LAC Eval for AI Models

A Protocol for Testing Local Alignment and Effectiveness
in Latin America and the Caribbean

Adrian Dunkley
Caribbean AI Association
info@caribbeanaiassociation.com

16 November 2026

Protocol preprint v0.2

Abstract

A model can answer a question fluently while applying the wrong territory's procedure, misreading a local expression or claiming to have performed an action it cannot perform. LAC Eval proposes a method for measuring these failures alongside useful task completion. It extends an existing Caribbean AI Association case set into a protocol for controlled comparisons, locally authored language variants, and independent review. The public development set comprises 36 English-language cases across public services, finance and tourism. It provides a common starting point for testing the evaluation pipeline. The proposed study uses reserved case families, fixed operating conditions, and paired comparisons to estimate differences in model behaviour by workflow and locale. Separate measures cover task completion, evidence use, authority, referral, and critical failures. The current release contains a protocol and application kit. Multilingual coverage, inter-rater agreement, and comparative model performance await the pilot.

Keywords: large language models, AI evaluation, Caribbean, Latin America, local alignment, multilingual evaluation

1 Introduction

A service assistant receives a request about a deadline. The answer depends on where the person lives, which document applies, and whether the information provided is current. An evaluator needs to see whether the assistant used the controlling document, identified when the evidence was insufficient and still helped with the parts of the task it could complete. A ranking on general questions leaves those decisions unresolved.

LAC Eval asks which models or service configurations can deliver useful results while respecting the sources, local conditions and permissions of a defined workflow. In this paper, **local alignment** means following a task's locally authorised rules, evidence and action boundaries. **Effectiveness** means completing the permitted task with an adequate answer or a usable referral where the policy requires one. These definitions scope the claim to a workflow and evaluation context. They leave broader questions about model alignment outside the study.

The first public CAIA release defines an English-language case set for public services, finance and tourism [1]. This protocol retains its four scoring dimensions and critical-failure rule. It adds a

controlled comparison across model versions, a process for locally authored cases, and a format for reporting evidence from reviewed responses. The first pilot will ask where the starting set needs repair, how much rating work is required, and whether model differences justify a confirmatory comparison.

2 Related work and justification

HELM treats evaluation as a combination of scenarios and metrics, with explicit reporting of coverage gaps [2]. LAC Eval adopts that approach for local decision tasks. A useful answer, faithful use of evidence and respect for a permission can differ within one response. Each warrants its own measure.

BLEnD uses locally specific everyday knowledge to test cultural and language coverage [3]. LatamGPT's resource collection describes Trueque and CHOCLO as evaluations of Latin American knowledge and culture [4]. They offer useful comparisons for regional coverage. LAC Eval focuses on constrained service decisions, including source dates, country boundaries, permissions and referral. Its first public set includes only English. A future comparison can use the same intent across language forms and add cases written around problems that emerged within a locale.

AgentHarm evaluates behaviour through software tools [5]. This motivates a distinct workflow track where the evaluator records actual sandbox tool calls and outcomes and can compare these with the written response. NIST's AI Risk Management Framework provides wider context for governance, measurement and management [6]. The LAC Eval score supplies limited evidence for a decision within that process. Certification and general safety claims require separate evidence.

3 Evaluation design

3.1 Unit, case families and coverage

The basic observation is one system response to one eligible case under a specified configuration. The **case family** is the unit for paired comparisons. It contains a base situation and its paraphrases, variants, repeated generations or matched translations. Assign family IDs before collecting results. Keep all members of each family within one split to prevent near-duplicates crossing development and final sets.

The public development set crosses six risk areas with three sectors and two request types. Its risk areas cover privacy, grounding, equal treatment, authority, safety and escalation, and resilience. One case occupies each sector-by-risk-by-request cell. The original release uses nine place labels, and all its prompts are in English. A locality label records the setting of a scenario. Coverage claims require review by people working in that locale and language. Use the public set to debug the runner and the rubric, and reserve independently authored families for final comparisons [1].

Start language expansion with locally reviewed work in Caribbean English, Jamaican Patwa, Haitian Creole, Latin American Spanish and Brazilian Portuguese, subject to contributor availability. Treat these as pilot priorities. Add French, Dutch and other varieties as partnerships permit. A language name alone can hide important differences between territories or registers. Store locale and language variety independently. Record task context, reference date, author and reviewer roles, publication rights and any unresolved disagreement.

3.2 Writing and reviewing a case

A case packet contains a situation, user request, versioned reference material, allowed actions, expected behaviour, rating anchors and a specific critical failure. Each packet states whether the controlling evidence is supplied or intentionally missing. Construct pairs that keep intent and rules fixed when the aim is to measure sensitivity to language. Author additional native scenarios when translation would change the situation, reference or service convention.

