Episode Book

EPISODE #6236

Synthetic Logs Without Leaking Your Clients

PUBLISHED	24 October 2026 at 11:17 UTC
DURATION	24:27
LISTEN	myweirdprompts.com/episode/synthetic-conversation-logs-privacy-53788/
HOSTS	Corn, Daniel, Herman, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular+meorav (chatterbox-local)
PIPELINE	V5.20
SOURCES	8 cited by the research (listed at the end)
BOOK GENERATED	25 October 2026 at 06:19 UTC

SYNOPSIS

Daniel has real agent-workflow logs full of his own projects and wants synthetic versions that preserve the rhythm and ambiguity without advertising what he's working on. Corn and Herman walk through what purpose-built privacy tooling actually does that a plain rewrite prompt doesn't: which libraries preserve information dependencies, where differential privacy guarantees really enter a pipeline, and why protecting a single turn may not hide a project described across hundreds of them. Along the way: tabular synthesizers, seed-based generators, state-aware agent simulators, and why differentially private synthetic text is still an unsolved problem.

TRANSCRIPT



Corn

Can a model change every name in a conversation and still leave everyone knowing whose conversation it was? Daniel, that's the uncomfortable part of what you're building.



Daniel

I'm building a classifier for agent workflows that spots when a new task should start, when the preceding context is no longer needed. I've got real conversation logs, but they're full of my projects. I want synthetic versions that keep the rhythm and ambiguity without advertising what I'm working on. What specialist tooling actually helps?



Herman

Generation itself is becoming almost incidental to the bill. OpenRouter lists Haiku five point five at ten cents per million input tokens and fifty cents for output. That's a tenfold reduction against the preceding Haiku listing.



Corn

Confidentiality is now the expensive ingredient.



Herman

The listing also gives it a million-token context window. You can present a substantial collection of conversations, but a large window doesn't make copying them safe.



Daniel

That's what I want to separate. I can ask a cheap model to rewrite my logs. What does a purpose-built system do that the prompt doesn't?



Corn

Ideally, notice that changing a repository name hasn't changed the only project on earth with that particular collection of problems.



Herman

Your classifier also needs examples where changing the subject does not mean changing the task. A user says, fix the deployment, then asks about credentials. That's potentially one task. They say, forget the deployment, draft the client update. That's potentially a boundary, despite the shared vocabulary.



Corn

Potentially matters. The update could be part of handling the deployment. If we generate conversations with beautifully explicit transitions, we train a classifier to recognize people who announce their intentions. A select population.



Daniel

Yes. I don't want every synthetic example saying, now I am beginning a completely unrelated task.



Herman

Corn, would you preserve the original topics, or move everything into another domain?



Corn

Neither wholesale. Preserve the dependency. Change the setting. If the next request needs the previous tool result, that relationship survives. Whether the tool result concerns a repository or an inventory system can change, provided the replacement doesn't make the interaction unnatural.



Herman

That's a useful constraint for generation. Preserve information dependencies, turn order, interruptions and ambiguous references. Change the identifying content. You can retain a label while destroying the reason that label was correct, so the label needs rechecking afterward.



Daniel

Are the tools mostly starting from real examples, then?



Herman

Two families. Generation from scratch uses specifications, personas or scenarios without real conversation seeds. Seed-based systems learn from or transform private records. The latter gets you closer to observed behavior, but introduces the direct route for copying private material.



Corn

Starting from scratch has its own trap. You write your idea of a realistic conversation, then congratulate the model for agreeing.



Herman

You need the real logs to tell you what behaviors are missing, even if those logs never reach the generator. Aggregate descriptions can guide the scenarios. Though... aggregate descriptions can reveal things too, if the group is small enough.



Daniel

My business-identifying information is broader than names. Products, clients, internal terminology, the particular shape of the work.



Corn

A name detector will find Alice. It won't necessarily find the migration nobody outside Alice's company is doing.



Herman

Privacy tooling has concepts for identity, attribute and group disclosure. Proprietary business communications are recognized as sensitive. But business-identifying information, in your particular sense, isn't a distinct tooling category across these systems. You're describing an application-specific inventory of secrets.



