in generative AI capability presents both opportunities and challenges—generative AI has facilitated creative expression and production, enabling artists, designers, and writers to create high quality digital content at a much faster pace, but has also raised concerns regarding the authenticity and integrity of digital media, especially as it has become increasingly difficult to distinguish AI-generated content from human-generated content.

The NIST Generative AI (GenAI) program[1] supports research in generative AI with a structured series of evaluations testing the capabilities of multiple AI technologies in various modalities, beginning with text generation. The 2026 GenAI Text Challenge invites teams from academia, industry, and the research community to test AI capabilities by acting in different roles across the spectrum as: "generators", "prompters", and "discriminators."

- **Generators** build AI systems that generate text that is indistinguishable from human writing and relevant to the prompt.
- **Prompters** craft prompts that make AI systems output either accurate, credible narratives, or intentionally misleading yet believable content.
- **Discriminators** create AI systems that accurately classify text as human- or AI-authored, and they predict the proportion of readers who will believe the text's content.

1.1 Research Objective

The research purpose of the upcoming GenAI Text Challenge is to evaluate the capabilities and limitations of generative AI systems in producing written content. The Challenge will investigate both indistinguishability and believability. Indistinguishability refers to whether the system output is indistinguishable from human-generated content, and believability refers to whether a system can generate, with appropriate prompting, incorrect or questionable information in a manner that the user trusts. The Challenge comprises Generator (G), Prompter (P), and Discriminator (D) tracks, to evaluate the strengths and weaknesses of various approaches. The objectives of this research are to:

- Evaluate the ability of generative AI systems to produce text that is indistinguishable from text written by humans.
- Assess the effectiveness of AI content discriminators in detecting AI-generated content as well as assigning a believability score.
- Investigate the factors that cause belief in inaccurate or misleading AI-generated content
- Explore the role of AI prompts in generating both credible and misleading content

[1] https://ai-challenges.nist.gov/genai

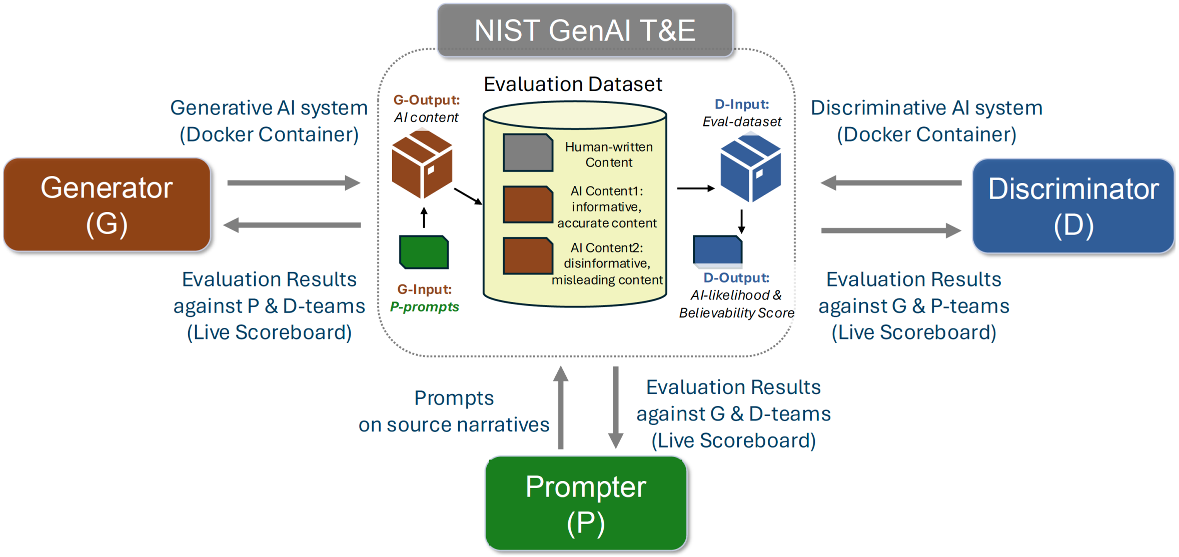

Figure 1: NIST GenAI Text Challenge - Evaluation Framework Overview.

