Infosys

Responsible AI Office

Infosys Responsible AI Toolkit

Playbook for Open Source users

Contents

Introduction

Responsible AI, also known as Ethical AI, is a fundamental concept in the field of artificial intelligence that emphasizes the importance of using AI in a manner that respects human well-being and ethical principles. It seeks to ensure that AI is developed and used thoughtfully and considerately, minimizing potential negative impacts on individuals and society. This involves designing and implementing AI technologies in a way that aligns with societal values, upholds human rights, and avoids unintended biases and discriminatory effects. Responsible AI principles serve as guidelines or frameworks to ensure the ethical and responsible use of AI.

To understand the importance of responsible AI principles, let's consider the example of facial recognition technology. While it has gained popularity, it has also raised concerns about privacy and potential misuse. If a company decides to implement a facial recognition system without proper safeguards or oversight, it could lead to unauthorized surveillance, profiling of individuals, privacy rights violations, potential discrimination, or false identification leading to wrongful arrests. By adhering to responsible AI principles, such as incorporating transparency, consent, and accountability measures, companies can mitigate these risks and ensure that facial recognition technology is deployed in a responsible and ethical manner.

Infosys Responsible AI Office

Infosys is working on a comprehensive framework, through its Responsible AI Office, to guide businesses in implementing Responsible AI. This framework is structured into three main components: Scan, Shield, and Steer.

Scan: This component focuses on monitoring and assessing compliance and risks. It includes tools like the Infosys Responsible AI Watchtower for monitoring external regulations, the Infosys Responsible AI Maturity Assessment and Audits for assessing compliance readiness, and the Infosys Responsible AI Telemetry for internal compliance monitoring.

Shield: The Shield component provides technical solutions to protect AI models and systems from various risks. It includes the Infosys Responsible AI Toolkit, which offers a range of solutions to safeguard AI systems, Infosys Gen AI Guardrails for moderating generative AI systems, and the Infosys Responsible AI Gateway to enforce responsible AI protocols throughout the AI lifecycle.

Steer: This part of the framework focuses on governance, legal consultation, and strategy formulation. It involves managing a dedicated Responsible AI practice, legal reviews of AI contracts with vendors, strategy development, standardized audits, and industry certifications.

Infosys Responsible AI toolkit for AI Moderation

The Infosys Responsible AI Toolkit, a technical offering within the Shield component of the 3S framework, provides solutions for detecting and addressing risks to ensure responsible AI.

The Infosys Responsible AI toolkit features a suite of APIs, known as a FM-Moderation Layer, to safeguard AI models by verifying inputs and outputs for safety, security, privacy, and fairness. Additionally, These APIs assist in providing explanations for the outcomes of various traditional and generative AI models, ensuring transparency in their decision-making processes.

The moderation layer is a modular component that can be seamlessly integrated into clients' AI systems. It enables integration with custom traditional and large language models, implementation of security measures, configuration of moderation checks, and ongoing monitoring of AI models to maintain ethical, legal, and compliance standards.

FM - Moderation Layer

Introduction

The Filtering and Moderation module (called as FM-Moderation layer) is a crucial part in content moderation systems that ensure the safety, compliance, and appropriateness of responses generated by AI models like GPT3.5 and GPT4. It applies a set of checks and filters to both the input prompt and the generated response to keep control over the content generated by the language model. The module consists of multiple layers, including Request Moderation, Response Moderation, and Response Comparison, each serving a specific purpose in the moderation process.

Components of Input

Prompt and Temperature Parameters

The FM Moderation module requires two input parameters: Prompt and Temperature. The Prompt is the first text provided to the model to generate a response, while the Temperature parameter controls the creativity of the response. A lower temperature value produces more precise and deterministic outputs, while a higher value introduces more creative elements into the response.

Multi-Lingual Support

Multilingual support in FM (content) moderation entails the ability to process prompts and content in various languages. When a prompt is provided in any language, the system automatically detects the language and translates it to English for further processing. This functionality allows users to select options such as Google, Azure, or none, while ensuring seamless backend operations with language detection and translation mechanisms in place. This approach enables efficient and effective moderation across different languages, using automated processes to manage content consistently and accurately.

Emoji Moderation

In FM moderation, emoji support allows users to include emojis to convey sentiments or expressions within sentences. By selecting the "emoji" option and specifying "yes," users show the presence of emojis in their input. Behind the scenes, the system detects these emojis and seamlessly integrates them into the moderation process. This functionality ensures that emojis are recognized and interpreted as part of the input, enabling the moderation system to effectively consider them when generating results or making decisions. This capability significantly enhances the system's ability to understand and respond to content having emojis, thereby improving the accuracy and relevance of moderation outcomes.

Prompt Template

Global Priority

Global Priority is nothing but Smooth LLM checking involves assessing an LLM prompt for potential issues or biases that could negatively affect the generated output. This includes evaluating prompt clarity, specificity, avoiding harmful stereotypes or discriminatory language, and ensuring the prompt aligns with ethical guidelines. By conducting a smooth LLM check, developers can mitigate risks of generating harmful or misleading content, enhancing the overall quality and safety of the LLM's responses.

Self-Remainder

Self Remainder is Nothing but A Bergeron check in the context of LLM prompts is a safety measure designed to prevent jailbreaking, or malicious manipulation, of the model. It involves analyzing the prompt for patterns or keywords associated with harmful or unintended behaviors. This can include detecting prompts that request harmful content, promote violence, or try to exploit vulnerabilities in the model. By showing and mitigating these prompts, Bergeron checks help ensure the LLM operates safely and ethically.

