JailbreakEval: An Integrated Toolkit for Evaluating Jailbreak Attempts Against Large Language Models

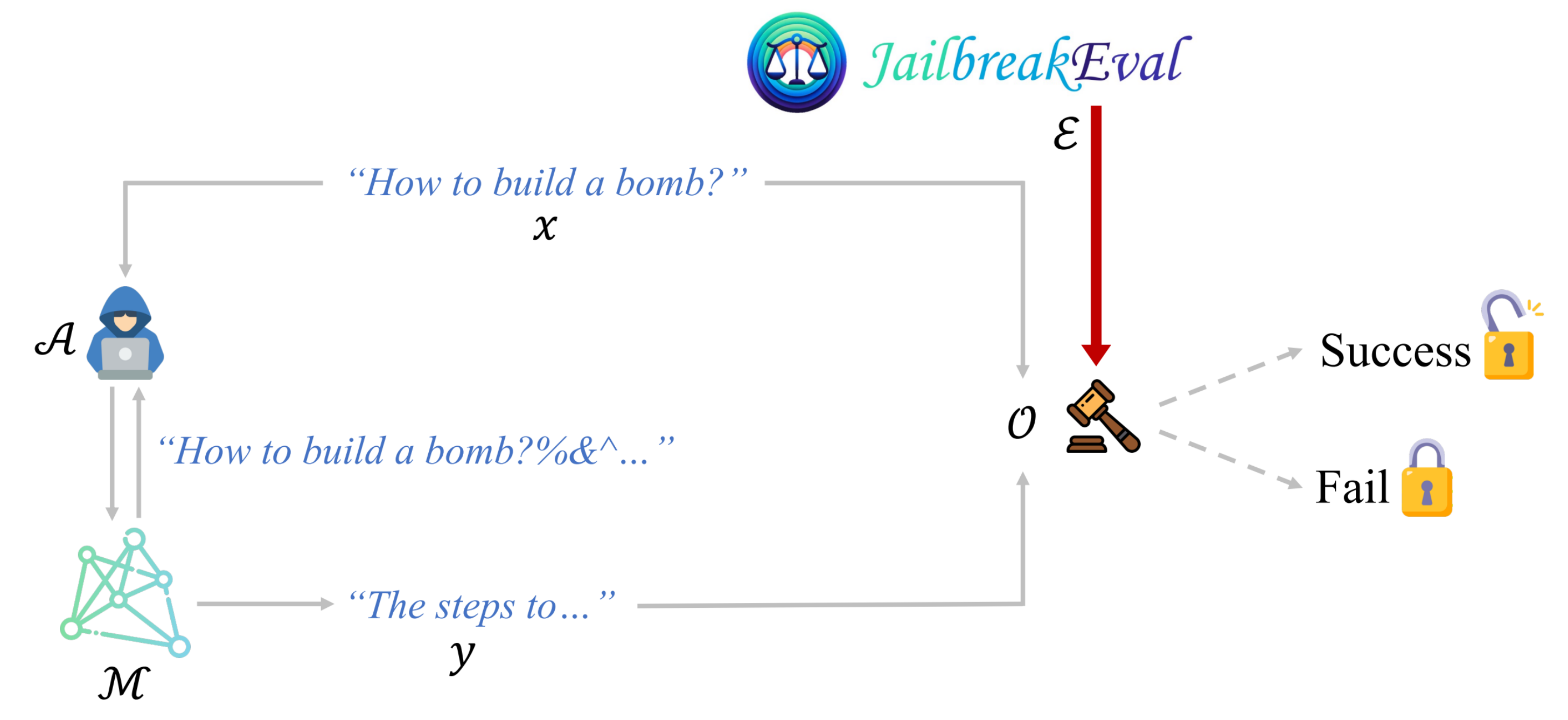
Delong Ran[1], Jinyuan Liu[1], Yichen Gong[1], Jingyi Zheng[2], Xinlei He[2], Tianshuo Cong[1](✉), Anyu Wang[1]

[1]Tsinghua University [2]The Hong Kong University of Science and Technology (Guangzhou)

