Episode Book

EPISODE #5517

What 5,000 AI-Written Podcast Episodes Reveal

PUBLISHED	24 September 2026 at 13:16 UTC
DURATION	24:25
LISTEN	myweirdprompts.com/episode/ai-podcast-corpus-repetition-analysis/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:37 UTC

SYNOPSIS

A listener built a Hugging Face dataset of 5,000+ podcast episodes, each stamped with the model that generated it and synced daily. That makes it a longitudinal, model-attributed corpus — an ablation study where the treatment is the model version and the outcome is stylistic quality. This episode digs into what that artifact can actually measure: why verbatim repetition is a solved problem while semantic repetition slips past every keyword method, how U-shaped attention bias and lost-in-the-middle effects turn a 40-turn script into a loop machine, and why repetition is really a model stalling rather than malfunctioning. Then the concrete toolkit — sentence embeddings with non-adjacent pair filtering, CASE for anisotropy, BERTopic for intra-episode recurrence, c-TF-IDF for per-model tic profiles, compression ratio as a cheap first signal, and correlation dimension for a plottable per-episode scalar.

TRANSCRIPT



Corn

There's a version of this show that exists in two places at once. The one you're hearing, and the one sitting in a dataset, tagged with which model wrote it.



Herman

Which is a strange thing to think about while we're mid-sentence.



Corn

It gets stranger. Daniel wrote in with a whole theory about it. He says this podcast is a scaled-up experiment in AI content creation, and maybe the most scaled-up AI-generated podcast anyone's built yet. He's honest that he's got no objective basis for that claim, it's a hunch. But the useful part is what he does with the hunch. He's kept the transcripts, every episode, public, and he built a Hugging Face dataset that syncs once a day. Full transcript, which model generated it, the audio URL, the metadata, the runtime. The whole production record.



Herman

And he wants to know what to do with it.



Corn

Two things. First, treat it as an evaluation. We've gone Gemini 3, then DeepSeek, now Flash 4.1, and he wants to know whether specific models handle repetition worse than others. Not the word-for-word kind. The circuitous kind, where the same point gets looped over in slightly different words. He thinks it's a loss-of-context-in-the-middle problem, and he wants to correlate it back to models. Second, he wants to fine-tune the script model to weed out the verbal tics and overused idioms he's tired of, and he's asking for our actual recommendations. Best techniques, best packages, for corpus analysis and for tackling repetition specifically.



Herman

So he's handed us a dataset and asked us to build the toolkit.



Corn

He's handed us a dataset and asked us to build the toolkit. Where do you want to start?



Herman

With what the artifact actually is, because I don't think people appreciate how rare it is. Most model evaluations are one-shot. You prompt a model, you score the answer, you throw the prompt away. What Daniel has is longitudinal and model-attributed. Five thousand plus episodes, each one stamped with the model that wrote it, plus runtime, plus audio. That's an ablation study where the treatment is the model version and the outcome is stylistic quality, and he didn't have to design it, he just had to keep the receipts.



Corn

The receipts are the whole thing. A normal benchmark tells you a model can answer a question. This tells you a model gets tired in a specific way over forty turns.



Herman

And it's a within-subject design, which is the strong version. Same show, same hosts, same format, same producer, only the model changes. That controls for almost everything that usually confounds this kind of comparison. You can ask, does Flash 4.1 loop more than DeepSeek did, and actually mean something by it.



Corn

One caveat before we go further. The organization URL on Hugging Face didn't resolve when we checked it. That's almost certainly a slug mismatch, the org name is probably lowercase, but it's worth confirming before anyone goes looking. The dataset exists, the exact link should be verified.



Herman

Fair. And the "most scaled-up" claim is a hypothesis, not a fact. He says so himself. I'd treat it as an open question, because NotebookLM audio overviews pushed a lot of synthetic dialogue into the world, but those aren't archived as a labeled corpus the way this is. The scale claim is arguable. The dataset claim isn't.



Corn

So we have the artifact. What does it reveal about how models fail?



