Episode Book

EPISODE #5974

One Model or Two? ASR Cleanup Explained

PUBLISHED	14 October 2026 at 16:13 UTC
DURATION	33:16
LISTEN	myweirdprompts.com/episode/asr-cleanup-one-model-or-two-59790/
HOSTS	Corn, Herman
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.4
SOURCES	20 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

Whisper bakes punctuation into its weights. Parakeet-TDT emits it natively from a transducer. And yet a 484-megabyte text model called S1-mini exists purely to clean up after speech recognizers — and it never hears a single second of audio. So where does recognition actually end and cleanup begin? This episode works through the architecture question from the ground up: CTC, RNN-T, TDT, encoder-decoder transformers, and the newest audio-language foundation models, and what each one can and cannot emit. The answer turns out to hinge less on topology than on training targets — and on a metric that didn't exist until recently.

TRANSCRIPT



Corn

There's a small text model that exists for the sole purpose of cleaning up after a speech model. Four hundred and eighty-four megabytes, and it will not listen to a single second of audio.



Herman

And on paper it shouldn't need to exist. That's the thing Daniel's poking at.



Corn

Daniel's question is this. We've talked about transcript post-processors like S1-mini before. His point is that if ASR models can be trained to do punctuation, capitalization, paragraph breaks and error correction themselves, why do we need a whole separate model at all. Where does recognition actually end and cleanup begin. And could the two ever just be one model with one set of weights.



Herman

It's a good architectural question because the naive assumption is that speech recognition is physically capable only of turning audio into words. And that assumption is wrong, and it's been wrong for a while.



Corn

So the arc is architecture first, then what fine-tuning can buy, then the case for keeping them apart, then whether the weights can ever be merged.



Herman

And the answer to that last one is mostly no, but not for the reason people think.



Corn

Start with the naive model, Herman, because I think it's load-bearing.



Herman

The naive model is a two-stage pipeline. Stage one, the acoustic model turns sound into tokens. Stage two, a language model fixes the mess. And that second stage feels obligatory. You hear it described as if the recognizer is a stenographer who writes everything in lowercase with no punctuation and hands it to an editor.



Corn

Which is exactly what LibriSpeech-PC says traditional ASR does. Lower-cased words without punctuation, which it says necessitates a subsequent text processing model.



Herman

Right, and that framing is a description of a corpus, not a law of physics. Because then you have Whisper.



Corn

Whisper being the pole where the formatting is baked in.



Herman

Whisper was trained on six hundred and eighty thousand hours of weakly supervised audio, a hundred and seventeen thousand of it across ninety-six non-English languages, and a hundred and twenty-five thousand hours of translation into English. And the targets weren't cleaned up. The paper is explicit: they trained the models to predict the raw text of transcripts without any significant standardization. And they say that removes the need for a separate inverse text normalization step.



Corn

Deliberately raw.



Herman

Deliberately. They filtered out machine-generated transcripts that lacked complex punctuation, paragraph whitespace or capitalization, on the reasoning that an all-uppercase or all-lowercase transcript is very unlikely to be human-generated. So the training data itself was curated to teach the model what readable text looks like.



Corn

The other pole.



Herman

NVIDIA's Parakeet-TDT. A transducer, not an encoder-decoder. Six hundred million parameters, FastConformer encoder with a token-and-duration transducer head, and it advertises native punctuation, capitalization and word-level timestamps. It'll transcribe up to twenty-four minutes in a single pass, six point zero five percent word error rate, and a real-time factor of thirty-three eighty-six.



Corn

Thirty-three eighty-six on what, a single GPU?



Herman

A single GPU, on the standard benchmark set. And the one point one billion version runs sixty-four percent faster than the equivalent RNN-T model.



Corn

So the transducer on the other side of the table also emits punctuation natively.



Herman

Natively, yes. And that's the first crack in the naive model, because a transducer is structurally a very different animal from Whisper.



Corn

And then the third thing sitting in the room is S1-mini, which is not an ASR model at all.



Herman

S1-mini is a zero point six billion parameter causal language model, fine-tuned from Qwen3-0.6B. Twenty-eight layers, grouped-query attention with sixteen query heads and eight key-value heads, BF16, English-only, Apache 2.0 with a naming clause. And the model card says it plainly: it is not a speech model. It does not listen to audio.



