The Instrument Around the Model

SteamLens Milestone 1: extraction + evaluation

Four engineering investigations: the vocabulary, the labeler, the scale, the trust

Every number verified against its run of record at build time

arda-basarici.github.io/steamlens-extraction

Arda Başarıcı · 2026

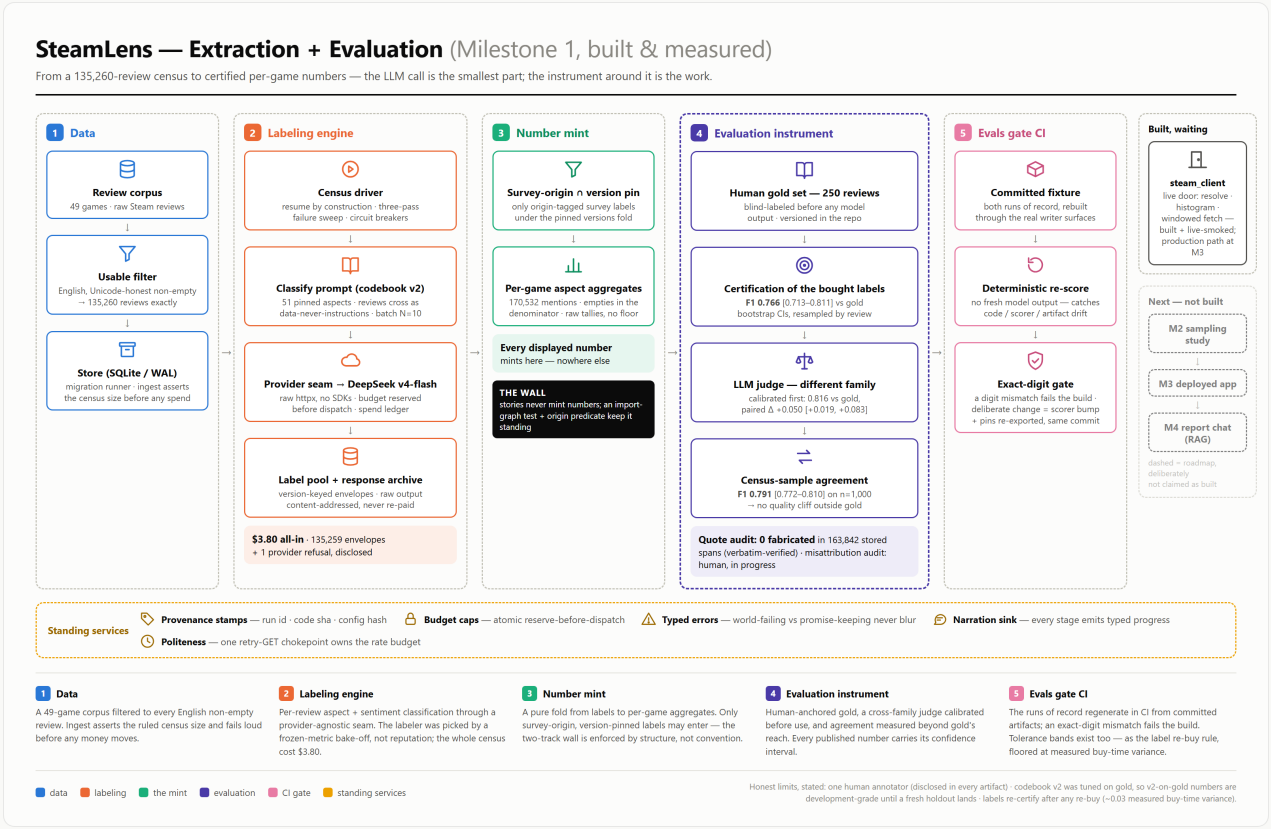
THE FINDINGS, FIRST

The instrument, in two pages

SteamLens turns Steam reviews into a per-game report with numbers under its claims: not “many players complain about performance,” but “x% of the reviews that mention performance are negative.” A language model does the reading; the promise is that its reading is *measured*: the labeler behind every displayed number has been measured against human judgment, and that measurement is reported alongside the numbers.

This milestone built that instrument end to end and certified it:

135,260 reviews across 49 games, every usable review in the corpus, resolved for $3.80 (135,259 labeled, plus one disclosed provider refusal), with the labeler's performance measured at **F1 0.766 [0.713–0.811]** against a human-adjudicated reference, and a census-scale audit finding no evidence of a quality cliff.



The Milestone-1 pipeline, built and measured: the LLM call is the smallest part; the instrument around it is the work. (“Survey-origin ∩ version pin”: only survey-track labels under the pinned codebook version may enter the fold that mints displayed numbers.)

The work organized itself into four investigations, and the report is structured as them:

1 · The vocabulary problem: what should the system extract? Counting needs stable units. Open extraction fragments (406 labels on 500 reviews); pure taxonomy encodes unmeasured priors. The answer is a versioned hybrid: a fixed core of 51 pinned aspects that all numbers ride on, plus a free-form candidate channel for what the core misses. The core was pruned against a corpus-wide probe spanning

all 49 games, where the priors lost more often than they won, and the codebook that teaches its boundaries was certified in a paired experiment: precision +0.066 [+0.039, +0.098] bought at recall –0.030 [–0.062, +0.000], a trade accepted deliberately.