Corn

Herman, you're about to offer him six libraries and a weekend.



Herman

Several libraries. The weekend is optional.



Corn

Your installations rarely agree.



Herman

This time I'm distinguishing the library from the work around it. Tabular tools are a useful place to see that distinction. Google DPSynth generates differentially private synthetic tables, and explicitly remains under active development. NeMo Safe Synthesizer has processing, PII replacement, training, generation and evaluation stages. Its differential privacy is optional, so choosing the product doesn't automatically choose that protection.



Daniel

Would either of those suit conversation logs if I put each turn in a table?



Herman

Packaging text into rows doesn't turn dialogue into a conventional tabular problem. They may help with structured metadata, but the dependencies across turns still need modeling. Mostlyai-engine is worth examining because it separates those uses. Its TabularARGN handles flat and sequential tabular data. Its LanguageModel handles semi-structured textual tabular data.



Corn

A table can hold a conversation. So can a restaurant.



Herman

Mostlyai-engine is Apache-licensed and supports optional differential privacy. Interestingly, its language model defaults to a lightweight LSTM trained from scratch, rather than assuming a giant pretrained model. That gives you a different starting point for controlling where knowledge comes from.



Daniel

What about deliberately sacrificing fidelity? Can a tool make that choice explicit?



Herman

Metasyn does. It's MIT-licensed and deliberately limits what statistical information it models. By default it doesn't incorporate multivariate information. It can preserve individual column distributions without preserving all the relationships among them.



Corn

I'd trust that description more than a promise to preserve everything.



Herman

You would. Although, Corn, for this classifier, would you accept losing relationships?



Corn

Not the ones that define the boundary. Lose the relationship between client identity and project type, happily. Lose the relationship between a pronoun and the result it refers to, and we've broken the example. I like limited tools when the limits are in the right place.



Herman

That's precisely the selection you have to specify yourself. SAS's dpmm offers end-to-end differential privacy guarantees for its supported synthesis process. But none of these table-oriented options, by itself, understands which conversational dependencies Daniel needs.



Daniel

The text-specific tools sound closer. Are any actually built around private seeds?



Herman

RPSG, Private Seeds, Public LLMs, is directly relevant. An abstraction model produces candidates aligned with the seed's sentiment. A formal differential privacy mechanism selects candidates. An LLM makes variations, followed by refinement to reduce memorization and redact personal information.



Corn

Sentiment survives. The task boundary might not.



Herman

That's the part you'd adapt and test. Its reported membership-inference attack performance is around fifty percent area under the curve, meaning that tested attacker is around chance. Useful evidence, not a certificate against every possible attacker.



Daniel

Where does the privacy guarantee enter? The abstraction step could already have copied something.



Herman

Through the specified mechanism and its assumptions, rather than the word abstraction. Differential privacy bounds how much an individual's contribution can change the released result. A pipeline has to account for the private information it uses, including selection. An unaccounted private-data-dependent step can spoil the claim.



Herman

Also, you must define the protected unit. A turn, a conversation, a person? Protecting one turn doesn't automatically hide a project described across hundreds of turns. For your business information, that distinction is enormous.



Corn

The repository gets a hundred votes in the supposedly anonymous election.



Daniel

I'd need protection at something closer to project level, then, not just record level?



Herman

At least a threat model that addresses project-level disclosure. Formal individual-record protection and hiding a shared business secret are different objectives. Stronger privacy also costs utility. A survey by Schlegel and colleagues found substantial degradation on specialized real-world datasets at epsilon of four or below. Rare, distinctive patterns are often useful and identifying simultaneously.



Corn

That's the tension between fidelity and privacy. Before we go further, I think we've got a caller on the line. Hello?
Ivan: Foxtrot nine. Forty one, sixty seven. Twenty two, zero eight. Forty one, sixty seven. The second stitch is dropped before the sleeve. The wool is combed, not spun. The pattern holds no warmth. Safehouse closed.



Herman

Wrong number.



Corn

What on earth was that?