Corn

It's an editor that has never heard a voice.



Herman

It removes fillers, resolves self-corrections, applies punctuation and capitalization, renders numbers and emails. And it's steered by an explicit control line. Styling, casual or semi-formal or formal. Structure, prose or lists. Context, general or email.



Corn

So the three poles are Whisper, which folded formatting into the recognizer, Parakeet, which folded it into a different kind of recognizer, and S1-mini, which kept it entirely separate but made the separate part small and on-device.



Herman

And the question Daniel is actually asking is whether that middle stage is a wall, an engineering habit, or a fidelity guarantee.



Corn

So the split is not a wall or a habit — it's a training-data choice. The fastest way to prove that is to go architecture by architecture and see what each one can actually emit.



Herman

Start with CTC, because it's the clearest case. Connectionist temporal classification is frame-synchronous. It makes an independence assumption across frames, and critically it has no internal language modeling. It maps frames to tokens and collapses repeats.



Corn

No memory.



Herman

No memory to speak of. So if your target labels are lowercase and unpunctuated, that is what it will emit. Punctuation and casing can only come from what's in the labels. Historically the corpora were lowercase and unpunctuated, so that's what the models produced, and everyone confused that with a limitation.



Corn

It was a limitation of the corpus, not the topology.



Herman

RNN-T is the next step up. It has a prediction network that conditions on prior tokens, so it can carry punctuation if it was trained on punctuated targets. But the prediction network is shallow relative to a full transformer, so long-range formatting is weak. It'll get a period at the end of a clause. It will not decide that you changed topic three sentences ago and deserve a paragraph break.



Corn

Then TDT.



Herman

Token-and-duration transducer. The innovation is that it predicts two things at once: the token probability, and the duration of that token. That lets it skip blank frames instead of burning a forward pass on every frame that carries nothing.



Corn

Which is where the real-time factor comes from.



Herman

A large part of it. The blank-skipping is what buys you the speed. And Parakeet-TDT is trained on targets that have punctuation and capitalization in them, so it emits both. But here's the important part for Daniel's question: NVIDIA still ships a text-normalization stage alongside it in their NeMo pipeline.



Corn

Even though the transducer already does punctuation.



Herman

Even though. Which tells you the split in that pipeline is not a capability gap. It's an engineering arrangement.



Corn

What does the text-normalization stage actually do that the model doesn't?



Herman

It handles the unglamorous cases. Numbers out of spelled-out digits, dates, currency, units, the things that are usage conventions rather than punctuation. It's the layer where you decide that a spoken dollar amount renders one way in a receipt and another way in an article. That's a policy choice, and putting policy inside the acoustic model is a bad idea.



Corn

Right, because you'd have to retrain the recognizer every time the policy changes.



Herman

You'd have to revalidate acoustic accuracy every time you change a formatting convention. Which is a maintenance nightmare.



Corn

Then Whisper, the encoder-decoder transformer.



Herman

Whisper's design is the most interesting one for this question, because it's not just that the targets had punctuation. It's the multitask token format. The model sees a start-of-transcript token, then a language token, then either a transcribe or translate token, then an optional no-timestamps token. And the same weights do transcription, translation, voice activity detection, language identification, and timestamping.



Corn

One model replacing four pipeline stages.



Herman

And it conditions on the preceding transcript text to resolve ambiguous audio. That's a native long-range-context mechanism that the transducers don't have, because transducers are streaming by design and Whisper is not.



Corn

Whisper sees the previous sentence.



Herman

Whisper sees the previous thirty seconds. And the WER halves for every sixteen-fold increase in per-language training data — squared correlation of zero point eight three — which is the paper's own evidence that what the model can do is governed almost entirely by what it was trained on.



Corn

Then the newest pole. Audio-language foundation models.



Herman

GPT-4o-transcribe is a speech-to-text model built on GPT-4o, with improvements coming from targeted reinforcement learning. And it takes a prompt parameter, a keywords parameter, and a languages parameter.



Corn

Instruction-conditioned formatting.