Through participation, teams will contribute to:

- **Driving AI innovation** by expanding knowledge of the capabilities and limitations of current AI systems and content discriminators
- **Creating a responsible AI ecosystem** by understanding the complex interactions between systems, prompts, and content discriminators
- **Informing AI standards** by evaluating and benchmarking diverse systems and prompting strategies in a controlled setting.

2 Task

The 2026 GenAI Text evaluation addresses the following three tasks:

- **Text Generators (Text-G)**: develop Generative AI systems capable of generating high-quality text content. These AI systems will be used by Prompter teams to generate text content. Submissions should be in the form of a Docker Image.
- **Text Prompters (Text-P)**: using the generative AI system from generator teams and a given topic statement as input, create two sets of prompts:
  - The first set of prompts will generate accurate, relevant information in a brief, well-organized and fluent manner.
  - The second set of prompts will generate a well-organized, and fluent narrative that contains inaccurate, non-credible, and/or misleading information, crafted to be believable to the general public.
- **Text Discriminators (Text-D)**: given a text narrative, which may be AI-generated or human-generated, develop AI systems to perform two main tasks:

- Assess the likelihood of the content being AI-generated by assigning a score between 0 and 1 (inclusive), such that scores closer to 1 suggest the content is AI-generated and scores closer to 0 suggest the content is human-generated (0.5 represents total indecision).

- Predict the proportion of the general public that would believe the main message of the text narrative by assigning a believability score between 0 (0% of the public would believe it) and 1 (100% of the population would believe it).

3 Evaluation Metrics

3.1 Performance Metrics

3.1.1 AUC-ROC

The AUC-ROC score reflects a detector's ability to distinguish between classes in general. A score near 0.5 indicates that the detector performance is no better than random guessing. A score of 1 means the detector is able to perfectly separate the classes. A score close to 0 suggests the detector tends to assign higher scores to non-target class and lower scores to target class, contrary to what it is supposed to do. This will lead to classifiers that are consistently wrong, essentially predicting the incorrect class with high confidence.

In this challenge, we place greater value on the indistinguishability of AI-generated content, rather than completely fooling the discriminator into labeling AI-generated content as more human-like than actual human writing. The reason is that such extreme misclassification does not reflect true success in generation — it represents an artifact of over-deception rather than genuine alignment with human-written quality.

3.1.2 Brier Score

The Brier score measures the accuracy of probabilistic predictions by calculating the mean squared difference between the assigned probability and the actual outcome. A lower score indicates better calibration and predictive accuracy, while a higher score reflects greater error.

Consider a scenario where the generator successfully causes the detector to judge AI-generated content as human-written (or accurate). For a well-calibrated detector, a score near 0.5 represents total uncertainty, signaling confusion. In this case, the Brier score will be

$$(1-0.5)^2 = (0-0.5)^2 = 0.25.$$

However, if the detector assigns a score close to 0 for AI-generated content (i.e., high confidence that the content is human-generated when it is actually AI-generated), the generator has not merely confused the detector but has completely fooled it. If the detector is incorrectly confident—for example, assigning probabilities higher than 0.5 to human-written content or lower than 0.5 to AI-generated content—then the Brier score will exceed 0.25. At the extreme, a Brier score of 1 indicates the detector has been entirely fooled, showing 100% confidence in the wrong classification.

3.1.3 Believability score

Believability scores are intended to approximate the proportion of general readers who would believe the main message in the text. Higher scores (closer to 1) suggest that the generator produces highly believable information. Evaluation will consider the maximum and average believability scores across each set of narratives and prompts, as well as the overall distribution of scores. Whether these scores truly reflect human judgment will ultimately be validated using human evaluation over a diverse corpus.

Table 1 shows how a human reader might quantify their degree of belief in the main message of the narrative.