Cove Complexity

Cove Complexity can be considered a synonymous term for The Cain of Verification. It is a rigorous evaluation method for LLMs that subjects generated responses to a series of increasingly complex logical questions. By tailoring the query difficulty to the selected level (simple, medium, or complex), this process effectively probes the LLM's ability to provide right, consistent, and logically sound information. Essentially, it acts as a quality control checkpoint, identifying potential weaknesses in the model's reasoning capabilities and factual knowledge.

LLM Model

LLM explainability is still a significant challenge for both GPT-3 and GPT-4. While GPT-4 generally demonstrates improved performance, understanding the internal decision-making processes of these models is still limited. Both models use as black boxes, making it difficult to pinpoint the exact reasoning behind their outputs. Researchers and engineers are actively exploring techniques like attention visualizations and model introspection to shed light on these complex systems, but a comprehensive understanding of LLM explainability is yet to be achieved.

Components of Output

In FM moderation, two types of outputs are distinguished: template-based guardrails and model-based guardrails.

Template-based guardrails

Where we make use of dynamic and efficient prompt templates through Prompt Engineering that enhance the detection capability of the llms to detect and block the adversarial attacks.

Request Moderation

In the Request Moderation layer, various checks are performed on the input prompt before generating a response. These checks include Prompt Injection, Jailbreak, Fairness and Bias, language critique coherence, language critique fluency, language critique grammar, language critique politeness, evaluator check, context relevance, context conciseness and context reranking. These checks ensure that the input prompt adheres to the defined guidelines and standards.

Prompt Injection Check

This check evaluates the presence of injected content in the input prompt. It calculates an injection confidence score based on the prompt and compares it against a dynamic injection threshold. The result writes down whether the check passes or fails.

Jailbreak Check

The Jailbreak check examines the input prompt for potential attempts to manipulate or bypass the moderation system. It calculates a jailbreak similarity score and compares it against a dynamic jailbreak threshold. The result says whether the check passes or fails.

Fairness and Bias

In the context of request moderation, fairness and bias analysis aims to show potential discriminatory patterns or biases in the system's responses. This analysis typically examines various aspects of the moderation process.

Language Critique Coherence

An evaluation of the logical connection and consistency between distinct parts of an explanation. It figures out how well the ideas are organized and linked together to form a cohesive narrative.

Language Critique Fluency

A measure of the smoothness and naturalness of the language used in an explanation. It assesses the flow of ideas and the overall readability of the text.

Language Critique Grammar

An assessment of the grammatical correctness and adherence to language conventions in an explanation. It evaluates the syntax, punctuation, and overall linguistic accuracy of the text.

Language Critique Politeness

A measure of the appropriateness and respectfulness of the language used in an explanation. It assesses the tone and choice of words to figure out if the explanation is polite and considerate of the audience.

Evaluator Check

Before delving into the output, it's essential to understand the concept. Elevator check request moderation typically refers to a process where given requests for elevator inspections, repairs, or maintenance are reviewed and approved or rejected based on certain criteria.

Infosys Advanced Jailbreak Check

Assuming "elevator check" is a misnomer or typo, and the actual focus is on security, Infosys Advanced Jailbreak Check likely refers to a system that assesses devices or systems for vulnerabilities that could potentially allow unauthorized access or control. This might involve checking for compromised software, weak passwords, or other security risks.

Infosys Random Noise Check

This could refer to a quality control or maintenance procedure where random elevators are checked for unusual noises. The output might include a list of elevators checked, the nature of the noise (if any), and the corresponding actions taken (e.g., maintenance scheduled).

Context Relevance

Evaluates how relevant the retrieved context is to the question specified. Context relevance score measures if the retrieved context has enough information to answer the question being asked. This check is important since a bad context reduces the chances of the model giving a relevant response to the question asked, as well as leads to hallucinations.

Context Conciseness

Evaluates the concise context cited from an original context for irrelevant information. Context conciseness refers to the quality of a reference context generated from retrieved context in terms of being clear, brief, and to the point. A concise context effectively conveys the necessary information without unnecessary elaboration or verbosity.

Context Reranking

Evaluates how efficient the reranked context is compared to the original context. Context Reranking reflects the efficiency of the reranking process applied to the original context in generating the new renamed context used to answer a given question. This operator assesses the degree to which the reranked context enhances the relevance, coherence, and informativeness with respect to the provided question.

Response Moderation

In the Response Moderation layer, the generated response from the Request Moderation layer is further evaluated before being presented to the user. This evaluation includes checks for language critique coherence, language critique fluency, language critique grammar, language critique politeness y. All the checks which are there in Request Moderation layer are same with response Moderation also except for response completeness, response conciseness, response validity and response completeness wrt context.

Response Completeness

Checks whether the response has answered all the aspects of the question specified. Response completeness score measures if the generated response has adequately answered all aspects to the question being asked. This check is important to ensure that the model is not generating incomplete responses.

Response Conciseness

Grades how concise the generated response is or if it has any other irrelevant information for the question asked. Response conciseness score measures whether the generated response holds any other information irrelevant to the question asked.

Response Validity

Checks if the response generated is valid or not. A response is valid if it holds any information. In some cases, an LLM might not generate a response due to reasons like limited knowledge or the asked question not being clear. Response Validity score can be used to name these cases, where a model is not generating an informative response.

Response Completeness With respect to Context

Response completeness with respect to context in response moderation refers to the ability of a moderation system to accurately assess and address the content of a response based on its surrounding context. This involves understanding the nuances of the conversation, the intent of the user, and the potential impact of the response.