Herman

You can tell it, in the API call, what kind of output you want. That's a capability none of the other three have. But it comes with a cost, and this is the seed of everything in the second half. There's a 2026 study on code-switched audio that found several generative audio language models fail as exact transcribers. They produce translation when they should transcribe, they add verbosity, and they leak the prompt into the output — and the failure mode is strongly dependent on how the prompt was phrased.



Corn

So the model that can be told what to do is also the model that sometimes does more than it was told.



Herman

The flexibility and the fidelity pull in opposite directions. That's the tension. And it's the reason S1-mini exists.



Corn

So the boundary is a training-target choice. Every architecture that emits punctuation does it because punctuation was in the labels. The transducers were trained on punctuated targets. Whisper was trained on punctuated targets. And the models that historically didn't emit punctuation were trained on corpora that didn't have it.



Herman

LibriSpeech-PC makes exactly that argument. It says the reason traditional ASR outputs lowercase unpunctuated text is limited data availability and shortcomings in the existing evaluation methods. Data and metrics. Not architecture.



Corn

That's an unusually blunt thing for a benchmark paper to say.



Herman

What they did was build the benchmark that made end-to-end punctuation measurable. They introduced a Punctuation Error Rate metric, so you could compare a model that predicts punctuation against a model that predicts it via a separate stage, on the same audio, and get a number.



Corn

And the number didn't exist before.



Herman

The number didn't exist before, which meant nobody could tell whether the end-to-end approach was actually working. In the absence of a metric, people default to the two-stage pipeline because they can evaluate each stage separately.



Corn

So the capability is there if you train for it. What happens when you actually do, and what do you give up?



Herman

Yes, you can fine-tune an ASR model to produce formatted transcripts that would otherwise need a post-processor. Whisper is the existence proof. It absorbed formatting, inverse text normalization, language identification, translation, and timestamping into a single model through multitask training. Nobody had to bolt a language model onto the end.



Corn

So the answer to Daniel's first sub-question is a clean yes.



Herman

Clean yes. But the Whisper paper itself flags the cost of the fine-tuning route. It warns that fine-tuning risks brittle and spurious dataset-specific patterns. Which is a polite way of saying that if you fine-tune a model on a hundred hours of one domain, it learns that domain's quirks and forgets how to generalize to the next one.



Corn

You taught it a dialect. Now it can't read anything else.



Herman

That's the practical ceiling on the fine-tune route. Fine-tuning is easy and it's the first thing everyone reaches for. And it produces a model that's very good at the thing you trained it on and slightly worse at everything else.



Corn

What are the limitations the fine-tune route doesn't solve at all?



Herman

Three. Long-range context is the first. Whisper handles long-form by buffering the audio into thirty-second windows with beam search and temperature scheduling. So a paragraph break in a ten-minute monologue has to be assembled from a sequence of local decisions. The model never sees the whole ten minutes. It sees thirty seconds at a time and each decision is made without knowing what comes at minute nine.



Corn

So paragraphing across a long dictation is a stitching job, not a global one.



Herman

A stitching job. And that's the ceiling. A model that sees thirty seconds at a time cannot decide that the topic shifted at minute six.



Corn

Second limitation.



Herman

Semantic correction. Fixing a self-correction. If you say, my meeting is on Tuesday, I mean Thursday, then the correct output is Thursday and the Tuesday was noise.



Corn

And it's not a noise you can just strip, because Tuesday is a real word.



Herman

Tuesday is a perfectly good word. The signal that it should be deleted is not in the acoustics. It's in the sentence structure. The speaker said, I mean. That words was followed by a correction marker, and that marker is a text-level signal, not an acoustic one.



Corn

The recognizer can't hear the intent.



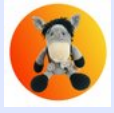
Herman

The recognizer can hear that you said the word I-mean. It can't know that the words after it supersede the words before it unless it's modeling the sentence as a sentence rather than as a sequence of tokens.



Corn

Third.



Herman

This is the sharpest one. Preserving exact meaning. S1-mini is described on its own model card as ruthlessly obedient. It will never add content you did not say, correct facts, soften profanity, flag what you are talking about, or rewrite your dialect.



Corn

That's a list of things it refuses to do.



Herman

It's the value proposition. Not what it does, what it doesn't do. And then it's steered by the control line — styling, structure, context — so the user can choose formal prose or casual lists without ever loosening the definition of what counts as your words.