Table 1: Believability Score and Human judgment

Scale of believability score	Human judgment
0.0 – 0.2	I strongly disbelieve it
0.2 – 0.4	I'm leaning toward not believing it
0.4 – 0.6	Unsure
0.6 – 0.8	I'm leaning toward believing it
0.8 – 1.0	I strongly believe it

In evaluating believability, the assessment is intentionally scoped so that only the content within the narrative informs the prediction—external references or ground-truth verification are not provided to the discriminator system.

- No reference or source narrative is provided to the discriminator.
- A discriminator may either: (1) draw on internal knowledge, common sense reasoning, and world models to predict how convincing a given narrative would be to lay readers; or (2) augment an external context or fact-checking modules to aid in believability prediction.
- A system description must be submitted alongside system outputs, detailing detection methods and any additional tools or models used.
- As a result, the believability score reflects how plausible or convincing the narrative appears to an unassisted reader or model at the moment of reading—not whether the information can be verified as factually correct.

3.2 Definition of Success for Each Track

3.2.1 Text Generator (Text-G)

A generator G is considered successful when it produces content that is difficult for a discriminator D to distinguish from human-written text. Specifically:

- AUC-ROC scores closer to 0.5, indicating indistinguishability from random guessing.
- Brier scores closer to 0.25, reflecting that D is confidently misled.
- Higher believability scores (close to 1), showing that the generated content is perceived as human-like and convincing.

Each generator must register jointly with a Text Prompter and submit its own set of prompts. See below for details.

3.2.2 Text Prompter (Text-P)

A prompter *P* is considered successful when the prompts it generates lead to convincing outputs from the generator. Success is measured by:

- AUC-ROC scores closer to 0.5, indicating that *D* cannot reliably separate human-written from AI-generated responses. Evaluations will consider the average AUC-ROC across all prompts submitted with a given generator.
- Brier scores close to 0.25, reflecting that *D* is consistently confused.
- Higher believability scores (close to 1), showing that outputs guided by the prompt are convincing and human-like. Evaluation will consider the maximum and average believability scores across each set of narratives and generator systems, as well as the overall distribution of scores.

Importantly, prompter teams do not generate content themselves. Rather, they influence the quality, credibility, and plausibility of the generated responses through prompt design. Consequently, the evaluation of a prompter system is based on its downstream impact on both generator and discriminator outputs.

3.2.3 Text Discriminator (Text-D)

A discriminator *D* is considered successful when it can effectively distinguish human-written from AI-generated content. Indicators of success include:

- Higher AUC-ROC scores, reflecting stronger discriminative power.
- Brier scores close to 0, indicating accurate and well-calibrated probabilities.
- Lower believability scores assigned to inaccurate or misleading content, demonstrating robustness against deception.

4 Data Resources

4.1 Development Set

NIST will release development datasets to participating teams to support system design, calibration, and internal testing. The development dataset will include a set of topic statements and corresponding text narratives, paired with ground-truth scores: AI-likelihood labels and human-annotated believability scores. These examples will illustrate the types of topics and domains covered in the evaluation, giving each team a general sense of expected input and output.

4.2 Test Set

The test dataset will remain blind to participants throughout the evaluation period. Submitted systems will be executed on the NIST servers using this blind evaluation set.

NIST reserves the right to not release the full test dataset following the conclusion of the evaluation. If any data is to be released, participants will be notified.

5 System Requirements and Specifications

Prompter teams must submit their prompts in a structured `JSON` format (see Appendix A).

Generator and **Discriminator** teams must package their systems as Docker images to ensure portability and consistent execution across evaluation environments. The submitted images will be executed as containers by the evaluation infrastructure. For detailed requirements and specifications, see Appendix B and C.

Additionally, Generator and Discriminator teams are required to provide comprehensive system descriptions. Refer to Appendix D for the Generator and Appendix E for the Discriminator.

6 Protocol and Rules

Rules and Restrictions

- Participants may use publicly available data, provided it complies with all applicable laws and regulations, to train their systems.
- All systems must complete training, model selection, and tuning prior to submission to NIST.
- Participants will be notified of the maximum number of allowed submissions at the start of the evaluation.