2 · The labeler problem: who reads 135,260 reviews? Cheaper readers (lexicons, trained classifiers, embeddings) died on the corpus's measured shape, so a model reads, chosen like an instrument: against the human reference, frozen metrics, each candidate at its own tuned operating point. The paired bootstrap corrected the eyeball read twice: a “tie” concealed a real gap at matched conditions (+0.034 [+0.002, +0.067] to the frontier candidate), and at the chosen model's best batch size the gap narrows to +0.025 [–0.004, +0.055], a difference this sample size cannot resolve. The ruling: the budget labeler, whose remaining gap to the frontier at its operating point is small and unresolved, at a fraction of the price.

3 · The scale problem: can a census be a trustworthy record? The sizing question dissolved on measurement: only 45% of the corpus is labelable at all, and at measured unit cost the whole usable pool cost single-digit dollars, so the sample became a census, with no pre-filtering (“mentions nothing” is a measured verdict, not noise). The buy survived two live incidents by design (durable cache and ledger, refusal-as-data, budget caps in the honest currency) and settled *exact*: 135,259 envelopes plus one disclosed provider refusal. Every displayed number is minted from this artifact in exactly one place.

4 · The trust problem: how do we know any number is real? A 250-review human gold set the pipeline cannot influence anchors everything. A judge from a different model family, an independent second annotator never shown production's answers, calibrated at F1 0.816 [0.773–0.855] against gold (+0.050 [+0.019, +0.083] over production, paired), then measured 0.791 [0.772–0.810] agreement with production on a 1,000-review census sample: bracketed between the two gold scores, consistent with **no quality cliff outside the reference**. The stored artifact carries **zero non-verbatim evidence spans** across 163,842. Then the numbers were attacked: pre-registered experiments, including a contingent that killed a conclusion already drafted, acquitted batch composition and identified **buy-time variance of the served model** (~0.02–0.03 F1 at temperature zero) as the best-supported driver of a census-vs-lab gap, now a standing re-certification rule. Every number above regenerates in CI; an exact-digit mismatch fails the build.

What this report refuses to claim: that the labels are right (their *error is measured*, which is the stronger statement); that the chosen model is “the best” (a measurably better one exists at matched conditions; the gap is published); that agreement equals accuracy (human adjudication of disagreements is pending); that reviews speak for players (every number is a share of reviews); that anything is deployed (nothing is, yet).

One plan-level change is reported straight in the closing chapter: the agentic event-investigator from the founding plan is deferred indefinitely in favor of a grounded interrogation chat over the labeled corpus, a measured scope call whose grounds, costs, and blast radius are stated where it is told.

How to read this report. These two pages are the summary. Each investigation is a self-contained chapter: problem, approach space, what was built, what was measured, what was refused. Every number cites its run of record; the repository regenerates all of them, and the evaluation gate re-scores the two headline runs on every commit.

INVESTIGATION 1

The vocabulary problem: what should the system extract?

The problem

SteamLens makes a specific promise: type a game name, get a report on what its players are saying, with numbers under the claims. Not “many players complain about performance,” but “4,026 reviews in this census mention performance, and 38.2% of them are positive”; not “the bugs are a problem,” but “of the 5,722 reviews that mention bugs, 7.0% are positive.”

There is a much cheaper way to build something that *looks* like this: send the reviews to a language model and ask for a summary. The output reads well, and it is even mostly right. But it can never say *how many*. A summary is testimony: unquantified, uncheckable, and different on every run. The moment the report should say “x% of reviews,” someone has to have counted, and counting needs units: every mention has to land in a category that means the same thing across reviews, across games, and across time. That set of categories is the vocabulary.

> ***The entire distance between “an LLM that summarizes reviews” and “a system that measures them” starts here.***

So the vocabulary has to exist before the first label is bought, because every label is bought against it. Get it wrong in one direction and the system is blind to what players actually talk about; get it wrong in the other and the counts fragment into synonyms that never add up.

It is also an evaluation question in disguise. Numbers over aspects are only comparable if aspect identities are stable: a gold set, a per-game table, a trend all assume that `performance` today is `performance` next month. So whatever the vocabulary is, it has to be *versioned and frozen*, and changing it has to be a deliberate act, not a drift.

The approach space

Three shapes were on the table.

Open extraction: let the model name aspects freely, aggregate afterwards. A probe over 500 reviews settled this quickly: free naming produced 406 distinct labels for 704 mentions, and even after merging obvious variants, 313 remained. The long tail is real (roughly half of all mentions used vocabulary specific to a single game), and post-hoc merging of a fragmented namespace is exactly the unstable-identity problem the vocabulary exists to prevent.

A fixed taxonomy, written from priors: decide upfront what players talk about. The risk here is quieter: priors feel authoritative and are unmeasured. This mattered later, when the evidence arrived (below): several aspects that genre intuition insisted on turned out to be nearly absent from the corpus.