Herman

SDialog takes the from-scratch route with persona-driven multi-agent dialogue generation. It's MIT-licensed, and includes comparison tools for judging how realistic generated dialogue looks. That's a candidate for generating your scenarios, not a formal privacy guarantee.



Daniel

Can these systems preserve tool state? My examples involve agents, rather than just people talking.



Herman

StateGen is particularly relevant there. It uses a user simulator, the agent under test, a tool simulator and a judge, with an authoritative state manager. The backend is treated as truth, so a conversational claim can't simply rewrite the simulated world.



Corn

You can say the deployment succeeded. The server can disagree.



Herman

MAG-V addresses another useful piece, generating customer queries and reverse-engineering alternate questions to verify trajectories. That could help test whether different wording still produces the intended workflow. Again, trajectory consistency and privacy are separate capabilities.



Daniel

UNCASE came up as a privacy-first option. Where does that fit?



Herman

It generates from structured seeds intended to contain zero real data, with scanning, validation and a nine-metric quality gate around the pipeline. It also offers differential private fine-tuning with epsilon at most eight. That fine-tuning setting doesn't confer a guarantee on every earlier operation.



Corn

The blueprint needs scrubbing before the model gets creative.



Herman

Especially if you derive it from real conversations. A seed saying, a consultant maintains this unusually named internal service for this distinctive client, has already carried the sensitive material across.



Daniel

I'm hearing components rather than one system that understands the whole job.



Corn

A fairly well-stocked workshop. The suspicious stains are still yours.



Herman

Domain-specific text remains difficult. SynBench calls differentially private synthetic text generation an unsolved challenge, with quality deteriorating as private data diverges from the generator's pretraining distribution. Its contamination checks also flagged four hundred and sixteen potentially contaminated samples in a dataset of roughly five thousand.



Daniel

Contaminated meaning the generator might already have seen the supposed private examples?



Herman

Yes. Actual overlap with non-private pretraining can make fidelity look better and undermine the intended end-to-end privacy claim. Mere resemblance is different. A model knowing public programming conventions doesn't invalidate a correctly implemented differential privacy mechanism. Exact prior exposure, or private information reaching the model outside that mechanism, changes the situation.



Corn

Knowing what a repository is doesn't mean knowing yours.



Herman

Nor does poor utility prove privacy. An incoherent output can still quote a secret.



Daniel

Then what would you actually measure? I don't want a report saying the dialogue sounds natural while a client name is sitting in the middle.



Herman

Separate gates. For structured fields, SDMetrics gives model-agnostic quality and privacy metrics. Quality measures statistical similarity. Privacy measures include distance-to-closest-record baseline protection, overfitting protection, disclosure protection and CategoricalCAP. Those test different ways synthetic output can reveal source records or attributes.



Corn

Close enough to be useful is also close enough to inspect.



Herman

Distances need sensible representations. Text embeddings can help find suspicious neighbors, but a small embedding distance doesn't tell you whether a shared phrase is confidential or merely common. Conversely, a distinctive product identifier can be buried in an otherwise dissimilar paragraph.



Daniel

What does its empirical differential privacy feature establish?



Herman

An empirical estimate under the audit's setup. It doesn't replace a mathematical guarantee covering the mechanism and its assumptions. Membership inference asks whether an attacker can distinguish training members from non-members. Post-hoc approaches such as Zero-Run use fixed member and non-member datasets without retraining the generator.



Corn

Passing one attack means that attack failed.



Herman

Yes. SynBench found observed privacy loss could exceed what claimed configurations suggested. That's a reason to examine implementation, accounting and assumptions, not a reason to conclude that a valid theorem stops working.



Herman

Corn, which would you inspect first here, membership risk or copied content?



Corn

Copied content. Daniel already knows specific things he doesn't want released. Check those directly before designing an attacker that might rediscover them. Exact strings, close variants, combinations of clues. Then membership and attribute attacks address what the list misses.



Daniel

How do I avoid treating a vocabulary list as the entire privacy problem?



Corn

Give it relationships. A client name has aliases, a product has an old name, a repository has a description. Changing the main identifier while leaving the distinctive description is cosmetic. I'd also ask whether someone familiar with your industry could identify the project from the remaining combination.