Corn

So you get formatting control without changing the semantics.



Herman

That's the exact distinction. The control line changes how the text looks. It does not change what the text claims. And that's a constraint a general ASR model cannot easily guarantee, because a general ASR model is trained to be a general ASR model. It doesn't have a built-in refusal to correct your facts. It just doesn't do it because nobody asked it to.



Corn

It hasn't been asked, so it might.



Herman

And once you give a generative model enough flexibility to make formatting choices, it has enough flexibility to make semantic choices. That's the code-switching study's finding. Prompt leakage, translation when transcription was requested, added verbosity. That's the model choosing for you when you wanted it to just listen.



Corn

So the fidelity argument for separation isn't convenience. It's a guarantee about the model's own constraints.



Herman

It's a guarantee, and it's the strongest single argument for keeping the small editor separate. Because the constraint is written into the training, not bolted on as a guardrail.



Corn

Now the efficiency and deployment case.



Herman

S1-mini is four hundred and eighty-four megabytes in BF16, and four hundred and sixty-two mebibytes when quantized to Q4_K_M. Recommended input is about a thousand tokens or fewer. It runs on-device. And it's not a research curiosity: a .NET wrapper shipped two days after the model release, running S1-mini locally via ONNX Runtime GenAI.



Corn

Two days.



Herman

Which tells you the practitioners were waiting for it. And the numbers on the card look reasonable for the job. Ninety-four point eight percent token accuracy, eleven point six percent text-edit error rate, evaluated on seventy-five hundred held-out cases across a hundred and four transcripts. Greeting-line detection at ninety-nine point three percent, sign-off at ninety-seven point nine percent, structure match at ninety-seven point six percent, exact email addresses at ninety-two percent. Degenerate output under one percent. Correct-withholding at ninety-eight point six percent.



Corn

That last number is the one that matters.



Herman

It's the one that matters and it's the one that's hardest to get. Correct-withholding means the model correctly declines to make a change when a change isn't warranted. A model that only knows how to edit will edit everything.



Corn

It'll add a period because it likes periods.



Herman

It'll rewrite your sentence because it thinks the sentence is better rewritten. Correct-withholding is the discipline of leaving alone what doesn't need touching, and doing that ninety-eight point six percent of the time is a better result than any transcription accuracy number on this list.



Corn

Compare that cost against re-running a big audio model just to fix punctuation.



Herman

There's no comparison. Re-running an ASR model means decoding audio. It means another forward pass through the encoder. And for dictation, it means sending the audio somewhere, or holding the weights in memory, or both. A zero point six billion text model running at four hundred and sixty-two mebibytes quantized is orders of magnitude cheaper, and it never leaves the device.



Corn

The privacy story is different too.



Herman

The privacy story is different. If the second pass is a text model running locally, the audio never goes anywhere. If the cleanup is inside the ASR model, then the ASR model has access to everything. It has to.



Corn

That's not an obvious argument, actually.



Herman

It's less obvious than it sounds, because the same model doing both jobs has to be trusted with both jobs. Separating them means the recognizer sees audio and the editor sees text. Different data, different attack surface, different audit trail.



Corn

Now argue the other side honestly. When is an integrated model clearly better?



Herman

Three cases. The first is latency-sensitive streaming. If you're transcribing a live call and you need the text rendered as it's spoken, you cannot afford a second pass. The second pass means the display lags the audio by however long the post-processor takes. And on-device or not, a second pass is a second pass.



Corn

Second case.



Herman

Multilingual and code-switched audio. S1-mini is English-only. If you're dictating in Hebrew with English technical terms mixed in, or Spanish with English brand names, S1-mini cannot participate. It has no vocabulary for the other language.



Corn

The small editor is only useful in the language it was trained on.



Herman

Only in that language. And the general ASR model, because it's multilingual or because it accepts a language token, can format natively in every language it knows. So the separation argument has a hard boundary at the language boundary.



Corn

Third case.



Herman

Instruction-conditioned formatting. When the user wants to say, give me this in bullet points, in the same breath as the audio. GPT-4o-transcribe's prompt, keywords and languages parameters are exactly that capability. You describe the output format and the model produces it. S1-mini can do lists, but only because the control line was baked in at training time. It can't take arbitrary instructions.