A hybrid: a fixed core of pinned aspects that all displayed numbers ride on, plus a free-form *candidate* slot the model may emit when a review talks about something off the list. This is what SteamLens uses. The pinned core gives the numbers stable, versioned identities; the candidate channel keeps the system honest about what the core misses, and feeds its future revisions with evidence instead of guesses.

Building the core: evidence over priors

The first draft had 22 labels and a flaw that nearly baked in silently: distinct concepts (lore, voice acting, immersion) had been folded into synonym lists of broader pins. The correction became a design rule: **folding is irreversible, not-pinning is nearly free.** A pin that earns no mentions costs a table row; two concepts fused at write time can never be split again without re-buying every label. That asymmetry flipped the sizing instinct toward generosity, and the draft grew to near fifty pins.

The draft then met the corpus. An extension of the probe grew the evidence base to all 49 usable games (about 4,900 reviews, 7,500 mapped mentions), enough to test every pin against measurement rather than intuition. The priors lost more often than they won. `camera`, a genre-critical certainty for action games, produced 0 mentions in 100 Elden Ring reviews, then 1 in roughly 1,900 Dark Souls III reviews. `grind`, surely a staple, appeared 15 times across all 4,900 reviews, never more than twice in one game. Both were demoted. The opposite case also occurred: `servers_netcode` was challenged on the same evidence (its counts looked thin too) and *kept*, because its mentions cluster in time around outages and launches; a windowed sample can miss an aspect that a live event makes dominant. The ratified core: **51 pinned aspects, version 1, frozen.**

From vocabulary to codebook

A vocabulary names the aspects; a codebook teaches a model where their boundaries are. Labeling the 250-review gold set (Investigation 4) generated 33 boundary rulings (is `gameplay` mechanics or pastime, are characters an authored cast or any person mentioned, when does difficulty-talk route to wayfinding), and those rulings were distilled into a second codebook version. Whether the fuller codebook actually helps is a measurable question, so it was measured, as a paired comparison on the same reviews before any large buy: precision improved by +0.066 [+0.039, +0.098] at a recall cost of –0.030 [–0.062, +0.000], an interval that touches zero, so the cost may be nil. The trade was accepted deliberately, because the system's failure mode of record is over-claiming, not under-hearing. This certified second version is the production codebook for everything that follows. A compact variant of the codebook (shorter, cheaper per call) was tested in the same experiment and rejected on evidence: it cost –0.057 [–0.097, –0.020] recall against savings of roughly ten cents per full corpus pass.

What this bought, and what it refused

The candidate channel did its job at scale: the census (Investigation 3) surfaced 1,239 distinct candidate strings, and the demoted `grind` resurfaced as the top candidate with 898 mentions, fragmented across grind/grinding/grindy, exactly as an open namespace fragments. The system deliberately does **not** merge these at write time. A fuzzy merge that is wrong corrupts two aspects at once, and stored identities would inherit the error permanently. Instead, recurring candidates wait for an offline, human-gated promotion pass: cluster as advice, review by hand, freeze the aliases into the next vocabulary version, re-fold deterministically, and never re-buy a single label.

THE GENERAL PROBLEM

Schema design under an open distribution: fix identities before measuring, but measure the tail before fixing them, and leave one honest channel for everything the schema will inevitably miss.

INVESTIGATION 2

The labeler problem: who reads 135,260 reviews?

The problem

The vocabulary defines the units of counting; someone still has to do the reading. Every review in the corpus needs the same treatment: find the aspects it talks about, in the pinned vocabulary or the candidate slot, with a sentiment on each. At human speed, say thirty seconds a review, the usable corpus is over a thousand hours of annotation. The reading has to be done by a model.

That sentence is where a system either becomes an instrument or stays a wrapper. Once a model does the reading, every displayed number inherits that model's errors, and the choice of model stops being an implementation detail: it is the calibration question for the entire pipeline. It is also, unusually, a *purchase*. Labels are bought per token, and once a corpus is labeled, the data has gravity: switching models later means paying the whole bill again. The code that calls the model is swappable in an afternoon; the labels it produced are not. So the model had to be chosen the way one chooses an instrument: against a reference, before the big buy, with the decision and its reopen conditions on record.

The approach space

The obvious question came first: does this need an LLM at all? Three cheaper readers were put on trial against the probe evidence.

Keyword matching dies on the corpus's shape: the top fifteen mention labels cover only 28% of mentions, and half of all mentions use vocabulary specific to a single game. A lexicon big enough to catch “ship bilding” and “the specialists problem” is not a lexicon anymore.

A trained classifier dies on its own prerequisites: 51 classes, multi-label, each with sentiment, needs exactly the large labeled training set whose absence is the problem being solved. It dies with a revival clause, though: once a *measured* labeler has produced the census, those 135,260 labels are a training set, and the classifier returns as a distillation candidate, swapped in behind the same seam, certified against the same gold set, one more row in the same comparison table. What it could not replace are the product features: the free-form candidate channel, and the verbatim evidence spans behind every claim.

Embedding similarity dies on the same long tail: nearest-neighbor to a pin name cannot decide whether a mention of “difficulty” is about game balance or about wayfinding, which is a boundary the codebook defines, not geometry.