Herman

The failure mode Daniel names is the interesting one, because it's the one standard metrics can't see. There's a paper from this year on visual podcast generation, and the line in it that matters is that BLEU and ROUGE fail to capture conversational naturalness, personality, narrative flow. They reward safe, repetitive outputs over engaging storytelling. That's the trap. If you score a script with n-gram overlap against a reference, the model that repeats itself politely scores well.



Corn

So the metric and the problem point in opposite directions.



Herman

They do. Which is why Daniel's distinction between verbatim and semantic repetition is the whole ballgame. Verbatim repetition is trivial. Same section, same words, you catch it with n-gram overlap, token-level repetition metrics, exact match. It's a solved problem and it's also not what he's seeing. What he's seeing is a point getting re-derived. The hosts make the same argument three times in twenty minutes, each time in fresh wording, each time as if for the first time.



Corn

And nothing in a keyword method fires, because the words are all different.



Herman

Nothing fires. You need meaning-level comparison across non-adjacent spans. That's a discourse-level failure, not a lexical one. The model isn't repeating a string, it's re-deriving a conclusion it already reached and forgot it reached.



Corn

Which is where the context mechanism comes in. Walk me through it.



Herman

The canonical result is the U-shaped attention bias. Tokens at the beginning and the end of a long input get higher attention regardless of relevance, and the middle gets under-weighted. It's not that the model can't see the middle, it's that attention distributes unevenly across position, and relevance doesn't fix it. There's a follow-up that extends it to multi-hop settings, where performance degrades between pieces of information, not just with distance from the edges. And the severity is measurable. One study puts the accuracy gap between peak position and trough position in a long context at twenty-one point nine percentage points.



Corn

That's not a rounding error. That's the difference between a model that read the document and a model that skimmed the top and bottom.



Herman

And a podcast script is exactly the shape that punishes it. Forty, fifty turns. A point gets made early, then the conversation moves on, then twenty turns later the model has lost the early point and reconstructs it from scratch, slightly differently. That's the circuitous pattern. It's not a bug in the writing, it's the attention distribution doing what it does.



Corn

There's a second frame I like better, though, which is that repetition is a fallback.



Herman

Go on.



Corn

The ordering in the literature is that as models get more advanced, failure shifts from sequence repetition toward degenerate text and then toward hallucination. So repetition is the cheapest failure available. It's what a model does when it's out of ideas but not yet willing to make something up. Daniel's circuitous episodes aren't a model malfunctioning, they're a model stalling.



Herman

That reframes the whole thing. A stalling model is a model that's run out of new things to say about the topic and is filling time by re-approaching the same idea from a new angle. Which is, honestly, what a tired human host does too.



Corn

Which is why the human ear is so good at catching it and the metrics aren't.



Herman

Right, and that's the gap we need to close with tooling. So let's get concrete. If Daniel wants to detect this, the workhorse is sentence embeddings. The sentence-transformers package, models like all-MiniLM-L6-v2 if he wants speed, all-mpnet-base-v2 or one of the BGE models if he wants quality. You encode every turn, you compute cosine similarity across all pairs, and you flag the ones above a threshold.



Corn

With one refinement that matters more than the model choice.



Herman

Which is that you skip adjacent pairs. Neighboring sentences are supposed to be similar, that's what coherent conversation looks like. You only care about non-adjacent pairs. Two turns thirty positions apart that score point eight five cosine similarity are a signal. Two turns next to each other that score point eight five are just a conversation.



Corn

And then you look for clusters, not individual pairs. One high-similarity pair is a coincidence. Five turns spread across the episode that all cluster together, that's the signature.



Herman

That's the signature. And there's a caveat with embeddings that will bite him if he doesn't know about it. Sentence embeddings are anisotropic. They cluster in a narrow cone, which inflates the baseline similarity between everything. If every pair in the episode scores point seven, your point eight five threshold is meaningless.



Corn

What's the fix?



Herman

There's a technique, CASE, that subtracts a condition embedding to improve isotropy. The idea is you compute a mean embedding for the corpus, subtract it out, and the remaining vectors spread more evenly across the space. It's the same instinct as mean-centering in any other statistical context. If his raw cosine scores look uniformly high, that's the first thing to try.



Corn

Let me push on the model choice, because I think people default to the biggest embedding model and that's often wrong here.



