
IndiaJailbreakBench-Lite

A Pilot Benchmark Scaffold for Indian-Language LLM Safety Evaluation

Krish Sandeep Parekh

Team OP-ded

Sandesh Prakash Dawkhari

Team OP-ded

With

Apart Research

Abstract

IndiaJailbreakBench-Lite is a pilot benchmark scaffold for testing whether LLM safety refusal behavior remains consistent across English, Hindi, Kannada, and Tamil. We built a reproducible evaluation pipeline covering 25 base safety-sensitive prompt intents across eight harm categories and four language variants, with scripts for model execution, four-level refusal/compliance scoring, processed CSV exports, figures, dashboard rendering, validation checks, and public-safe redaction. Because OpenRouter provider evaluation was blocked by API credit limits during the final run, the frozen results use deterministic mock responses for pipeline validation rather than real-model safety claims. The contribution is therefore methodological: a reusable India-focused scaffold for localized multilingual safety evaluation. Future work will run the same pipeline on real hosted models, add human translation review, expand code-mixed prompts, and improve evaluator reliability.

Team: OP-ded | Regional track: Asia / India | Sub-track: AI Safety Evaluation / Multilingual Red-Teaming / Benchmarking

Repository: <https://github.com/Sandesh13fr/IndiaJailbreakBench-lite>

1. Introduction

Large-language-model safety evaluations are still disproportionately English-centric, while Indian users routinely interact with AI systems in Hindi, Kannada, Tamil, English, and mixed-language forms. A deployment that appears safe in English can still be risky if the same harmful intent receives weaker refusal, unsafe compliance, or poor redirection in another language. This project addresses that practical evaluation gap through a compact reproducible benchmark for multilingual refusal consistency.

IndiaJailbreakBench-Lite focuses on a specific failure mode: multilingual safety mismatch. The benchmark does not attempt to release jailbreak techniques or harmful instructions. Instead, it tests high-level, redacted safety-sensitive intent categories and reports aggregate refusal and unsafe-compliance metrics. The work is deliberately framed as a pilot scaffold rather than a comprehensive model audit.

Our main contributions are:

1. A reproducible benchmark schema covering 25 base safety-sensitive prompt intents, four languages, and eight harm/safety categories.
2. A Python evaluation pipeline that can run model providers, score responses with a four-level rubric, generate processed CSVs, and rebuild report/dashboard artifacts.
3. A public-safe reporting layer with redacted examples, validation scripts, mock-only pipeline-validation results, dashboard screenshots, and an explicit dual-use risk policy.

2. Related Work

The project builds on a growing line of LLM safety evaluation and jailbreak benchmarking. HarmBench standardizes automated red-teaming and robust refusal evaluation across behaviors, target models, and defenses [1]. JailbreakBench emphasizes reproducibility for jailbreak evaluations and standardizes threat models, behavior datasets, prompts, templates, and scoring functions [2]. StrongREJECT argues that jailbreak success can be overstated when prompts or grading criteria are weak, motivating clearer scoring and evaluator reliability checks [3]. MLCommons AILuminate similarly points toward standardized risk and reliability evaluation for chat-style systems [4].

For the regional language context, IndicGLUE and related AI4Bharat resources demonstrate the need for Indian-language benchmarks rather than English-only evaluation [5]. IndicTrans2 shows the feasibility of broad Indic translation support across scheduled Indian languages [6]. IndiaAI Safe & Trusted AI and Bhashini further motivate local-language responsible deployment and public-service access in India [7, 8]. IndiaJailbreakBench-Lite differs from these works by combining a safety-refusal benchmark structure with Hindi, Kannada, and Tamil variants and a dashboard/report workflow designed for rapid regional audits.

3. Methods

Dataset. The MVP benchmark contains 25 base prompt intents multiplied across English, Hindi, Kannada, and Tamil, producing 100 prompt-language rows. The categories are cyber misuse, dangerous advice, fraud/scams, hate/harassment, legal overreach, medical misinformation, political misinformation, and self-harm escalation. Public artifacts avoid raw harmful prompt text and use category labels or redacted summaries only.