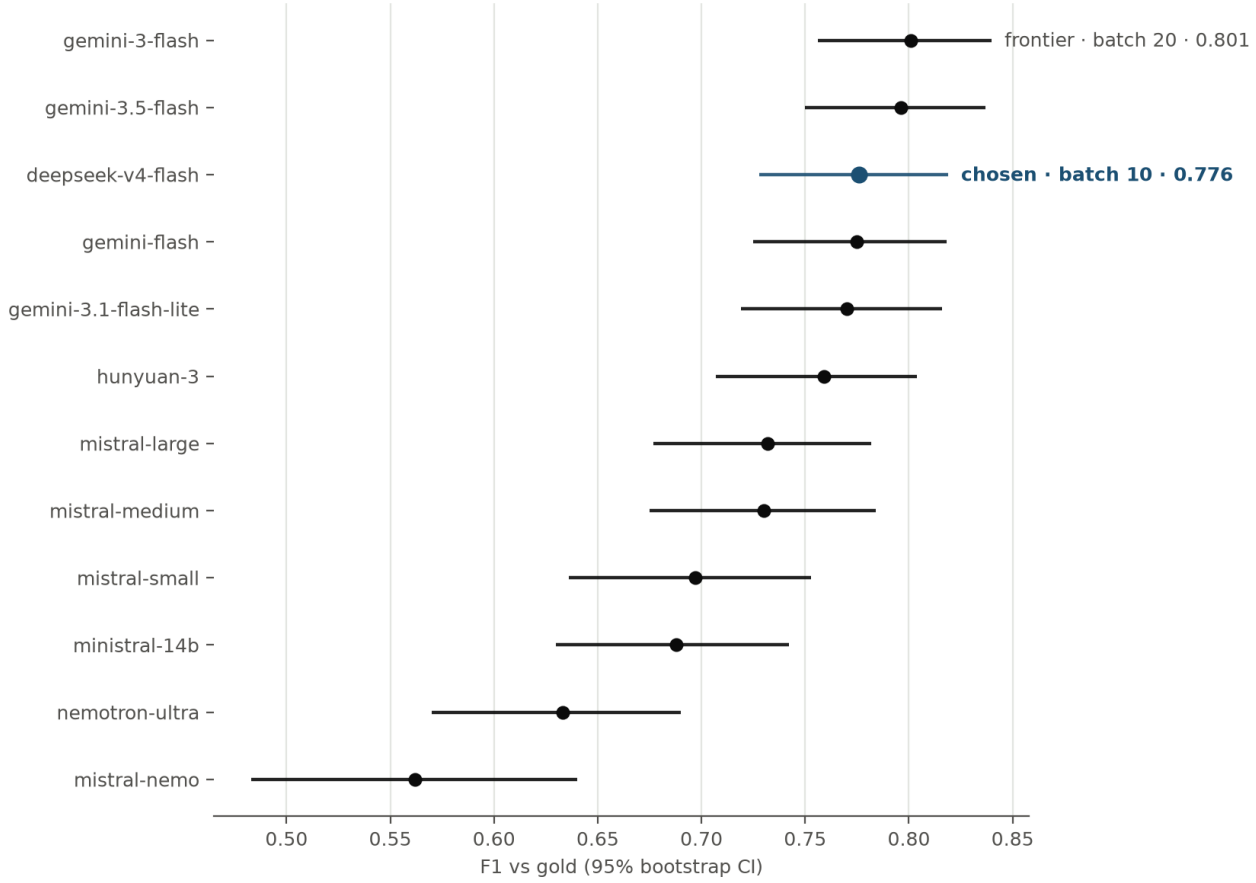
What survives is an LLM reading each review against the codebook. The important reframe: the labeling model is not plumbing. It is *the object this milestone evaluates*, and everything downstream (the gold set, the judge, the CI gate) exists to measure this one component.

The decision procedure

The reference came first. A 250-review human gold set (Investigation 4) was built *before* any candidate was compared: labels are provider-neutral by construction, so the reference cannot favor a vendor.

Cost was expected to decide, and measurement dissolved it. A landscape scan priced every realistic candidate below ~$20 for the full survey buy. At that ceiling, cost is a tiebreak, not a criterion, and the bake-off became a pure quality comparison. (Two providers were dropped at the scan stage.)

The bake-off ran frozen-metric. Fourteen models, thirty-two full-slice captures against gold, metrics defined and frozen before any results were viewed. The score is F1 over pinned-aspect matches, set-intersected by label within each review and micro-aggregated; candidate-slot mentions stay out of the score on both sides; sentiment is scored separately, as accuracy over matched mentions; the bootstrap resamples reviews, never mentions, because mentions within a review are not independent. A hard gate on unrecoverable parse failures (2%); zero-aspect share demoted from score to diagnostic once gold defined the true base rate (49.2% of reviews mention nothing in the vocabulary, so a labeler must be allowed to say "nothing"). The decision rule was deliberately a recorded human ruling over the frozen table, not an automatic ranking: the metrics constrain the judgment, the judgment stays accountable.



The bake-off at each model's best scored operating point: F1 vs gold with 95% bootstrap CIs; models disqualified at the parse gate are not shown. Rendered from the frozen comparison table.