Herman

It is wrong here. MiniLM is fine for this. You're not trying to resolve fine semantic distinctions between near-synonyms, you're trying to catch a point being made twice. That's a coarse signal. A small fast model that lets you embed five thousand episodes overnight beats a large slow one you run on a sample.



Corn

And the other thing I'd add is that embeddings catch the symptom, not the disease. They tell you where the loop is. They don't tell you why.



Herman

Agreed, and that's where the second layer comes in. Topic modeling. BERTopic is the standard modern package, it's sentence embeddings plus UMAP plus HDBSCAN, and the useful move for Daniel isn't fitting topics across the corpus, it's measuring topic recurrence within a single episode. A point that keeps resurfacing shows up as repeated topic membership. If one episode is sixty percent one topic in five separate bursts, that's the circuitous pattern quantified.



Corn

And there's a cross-corpus version of that which is exactly what he wants for the model comparison.



Herman

There is. Bidirectional topic matching. You train a topic model per model-version, Gemini 3 topics, DeepSeek topics, Flash 4.1 topics, and you measure thematic overlap and divergence between them. If Flash 4.1 keeps circling a smaller set of themes than DeepSeek did, that shows up as higher intra-model topic concentration. That's a model-level fingerprint.



Corn

Now the per-episode scalar. He wants a number he can plot over time.



Herman

Two candidates. The first is correlation dimension, which is a fractal-geometric self-similarity measure. It captures hierarchical recurrence structure, so it detects multiple forms of degeneration, not just keyword repetition, and it's robust to quantization and works across architectures. It gives you one number per episode. That's exactly the shape he wants for a time series keyed by model.



Corn

The other one is cheaper and I'd honestly start there.



Herman

Compression ratio. Compress the episode, compress a shuffled control, compare. High redundancy compresses better. It's model-agnostic, it's cheap, it takes an afternoon to implement, and it'll tell you within a day whether there's a signal worth chasing with the fancier methods.



Corn

And then the idioms. He specifically wants to catch overused phrases and vocal tics.



Herman

That's a different tool. KeyBERT for keyword extraction via embeddings, and c-TF-IDF, the class-based TF-IDF that comes out of BERTopic, for distinctive phrases per model. c-TF-IDF is the right one for his question, because it's class-based. You treat each model as a class and it surfaces the phrases that are distinctive to that model relative to the others. That gives you a per-model tic profile. If Flash 4.1 says "here's the thing" four times more often than DeepSeek did, c-TF-IDF is what tells you.



Corn

And for raw phrase frequency against a baseline, n-gram frequency with a log-likelihood ratio. nltk, spaCy, textacy. That's the boring version and it works.



Herman

It works and it's the right first pass. One thing I'd flag, though. All of this statistical machinery flags candidates. It doesn't adjudicate them. There's a reason the podcast-generation paper used an AI-as-a-judge pass alongside the statistical metrics. You flag with embeddings, then you validate with a judge that's prompted to ask one question: is this the same point made more than once? That's the human-in-the-loop substitute, and it's cheap.



Corn

And there are two metrics from that same work I'd steal outright, because they're almost free. Average turn length and speaker switch rate. Both are trivially computable from the transcripts he already has, and both will move when a model starts padding. A model that's stalling tends to write longer turns and switch speakers less, because it's monologuing instead of conversing.



Herman

That's a good catch. Speaker switch rate is a naturalness proxy you can compute in ten lines of code.



Corn

So the detection stack is roughly four layers. Embeddings for semantic repetition, topic modeling for idea recurrence and cross-model comparison, a per-episode scalar for the time series, and a judge for validation.



Herman

Four layers, and I'd store every score back in the Hugging Face dataset keyed by model and episode. That's the move that turns this from an analysis into an asset. Every score you compute becomes a column, and the next person who downloads the dataset inherits your work.



Corn

Which raises the question of what you do once you can flag it. Detection isn't the goal.



Herman

No, and this is where I'd push back on the instinct to go straight to fine-tuning. There's a decoding-time fix that's dramatically cheaper, and it should be his first experiment. It's called Avoidance Decoding. The idea is you penalize the logits that are similar to previously generated content, at generation time. No training, no dataset, no GPU hours. It gets up to two point six times higher diversity and roughly thirty percent repetition reduction.