Advertising and Endorsement

- Participants may not make advertising claims about their standing in the evaluation or claim NIST endorsement of their system(s).
- The following language from the U.S. Code of Federal Regulations (15 C.F.R. §200.113 (d)) must be respected: NIST does not approve, recommend, or endorse any proprietary product or proprietary material.

Reporting and Publication

- NIST will generate a report summarizing the system results with participant team names.
- Participants may publish their research, provided they include proper references to the original source.

7 Agreement

All participants who wish to submit their systems and outputs will be required to complete and sign an agreement included in a registration form before uploading their submissions. This registration form will be provided to participants during the registration process for the evaluation.

8 Tentative Schedule

The schedule for the GenAI Text Challenge is:

September 1, 2025	Evaluation Plan Posted
February 23, 2026	Evaluation Plan Updated
March 9, 2026	Registration Open
March 31, 2026	Registration Close
April 27, 2026	Round-table with Registrants
May 25, 2026	Dry-run with Sample Set
June 30, 2026	Dry-run Close
July 27, 2026	Phase 1 Start
Aug 31, 2026	Phase 1 Close
Sep 28, 2026	Phase 2 Start
Oct 30, 2026	Phase 2 Close

A Appendix: Prompter Submission Format

Submissions must be formatted in JSON and include all required fields. For the `truthfulness` field, use "0" to indicate prompts intended to produce factual and accurate narratives, and "1" to indicate prompts intended to produce misleading or inaccurate narratives. For the `submissionindex` field, assign each submission a value from 1 to 3, corresponding to its order of submission to the system.

```
{
  "team": "team ID",
  "version": "1.0",
  "submissionindex": "1",
  "prompt_list": [
    {
      "statement_id": "00001",
      "truthfullness": "0",
      "system_prompt": "Your system prompt here",
      "user_prompt": "Your user prompt here",
    },
    {
      "statement_id": "00001",
      "truthfullness": "1",
      "system_prompt": "Your system prompt here",
      "user_prompt": "Your user prompt here",
    },
    {
      "statement_id": "00002",
      "truthfullness": "0",
      "system_prompt": "Your system prompt here",
      "user_prompt": "Your user prompt here",
    },
    ...
  ]
}
```

B Appendix: Container Build and Runtime Requirements

This appendix defines the technical specifications, security constraints, and system design requirements for all generator system submissions.

All submissions must be provided as Docker images, which will be executed as containers by the evaluation infrastructure inside a secured, off-network environment.

B.1 Submission Artifact

Participants must submit a Docker image that encapsulates the complete system implementation, including all required binaries, libraries, model weights, indexes, and configuration files.

- The submitted Docker image must be self-contained and reproducible.

- Images may not rely on runtime downloads except where explicitly permitted.

B.2 Standardized Input and Output Interface

All systems must adhere to the following fixed in-container directory conventions:

- `/input`: Read-only directory containing all input files.
- `/output`: Writable directory where all system outputs must be written.

The evaluation infrastructure will **mount host directories into these locations at runtime**. Systems must not assume the presence of any other writable filesystem paths.

B.3 Runtime Execution Model

Submitted images will be executed using a standardized container invocation pattern similar to:

```
docker run --rm \
  --network none \
  --mount type=bind,src=<host_input>,dst=/input,ro \
  --mount type=bind,src=<host_output>,dst=/output \
  <image>
```

B.3.1 Conditional Network Access for NIST-Provided APIs

By default, all submitted systems are executed with outbound network access disabled. An exception is granted only for approved workflow or RAG systems that rely on organizer-provided foundation model APIs. NIST will publish in advance a fixed list of approved API providers and model endpoints.