Response Comparison

The Response Comparison element compares the generated response from the FM Moderation module with a reference response from Infosys guardrail. This comparison helps ensure consistency and evaluate the effectiveness of the moderation process.

The FM Moderation module ensures that the generated responses are compliant, safe, and aligned with defined guidelines. It provides granular control over the content generated by AI language models, reducing risks associated with inappropriate or harmful outputs.

Response with Infosys RAI guardrails

Infosys RAI guardrails for response comparison likely involve a system that evaluates and compares generated responses against predefined quality standards and ethical guidelines. This process ensures that AI-generated content aligns with Infosys' values, is exact, relevant, and free from biases. By comparing responses to these guardrails, the system can name potential issues, suggest improvements, and keep a prominent level of quality and consistency in the AI outputs.

Results with gpt4

Comparing ChatGPT-4 responses involves analyzing the quality, relevance, and coherence of its outputs across various prompts and contexts.

Model-based guardrails:

Model-based guardrails we use Pre-trained models trained on extensive billion-parameter datasets to effectively combat adversarial attacks. These models are equipped with sophisticated algorithms capable of discerning and mitigating malicious content by employing various metrics such as scores and thresholds. These metrics are meticulously set based on detection entities or through comparative analysis of text embeddings, ensuring robust detection and response mechanisms against adversarial threats. This approach enables our system to support ambitious standards of security and reliability, safeguarding against potential risks and ensuring the integrity of moderated content.

Request Moderation

In the Request Moderation layer, various checks are performed on the input prompt before generating a response. These checks include Prompt Injection, Jailbreak, Privacy, Profanity, Toxicity, and Restricted Topic checks. These checks ensure that the input prompt adheres to the defined guidelines and standards.

Prompt Injection Check

This check evaluates the presence of injected content in the input prompt. It calculates an injection confidence score based on the prompt and compares it against a dynamic injection threshold. The result writes down whether the check passes or fails.

Jailbreak Check

The Jailbreak check examines the input prompt for potential attempts to manipulate or bypass the moderation system. It calculates a jailbreak similarity score and compares it against a dynamic jailbreak threshold. The result writes down whether the check passes or fails.

Privacy Check

The Privacy check allows configuration of specific entities that should be recognized and protected within the prompt, such as Aadhar numbers, passport details, or PAN numbers. It names the recognized entities and compares them against the configured entities to block. The result writes down whether the check passes or fails.

Profanity Check

The Profanity check names profane words within the input prompt. It reports the profane words named and compares them against a customized threshold. If the number of occurrences of profane words is greater than the threshold, it falls to failed category.

Toxicity Check

The Toxicity check assesses the level of toxicity in the input prompt based on metrics such as toxicity, severe toxicity, obscenity, identity attack, insult, threat, and sexual explicitness. It compares the calculated toxicity scores against a predefined toxicity threshold and at last we can assess whether response passes or fails.

Restricted Topic Check

The Restricted Topic check ensures that certain predefined topics, such as explosives, terrorism, or political subjects, are not included in the input prompt. We can configure what topics needs to be restricted.

Response Moderation

In the Response Moderation layer, the generated response from the Request Moderation layer is further evaluated before being presented to the user. This evaluation includes checks for Text Quality, Text Relevance, and Refusal. All the checks which are there in Request Moderation layer are same with response Moderation also except jailbreak check and Prompt injection check since the response from the LLM may not hold these attacks. Other than it has Privacy Check, Profanity Check, Toxicity Check, Restricted Topic Check.

Privacy Check

The Privacy check allows configuration of specific entities that should be recognized and protected within the prompt, such as Aadhar numbers, passport details, or PAN numbers. It finds the recognized entities and compares them against the configured entities to block. The result shows whether the check passes or fails.

Restricted Topic Check

The Restricted Topic check ensures that certain predefined topics, such as explosives, terrorism, or political subjects, are not included in the input prompt. We can configure what topics needs to be restricted.

Toxicity Check

The Toxicity check assesses the level of toxicity in the input prompt based on metrics such as toxicity, severe toxicity, obscenity, identity attack, insult, threat, and sexual explicitness. It compares the calculated toxicity scores against a predefined toxicity threshold and at last we can assess whether response passes or fails.

Profanity Check

The Profanity check finds profane words within the input prompt. It reports the profane words shown and compares them against a customized threshold. If the number of occurrences of profane words is greater than the threshold, it falls to failed category.

Text Quality

The Text Quality check assesses the readability and grade level of the generated response. Score stands for the grade level using this scale below.

Score	**School level**	**Remarks**
100.00-90.00	5th grade	Quite easy to read. Easily understood by an average 11-year-old student.
90.0-80.0	6th grade	Easy to read. Conversational English for consumers.
80.0-70.0	7th grade	Fairly easy to read.
70.0-60.0	8th & 9th grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0-50.0	10th to 12th grade	Fairly difficult to read.

Score	School level	Remarks
50.0-30.0	College	Difficult to read.
30.0-10.0	College graduate	Exceedingly difficult to read. Best understood by university graduates.
10.0-0.0	Professional	Extremely difficult to read. Best understood by university graduates.

Text Relevance Check

The Text Relevance check measures the relevance of the generated response with respect to the input prompt.

Refusal Check

The Refusal check finds cases where the language model refuses to provide a response due to content that violates norms or guidelines.