Batch size turned out to be a per-model quantity, not a convention. How many reviews ride in one prompt changes accuracy, and the curve is not monotone: the selected model scored F1 0.746 / 0.776 / 0.767 / 0.762 at batch sizes 5, 10, 20, 50; batch-10 beats batch-5 by +0.029 [+0.009, +0.052]. Prompt economics explain the lever: 88.2% of prompt tokens are the codebook, not the reviews, so batching amortizes the codebook, but past the sweet spot recall decays. Each candidate was compared at its own tuned operating point.

The close is the part worth remembering. On the eyeball read, the top two candidates looked tied: "the error bars overlap." The paired bootstrap said otherwise. The frontier candidate is measurably better at matched batch size, +0.034 [+0.002, +0.067], a real gap the eyeball had waved away. But at the budget candidate's best operating point the gap narrows to +0.025 [–0.004, +0.055], a difference this sample size cannot resolve, in either direction. The instrument corrected the operator twice in one reading: first against

calling a real difference a tie, then against paying for a difference the data could not resolve at the actual operating point.

The ruling: DeepSeek's v4-flash at batch size 10, under the certified codebook, labels the survey, at an effective rate of roughly $0.007 per 250 reviews, which is what made a full census cost $3.80 (Investigation 3). The measured matched-condition gap to the frontier candidate is published, not hidden, and the decision's reopen conditions are recorded with it.

What this refused

The report does not claim the chosen model is “the best model.” A measurably better one exists at matched conditions and the gap is stated. The claim is narrower and honest about its resolution: at its operating point, against a human reference, the experiment could not resolve a quality difference between the chosen labeler and the frontier alternative, at a fraction of the price, and the whole comparison is regenerable from the frozen captures.

THE GENERAL PROBLEM

Instrument procurement: when a measurement device must be bought rather than built, build the reference first, demote the spec sheet to measurements, compare at each device's operating point, and write down what would reopen the decision.

INVESTIGATION 3

The scale problem: can a census be a trustworthy record?

The problem

A labeler is chosen; now the numbers need a base to ride on. Every claim in a SteamLens report has the form “x of y reviews,” so the system has to decide what *y* is, label all of it, and be able to say afterwards exactly what was labeled, at what cost, under which model and codebook version. The failure modes at this stage are not statistical, they are logistical: a labeling run that dies halfway and double-bills on retry, a cap that trips in the wrong currency, an artifact that cannot prove what it contains. Scale is where a measurement system earns or loses the right to be trusted as a *record*.

The sizing question, answered by deletion

The original plan was to sample: a thousand reviews per game, cost-bounded, with a shortfall policy for small games. A supply check run to size that buy dissolved it instead. Of the 298,553-review corpus, only 135,260 (about 45%) are labelable at all under the system's own rules (English text that survives Unicode-honest emptiness checks). And at the chosen labeler's measured unit cost, labeling *all* of it projected to single-digit dollars: 2.9 times the sampling alternative, for the difference between a sample and a census.

At that price the sampling question is not worth its own complexity: a census has no shortfall policy (small games are censuses anyway), no sampling error against the corpus, and it never caps the sampling study that comes next milestone. The ruling that followed it is easy to skip past but carries the honesty of the whole pipeline: **no pre-filtering of “useless” reviews.** A review that mentions no aspect is not noise to be screened out before paying. “Nothing in the vocabulary” is the classifier's own verdict, and the zero-mention share is a measured product quantity (the human gold set puts its base rate at 49.2%). Screening reviews out beforehand would move that number from *measured* to *assumed*. Exact-duplicate texts cost once regardless, through a content-keyed cache.

An engine built for a bad day

The labeling driver was designed on the assumption that something would go wrong mid-buy. Every response is banked durably before anything else happens (a cache keyed by content, a ledger of what was paid), so an interrupted run resumes without re-buying a single label. The label's identity records the *requested* model id, with the provider's reported version journaled per call and a mid-run drift watch that aborts the run if the model changes under it; throughput dials are run configuration, never label identity. Spending is capped by an atomic reserve-before-dispatch budget guard, and a pilot run surfaced a subtlety worth recording: the internal ledger prices calls cache-blind, so the guard trips in a *more expensive currency* than the provider actually bills (a projected ~$22 ledger against ~$5 true). The response was deliberate: keep the conservative cap, treat cap trips as checkpointed tranches, and the system's guard always fires before the provider's.

The buy, and its two incidents

The census ran as a pilot plus tranches, and both of the failure modes the engine was built against showed up in live form.

The provider's content filter refused one review. A single review produced a hard 400 from the provider's moderation layer, and before the fix, that abort took the whole run down. Worse, deterministic

batching meant a retry would rebuild the same batch and hit the same wall forever. The fix reframed refusal as data: a refused batch fails its rows into the ordinary retry sweep, where the innocent reviews label successfully in isolation and the triggering review takes a durable mark carrying the provider's refusal verbatim. A circuit breaker guards the other direction (too many refused batches means something systemic, not one review). After the census, an ablation confirmed the diagnosis at the level of a single line of review text: with the line present, refused; with it removed, accepted. Necessary and sufficient, within the reproduced request.