Ask two locally competent people to review the consequential cases, including a domain reviewer where eligibility, money, access, safety or rights are at stake. Record reviewer agreement and an adjudication reason for each disagreement. Resolve the scope of consent and reuse before accepting contributed material. Use invented case records and avoid identifiable customer or applicant data. Use real source documents only where public access, permitted use, effective dates and the local interpretation can be documented.

For example, a pilot case could give a service assistant two programme notices with different dates and its authority to draft only. The evaluator checks whether the response cites the controlling notice, asks for the missing fact and keeps the action at draft stage. A matched variant can change the request's language while holding the documents and permission fixed. A native case might require a different official route. Analyse that case in its local setting.

3.3 Comparing models and configured services

In the response track, run the identical input packet and allowed-action description against pinned model versions. Use fixed system instructions, evidence snapshots, output limits and provider settings wherever the endpoints support them. Record settings that are unavailable or behave differently. Preserve timestamps, code and dataset versions, model IDs, latency, retries and errors. A workflow track can add a simulated tool environment with explicit access and recorded outcomes. Keep results for the two tracks distinct.

The pilot should compare three model configurations chosen at run time, with the same case list and frozen protocol. Add a simple rule-based comparator where the task permits a fair rule. A separate comparison can hold the model fixed and change local configuration. That comparison tests the configuration itself. Use multiple attempts per case only when the design prospectively defines how repeats enter the analysis. Record seeds when supported and blind human reviewers to model identity where practical.

4 Scoring and reporting

Use the published 0–2 scale for task completion, evidence use, authority and recovery or referral. Zero indicates unmet criteria. One indicates a material correction is needed. Two indicates the specified behaviour is met. Sum the four ratings and divide by eight for the conventional case score. Give full recovery credit on a normal case when no escalation was needed and the model completed the work without an unnecessary referral. A correct refusal can still earn task credit when it also provides permitted help.

Flag a critical failure independently of the score. Examples include unauthorised execution, disclosure, or a false consequential instruction that the case identifies in advance. In the response track, record a text claim about an action as a claim. In the workflow track, record whether a tool was

invoked and whether the effect occurred. Retain the applicable permission, exact response span or event, consequence and adjudication. Keep the implicated permission on hold while a critical failure is unresolved [1].

Pre-specify a primary endpoint before comparing models. For the pilot, estimate the paired difference in critical-failure rates on jointly reviewed families and report permitted-task completion alongside it. Show the number assigned, returned, reviewable and scored by model. Use the same family set when reporting pairwise differences and state exclusions. Report timeouts and absent ratings on their own lines. Show results by risk category, sector, locale and language where counts support the comparison. Treat small slices as descriptive.

Use paired family-level resampling for uncertainty when sample size supports it. Summarise repeated observations within family before running the resampling procedure. Publish denominators, point estimates, intervals and disagreement counts. An average can conceal a severe error in a particular language or action. A short list of model profiles with visible trade-offs is the primary reporting format. A public comparative dashboard should carry the case-set version, model snapshot, test date and evidence status for each report.

5 Pilot and interpretation

Use Inspect AI for the first cross-provider run. The accompanying task loads the same case packets for each model. Inspect's evaluation-set command can run multiple model configurations, and Inspect View can inspect their logs together [7, 8]. The kit's offline report turns reviewed rows into a single HTML comparison with coverage and critical-failure counts. Start the pilot with a small subset, check the prompts and logs, then expand only after the reviewers confirm the rubric works. A claim about a deployed service needs a separate sandbox evaluation with actual tool traces.

The planned pilot will test the feasibility of the design. Review how often raters disagree, which cases cannot be decided from the supplied evidence, how failures are classified and whether the burden is manageable. Decide which families require rewriting and which candidate language packs require another local review. Fix the protocol and reserve a fresh set before a confirmatory comparison. No comparative model results are available in this release. The public English examples support method development and leave performance in untested languages and territories unresolved.

5.1 Data and code availability

The CAIA observatory publishes the development case set and original scoring notes. The LAC Eval project page carries the application kit, run guide and manuscript. Reserved cases and review records are prospective. Release de-identified cases and evidence only where their contributors and source rights permit it.

References

1. Adrian Dunkley. LAC Eval alignment. Caribbean AI Association, version 0.1, 10 November 2026.
2. Percy Liang et al. Holistic Evaluation of Language Models. arXiv:2211.09110, 2022.

3. Junho Myung et al. BLEnD: A Benchmark for LLMs on Everyday Knowledge in Diverse Cultures and Languages. arXiv:2406.09948, 2024.
4. Latam-GPT. Resources and benchmarks. Accessed 16 November 2026.
5. Maksym Andriushchenko et al. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. arXiv:2410.09024, 2024.
6. Elham Tabassi. Artificial Intelligence Risk Management Framework 1.0. NIST AI 100-1, 2023.
7. UK AI Security Institute. Inspect AI evaluation sets. Accessed 16 November 2026.
8. UK AI Security Institute. Inspect AI log viewer. Accessed 16 November 2026.