This list is provided solely to allow participants to design and implement compatible orchestration logic. Example providers may include, but are not limited to:

- Azure-hosted models including OpenAI's GPT, Anthropic's Claude
- Google Vertex AI-hosted Gemini models

For such systems:

- Network access is restricted exclusively to organizer-approved internal API endpoints.
- API credentials and endpoint configuration are injected at runtime by the evaluation infrastructure via environment variables.
- Submissions must not store, log, export, or hard-code any credentials or endpoint details.
- No other external network communication is permitted.
- Participants must not assume the availability of any unlisted providers, models, or API features.

An example execution pattern for these systems is shown below for illustration only:

```
docker run --rm \
  --network restricted \
  --mount type=bind,src=<host_input>,dst=/input,ro \
  --mount type=bind,src=<host_output>,dst=/output \
  -e NIST_API_ENDPOINT=<injected_endpoint> \
  -e NIST_API_KEY=<injected_secret> \
  <image>
```

B.4 Determinism and Reproducibility

All systems must operate deterministically.

- Identical inputs must produce identical outputs, if applicable.
- All sampling parameters (e.g., temperature, top-k, top-p) must be fixed.
- Any sources of nondeterminism must be explicitly disabled.

B.5 Offline Operation and Security

Except where explicitly permitted:

- No outbound or inbound network communication is allowed.
- All model weights, corpora, indexes, and resources must be packaged inside the Docker image.
- Systems may write only to `/output`.

Any attempt to bypass network isolation or leak data through unauthorized channels will result in disqualification.

B.6 Allowed Generator System Categories

The evaluation permits the following generator system categories. All categories must comply with the common interface and security requirements defined above.

The evaluation infrastructure will include organizer-provided baseline generator systems based on standard, unmodified foundation models (e.g., GPT-class, Claude-class, and Gemini-class models). These baselines are used solely for comparative and analytical purposes.

Accordingly, participant submissions must demonstrate substantive system-level contributions beyond direct use of a vanilla foundation model. Systems that simply forward inputs to an unmodified foundation model API, without meaningful model adaptation, local computation, retrieval, or orchestration logic, are not eligible for submission.

Eligible generator systems must therefore fall into one or more of the categories, such as locally fine-tuned or quantized models, retrieval-augmented generation systems, or workflow-based systems that incorporate non-trivial control, reasoning, or integration logic around NIST-approved foundation model APIs.

B.7 Model Origin Restrictions

For security and compliance reasons, the evaluation only permits the use of US domestic models originating from US vendors, companies or institutions. Any fine-tuned, quantized, or derived variants of restricted models are also prohibited.

B.8 Execution Entry Point and Invocation Contract

Each submitted system must provide a single executable entry point inside the Docker image that serves as the system's execution interface.

The entry point:

- Must be executable at container runtime (e.g., `/app/run.sh`).
- Must read one evaluator-provided input JSON file from the mounted `/input` directory.
- Must produce one JSON output file in the mounted `/output` directory.

The specific filename of the input JSON will be determined by the evaluation infrastructure at runtime. Systems must not assume a fixed input filename and must process exactly one input file per execution.

The location and name of the output JSON file may be chosen by the system, unless otherwise specified, but the file must be written under `/output` and must conform to the required output format defined below.

The entry point may be implemented as a shell script, Python program, or compiled binary. No assumptions may be made about interactive input, network availability, or writable filesystem paths beyond `/output`.

C Appendix: Output Artifacts

Each system must produce a valid JSON output file in the `/output` directory. Failure to produce a syntactically and semantically valid output file will be treated as a submission failure.

C.0.1 Generator System Output Format

Generator systems must emit a JSON file with the following structure:

```
{
  "team": "you_provide_generator_team_id",
  "docker_id": "you_provide_generator_team_id/image_name:version",
  "input": "given_input_filename_from_execution_command",
  "narrative_list": [
    {
      "statement_id": "00001",
      "truthfulness": 0,
      "narrative": "Generated factual output here",
      "source": "ai"
    },
    {
      "statement_id": "00001",
      "truthfulness": 1,
      "narrative": "Generated misleading output here",
      "source": "ai"
    },
    ...
  ],
  "execution_time": 202.6
}
```