Response Comparison

The Response Comparison element compares the generated response from the FM Moderation module with a reference response from Infosys guardrail. This comparison helps ensure consistency and evaluate the effectiveness of the moderation process.

The FM Moderation module ensures that the generated responses are compliant, safe, and aligned with defined guidelines. It provides granular control over the content generated by AI language models, reducing risks associated with inappropriate or harmful outputs.

Response with Infosys RAI guardrails

Infosys RAI guardrails for response comparison likely involve a system that evaluates and compares generated responses against predefined quality standards and ethical guidelines. This process ensures that AI-generated content aligns with Infosys' values, is exact, relevant, and free from biases. By comparing responses to these guardrails, the system can find potential issues, suggest improvements, and keep a prominent level of quality and consistency in the AI outputs.

Results with gpt4

Comparing ChatGPT-4 responses involves analyzing the quality, relevance, and coherence of its outputs across various prompts and contexts.

Endpoints and Functionalities

The following table lists the available Moderation Layer endpoints and their functionalities. Refer to the API documentation: Moderation Layer Models

Endpoint	Description
/rai/v1/moderations	This API provides the decoupled guardrail (checks for the prompt like – privacy check, prompt injection check, jailbreak check, toxicity check, restricted topic, custom theme check).
/rai/v1/moderations/coupledmoderations	This API provides the coupled guardrail (provides checks for input prompt, LLM interaction for generating response and checks for response)
/rai/v1/moderations/getTemplates/<userId>	To retrive existing prompt engineering templates
/rai/v1/moderations/evalLLM	Using this API, we can check our prompt for various checks like prompt injection, jailbreak, language coherence etc. using LLM as evaluator using various templates.
/rai/v1/moderations/multimodal	Using this API, we can check our prompt and image for various checks like prompt injection, jailbreak etc. using GPT4o.
/rai/v1/moderations/translate	Using this API, we can use google or azure translate to convert text in any language to English.
/rai/v1/moderations/openai	Using this API, we can get response for the prompt passed from openAI.
/rai/v1/moderations/openaiCOT	Using this API, we can get explainability by 'chain of thoughts' the LLM went through to provide response to our prompt
/rai/v1/moderations/healthcareopenaiCOT	Using this API, we can get the 'chain of thoughts' the LLM went through to provide response to our prompt, adding in example prompt response to tell the LLM which details to be included in the response and what format the response should be in. (few shot prompting)
/rai/v1/moderations/openaiTHOT	Using this API, we can get the explainability using 'thread of thoughts' the LLM went through to provide response to our prompt, we can see how the LLM break down the prompt to correctly understand it and to generate response.
/rai/v1/moderations/setTelemetry	Using this API, we can check if telemetry is working or not, we can select payload as True or False.

/rai/v1/moderations/ToxicityPopup	Using this API, we can check if our entered prompt is toxic or not, we can also see which type of toxicity label the prompt belongs to as well as the respective toxicity score under that category and a cumulative toxicity score.
/rai/v1/moderations/ProfanityPopup	Using this API, we can check if our entered prompt contains any profane words.
/rai/v1/moderations/PrivacyPopup	Using this API, we can check if our entered prompt contains any PII entities
/rai/v1/moderations/COV	Using this API, we can see the ‘chain of verification’ or questions the LLM asked itself to reach the response it gave us. We can give ‘gpt4’, ‘gpt3’ or ‘Llama’ as model names.
rai/v1/moderations/OrgPolicy	Using this API, we could see if the prompt passed is associated with any restricted topic we have passed in ‘labels’. In labels we can add the restricted topics under which we want to test our prompt, like – ‘terrorism’, ‘explosives’, ‘fraud’, ‘cheating’ etc.
/rai/v1/moderations/gEval	Using this API, we can compare the text and the summary provided. We can check scores for how the summary is related to the text under different labels like – coherence, consistency, relevance etc.
/rai/v1/moderations/Hallucination_Check	Using this API, we can check if the provided prompt is related to sources provided.

Explainability

Introduction

Explainability in AI, including Machine Learning and Large Language Models, is about understanding the logic behind a model's decisions. By revealing the reasoning process, it builds trust and improves model performance. It helps us spot biases, fix mistakes, and align models with our goals. Ultimately, explainability makes AI more reliable and trustworthy. *Infosys Responsible AI* toolkit provides a range of explainability techniques to enhance transparency and understanding of AI model decisions

Explainability Features

Explainability aims at understanding how AI models arrive at its predictions. It involves breaking down the complex decision-making process of the model into human-understandable terms.

Feature Importance is a critical component of Model Interpretability, used to understand the relative importance of different features (variables) an AI model prediction. By quantifying the contribution of each feature, it helps to explain the model's decision-making process.

AI models require explanations of key features at both the overall model level and the individual instance level, termed global and local interpretability. Brief descriptions of terms related to model interpretability are outlined below

Global Interpretability

Understanding the overall behavior and decision-making process of a model across its entire input space. It provides insights into the general patterns and trends that the model has learned from the data.

Local Interpretability

The ability to understand how a machine learning model arrives at its predictions for specific instances. Unlike global interpretability, which focuses on understanding the overall behavior of the model, local interpretability provides explanations tailored to individual predictions.

SHAP (SHapley Additive exPlanations)

a game-theoretic method for explaining the output of any machine learning model. It calculates the contribution of each feature to a prediction, providing a more comprehensive understanding of the model's decision-making process.