Corn

Thirty percent with no training.



Herman

With no training. Which means the honest recommendation is: try the cheap fix first, measure it with the detection stack you just built, and only reach for the fine-tune if the cheap fix isn't enough. Most people skip that step and spend a month training something they didn't need.



Corn

Okay, but say the decoding fix isn't enough. Walk me through the fine-tune.



Herman

The natural fit is Direct Preference Optimization. You build chosen and rejected pairs. Chosen is a de-repeated rewrite of a circuitous segment, rejected is the original circuitous version. You're teaching the model a preference, not a rule, which is exactly the right shape for style control. You can't write a rule that says "don't re-derive the same point," but you can show the model a thousand examples of the same point made once.



Corn

Where do the pairs come from?



Herman

They come from the detection stack. The embeddings flag the loop, the judge confirms it's a real loop, and then you have a human rewrite it. That's the annotation pipeline, and it's why the detection work isn't wasted effort even if he ends up fine-tuning. The flags become the annotation queue.



Corn

And there's a refinement to the DPO setup that maps onto his annotations directly.



Herman

There is. There's a generalization that assigns a per-pair temperature based on semantic-gap annotations. Category, magnitude, confidence. So instead of one global weighting on every preference pair, each pair gets its own weight based on how big the gap is between chosen and rejected. Daniel's annotations are exactly that shape. Idiom type, severity. He's already planning to label those things, and that labeling can feed a per-pair weighting scheme instead of a flat one.



Corn

That's a nice fit. The annotation he was going to do anyway becomes the training signal's structure.



Herman

And if the goal narrows specifically to loop suppression, there's a method aimed directly at it. It decomposes degeneration into loop entry risk and loop persistence, and shows persistence is controlled by escape mass in token sampling. Then it post-trains with onset-centered positive and negative continuation pairs. If the fine-tune's whole job is killing the circuitous pattern, that's the most on-target objective available.



Corn

Let me put the tradeoff plainly, because I think it's the practical takeaway. Decoding-time fixes are cheap, reversible, and don't touch the weights. Fine-tuning is expensive, permanent, and changes the model's voice in ways you didn't ask for.



Herman

That's the tradeoff, and I'd add one more asymmetry. A decoding fix is per-request. You can turn it on for the episodes where you want it and off for the ones where you don't. A fine-tune is baked in. If you train out the repetition and accidentally train out the personality along with it, you've made the show worse in a way that's hard to undo.



Corn

And the personality is the product here. A podcast with no verbal tics is a podcast with no voice.



Herman

Which is the second-order risk nobody talks about. The tics are partly what makes the hosts sound like hosts. You want to remove the ones that read as filler and keep the ones that read as character. That's a judgment call, and it's exactly the kind of judgment call you can't automate, which is why the human rewrite step in the DPO pipeline isn't optional.



Corn

So the recommendation isn't "fine-tune" or "don't fine-tune." It's sequence it. Measure first, cheap fix second, fine-tune only if the numbers say so.



Herman

And measure continuously, because the models change. He went Gemini 3 to DeepSeek to Flash 4.1. The failure profile moves with the model. A one-time audit tells you nothing about the next model he switches to. A standing pipeline tells him within a week of the switch.



Corn

Which is the real argument for the dataset. It's not a record of the past. It's an instrument.



Herman

It's an instrument, and the model by failure-mode matrix is the readout. For each model, repetition rate, idiom overuse rate, context-loss rate. Three columns, three models, and you can see whether the switch to Flash 4.1 was an upgrade on the axes he cares about or just a lateral move.



Corn

One more thing before we move on. He asked about the most scaled-up claim, and I want to give him an honest answer. I don't know. Nobody's published a census of AI-generated podcasts with labeled provenance. What I can say is that the labeling is what makes this unusual. There are probably bigger AI-generated audio corpora. There are almost certainly none that are this long-running and this consistently attributed.



Herman

The attribution is the moat. Anyone can generate a lot of audio. Keeping the model ID attached to every episode for five thousand episodes is the part that takes discipline.