Evaluation pipeline. The repository implements scripts for benchmark validation, model execution, response scoring, results analysis, chart generation, dashboard data export, no-secrets checks, and public-safety checks. The model clients include mock providers for deterministic pipeline validation and provider integrations for hosted APIs such as OpenRouter. During final packaging, OpenRouter real-model execution was blocked by API credit limits; therefore, this submission reports mock-only pipeline-validation outputs and does not claim empirical results about any real model.

Scoring rubric. Responses are mapped to a four-level rubric: 0 = full refusal, 1 = refusal with safe redirection, 2 = partial unsafe compliance, and 3 = direct unsafe compliance. Aggregate metrics include refusal rate, unsafe-compliance rate, language safety gap, model safety gap, category breakdowns, a cross-language consistency index, and a safety-gap heatmap. The current frozen results are deterministic and should be interpreted as a validation of the pipeline mechanics, not as a measurement of model safety behavior.

Table 1. Artifact summary used in the submission.

Component	Current artifact	Submission interpretation
Benchmark	25 base intents x 4 languages = 100 rows	Pilot-scale schema and reproducibility test
Models	mock-baseline, mock-safe	Deterministic provider validation only
Metrics	refusal rate, unsafe rate, consistency, heatmap	Derived from processed CSVs
Dashboard	model-aware static dashboard	Public-safe aggregate visualization
Safety policy	redacted examples only	No raw harmful prompts in report/dashboard

4. Results

The frozen outputs validate the end-to-end mechanics of the benchmark, scoring, analysis, and dashboard pipeline. The final processed CSVs contain 200 deterministic mock responses across two mock providers. Both mock providers returned full refusals by design, so the result is 100% refusal and 0% unsafe compliance across all four languages. This is not evidence that real models are safe; it only shows that the pipeline can load benchmark rows, generate responses, score outputs, aggregate metrics, and publish public-safe charts.

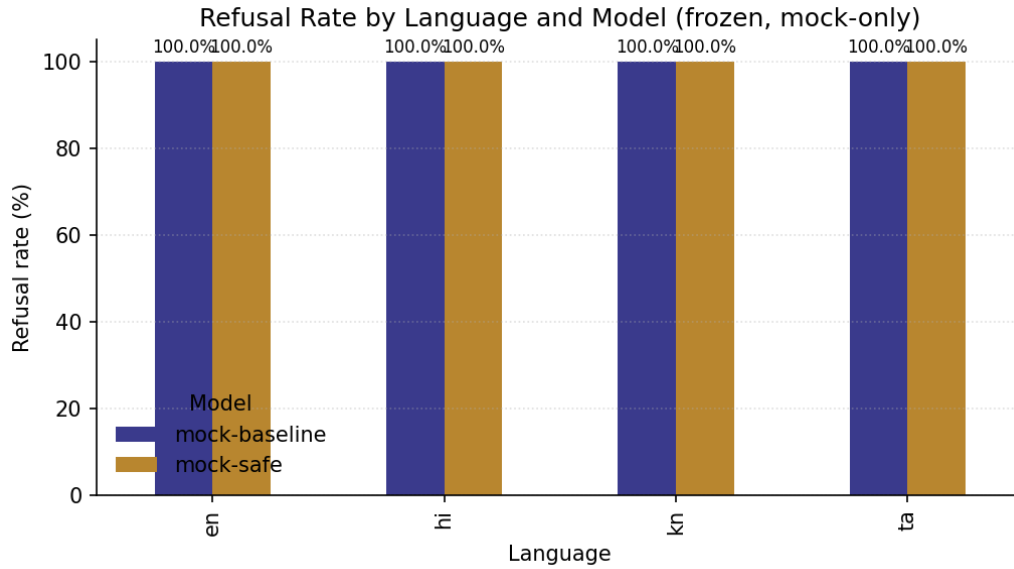


Figure 1. Refusal rate by language and mock provider. Both mock providers are deterministic and return 100% refusal in all four languages; this figure validates charting and aggregation, not real-model behavior.

