Moodinary

RESEARCH RETROSPECTIVE · OCTOBER 2026

Why We Stopped ECO-1

Evidence from two small model experiments

Manuel Nicastro · Moodinary
October 3, 2026

Abstract

ECO-1 explored two uses: extracting contextual entities from organizational comments and classifying moderation-related signals. The entity system used LACE, a cross-review method for proposing candidates, reviewing disagreements without revealing which system proposed them, allowing one bounded reasons round and retaining a conservative floor of flagged items. The moderation system was a separate fine-tuned classifier. Both were trained and checked on synthetic data, but those numbers did not answer the product question we needed to answer. On a small, manually reviewed set of real anonymous comments, the newer entity system matched only 6 of 28 specific indirect references to people or groups, compared with 15 of 28 for its predecessor. These counts are coverage observations, not accuracy estimates: the references were inventoried manually, not independently annotated as a gold test set. The moderation system underperformed the existing baselines, and its early development score was contaminated by training or selection on the same examples. The later version also failed to beat the baselines on a previously exposed test slice. A planned low-rank adaptation pilot was stopped before paid launch because gated-base access, license acceptance, cost limits and a clean validation gate could not all be established. We cancelled the model-building program rather than turn synthetic scores into a product claim. The work now informs two narrower directions: pulse-climate forecasting and reducing platform costs. This paper reports the evidence, the mistakes and the accounting boundaries. It contains no participant text.

1. What ECO-1 was meant to do

Moodinary collects short, anonymous pulses about how people are doing in an organization. The original model idea was to make those responses more useful by identifying contextual entities and detecting moderation or safety signals. The entity-extraction system used LACE: candidate proposals, blind cross-review of disagreements, one bounded reasons round and a conservative floor of flagged items. The moderation model was a separate fine-tuned classifier.

LACE could in principle be used for other tasks that need a candidate reviewed and then kept, dropped or flagged. But the evidence gathered here only concerns entity extraction. It does not establish that LACE is effective across language tasks, or that cross-review improves quality generally.

On October 2, we publicly introduced the two experiments as Nubi previews, under the names nubi-moderation-preview and nubi-entity-preview. That announcement said they were being evaluated and were not integrated into Moodinary. This paper is the later retrospective and records the decision to stop.

We also began with the wrong product emphasis. A custom privacy extractor was not the core need. The useful interface was closer to contextual highlights about time, people and roles, places, and topics. A privacy detector cannot supply that whole experience, and our results do not show that custom development justified its evaluation and operational burden for the narrower PII task.

Figure 1. Sequence only. Dates are shown only where documented; the early experiment stages are intentionally undated.

Figure 2. LACE as used for entity extraction in these experiments. This diagram describes the tested process, not a validated general-purpose method.

2. Entity extraction: strong synthetic scores, weak transfer

The development results

The first entity system was trained on 3,000 authored synthetic examples. Its synthetic development set contained 200 comments, 300 labeled spans and 60 hard-negative comments. The training and holdout rows did not share exact normalized text, template skeletons, family IDs or entity values. They did, however, share the same renderer, ontology, Spanish organizational settings and recognizable formats. One hundred fifty of the 200 development comments shared an entity-type slot pattern with training. The examples were generated by the same small authoring system, not independently written or annotated by people.

There was also an output-format bug. The original frozen exact-span scorer reported an F1 of 0.000 because generated character offsets were wrong, leaving all 300 gold spans unmatched. A later code-only diagnostic located predicted text back in the source and produced an F1 of 0.879 on interpretable outputs; charging malformed and unlocatable items as false positives lowered it to 0.861. A separately authorized serving run on the same synthetic development set scored 0.904 under the source-relocation protocol, compared with 0.672 for the selected cross-review system and 0.509 for the checked baseline on that same development set. The original zero score remains part of the record; the later result does not replace it. The relocation protocol was introduced after the initial failure and is not independent confirmation.

A second version used a new synthetic development set of 280 comments and 259 gold spans. Its exact text-and-type score was 0.903 F1. Charging five unlocatable or repeated items as false positives reduced the supplementary score to 0.894. The 98 hard-negative comments had no accepted false-positive spans. These are useful checks of a controlled pipeline, not evidence of real-world accuracy. Synthetic labels were assigned by the generator, not independently reviewed. The two development results used different sets and scoring histories, so their similarity should not be read as a controlled comparison.