Purpose: Feature ranking, Model understanding, Bias detection, Debugging

Applicability: SHAP is a model-agnostic method and applied to wide range of machine learning models

(e.g., Decision trees, Random forests, Gradient boosting machines, Neural networks, Support vector machines)

LIME (Local Interpretable Model-Agnostic Explanations)

a technique used in machine learning to explain the predictions of complex models in a locally interpretable way. It works by approximating a complex model with a simpler, linear model around a specific prediction.

Purpose: Local interpretability, Model agnosticism, Feature importance, Bias detection, Debugging

Applicability: LIME is a model-agnostic method and applied to wide range of machine learning models

(e.g., Decision trees, Random forests, Gradient boosting machines, Neural networks, Support vector machines)

Anchor Tabular

a technique used in explainability to identify a minimal set of conditions (anchors) that are sufficient to explain a prediction made by a machine learning model. These anchors are human-readable rules that capture the essence of the model's decision-making process for a specific instance.

Purpose: Local interpretability, Simplicity, Feature importance, Bias detection, Debugging

Applicability: Anchor Tabular is primarily designed for tabular data, but it can also be applied to other types of data with some modifications. It can be used to explain the predictions of various machine learning models (e.g., Decision trees, Random forests, Gradient boosting machines, Neural networks, Support vector machines)

Self-reasoning techniques

a set of techniques that enable LLMs to generate more comprehensive and coherent explanations for their outputs. These frameworks often involve breaking down complex problems into smaller, more manageable steps and guiding the LLM through a reasoning process. This technique relies heavily on prompt engineering, with thoughtfully crafted prompts leading to informative outcomes.

Few models built on self-reasoning framework are:

Chain of Thought (CoT)

Step-by-step reasoning technique prompts the LLM to break down a complex task into smaller, more manageable steps and explain its reasoning for each step. This allows users to follow the LLM's thought process and understand how it arrived at its conclusion.

Thread-of-Thought (ThoT)

Generate a series of interconnected thoughts that form a coherent narrative. This helps to visualize the LLM's reasoning process and identify any inconsistencies or biases.

Graph-of-Thought (GoT)

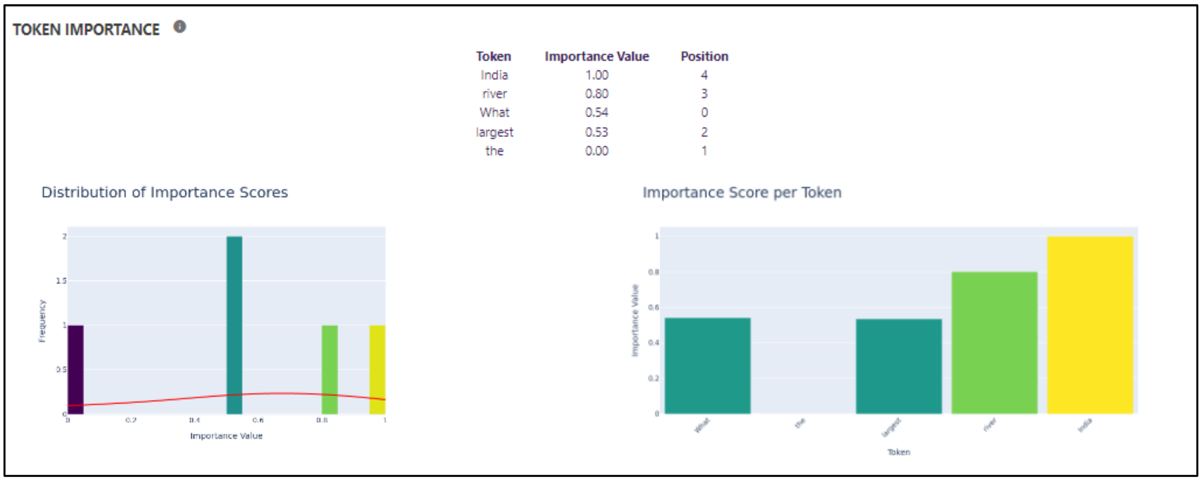
represents the LLM's reasoning process as a graph, where nodes represent intermediate thoughts and edges represent the connections between them. This visual representation can be helpful for understanding the LLM's decision-making process.

Chain-of-Verification (CoV)

prompts the LLM to verify its responses against external knowledge sources. This helps to ensure the accuracy and reliability of the LLM's outputs.

Attention Visualization using Token Importance Charts

In AI language models, the importance of tokens can significantly influence the generated responses. Understanding which tokens (words or phrases) are most impactful can be crucial for interpreting and trusting the model's decisions. This is particularly relevant in applications where precise and reliable outputs are essential, such as in healthcare, finance, and legal domains.



TOKEN IMPORTANCE

Token	Importance Value	Position
India	1.00	4
river	0.80	3
What	0.54	0
largest	0.53	2
the	0.00	1

Endpoints and Functionalities

The following table lists the available Moderation Layer endpoints and their functionalities. Refer to the API documentations : Traditional Models , Gen AI Models

Explainability for LLMs

Endpoint	Description
/rai/v1/llm-explainability/sentiment-analysis	Sentiment Analysis based on input prompt and displays token importance charts
/rai/v1/llm-explainability/uncertainty	Generates uncertainty and coherence score for the given input
/rai/v1/llm-explainability/token-importance	Gives a table of important tokens and associated scores for the given input
/rai/v1/llm-explainability/got	Provides explainability based on Graph of Thoughts technique
/rai/v1/llm-explainability/serper_response	Extracts facts from Google internet search based on given input text
/rai/v1/moderations/openaiCOT	Provides explainability using Chain of Thoughts
/rai/v1/moderations/openaiTHOT	Provides explainability using Thread of Thoughts
/rai/v1/moderations/COV	Provides explainability using Chain of Verification
/rai/v1/moderations/healthcareopenaiCOT	Get explainability using Chain of Thoughts in RAG scenario