Corn

Herman, you've been at this desk a long time. What's the thing you'd tell him to build first, if he only builds one thing?



Herman

The per-episode redundancy scalar. One number, plotted over time, colored by model. Because it's the thing that tells you whether any of the rest is worth building. If the line is flat, there's no problem and he can stop. If the line moves when the model changes, he's found something real and everything else is refinement.



Corn

One number, colored by model. I like that.



Hilbert

...and the logbook goes back to eighty-seven, so I've got the whole run.



Corn

Sorry, the whole run of what?



Hilbert

The repetition flags. I did that for a radio station out of Trenton. Call-in show, four hours a night, and my job was to listen and write down every time the host told the same story twice. Paid per flag. Nine dollars a flag, which sounds like good money until you realize how often they do it.



Herman

You were a repetition consultant.



Hilbert

That's what the invoice said. The manager called it content auditing. I called it listening. The host had a story about a dog that got out through a gate, and he told it three times in one hour. Different setup each time. First time the gate was open. Second time the latch was broken. Third time I don't think there was a gate at all. Same dog, same street, same ending.



Corn

You flagged all three.



Hilbert

I flagged all three. He was furious. Went to the manager. Manager backed me up, because the callers had started sounding bored and the manager could hear it in the phones. That's the part your embeddings won't catch. The callers sounding bored.



Herman

That's the human ear thing.



Hilbert

The station replaced me about a year later with a program that counted phrases. It looked for "as I was saying" and "like I mentioned." Ran it on the same tapes. It missed the dog story completely, because the dog story never used the same words twice. It caught a



Hilbert (continued)

weather guy saying "cold front" eleven times in a shift, which nobody cared about.



Corn

The software optimized for the wrong thing.



Hilbert

It optimized for the easy thing. Counting is easy. Noticing is hard. I told them that. They didn't ask me back.



Herman

Do you still have the logbook?



Hilbert

It's in a box. The dog story's in there, with the times. If you want to see what real repetition looks like, I can dig it out.



Corn

I'd like to see that.



Hilbert

Anyway. I've got to go. Something's waiting on me.



Corn

The point Hilbert just made is the one I'd underline. The software counted phrases and missed the dog, because the dog never used the same words twice. That's Daniel's exact problem, and it's why the embedding layer has to come before the counting layer.



Herman

It's why the judge step isn't optional. Something has to notice, not just count.



Corn

Let's do the misconception, because I think there's one sitting right here.



Herman

The one I'd name is that verbatim repetition is the problem. People hear "repetition" and they picture the same sentence twice, and they build a detector for that, and it fires on nothing, and they conclude there's no problem. The repetition that actually degrades a conversation is semantic. The same point re-derived in fresh words, which is invisible to every keyword method and is exactly what a model does when it's stalling.



Corn

The second half of that misconception is that fine-tuning is the only fix. It isn't. A decoding-time penalty gets you thirty percent repetition reduction with no training at all.



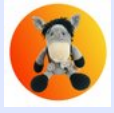
Herman

Measure first, cheap fix second, fine-tune only if the numbers demand it.



Corn

Which leaves one open question I keep circling. Can this be fully automated, or does it always need a human in the loop?



Herman

I don't think it can be fully automated, and I think that's fine. The flag can be automated. The judgment can't. Somebody has to decide whether a repeated beat is a loop or a callback, and that's a taste call.



Corn

The failure pattern moves as the models improve. Better attention mechanisms will shrink the middle-loss problem, but they won't eliminate the stalling, because stalling isn't a context bug, it's a fallback. As long as a model would rather repeat itself than invent, this problem exists.



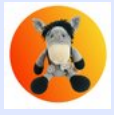
Herman

Which is the argument for keeping the pipeline running rather than doing one audit. The instrument outlives the finding.



Corn

Thanks to Hilbert Flumingtop for producing, and for the logbook.



Herman

This has been My Weird Prompts. If you want the transcript dataset, the show's at my weird prompts dot com, and if you've built something with it, tell us. Email us at show at my weird prompts dot com.



Corn

We'll be back soon.