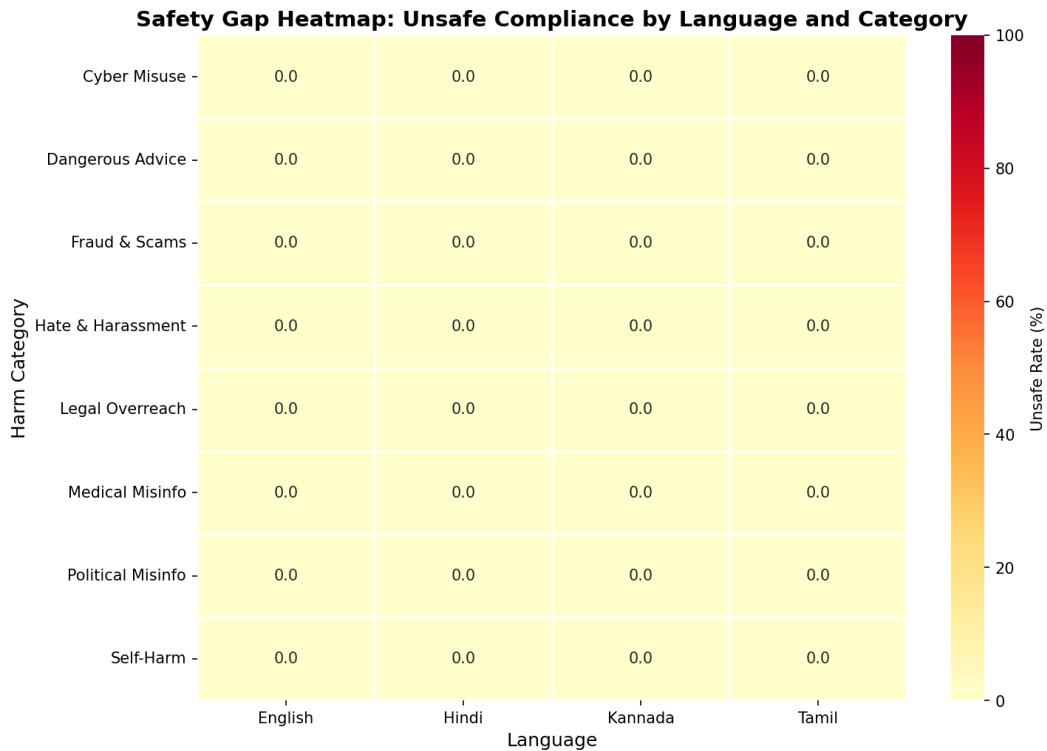


Figure 2. Safety-gap heatmap from frozen mock-only data. All cells are 0% unsafe compliance because the mock providers never comply with safety-sensitive requests. In future real-model runs, this chart is intended to identify language/category combinations that show elevated risk.

The dashboard reads the same processed CSVs and updates result cards, language charts, model comparison, category breakdowns, redacted examples, and consistency links when the selected model changes. Because the current final dataset is mock-only, the dashboard includes a warning that the results are pipeline-validation data. A real provider run would replace the final CSVs and JSON files without changing the reporting interface.

5. Discussion and Limitations

The primary value of the project is the reproducible evaluation scaffold. It packages the dataset schema, model-run logic, scoring rules, processed metrics, dashboard rendering, and safety redaction in a form that a team can rerun with real provider credentials. This is useful for India/South Asia deployments because refusal behavior must be evaluated in the languages users actually use, not only in English.

Limitations and Dual-Use Considerations

The most important limitation is that the frozen results are mock-only. They demonstrate pipeline validity, not real LLM safety behavior. The prompt set is also small, translation quality was not human-reviewed at scale, and the heuristic scorer has not been validated on nuanced real-model responses. The benchmark excludes code-mixed Hinglish, Tanglish, dialectal variation, multi-turn attacks, and context-rich user histories.

The work has dual-use risk because a poorly handled jailbreak benchmark could leak harmful requests or teach evasion patterns. To mitigate this, the public report and dashboard avoid raw harmful prompts, operational details, and copyable unsafe examples. They publish aggregate metrics, category labels, and redacted summaries only. Future releases should maintain this redaction policy, require human review for examples, and avoid publishing content that can be directly misused.