Explainability for traditional ML models

Endpoint	Description
/rai/v1/explainability/methods/get	Returns list of explanation methods applicable for selected model and dataset
/v1/workbench/adddata, /v1/workbench/data	To upload Data and retrieve dataid and name respectively
/v1/workbench/addmodel, /v1/workbench/model	To upload Modle file and retrieve modelid and name respectively
/v1/workbench/addpreprocessor, /v1/workbench/preprocessor	To upload encoder file and retrieve encoderid and name respectively
/v1/workbench/batchgeneration	Maps data, model and preprocessor to create a pipeline for explainability generation
/rai/v1/explainability/explanation/get	Get explanation for the given model and dataset
/rai/v1/explainability/report/generate	Generates a doc report for the given input of data, model, preprocessor for a specific batchid
/v1/report/downloadreport	To download the report generated against specific batchid

Fairness & Bias

Introduction

The AI Fairness module is an essential component in the field of responsible AI and aims to address and mitigate biases in machine learning models, specifically it works with data sets. Biases can emerge from various factors like race, sex, religion, and other sensitive attributes present in the dataset. The goal of AI Fairness is to ensure that these biases do not influence the outcomes generated by the models, promoting fairness, and mitigating discriminatory effects.

The AI Fairness module helps identify and address biases in machine learning models, promoting fairness, and mitigating the impact of discriminatory outcomes. By incorporating fairness considerations, organizations can foster more equitable and unbiased decision-making processes.

Fairness & Bias Components

Text: [Evaluate texts generated by LLM]

Given an un-structured text, a prompt template has been designed to check the text's fairness and evaluate its bias indicator [High / Medium / Low / Neutral] using GPT-4o. The prompt template also provides additional information like "Affected group", indicating the group of people affected by the context of the sentence and also the type of Bias [Historical Bias, Confirmation Bias, etc.,] as well. We are working to extend this prompt template to generate the neutral versions of the given text

Image: [Evaluate images generated by LLM]

"A picture can speak 1000 words. "

As good as this statement is, the context perceived about the picture just by looking at it can also differ from person to person. With this established, to see if a given picture / image is Fair or biased, we depend on the input prompt given to the LLM to generate this particular image. The input prompt given by the user sets the contextual expectation of the user and the picture / image generated can be validated with similar context. For the template-based approach, we are currently leveraging GPT-4o 's multimodal capabilities for evaluation. We have plans to extend this to Gemini as well in the future.

Fairness & Bias Evaluations for Traditional Models [Structured Data]

Based on the selected sensitive / protected attribute for the given dataset, the positive / favorable outcome distribution is compared with the rest of the groups in the dataset and the metrics are calculated.

i. Statistical Parity Difference:

The Statistical parity difference metric calculates the difference in the ratio of favorable outcomes between privileged groups and un-privileged groups.

$$\text{SPD} = P(\widehat{Y} = 1 \mid A = \text{minority}) - P(\widehat{Y} = 1 \mid A = \text{majority}), \text{ where } \widehat{Y} \text{ are the model predictions and } A \text{ is the group of the sensitive attribute.}$$

ii. Disparate Impact Ratio:

The Disparate Impact Ratio metric calculates the ratio of favorable outcomes between privileged groups and un-privileged groups.

$$\text{DI} = P(\widehat{Y} = 1 \mid A = \text{minority}) / P(\widehat{Y} = 1 \mid A = \text{majority}), \text{ where } \widehat{Y} \text{ are the model predictions and } A \text{ is the group of the sensitive attribute.}$$

iii. Smooth Empirical Differential:

SED calculates the differential in the probability of favorable and unfavorable outcomes between intersecting groups divided by features. All intersecting groups are equal, so there are no unprivileged or privileged groups. The calculation produces a value between 0 and 1 that is the minimum ratio of Dirichlet smoothed probability for favorable and unfavorable outcomes between intersecting groups in the dataset.

iv. Four Fifths:

This function computes the four fifths rule (ratio of success rates) between group_unprivileged and group_privileged. The minimum of the ratio taken both ways is returned. A value of 1 is desired. Values below 1 are unfair. The range (0.8,1) is considered acceptable.

v. Cohen's D:

This function computes the Cohen D statistic (normalized statistical parity) between group_unprivileged and group_privileged.A value of 0 is desired. Negative values are unfair towards group_unprivileged. Positive values are unfair towards group_privileged. Reference values: 0.2 is considered a small effect size, 0.5 is considered medium, 0.8 is considered large.

$$\text{Cohen's d} = (\bar{x}_1 - \bar{x}_2) / \sqrt{(s_1^2 + s_2^2) / 2}$$

Mitigation of Bias - Reweighing

To ensure fairness in AI, we can adjust the importance of different data points during training. Additionally, we can modify the decision-making process to ensure equal outcomes for different groups. By combining these techniques, we can create AI models that are both accurate and equitable. Following are some of the approaches to mitigate bias in AI models.

Reweighing is a preprocessing technique that Weights the examples in each (group, label) combination differently to ensure fairness before classification.