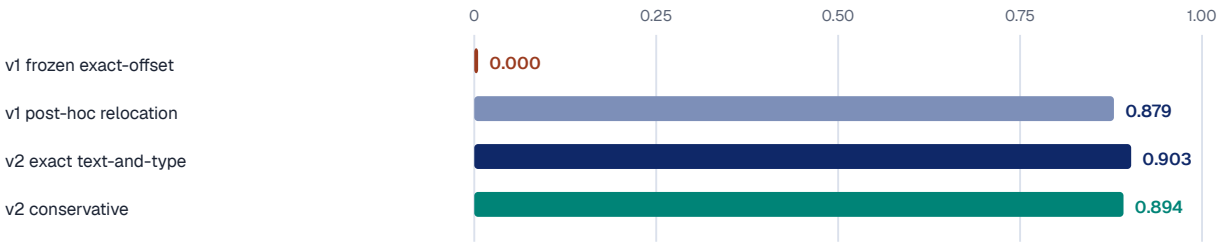

Figure 3. Synthetic development scores only. Versions, sets and scoring protocols differ; these bars are not a controlled comparison and are not real-world accuracy.

The real-comment review

We reviewed an authorized sample of 90 nonblank anonymous comments and manually inventoried 28 specific indirect references to people or groups. The first system's accepted PERSON spans overlapped 15 of those 28 references. The broadened second version overlapped 6 of 28. The second version's normalized-text diagnostic

matched 7, but the primary count remains 6. Missed cases included references in teacher and friend roles; these are category-level descriptions, not excerpts.

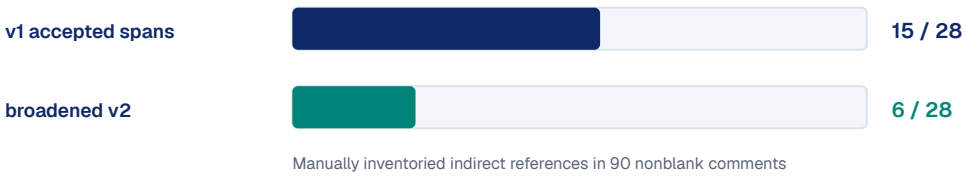

Figure 4. Descriptive overlap counts, not recall, precision or F1. No independently annotated gold set was available.

These are descriptive coverage counts only. There was no independent annotation of every comment, no adjudicated gold dataset and no valid real-data precision or F1 estimate. We do not publish comment text, names, organizations, identities or raw model outputs. The review also exposed a definition mismatch: the earlier benchmark treated PERSON mainly as names and nicknames, while the product need included references expressed through roles, relationships and bounded groups. Broadening the label definition did not improve the observed coverage.

The first attempted private export had selected a demo/test organization and returned 59 comments. We discarded that as evidence about real student responses; none of those rows was marked mock, so a flag alone could not verify provenance. We then confirmed the intended organization and exported only comment IDs and text inside a read-only transaction. That source contained 127 pulses, 90 nonblank anonymous comments and 37 empty comment fields. The counts were checked again against the same source. The earlier mistake is included because it changed what we could validly claim, not because it changed the product data.

The synthetic-to-real gap was not explained by a single bug. The first experiment had a bad offset contract; later evaluation repaired location deterministically, but that did not solve the difference between patterned synthetic mentions and real indirect references. The second system's roughly 0.90 synthetic score and 6-of-28 observed coverage are not contradictory. They measure different things on different data, and only the synthetic score has a structured reference set.

What we can and cannot claim

The training-to-development audit found no direct text, family or exact value overlap. That is narrower than demonstrating independence. Both collections used the same rendering logic and entity ontology. An exploratory subset with unseen combinations of entity types scored 0.817 F1, with lower recall than the full development set; it contained only five families and no hard negatives, so this is a warning, not a clean causal experiment. No held-out test score is reported here. The sealed test set was not evaluated for this result.

The evidence supports a limited conclusion: the system learned useful patterns in its synthetic task, but we did not establish that it could surface the contextual references people needed in actual organizational feedback. A privacy pipeline should not be sold on synthetic development performance alone.