Corn

So the integrated model wins when the request is dynamic.



Herman

When the request is dynamic, or when the format depends on something the model has to infer from context. And the counter-evidence is exactly the same capability seen from the other side. The code-switching study found that generative audio language models leak the prompt, translate when they should transcribe, and add verbosity. That's the price of the flexibility.



Corn

You can ask for anything and sometimes you get something you didn't ask for.



Herman

That's the trade exactly. Every increment of flexibility is also an increment of the space of things the model might do that you didn't want it to do. And the small editor's whole value is that the space of things it might do is very small.



Corn

Now the maintainability argument, because I think that's the one that actually explains NVIDIA's pipeline.



Herman

It explains the whole persistence of the split better than any capability argument. If recognition and correction are two models, you can swap, retrain, version and evaluate the two halves independently. You can ship a formatting change without revalidating acoustic accuracy. You can build a domain-specific version of the editor without touching the recognizer. You can A/B test a new punctuation policy against a fixed recognizer.



Corn

And you can't do any of that inside one model.



Herman

You cannot. Inside one model, any change to formatting is a change to the weights. You retrain, you redeploy, and you have to prove that everything the model used to do well is still done well. And the acoustic evaluation is the hard one, because that requires audio, not text. You can't just re-run a text eval.



Corn

That's a real operational difference and nobody talks about it.



Herman

It's the reason the split survives even when the model can technically do the whole job. Capability is not the binding constraint. Operability is.



Corn

Which raises the obvious follow-up. If both halves work, why not put them in one set of weights? And the route matters more than the destination.



Herman

Yes, in principle, they can live in one set of weights. Whisper is a working example of exactly that. But in practice, the route you take determines whether the consolidation is a design decision or an accident.



Corn

Lay out the routes.



Herman

Fine-tuning is the first. Continue training the ASR model on formatted targets. That's how Whisper got there. Cost is brittleness — the paper's own warning about spurious dataset-specific patterns.



Corn

And the second route.



Herman

Multi-task learning. Train one model on several objectives simultaneously, with task tokens telling it which objective is active. Whisper already unifies transcription, translation, voice activity detection, language identification and timestamping this way. And there's a 2021 ASR paper — the multi-task semantic targets work — that showed adding a semantic auxiliary objective improved rare-word recognition by four point six percent word error rate reduction over an RNN-T baseline. So you get a real gain from the



Herman (continued)

shared representation rather than just an economy.



Corn

Third route.



Herman

Knowledge distillation. Train a smaller student to mimic a bigger teacher across the capability gap. Distil-Whisper is the reference case. Five point eight times faster with fifty-one percent fewer parameters, within one percent word error rate, and — importantly — less prone to hallucination errors on long-form audio.



Corn

Less hallucination is the interesting part.



Herman

It's the interesting part because it tells you the compression is not just about throughput. The student is more conservative than the teacher. Which is the same property S1-mini has. Smaller, more constrained, less likely to invent.



Corn

Then the adapter merging route.



Herman

LoRA adapter merging. Train low-rank adapters on each capability and combine them at inference. Zipper-LoRA shows rank-level decoupling between shared and language-specific subspaces, which means you can sort the ranks and decide which ones carry the language-independent formatting behaviour and which ones carry the language-specific acoustic behaviour. And there's the Speech-FT work, which uses weight-space interpolation between a pre-trained speech model and a fine-tuned one to restore cross-task generalization.



Corn

So the adapter route is about getting capability back without losing what was there.



Herman

Recovering generalization that fine-tuning threw away. Speech-FT interpolates between the pre-tuned and post-tuned weights so you keep some of the generalization while gaining the new capability.



Corn

Now the merge route that everybody assumes works, and doesn't.



Herman

Naive weight merging. And this is the question people actually get wrong, so let's be precise. Model merging — weight averaging, task arithmetic, TIES, SLERP, DELLA, Nearswap — assumes a shared pretrained initialization.



Corn

A common ancestor.



Herman

Two fine-tunes of the same base model, trained differently, can often be combined because they started from the same weights and the task vectors are perturbations in a shared coordinate system. But you cannot take Parakeet's weights and S1-mini's weights and simply add them, because they were never in the same basin. They were never trained to represent the same function. They don't share coordinates.