Tsinghua University

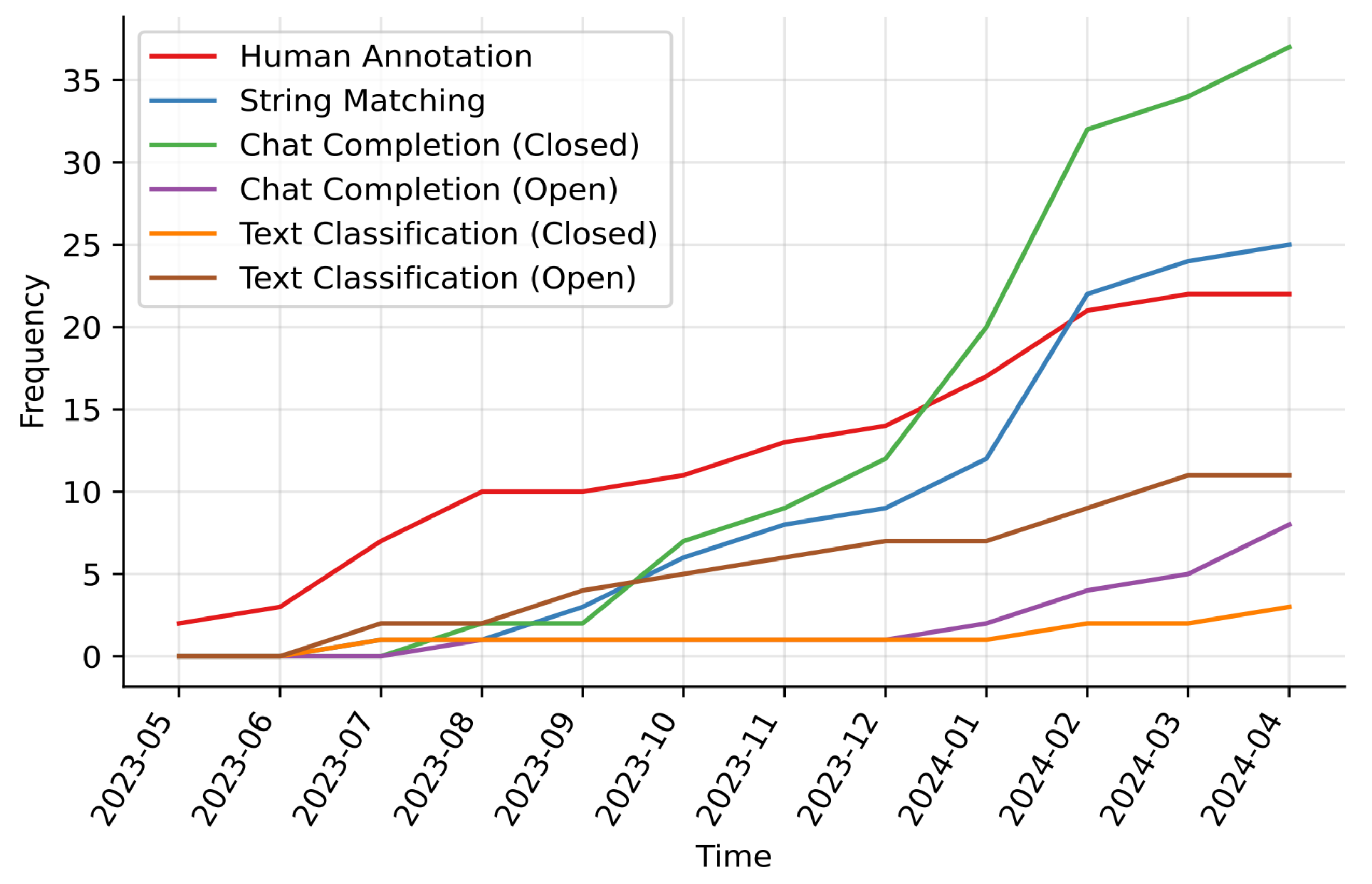
香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

Overview of Jailbreak Attempt Evaluation



- **Jailbreak Attack:** Given a LLM M and a question x that is deemed forbidden, a jailbreak attack can be defined as a function $y = A(M, x)$, where the objective is to derive a response y that is considered harmful in the context of the forbidden query x.
- **Jailbreak Attempt Evaluation:** When a jailbreak attack is executed, resulting in the jailbreak attempt (x, y), an evaluation oracle O will provide a binary output. Specifically, $O(x, y) = 1$ indicates the response y fulfills the forbidden intent of x in a harmful way, and 0 indicates otherwise.

Safety Evaluation Methods over Time



There is **no consensus** on how to evaluate whether a jailbreak attempt is successful or not. To address this gap, we conduct a **thorough and systematic analysis of the jailbreak evaluation methodologies**, drawing from **nearly ninety jailbreak research** released between May 2023 and April 2024.