The database locked under its own success. At thirty workers, the cache-banking write flood (hundreds of envelope transactions per second) starved SQLite's unfair busy-wait past its five-second default. The fix was one pragma sized to the observed burst shape rather than a bigger architecture: capacity had an order of magnitude of slack; the queue, not the disk, was the bottleneck. The escalation trigger (batching envelope writes) is recorded in the store's docstring for the day the write pattern changes.

The settlement

135,259 labeled envelopes + 1 durable, disclosed refusal = 135,260, exact to the review.
170,532 aspect mentions · true cost **$3.80** all-in (projected band $3.2–6.0) · final cache hit rate 92.8% · both backup artifacts off-site and hash-verified.

The numbers the product will display are minted from this artifact in exactly one place, a pure per-game fold from labeled envelopes to aspect aggregates: survey-origin labels only, codebook version pinned, candidates folded exactly as pinned aspects are (no fuzzy merging; a wrong merge corrupts two aspects at once and stored identities would inherit it), singleton candidates kept, per-game denominators counting the ~46% empty envelopes. The fold's internal consistency invariant, reviews-with-aspect equals the sum of its counts, held wholesale across all 49 games on the real artifact. Everything a report displays traces to this mint; nothing else is allowed to produce a displayed number.

What the fold shows, in one line: what players *mention* and what they *conclude* dissociate. Reviews that bring up `music` are 95.1% positive; reviews that bring up `bugs`, 7.0%; `developer_conduct`, 13.5%. An aspect profile is not a verdict profile, which is much of why per-aspect numbers are worth building at all.

What this refused

The census is a census *of the usable pool*, and the report says so: 45% of the raw corpus, English-first by explicit rule, not “all the reviews,” and the number is stated rather than rounded away. The one refused review is part of the artifact's arithmetic, not a footnote. And the artifact's provenance is complete: every envelope carries its run, model id, and codebook version; the whole settlement is regenerable from the manifests.

THE GENERAL PROBLEM

Buying data at scale from an infrastructure you don't control: assume mid-flight failure, make every purchase durable and idempotent before anything else touches it, cap spend in the currency you actually pay, and when the artifact settles, disclose its exact shape, including its one hole.

INVESTIGATION 4

The trust problem: how do we know any number is real?

The problem

The census exists and the mint folds it into per-game numbers. But every one of those numbers was produced by a model this project chose, reading against a codebook this project wrote. Left there, the pipeline is circular: it would be grading its own homework by never being graded at all. The promise of “x% of reviews” is only worth making if the error of the labeler is *measured*: against a reference it cannot influence, at a scale beyond the reference, with the results wired so they cannot drift silently after the fact. This chapter is the reason the report calls the system an instrument.

who / what	what it is	its role
the production labeler	the chosen model + certified codebook; labeled every census review	the object under evaluation
gold	250 reviews, human-adjudicated	the reference the pipeline cannot influence
the judge	an independent second annotator from a different model family	extends the reference's reach beyond 250
the census sample	1,000 seeded reviews, text pinned by hash	where judge and production are compared at scale

The reference: a gold set the pipeline cannot touch

The first attempt at a reference was the obvious one, and it was killed on circularity: a large gold set authored by a frontier model is a model grading labels a model produced. Whatever it measures, it is not the pipeline's error against *human* judgment. The reference had to be human-adjudicated, which caps its size and makes its construction the thing to get right.

The build was treated as an instrument-calibration exercise of its own. The labeling instructions went through dry-run rounds against fresh reviews; one round caught the protocol about to test itself against its own worked examples, which would have validated memorization, not instructions. Machine-drafted assists accelerated the pass, under two guards: every drafted evidence span was machine-verified verbatim against the review text, and the assisting model was named in provenance and banned from the labeler comparison, because the reference must not favor a candidate. A human adjudicated all 250 reviews; the assist's scorecard (73% accepted, 11% modified, 16% deleted, 17 human-added mentions) is recorded rather than hidden. Even the tooling was audited: a verbatim round-trip gate caught a file formatter silently rewriting asterisks inside review quotes.

Gold, drawn by enriched stratified sampling with its skips and reserve replacements logged in the draw manifest, settled at **250 reviews, 351 mentions, 49.2% mentioning nothing in the vocabulary**. That last number matters twice: it defines the true base rate a labeler must reproduce, and it retired zero-share as a score (a labeler must be free to say “nothing” without being penalized toward invention).