3. Moderation: the benchmark was not enough

The first moderation model was a compact fine-tuned encoder and classifier trained on 212 synthetic examples. That same small development set had been used to train the model, so its reported development score was in-sample and not a valid estimate of generalization. Its separate evaluation on 212 examples was also not pristine because those examples had already been used in earlier baseline work. The second version was evaluated on the same previously exposed 212-example slice.

On that reused evaluation slice, the first model had lower F1 than the supplied nearest-neighbor and full-stack baselines for all four labels. A second moderation version combined 3,000 synthetic examples with 2,968 public examples. The public labels were used only for the moderation target; unobserved labels were masked rather than treated as negatives. Synthetic labels remained author-assigned and the synthetic source had only 30

template families. Training completed on CPU at no compute charge. The selected epoch was chosen using the same 212-example development set, and the subsequent 212-example test was reused and non-pristine. Both model versions and the supplied baselines are shown below. The baseline numbers were supplied from prior work and were not rerun for this comparison.

Label	First version F1	Second version F1	Existing nearest-neighbor F1	Existing full-stack F1
Moderation	0.837	0.462	0.896	0.905
Sarcasm	0.686	0.457	0.797	0.715
Risk	0.942	0.389	0.959	0.964
Fake	0.851	0.635	0.905	0.958

The second version was below both supplied baselines on every label. These results do not say that moderation models are unnecessary. They say this experiment did not produce a better one. The comparison was limited by a small and reused evaluation set, synthetic supervision, incomplete label coverage and different data sources. We stopped rather than continuing to optimize against the same examples.

The October 2 preview page publicly reported F1 values of 0.905, 0.715, 0.964 and 0.958 as the preview system's results. We later found that those values match the supplied full-stack baseline in the internal experiment record, not the preview model's own scores. The model's first-version results were 0.837, 0.686, 0.942 and 0.851; the second version scored 0.462, 0.457, 0.389 and 0.635 on the reused slice. We are correcting the attribution here. Both model-score sets are synthetic and limited; neither is a clean estimate of real-world performance.

4. Why the paid adaptation pilot did not launch

After the CPU experiments, we considered adapting a larger base model for the moderation task. The proposed pilot never reached paid training or hosted evaluation. Access to the candidate bases was gated, license acceptance had not been established, and the team could not verify a mechanical all-in spending limit that covered downloads, startup, training, upload, inference and shutdown. A trustworthy held-out validation gate was also missing. The owner's instruction was to keep the pilot on hold; no paid launch, adapter upload, smoke request or production change followed.

This was a stop decision, not a failed GPU training result. No score from that planned pilot exists. We do not report a speculative quality comparison or describe an unlaunched experiment as completed.

5. Cost accounting

The cost record separates resource usage, billed amounts, usage-derived estimates and unused reservations. They are not interchangeable. In particular, credits can make a bill zero while metered resource usage remains nonzero; an API price calculation is not proof of an invoice; and a reservation is not money spent.

Category	Metered or observed usage	Billed or estimated amount
GPU workspace snapshot after entity v2 training	$1.54 metered at workspace level	$0.00 billed after $1.54 in credits in the returned account snapshot
Named entity-training job resource rows	$1.49 across the listed rows: $0.04, $0.82 and $0.63	No separate per-job billed receipt; the itemized rows and workspace-level summaries do not reconcile
Main text-inference ledger	2,326,196 input tokens over 2,000 completed requests; five started requests have unknown usage	About $0.56 at the archived input-token rate; a separate uncertainty-inclusive estimate also rounds to about $0.56. They overlap and are not invoices
Hosted entity-preview DEV evaluation	43,800 returned tokens; one request's usage is uncertain	No verified invoice

Category	Metered or observed usage	Billed or estimated amount
One isolated entity-preview smoke	224 tokens	Billed dollars were not separately obtained
Planned paid moderation adaptation pilot	No paid training or hosted evaluation launched	No pilot compute charge is known because it did not launch

These figures are not a project-wide total. The workspace snapshot is account-level, not a project invoice. Rounded to cents, the reported ephemeral-app component and aggregate meter are both $1.54; in the underlying records, the component is slightly higher. The three identified job rows round to $1.49, leaving an approximately $0.05 difference against the aggregate meter with no demonstrable allocation. We preserve the discrepancy rather than reconcile it by guess. The inference dollar figures are two estimates of overlapping token usage; rounded to cents, both are about $0.56, and they are not amounts to add. Billing analytics were unavailable, so actual charges for those API requests are unverified. Historical budgets, forecasts and reserves are not expenditure.