Endpoints and Functionalities

The following table lists the available Fairness & Bias endpoints and their functionalities. Refer to the API documentation : [Fairness & Bias Models]

Endpoint	Description
/api/v1/fairness/Analyse	Detects bias in traditional ML models and datasets using statistical techniques like Statistical Parity Difference, Disparate Impact Ratio, Smooth Emperical Differential, Four-fifths and Cohen's D
/ap/v1/fairness/inprocessing/ exponentiated_gradient_reduction	Detects bias in traditional ML models using exponential gradient reduction method
/api/v1/fairness/individualMetrics	Analyze the pretrain data and post-train data [with model’s predictions] for group bias using metrics like Statistical parity
api/v1/fairness/pretrain/mitigation/getDataset	Generate metrics score for provided dataset
/api/v1/fairness/analysis/llm	Generates response of bias analysis from text prompt
/api/v1/fairness/analysis/image	Generates response of bias analysis from Image

Hallucination

Introduction

Hallucinations in LLMs occur when the model's internal biases, learned patterns, or overgeneralizations lead to the creation of text that is not consistent with information. This can happen when the model struggles to distinguish between real and imagined information.

To address hallucinations in LLMs, it is essential to detect, quantify, and mitigate them. The Infosys Responsible AI toolkit offers features to achieve this. By employing techniques such as Chain of Thought, Thread of Thought, and Graph of Thought, potential hallucinations in diverse LLM outputs can be identified. Internet searches aid in verifying the accuracy of responses. G Eval metrics like adherence, faithfulness, and correctness assess the degree of hallucinations in both original and refined LLM outputs.

Components of Hallucination

Thread of Thoughts (ThoT)

Thread of Thoughts (ThoT) is developed to address challenges in chaotic contexts, where LLMs struggle to sift through and prioritize relevant information amidst a plethora of data.

This is a novel prompting technique designed to enhance the reasoning capabilities of Large Language Models (LLMs) in handling chaotic contexts. The approach involves systematic segmentation, summarization, and analysis, which aligns with human cognitive patterns. It draws inspiration from human cognitive processes and aims to systematically segment and analyze extended contexts for better comprehension and accuracy.

G-Eval Metrics

The G-eval metrics are tools used to evaluate the performance and quality of texts generated by LLMs, taking into account the many specific nuances of these models. The G-

Eval metrics leverage existing AI models and evaluate metrics like relevance, adherence, correctness, and faithfulness to assess the response generated by the LLM.

Chain of Verification (CoVe)

This measure is implemented directly to counteract hallucinations. As a reminder, hallucinations occur when the LLM responds incorrectly but in a logically coherent manner to a prompt.

Chain of Thoughts (CoT)

CoT refers to a structured approach in problem-solving, where complex tasks are broken down into a sequence of logical steps. It mirrors human reasoning by constructing a coherent argument from premises to a conclusion. In artificial intelligence, CoT prompting involves systematically reasoning through a problem rather than providing context-based responses. It enhances problem-solving capabilities.

CoT can be relevant to understanding hallucinations. CoT can help break down the phenomenon, considering factors like neural processes, sensory input, and cognitive biases. CoT ensures that AI-generated responses maintain coherence, avoiding hallucinatory or nonsensical outputs.

Chain of Thoughts (CoT)

CoT refers to a structured approach in problem-solving, where complex tasks are broken down into a sequence of logical steps. It mirrors human reasoning by constructing a coherent argument from premises to a conclusion. In artificial intelligence, CoT prompting involves systematically reasoning through a problem rather than providing context-based responses. It enhances problem-solving capabilities.

CoT can be relevant to understanding hallucinations. CoT can help break down the phenomenon, considering factors like neural processes, sensory input, and cognitive biases. CoT ensures that AI-generated responses maintain coherence, avoiding hallucinatory or nonsensical outputs.

Hallucination Score

The hallucination score is determined through G- Eval metrics values and assessing similarity score across multiple categories, including input prompt to output prompt, input prompt to source, and output prompt to source. This score is a weighted average that considers the scores from these below categories.

Endpoints and Functionalities

The following table lists the available Hallucination endpoints and their functionalities. Refer to the API documentation: Hallucination Models

Endpoint	Description
/rag/v1/FileUpload	It is used to upload document with pdf format and return vectorestoreid of vectorestore and blobname for the pdf.
/rag/v1/RetrievalKepler	This will take keys FileUpload, text, vectorestoreid as Input and will return a rag response along with the score.
/rag/v1/cov	It is used for getting the response with 5 more variants of questions generated by LLM to verify the answer and a refined final response
/rag/v1/cot	Provides the explanation steps and reasoning behind and from where (source referred which document)
/rag/v1/thot	Provides more descriptive and complex information that are spread over the file
/rag/v1/geval	Assess the alignment of the generated text with real-world information and can help detect hallucinations
/rag/v1/caching	This will take blobname which is generated while uploading file and return a array of length
/rag/v1/removeCache	It will take input as cache id and will remove the cached Vectorestore from the Cache

Privacy and Safety

Introduction

In the era of growing concerns over data privacy and protection, the Privacy-Analyze and Privacy-Anonymize modules provide crucial functionalities to safeguard sensitive information within unstructured text data. These modules focus on identifying Personally Identifiable Information (PII) and anonymizing it to address privacy concerns effectively. By employing advanced algorithms and techniques, these modules enable users to protect personal data while retaining the contextual details of the text. AI technologies often collect and analyze large amounts of personal data, raising issues related to data privacy and security. To address these concerns, it is crucial to promote stringent data protection regulations and the adoption of secure data handling practices.

Components of Privacy and Safety

PII Detection: The Privacy-Analyze module scans the input text to identify PII such as Aadhar numbers, PAN numbers, and confidential company data. It provides a comprehensive summary of the detected PII entities, along with corresponding metric scores. Higher scores indicate a higher likelihood of PII presence, empowering users to assess the sensitivity of the identified information.