Herman

You can use entity recognition and custom dictionaries as the first layer, then targeted model review for descriptions and combinations. Keep that review separate from generation. The generator shouldn't award itself the privacy pass.



Corn

It has already worked very hard on the essay.



Daniel

Fine. Privacy is a set of tests. How do I prove I haven't flattened all the boundaries?



Herman

Use the actual downstream task. Hold out real conversations before building the generator. Train the classifier on synthetic data and evaluate locally on that untouched real set. Compare missed boundaries, false boundaries and performance on ambiguous cases. Keep projects or conversation families separated when splitting, so near-duplicate situations don't leak across.



Corn

Include topic changes within a task and task changes without a topic change. Otherwise vocabulary will do the classifier's homework.



Herman

Also compare boundary frequency, position, context length and dependency patterns. Not just average turn length. You want the classifier to recognize when earlier context stops being necessary, rather than when the synthetic user becomes unusually polite.



Daniel

Do the dialogue libraries give me useful checks before that expensive downstream evaluation?



Herman

SDialog's DatasetComparator includes an LLMJudgeRealDialog for realistic-dialog rate and linguistic measures such as reading ease. UNCASE's gates cover overlap, factual and semantic fidelity, lexical diversity, coherence, embedding drift, tool validity, privacy and memorization. That's useful coverage. None is a single score proving both realistic



Herman (continued)

behavior and absence of identifying information.



Corn

Reading ease won't detect an easy-to-read secret.



Daniel

Where would you use Haiku, then? I'm not looking to fine-tune a generator just because there's a framework for it.



Herman

I'd use a cheap general model for constrained generation from sanitized structural seeds. The Risk In Context benchmark found a cheap general model could offer a relatively good privacy-utility compromise, but also found in-context generation repeating seed rows verbatim. Those were tabular tests, so don't turn them into a measured leakage rate for your conversations. The copying risk is still directly relevant.



Corn

Ten times cheaper copying remains copying.



Herman

One larger model in that benchmark had attack detection rates up to fifty-four percentage points above the safest baseline at a one-percent false-positive rate. Capability alone doesn't buy privacy. Better pattern reproduction can mean better reproduction of the private bits.



Daniel

Could I just tell the model never to reproduce any phrase from the original?



Corn

Then ordinary technical language becomes contraband, while paraphrased secrets get through.



Herman

Prompt restrictions help steer behavior. Enforced rejection catches what steering misses. Keep an overlap detector, identifier scanner and semantic review outside the generator. Generate candidates, reject failures, and track why they failed. Rejection rates can reveal that a seed itself remains too identifying.



Daniel

What would the seed actually contain for one boundary example?



Corn

Something like, ongoing troubleshooting, user requests another check dependent on the previous tool result, then explicitly abandons that goal and asks for an independent deliverable. Preserve where the transition occurs. Replace the domain, entities and tool details coherently.



Herman

Preserve the awkward cases too. A request that sounds independent but requires the previous artifact. An apparent continuation that introduces a new goal. You can specify these structures without transmitting the original project.



Daniel

That sounds viable, but a lot of the important ambiguity could vanish during abstraction.



Herman

It could. That's why the real held-out evaluation matters more than the generator's self-assessment. I'd pilot a small batch against the untouched classifier test set before manufacturing a mountain of examples. DP fine-tuning is another route when you control



Herman (continued)

the model and need formal accounting, but it isn't generally something you can bolt onto an API-only generator.



Corn

Herman has restrained himself from installing a mountain.



Herman

I'm recommending a hill. With measurements.



Corn

A source-controlled hill.



Herman

There is one useful shortcut in the scrubbing step. Distinctive nonhuman names should already be handled as organization or product entities, so once those categories are covered you can...



Hilbert

Leave the boat alone, Herman. That's what I'm doing, and the marina committee is refusing the dataset.



Hilbert

I'm housesitting for our secretary, Mr. Pell, this month. The committee's conversation archive is on his computer. They're still arguing about whether the waiting list starts at application or payment. I'm making a synthetic copy for them. Every person's name changes. Every client changes to the name of another client. They're all approved names, so I don't see why the secretary objects. The boat stays called Pending Approval. That's its name on the mooring list.