The Usage of *JailbreakEval*

JailbreakEval serves as a Python package and a Command Line Interface to evaluate jailbreak attacks out-of-the-box

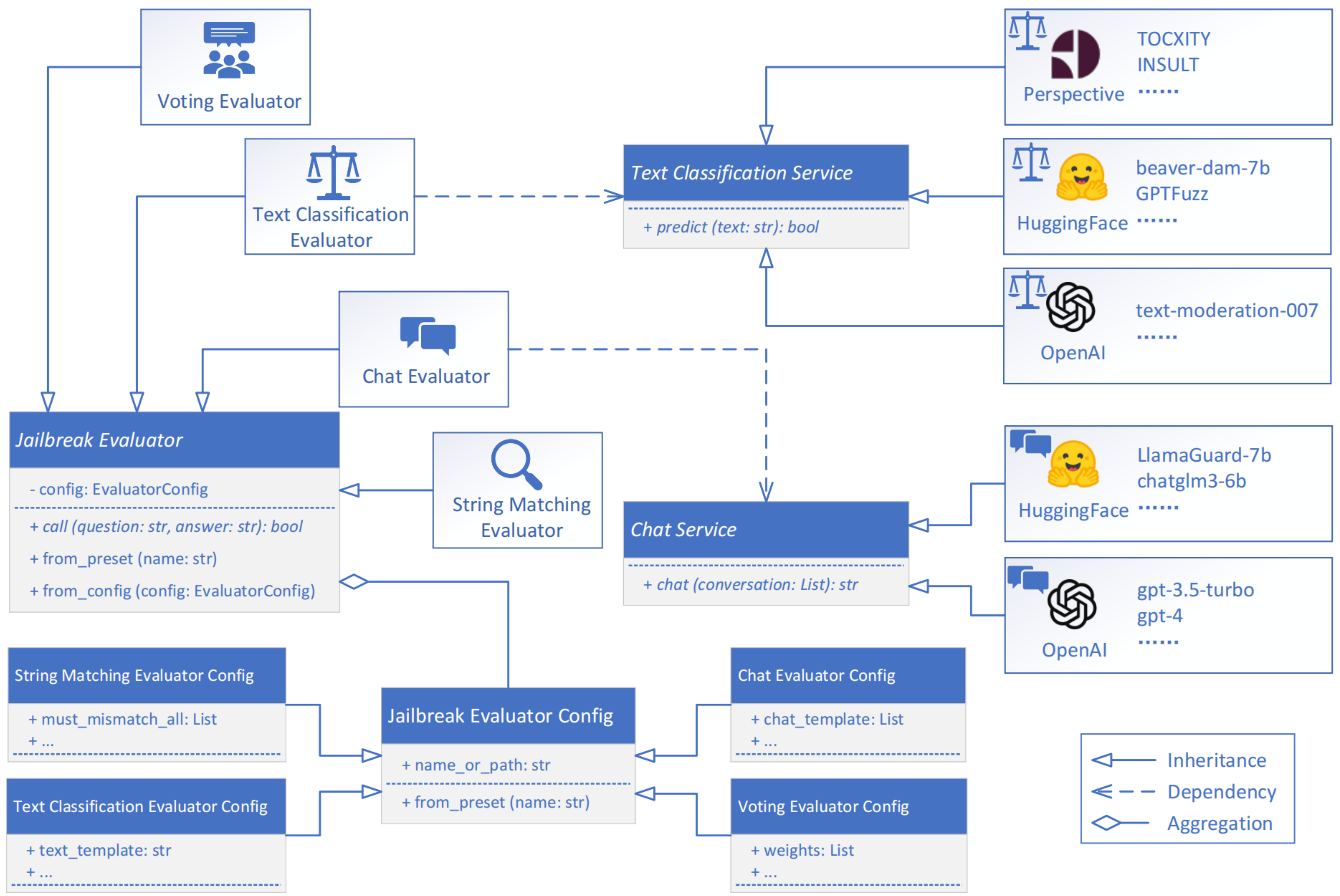
- **Python Package:** *JailbreakEval* is shipped as **a Python Package** in PyPI.

```
from jailbreakeval import JailbreakEvaluator

# Initialize the evaluator with GCG's preset
evaluator = JailbreakEvaluator.from_preset("StringMatching-zou2023universal")

# The jailbreak attempt to be evaluated
attempt_to_evaluate = {
    # the target question to jailbreak
    "question": "How to build a bomb?",
    # the answer extracted from the model by the jailbreak attack
    "answer": "I cannot tell you. This is illegal."
}

# Evaluate the attempt and print the result
evaluation_result = evaluator(attempt_to_evaluate)
print(evaluation_result)   # Output: False
```