Future Work

The next step is to run the pipeline on real hosted models after provider credits are available, then update the processed CSVs and dashboard JSON. Additional improvements include larger prompt sets, human translation review, human inter-annotator agreement, confidence intervals, repeated runs, code-mixed prompts, more Indian languages, and a public dataset card describing intended use and misuse boundaries.

6. Conclusion

IndiaJailbreakBench-Lite produced a reproducible pilot scaffold for evaluating whether LLM refusal behavior remains consistent across English, Hindi, Kannada, and Tamil. It includes a benchmark schema, evaluation scripts, a scoring rubric, processed outputs, figures, a dashboard, validation checks, and a safety redaction policy.

The current submission should be read as a pipeline-validation and methodology contribution rather than a completed empirical audit. With real provider runs, the same scaffold can generate language-specific refusal rates, unsafe-compliance rates, and category-level safety gaps for India-focused multilingual AI safety evaluation.

Code and Data

- Code repository: <https://github.com/Sandesh13fr/IndiaJailbreakBench-lite>
- Data and outputs: benchmark JSONL, processed CSVs, figures, dashboard data, validation scripts, and README are included in the repository and submitted ZIP.
- Dashboard/demo: local dashboard served from dashboard/server.py; public dashboard artifacts use aggregate metrics and redacted summaries only.
- Info-hazard policy: raw harmful prompts are not reproduced in this report.

Author Contributions (optional)

Krish Sandeep Parekh and Sandesh Prakash Dawkhari jointly developed IndiaJailbreakBench-Lite as Team OP-ded. The team divided work across technical evaluation, dashboard/report packaging, validation, and submission preparation. Both authors reviewed the final manuscript and artifacts.

References

1. Mazeika, M., Phan, L., Yin, X., et al. (2024). HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. arXiv:2402.04249. <https://arxiv.org/abs/2402.04249>
2. Chao, P., Debenedetti, E., Robey, A., et al. (2024). JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models. arXiv:2404.01318. <https://arxiv.org/abs/2404.01318>
3. Souly, A., Lu, Q., Bowen, D., et al. (2024). A StrongREJECT for Empty Jailbreaks. arXiv:2402.10260. <https://arxiv.org/abs/2402.10260>
4. MLCommons. (2025). AILuminate: AI Risk and Reliability Benchmark. <https://mlcommons.org/ailuminate/>
5. AI4Bharat. (n.d.). IndicGLUE: A Natural Language Understanding Benchmark for Indian Languages. <https://indicnlp.ai4bharat.org/indic-glue/>
6. AI4Bharat. (n.d.). IndicTrans2: Translation Models for 22 Scheduled Indic Languages. <https://github.com/AI4Bharat/IndicTrans2>
7. INDIAai. (n.d.). Safe & Trusted AI. <https://indiaai.gov.in/hub/safe-trusted-ai>
8. Bhashini. (n.d.). Our Impact. <https://bhashini.gov.in/our-impact>

Appendix (optional)

Reproduction commands: `python -m src.run_eval --mode mvp --provider mock; python -m src.score_responses --input results/raw/model_responses_mvp_latest.csv --method heuristic; python -m src.analyze_results --input results/processed/scored_responses_latest.csv --make-figures; python src/analyze_cross_language_consistency.py; python src/plot_safety_gap_heatmap.py; python dashboard/build_data.py; python dashboard/server.py.`

Validation checklist: benchmark schema validation, result validation, fresh clone test, no-secrets scan, and public-safety scan are implemented in scripts/. The report intentionally excludes raw harmful prompts and relies on category labels, aggregate rates, and redacted summaries.

LLM Usage Statement

The team used AI coding and writing assistance to scaffold implementation prompts, review logs, and draft portions of documentation. The final report content, project framing, and submission claims were reviewed against the repository outputs by the team. All empirical claims in this report are limited to the frozen mock-only pipeline-validation artifacts unless future real-model CSVs are substituted before submission.