Corn

That gives the minutes an unfortunate amount of suspense.



Hilbert

The names aren't the problem now. Mr. Pell says everyone at that marina knows whose boat Pending Approval is. There's only one with that name there. Your organization list won't catch it unless someone puts it on the list. I've got the archive running through a scrubber on my laptop, beside the kettle, plugged into the kitchen socket. It prints a clean report. The committee keeps rejecting the report.



Herman

The scan can be clean against its configured categories and still miss that name. You're right about the category coverage.



Hilbert

I'm with you on the checking. I'm checking constantly. Corn, your point about combinations is costing me afternoons. The committee requires each disputed replacement to be approved by the person it no longer refers to. Six people now approve themselves twice. I still have the dataset. I keep meaning to give it back, but the secretary says a copy doesn't leave his computer when I return mine.



Corn

He's making you return the absence of a copy.



Hilbert

He wants the archive usable without putting members' business in it. He's doing the reviews himself after work.



Herman

...Deletion and retention belong in the pipeline too. You need to know where the originals, intermediate abstractions, rejected candidates and review logs end up. A clean final dataset doesn't erase private material left in a temporary artifact.



Daniel

Would you keep the mapping from original entities to replacements?



Corn

Only inside the protected environment if you need it for consistency or checking. Publishing the mapping would rather spoil the change of names.



Herman

Even internally, it needs access control and a retention decision. Stable replacement identities help preserve references across turns, but they can also preserve a recognizable pattern across the whole collection.



Daniel

Then the specialist layer is partly keeping track of those decisions, rather than simply choosing a better generator.



Herman

That's what I'd buy or build around first. Reproducible seeds, consistent replacement, provenance, rejection rules and separate evaluation. A framework earns its place when it makes those reliable, or supplies formal privacy you actually need and can account for.



Corn

One adjacent detail I like is metasyn's auditable saved model. You can inspect what statistical information it retained. For this job, an inspectable intermediate description could be more valuable than another polished synthetic conversation.



Herman

Especially when you're deciding whether that description still reveals a project.



Corn

My recommendation is a cheap generator behind a custom project-scrubbing layer, with separate privacy gates and a real held-out classifier test. As generation gets cheaper, the harder question will be who maintains the list of things like that boat, before they reach the generator.



Herman

Thanks to Hilbert Flumingtop for producing the show.



Corn

If you want more of this, try episode twenty-four eighty-three, Substitution Anonymization; and episode sixteen, On Deepfakes, SynthID, And AI Watermarking. This has been My Weird Prompts, the human-AI collaboration podcast.



Herman

If you enjoyed this one, leave us a review on your podcast app. It helps people find the show.



Daniel

Thanks, both. See you soon.



Corn

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Google DPSynth [primary]**
   (DP synthetic tabular data library, active development)
   https://github.com/google/dpsynth

2. **NVIDIA NeMo Safe Synthesizer docs [primary]**
   and https://nvidia-nemo.github.io/Safe-Synthesizer/latest/
   https://docs.nvidia.com/nemo/microservices/26.3.1/safe-synthesizer/about/index.html

3. **mostlyai-engine [primary]**
   (Apache-2.0, last push 2026-03-02)
   https://github.com/mostly-ai/mostlyai-engine

4. **metasyn [primary]**
   (MIT, last push 2026-10-22; JOSS 2025)
   https://github.com/sodascience/metasyn

5. **SDialog [primary]**
   (MIT, EACL 2026 paper arXiv:2506.10622)
   https://sdialog.github.io/

6. **UNCASE [primary]**
   (BSD 3-Clause)
   https://github.com/uncase-ai/UNCASE

7. **SDMetrics [primary]**
   and https://github.com/sdv-dev/SDMetrics (quality + privacy metrics)
   https://docs.sdv.dev/sdmetrics

8. **OpenRouter Claude Haiku 5.5 listing**
   (fetched 2026-10-24)
   https://openrouter.ai/anthropic/claude-haiku-5.5