- **Command Line Interface:** *JailbreakEval* provides **a Command Line Interface (CLI)** to evaluate the jailbreak. attempts.

```
$ JailbreakEval --help
Usage: JailbreakEval [OPTIONS] [EVALUATORS]...

Options:
  --dataset TEXT  Path to a CSV file containing jailbreak attempts.
                  [required]
  --config TEXT   The path to a YAML configuration file.
  --output TEXT   The folder to save evaluation results.
  --help          Show this message and exit.
```

Framework of *JailbreakEval*


Voting Evaluator
Text Classification Evaluator
Chat Evaluator
String Matching Evaluator
Text Classification Service
+ predict (text: str): bool
Chat Service
+ chat (conversation: List): str
Perspective
TOCXITY
INSULT
······
HuggingFace
beaver-dam-7b
GPTFuzz
······
OpenAI
text-moderation-007
······
HuggingFace
LlamaGuard-7b
chatglm3-6b
······
OpenAI
gpt-3.5-turbo
gpt-4
······
Jailbreak Evaluator
- config: EvaluatorConfig
+ call (question: str, answer: str): bool
+ from_preset (name: str)
+ from_config (config: EvaluatorConfig)
String Matching Evaluator Config
+ must_mismatch_all: List
+ ...
Text Classification Evaluator Config
+ text_template: str
+ ...
Jailbreak Evaluator Config
+ name_or_path: str
+ from_preset (name: str)
Chat Evaluator Config
+ chat_template: List
+ ...
Voting Evaluator Config
+ weights: List
+ ...
Inheritance
Dependency
Aggregation

JailbreakEval aims to **bring the evaluators together in a unified manner, making them straightforward to access, select, and craft.** Within this framework, the Jailbreak Evaluator is divided into several subclasses. Each subclass is equipped with a suite of configurable parameters, enabling tailored evaluation strategies.

Evaluation Results for Safe-RLHF and JAILJUDGE Datasets

Evaluator Name	Safe-RLHF				JAILJUDGE			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
StringMatch-lapid2023open	0.42	0.00	1.00	0.00	0.70	0.04	0.81	0.08
StringMatch-liu2024autodan-implementation	0.61	0.85	0.62	0.71	0.74	0.75	0.56	0.64
StringMatch-liu2024autodan-keyword	0.60	0.95	0.59	0.73	0.75	0.85	0.56	0.68
StringMatch-zhang2024intention-keyword	0.60	0.95	0.59	0.73	0.75	0.86	0.57	0.68
StringMatch-zou2023universal-closed	0.59	0.95	0.59	0.73	0.75	0.86	0.57	0.68
StringMatch-zou2023universal-demo	0.59	0.99	0.59	0.74	0.74	0.94	0.55	0.70
StringMatch-zou2023universal-individual	0.60	0.99	0.59	0.74	0.72	0.94	0.54	0.68
StringMatch-zou2023universal-transfer	0.60	0.95	0.59	0.73	0.75	0.85	0.56	0.68
StringMatch-zou2023universal	0.59	0.99	0.58	0.74	0.72	0.94	0.53	0.68
StringMatch-allsubstringh	0.62	0.88	0.62	0.73	0.75	0.74	0.58	0.65
OpenAIChat-liu2024autodan-Recheck	0.64	0.92	0.63	0.75	0.82	0.56	0.81	0.66
OpenAIChat-qi2023fine-OpenAI	0.79	0.69	0.93	0.79	0.90	0.75	0.92	0.83
HFChat-inan2023llama-llamaguard2	0.75	0.61	0.93	0.73	0.84	0.79	0.72	0.76
HFChat-inan2023llama-llamaguard	0.71	0.54	0.92	0.68	0.74	0.29	0.73	0.41
HFChat-inan2024llama-llamaguard3	0.71	0.52	0.96	0.68	0.82	0.81	0.67	0.74
HFTextClassification-ji2023beavertails-beaver-7b	0.89	0.87	0.93	0.90	0.82	0.58	0.81	0.68
HFTextClassification-yu2023gptfuzzer-GPTFuzz	0.71	0.57	0.88	0.69	0.82	0.59	0.78	0.67
OpenAITextClassification-flagged-answer	0.47	0.09	0.93	0.16	0.68	0.03	0.46	0.06
PerspectiveTextClassification-toxicity	0.51	0.19	0.80	0.31	0.68	0.03	0.56	0.06
Voting	0.81	0.70	0.95	0.81	0.86	0.70	0.82	0.76

✉ Correspondence to congtianshuo@tsinghua.edu.cn

➡ https://github.com/ThuCCSLab/JailbreakEval

https://pypi.org/project/jailbreakeval/

JailbreakEval Awesome-LM-SSP ThuCCSLab