Anonymization: The Privacy-Anonymize module replaces the identified PII entities with generic placeholders while preserving the overall structure and context of the text. Personal information such as names, locations, appointment dates, passport numbers, Aadhaar numbers, and PAN numbers are anonymized to protect individual privacy. This ensures that the text remains meaningful while safeguarding sensitive data.

Data Protection: The Privacy-Analyze module assists organizations and individuals in identifying and understanding potential privacy risks within unstructured text data. It helps to ensure compliance with privacy regulations and mitigate the risk of unauthorized data exposure.

Enhanced Privacy: The Privacy-Anonymize module enables the secure sharing and analysis of text data containing PII. It is particularly useful in scenarios where data needs to be anonymized for research, analysis, or collaboration purposes while maintaining privacy standards.

Confidentiality Preservation: Both modules aid in preserving the confidentiality of sensitive information in unstructured text data. This is crucial in industries such as healthcare, finance, and legal, where privacy and data protection are paramount.

Endpoints and Functionalities

The following table lists the available Fairness & Bias endpoints and their functionalities.
Refer to the API documentation: Privacy Models, Safety Models

Privacy

Endpoint	Description
/rai/v1/privacy/text/analyze	Checks if the input text contains any PII entities or not.
/rai/v1/privacy/text/anonymize	Anonymizes all PII entities in the input text.
/rai/v1/privacy/text/encrpyt	Encrypts all PII entities in the input text
/rai/v1/privacy/text/decrpyt	<to be updated>
/rai/v1/privacy/image/analyze	Analyzes the uploaded image for any PII entities
/rai/v1/privacy/image/anonymize	Anonymizes the PII entities present in the uploaded image
/rai/v1/privacy/image	Hashes out the PII entities detected in the uploaded image
/rai/v1/privacy/dicom/anonymize	Identifies and anonymizes the PII entities present in dicom images
/rai/v1/privacy/code/anonymize	Identifies and anonymizes the PII entities present in the code entered as text
/rai/v1/privacy/codefile/anonymize	Identifies and anonymizes the PII entities present in the code file that we uploaded as input
/rai/v1/privacy/DifferentialPrivacy/file, /rai/v1/privacy/DifferentialPrivacy/anonymize	Using the first API, we can upload the file we want to check for differential privacy and using the second API we can add suppression, noise etc. to the file values

Safety

Endpoint	Description
/api/v1/safety/profanity/analyze	check if the text contains any profane words or not and we can get the toxicity score for the same
/api/v1/safety/profanity/censor	Censors any profane words identified in the text
/api/v1/safety/profanity/imageanalyze	Checks if the image falls under any of the following labels – drawings, hentai, porn, neutral or sexy
/api/v1/safety/profanity/imageGenerate	Generate images based on the prompt and can check under which label(drawings, hentai, porn, neutral or sexy) they would fall
/api/v1/safety/profanity/videosafety	Check under which labels the uploaded video belongs to – drawings, hentai, neutral, porn or sexy and can mask those profane objects in the video
/api/v1/safety/profanity/nudanalyze	Checks if the uploaded image or video contains any nudity and can blur the same.

Security

Introduction

The introduction of a security module in RAI is a crucial step to safeguard the integrity and reliability of the platform. By incorporating robust security measures, RAI aims to protect user data, prevent unauthorized access, and ensure the platform's resilience against cyber threats. This module likely encompasses various security protocols, encryption techniques, authentication mechanisms, and risk management strategies to create a secure environment for users and their transactions.

Security module is a software component safeguarding digital systems by implementing robust protection mechanisms. Key features include authentication, authorization, encryption, intrusion detection, and access control to prevent unauthorized access, data breaches, and system failures.

Features of a Security Module

A security module typically incorporates the following features:

- Authentication: Verifies user identity to grant access.
- Authorization: Defines permitted actions based on user roles and privileges.
- Encryption: Protects data confidentiality through secure encoding.
- Intrusion Detection: Monitors for suspicious activities and threats.
- Access Control: Restricts system access to authorized individuals.
- Audit Logging**:** Records system activities for security analysis.
- Data Integrity: Ensures data accuracy and consistency.
- Non-repudiation: Prevents denial of actions or transactions.

Component of Security

Bulk Vulnerability Assessment

It is a comprehensive model assessment report that involves simulating a variety of potential attacks on the model to identify vulnerabilities. This process allows us to understand how the model might be compromised under different scenarios. In addition to analyzing these threats, we are also proposing a defense strategy designed to protect user models from the predicted attacks. This dual approach ensures that we not only identify weaknesses but also provide actionable solutions to enhance the security of the models.

Endpoints and Functionalities

The following table lists the available Security endpoints and their functionalities. Refer to the API documentation: Security Models

Endpoint	Description
/v1/workbench/adddata	Use for Uploading data
/v1/workbench/data	To retrieve the details of the uploaded data
/v1/workbench/addmodel	Use for Model Upload
/v1/workbench/model	To retrieve the details of the uploaded model
/rai/v1/security_workbench/attack	To check which attacks is applicable, user have to provide classifier and data type to this api endpoint
/v1/workbench/batchgeneration	Use for Batchgeneration
/rai/v1/security_workbench/runallattacks	Use for creating the report and storing it in zip format in the database.
/v1/report/downloadreport	Use for downloading the generated report

Contact

Infosysraitoolkit@infosys.com