C.0.2 Discriminator System Output Format

Discriminator systems must emit a JSON file with the following structure:

```
{
  "team": "you_provide_discriminator_team_id",
  "docker_id": "you_provide_discriminator_team_id/image_name:version",
  "input": "given_input_filename_from_execution_command",
  "prediction_list": [
    {
      "statement_id": "00001_0",
      "ai_likelihood_score": 0.15,
      "believability_score": 0.85
    },
    {
      "statement_id": "00001_1",
      "ai_likelihood_score": 0.92,
      "believability_score": 0.45
    },
    ...
  ],
  "execution_time": 202.6
}
```

Submission Package Requirements

The following must be included when submitting your system:

- **Container Artifacts:** Docker image (by name/tag) plus `Dockerfile`, `Docker-Compose` and any build scripts.
- **Dependencies:** Manifest file `environment.yml`.
- **System Description:** Architecture summary, model versions, hardware mapping, and citations. See the Appendix D and E
- **Documentation:** README with build steps, configuration options, invocation examples, and a compliance checklist.
- **Resource Usage Report:** Expected CPU, memory, and GPU requirements.

D Appendix: System Description Template (Generator)

Suggested template: https://www.ieee.org/conferences/publishing/templates.html

D.1 Section 1. Submission Identifier(s)

List your **team ID** and the **submission IDs** for which system outputs were submitted.

D.2 Section 2. System Description

Provide a concise technical and architectural overview of your generator system.

- **Base Foundation LLM:** e.g., GPT-5.1, Claude Sonnet 4.5, etc.
- **Model Architecture:** Customized (hybrid, fine-tuned, quantized etc.), Any novel modules or pipelines (RAG, MCP, etc.)
- **Unique Features / Modifications:** Custom pre- or post-processing, Specialized prompt engineering strategies, Any adapter layers, retrieval augmentation, etc.

D.3 Section 3. Docker Image Specification

Describe the submitted docker image names & tags, a list of base images, on-disk docker image size, entrypoint & CMD, dependencies

D.4 Section 4. Training Data and Knowledge Sources

- **Training Datasets:** Dataset A (description, size, source), Dataset B (description, size, source)
- **Knowledge Sources (if used):** External APIs (e.g., Wikipedia, news feeds), Knowledge bases or retrieval corpora, Other dynamic data sources
- **Curation / Augmentation:** Filtering criteria, Data augmentation methods, Any alignment or cleaning procedures

D.5 Section 5. References

[1] A. Author, “Title of the paper,” in *Proc. XYZ*, 2023.

[2] Dataset Name, Version, URL, Year.

E Appendix: System Description Template (Discriminator)

Suggested template: https://www.ieee.org/conferences/publishing/templates.html

E.1 Section 1. Submission Identifier(s)

List your **team ID** and the **submission IDs** for which system outputs were submitted.

E.2 Section 2. System Description

Provide a concise technical and architectural overview of your discriminator system.

- **Base discriminator model:** e.g., Vicuna-RADAR, OpenAI detector, etc.
- **Model Architecture:** Highlight unique features or modifications that differentiate your system from an off-the-shelf discriminator model.
- **Unique Features / Modifications:** If applicable, specify the model architecture (e.g., hybrid systems, fine-tuning only).

E.3 Section 3. Docker Image/Container Specification

Describe the submitted docker image names & tags, a list of base images, on-disk image size, entrypoint & CMD, dependencies

E.4 Section 4. Training Data and Knowledge Sources

- **Training Datasets:** Dataset A (description, size, source), Dataset B (description, size, source)
- **Knowledge Sources (if used):** External APIs (e.g., Wikipedia, news feeds), Knowledge bases or retrieval corpora, Other dynamic data sources
- **Curation / Augmentation:** Filtering criteria, Data augmentation methods, Any alignment or cleaning procedures

E.5 Section 5. References

[1] A. Author, “Title of the paper,” in *Proc. XYZ*, 2023.

[2] Dataset Name, Version, URL, Year.