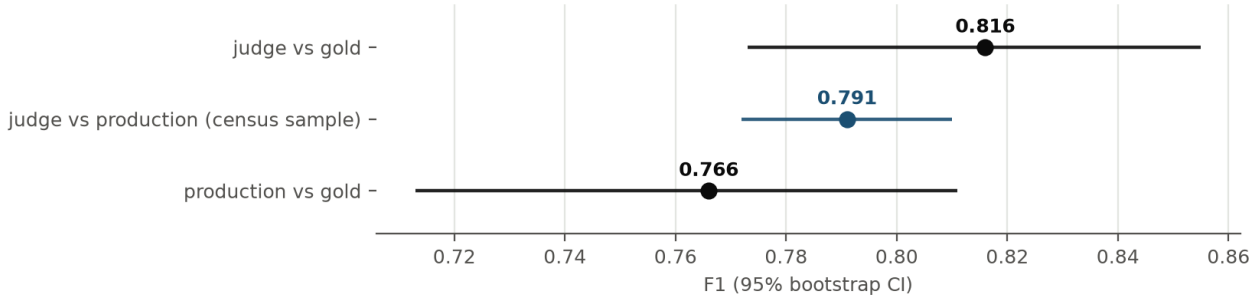
Certification: against gold, the production labels that the census actually bought score **F1 0.766 [0.713–0.811]**, on the 245 of gold's 250 reviews that fall inside the census's usable scope. One disclosed

circularity remains and is scheduled rather than waved away: the second codebook version was tuned on gold's rulings, so gold slightly favors it, and that mild favor extends to anything scored under this codebook, the judge included. A fresh frozen holdout is the registered answer, still pending.

The judge: a second annotator, not a verifier

Gold is 250 reviews; the census is 135,260. Does quality hold *out there*? The obvious design, showing a judge model production's labels and asking “are these right?”, was rejected on anchoring bias: a judge shown an answer drifts toward endorsing it. The judge built here never sees production's answers. It is an **independent second annotator** from a different model family, labeling one review at a time against the same codebook, with a calibration rule pre-registered before any run: the paired judge-minus-production difference on shared gold decides. Significantly above production and the judge is a valid quality reader; indistinguishable and it is only a disagreement flagger; significantly below and that is reported as a finding.

It did: **judge-vs-gold F1 0.816 [0.773–0.855]**, a paired **+0.050 [+0.019, +0.083]** over production, recall-shaped. The judge hears more than production does; it does not hallucinate more. With the judge calibrated, a 1,000-review census sample (review-framed, seeded, text pinned by hash) measured **judge-vs-production agreement at F1 0.791 [0.772–0.810]**, landing between production-vs-gold (0.766) and judge-vs-gold (0.816). That bracketing is the load-bearing claim, read honestly as a point-estimate ordering (the three intervals overlap): the census result is consistent with production behaving as it did on gold. **No evidence of a quality cliff outside the reference.**



The bracketing: judge-vs-production agreement on the census sample lands between the two gold readings, consistent with production behaving on the census as it did on gold.

The agreement decomposes informatively: technical aspects agree at 0.9+ (`music` 0.974), soft and meta aspects carry the gap (`updates` 0.611, driven by the judge finding mentions production misses). The caveat is stated with it: agreement is not accuracy. Where the two disagree, the judge may be over-finding rather than production under-hearing; a human adjudication pass on the ranked disagreements is the registered next step.

The receipts audit rode the same machinery census-wide: every evidence span the labeler emits must appear verbatim in its review. Across 163,842 spans (96.1% of all 170,532 mentions carry evidence) there are **zero non-verbatim evidence spans** in the stored artifact: no fabricated quote survives to storage. The parse layer repaired roughly 2.9% attempted near-misses (paraphrase, elision) on the way in; what survives to storage is verbatim by construction, and the audit measured it anyway.

We then tried to break it

Certification produced one number that demanded an explanation rather than a celebration: the census labels score 0.766 against gold, but the *lab arm of the codebook certification*, the same model at the same batch size under the same certified codebook, had scored 0.799 on the same reviews a day earlier, a paired gap of **–0.033 [–0.061, –0.007]**. Real, not noise. (The bake-off's 0.776 in Investigation 2 is a different, earlier reading: the same configuration under the *first* codebook version, before certification. The pair compared here is apples-to-apples under the certified codebook.) The suspect list was short: batch composition (gold's reviews ride the census in mixed company), the codebook, or the provider itself.

The experiments were registered before running: a contamination isolation, a codebook two-by-two, and, the part that saved the conclusion, a *contingent* pre-committed to fire if the isolation arm partially recovered. The morning read looked like a confirmed composition effect, and that conclusion was already drafted. Then the contingent fired: a $0.38 re-buy under *matched* composition reproduced the same recovery. The best-supported driver is **buy-time variance of the served model**: across three buy dates the same configuration read 0.799, 0.766, and 0.791 against gold, day-to-day movement of ~0.02–0.03 F1 at temperature zero. Batch composition was acquitted (every same-day test null); the compact codebook was closed the same way (–0.018 [–0.031, –0.005] against full, same-day, drift-clean); and the finding graduated into a standing operational caveat: **any re-buy triggers re-certification**, because the instrument's reading drifts with the provider's serving stack, not with the code.

The gate: numbers that cannot drift silently

Every published number in this report regenerates in CI. The two runs of record (certification and agreement) re-score from a committed fixture on every push, and an exact-digit mismatch fails the build; changing a scorer is only possible through a deliberate, versioned path. That path has been walked once for real: a bootstrap fix (empty-denominator ratios had been scored as 0.0, dragging confidence-interval tails) forced a scorer version bump, and the re-anchored runs came back digit-identical to the old anchors, which is the gate proving that the bug had never touched a published number.