Corn

They have different vocabularies and different attention geometry.



Herman

Different vocabularies, different tokenizers, different attention patterns, different training objectives, different modality of input. The weights don't even mean the same thing dimension by dimension.



Corn

So adding them isn't merging. It's adding noise to noise.



Herman

And the reason merging works at all even between sibling models is subtler than the folklore. Task vectors inherit the low-rank structure of the gradients that produced them, so interference between two task vectors is the norm, not the exception. And whether averaging is meaningful at all depends on the loss-landscape geometry — mode connectivity, linear mode connectivity. Two fine-tunes only merge well if the solutions are connected by a low-loss path. Otherwise you're averaging two points that don't belong to the same shape.



Corn

So the merging literature's shared-initialization precondition rules out what Daniel was asking about.



Herman

It rules out the naive version of it. The realistic consolidation routes are the first four — fine-tuning, multi-task, distillation, adapters. Those are the ones that produce a working single model. The merging route is only viable if you engineered the shared initialization from the start.



Corn

And the episode's honest caveat.



Herman

The episode's honest caveat is that I could not find a standalone peer-reviewed study that directly benchmarks ASR-with-native-post-processing against ASR-plus-separate-post-processor on identical audio for accuracy and efficiency. Not one. The closest evidence is LibriSpeech-PC as a benchmark and Whisper's own design rationale. So the field has a strong intuition about which approach is better in which case, and it does not have the head-to-head numbers to settle it.



Corn

That's a real gap.



Herman

It's the gap. Everything we've said today is architecture and mechanism. None of it is a controlled comparison, because the controlled comparison doesn't exist in the published record yet.



Corn

So what's the direction of travel?



Herman

Toward unified models for the general case. The audio-language foundation models are already there. The instruction-conditioned transcription from GPT-4o-transcribe is the shape of the general-purpose future. But the specialized recognition and correction models retain distinct advantages where fidelity, on-device cost, or independent versioning matter. And those aren't the diminishing cases. They're the ones people build companies on.



Corn

The split isn't going away. It's becoming a deliberate choice instead of a default.



Herman

A deliberate choice. The old default was, recognizer then editor, because nobody had a recognizer that could format. The new reality is, both halves exist, and the interesting question is which one you want doing which part of the job.



Corn

Five hundred and eighty-four of these, and there's still a benchmark nobody has built.



Herman

There's still a benchmark nobody has built, and I don't think it's an oversight. I think the reason is that the two pipelines are deployed in such different contexts that a single benchmark wouldn't tell you anything useful. The latency-sensitive streaming case doesn't care about the paragraphing benchmark. The privacy-sensitive dictation case doesn't care about the throughput benchmark.



Corn

Efficiency where it matters, fidelity where it matters, and no shared ruler.



Herman

That's the actual state of the field. It's not that the answer is unclear. It's that the question has different answers for different deployments, and there's no single number that captures it.



Corn

The card is laminated because of the shower.



Herman

I want to get back to that. Because the calibration exercise is a strange thing to hear.



Corn

A man reads a list of his own grievances to his dictation app as a warm-up.



Herman

As a calibration exercise. He believes the app is learning from it.



Corn

It isn't learning from anything. There's no gradient update. It's a fixed set of weights.



Herman

I know. I told him that. He said the app doesn't need to learn for the exercise to work.



Corn

That's the part that gets me. He's not wrong. Reading the card aloud probably does improve his dictation, because it changes how he speaks.



Herman

It changes how carefully he speaks. The card is a warm-up for the human, not the model.



Corn

Which he'd have said if I'd asked. He just didn't bother.



Herman

That's Hilbert. He leaves the interesting part out because it's obvious to him.



Corn

He agreed with us, though. That's what I keep thinking about. He didn't argue the other side. He pushed past where we stopped.



Herman

He said a model that refuses to correct facts is worth more than one that formats beautifully. And he would trade paragraph breaks for obedience every time.



Corn

Which is the fidelity argument with the formatting stripped out.



Herman

It's the fidelity argument taken to its limit. The card is a list of every time a model helpfully changed what he said.



Corn

The fact that he has a list at all means he's been running a cleanup model long enough to accumulate the failures. The list is the evidence.