The planned adaptation pilot never launched, and its earlier cost ceiling is not spend. We cannot establish an exact all-in dollar total across services from the evidence available.

Production-scale inference: historical input-only proxy

For a rough API comparison, the archived request ledger gives an average of 824.985849 input tokens per moderation call (349,794 tokens across 424 calls) and 4,130.858182 input tokens per full-comment entity pass (1,135,986 tokens across 275 passes). Adding one call of each gives 4,955.844031 input tokens per nonblank comment. At the archived historical rate of $0.24 per million input tokens, that is about $0.0012 per comment. The exact dollar calculations are kept in the internal claims ledger. This is a dated historical assumption, not a current quote or invoice.

Nonblank comments per month	Moderation calls	Entity passes	Combined input-only proxy
1,000	$0.20	$0.99	$1.19
10,000	$1.98	$9.91	$11.89
100,000	$19.80	$99.14	$118.94

Figure 5. Linear-scale illustration from the same historical input-only proxy. It is not a production bill or a forecast of total operating cost.

This is deliberately narrower than a production estimate. It assumes one nonblank comment per pulse and exactly one moderation call plus one full-comment entity pass. It excludes output tokens, retries, extra LACE rounds, aggregation, platform requests, storage, monitoring, human review and support. It does not model whether the current product path would make those calls at this frequency. Empty comments would not incur these assumed model calls, but their share in future traffic is unknown.

Hosted accelerator serving, warm capacity, batching, latency, utilization and storage prices are not established here. A generic compute formula would be hourly accelerator price multiplied by active request time divided by utilization, plus warm-idle hours; the required price, production latency, utilization and idle-capacity inputs are unknown. Retraining and evaluation costs are also unknown because a future dataset, number of runs, serving path and review protocol have not been selected. These costs remain unknown. The historical API proxy should not be compared as if it included GPU hosting, retraining or a complete evaluation program.

6. What changed

We stopped treating a custom fine-tuned model as the default way to improve Moodinary. The next research direction is pulse-climate forecasting, which asks whether patterns across aggregate responses can provide useful early warning without pretending that generated comments are representative data. A second direction is platform cost reduction. Both can be tested against the product's real needs without keeping a weak detector alive just because training already happened.

The work remains useful as engineering evidence. It surfaced the offset bug, showed how an in-sample score can look persuasive, exposed a mismatch between the label definition and the product need, and forced a more careful separation between metered usage and billed cost. Preserving these failures is more valuable than relabeling the experiment as a launch.

7. Limitations and data handling

All reported entity development data were synthetic and authored by a generator. The moderation experiment mixed synthetic data with a public labeled source, but its evaluation remained small and reused. Neither result establishes performance on representative Argentine users. The manually reviewed comment sample is summarized only with aggregate counts. No raw text, names, organizations, identifiers, identifying anecdotes or generated private outputs are included in this paper.

This is a retrospective project note, not a preregistered benchmark paper. Some scoring choices and diagnostics followed observed failures. The reporting distinguishes original frozen results from post-hoc diagnostics, but those choices limit confirmatory interpretation. The project did not demonstrate a deployment-ready system, safety guarantee or general advantage for cross-review.

Conclusion

ECO-1 began with a plausible idea: fine-tune small task-specific systems and use cross-review to make outputs more dependable. The evidence did not justify continuing. Synthetic development scores near 0.90 concealed a transfer problem; the moderation model lost to the existing baselines; and the proposed paid pilot could not clear its access, budget and validation gates. Without appropriate consented real data, or synthetic data that transfers much better, we had no credible next evaluation. We cancelled the model-building program and kept its artifacts as research records. The next work is aimed at the product questions that remain, not at defending the experiment.

Source notes

The detailed evidence, hashes, protocols and accounting source map remain in an internal research ledger. The real-comment sample is not available for reproduction. No participant text is reproduced here.