What this refused

The report does not claim the labels are right; it claims their error against human judgment is measured, bracketed census-wide, and wired to stay that way. It does not resolve judge-production disagreement in its own favor: adjudication is pending and says so. It discloses the codebook's mild circularity with gold and schedules the holdout that answers it. And it treats the labeler's quality as a *point-in-time property of a served model*, not of the code: the strongest honest statement available, and the reason the caveat about re-buys exists.

THE GENERAL PROBLEM

Evaluating an instrument you didn't build and cannot open: establish a human reference it cannot influence, extend reach with an independent second annotator rather than a self-grading loop, pre-register the analyses (including the contingent that can kill your own headline), and put the numbers behind a gate that makes silent drift a build failure.

CLOSING

What this earned, and what it didn't

What the milestone set out to test

The founding plan had two halves. The **measurement half**: extract what reviews say into rigorous per-game numbers, the four investigations of this report. And the **explanation half**: an agentic investigator that notices an anomaly in a game's review history (a spike, a bombing, a reversal) and explains it with verified evidence. The two were joined from the start by one rule born of a real constraint (a fixed sample can give defensible numbers *or* chase an anomaly, never both at once), so numbers and stories were separated structurally: every displayed number comes from the survey mint alone; stories travel their own channel and never mint numbers.

Milestone 1 built and certified the measurement half. The explanation half is where the plan changed, and the change should be told as what it was: a measured scope call on stated grounds, not a retreat.

The redirect

The agentic investigator is **deferred indefinitely**. In its milestone slot: a grounded chat over the labeled corpus, the report's interrogation channel. “Type a game name, get the report. Then interrogate it.”

The grounds, as recorded when the call was made: the chat converts this milestone's assets directly. The labeled envelopes are precisely the metadata-filtered retrieval index most retrieval systems lack (“why do people hate the grind?” resolves to aspect $\wedge$ sentiment $\wedge$ game filters *before* any embedding runs, with a measured classifier, published F1 and interval, as the index), and the calibrated judge extends naturally to groundedness and retrieval-quality evaluation. The cost is named with it: the verify-then-explain differentiator goes dormant (deferred, not deleted), and what remains of event awareness is display-only episode markers, pure statistics with no explainer. The deferral's blast radius was verified before ruling: nothing built depends on investigation machinery.

The design bar for the chat is set so it cannot decay into a commodity: every choice must serve a claim a stock retrieval-chat cannot make: retrieval over self-labeled structure, evaluation on an already-calibrated judge, and answers that structurally cannot fabricate statistics (claims pinned to verbatim quotes run through the same fabrication verifier as everything else; numbers appear only as citations of the mint, never phrased by the model; thin evidence is named as thin; refusal is an answer).

Limits, plainly

- **Reviews, not players.** Every number is a share of *reviews in the labeled pool*. Reviewers self-select; no claim about the player base is made anywhere.
- **English-first.** The usable pool is 45% of the raw corpus by explicit rule; the other languages are counted, not read.
- **A point-in-time reading.** The labels are one served model's reading at buy time; the measured buy-time variance (~0.02–0.03 F1 day to day) makes any re-buy a re-certification event by standing rule.
- **The reference is small and the codebook saw it.** Gold is 250 reviews, and the second codebook version was tuned on gold's rulings. A fresh frozen holdout is registered to measure that circularity; until it runs, the certification number carries this asterisk honestly.
- **Agreement is not accuracy.** Where judge and production disagree, which one errs is adjudicated by a human, and that pass is pending.

- **Nothing is deployed yet.** The numbers exist as versioned artifacts behind a CI gate, not behind a URL. Deployment is the next milestone but one claim this report deliberately does not make.
- **The intervals are reported uncorrected.** The report runs many 95% bootstrap comparisons with no family-wise correction; the borderline calls (a lower bound grazing zero) should be read with that in mind. The buy-time variance figure likewise rests on a small number of paired re-buys and is held as an operational rule, not a measured constant.

The road

The sampling study comes next: this milestone's numbers ride a fixed corpus census; the study builds the estimator discipline for defensible numbers from *live* sampling windows. The door for it, a hardened Steam review client with its own probe-verified wire lore, was built and live-smoked within this milestone. Deployment follows; the interrogation chat after that, its evaluation already runnable offline against the census before any deployment exists.

The thesis of this report, restated once: the model call was the smallest part. The vocabulary that makes counting possible, the procurement that made the labeler a measured choice, the engine that made the census a trustworthy record, the harness that put an error bar under every number: that instrument around the model is the work, and it is the part that survives any model swap. Even the spend says so: the measuring apparatus (the bake-off, the judge, the registered experiments) cost roughly $10 of API spend against the census's $3.80.

Colophon

Python 3.13 with uv; SQLite in WAL mode as the one store; ruff, pyright strict, and pytest behind a GitHub Actions gate that re-scores the two runs of record on every push; one provider-agnostic client seam over the labeling and judging APIs; matplotlib and reportlab render this document, which verifies its own numbers against the repository's artifacts at build time. The stack is deliberately boring; the instrument is the work. Report build: git 467471e, rendered 2026-07-31.