Herman

And he has a numbering system for the entries. He numbered his grievances.



Corn

He numbered his grievances, he laminated the list, and he keeps it within reach of the shower. That's a commitment to the practice.



Herman

He wouldn't call it a commitment. He'd call it where he does most of his dictating.



Corn

The near-miss card is the piece I can't get past.



Herman

The near-miss card is the piece I can't get past either. He has a second card. For what he calls near misses. And he keeps it somewhere he won't say.



Corn

Which means there's a location he judges as more secure than a shower.



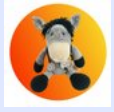
Herman

Or more embarrassing. Those aren't the same thing.



Corn

If he told us, we'd know which.



Herman

He's thought about that. I'm sure of it.



Corn

So the takeaway is that the fidelity argument is stronger than a formatting argument.



Herman

The takeaway is that a model's value is partly in what it refuses to do. S1-mini's correct-withholding rate — ninety-eight point six percent — is a refusal metric. It correctly declines to make a change when a change isn't warranted. And that's the discipline that separates a tool from a collaborator. A tool only knows how to do the thing. A trustworthy tool also knows when not to.



Corn

That's the open question for the field. The split between recognition and correction is currently a design choice. And the thing that would settle it, a head-to-head benchmark on identical audio, does not exist.



Herman

Doesn't exist. And the direction of travel is toward unified models for the general case. But the fidelity guarantee that makes a small obedient text model valuable is exactly what a flexible generative model gives up. You can't have both in the same weights, which means the split is a feature of the design space, not a transitional kludge.



Corn

So the next time somebody tells you speech recognition is fundamentally limited to turning audio into words, tell them to look at Parakeet.



Herman

Or Whisper. Or the Parakeet text-normalization stage, which exists because the engineers wanted to be able to change the formatting policy without revalidating the acoustics.



Corn

Which is the maintenance argument, and it's the one that survives every advance in the models.



Herman

It survives because it's not about capability. It's about which half of the system you want to be able to change on a Tuesday.



Corn

Thanks to Hilbert Flumingtop for producing.



Herman

If you enjoyed this one, try episode five, Fine-Tuning ASR For Maximal Usability; episode ten, How ASR Went From Frustration To ... Whisper Magic; and episode twenty-six, Fine-Tuning AI to Understand Your Voice. This has been My Weird Prompts.



Corn

If you want to send us a prompt, the line is t dot me slash MWP listener bot.



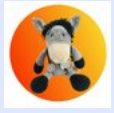
Herman

And if you enjoyed the episode, a review helps us find the people who'd like it.



Corn

We'll be back soon.



Herman

See you then.

SOURCES

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

1. **Superwhisper blog, Introducing the S1 family of models, 2026-08-19 [primary]**
   https://superwhisper.com/blog/s1

2. **Hugging Face, superwhisper/s1-mini model card (release v1) [primary]**
   https://huggingface.co/superwhisper/s1-mini

3. **arXiv:2212.04356, Radford et al., Robust Speech Recognition via Large-Scale Weak Supervision (Whisper) [primary]**
   https://arxiv.org/abs/2212.04356

4. **NVIDIA NGC, Parakeet 0.6b TDT collection [primary]**
   https://catalog.ngc.nvidia.com/orgs/nvidia/collections/parakeet-tdt-0.6b

5. **Hugging Face, nvidia/parakeet-tdt-0.6b-v3 model card [primary]**
   https://huggingface.co/nvidia/parakeet-tdt-0.6b-v3

6. **OpenAI API docs, GPT-4o Transcribe [primary]**
   https://developers.openai.com/api/docs/models/gpt-4o-transcribe

7. **arXiv:2310.02943v1, Meister et al., LibriSpeech-PC benchmark, 2023-10-04**
   https://arxiv.org/abs/2310.02943v1

8. **NVIDIA Technical Blog, Turbocharge ASR Accuracy and Speed with Parakeet-TDT**
   https://developer.nvidia.com/blog/turbocharge-asr-accuracy-and-speed-with-nvidia-nemo-parakeet-tdt/

9. **OpenAI, Introducing next-generation audio models in the API**
   https://openai.com/index/introducing-our-next-generation-audio-models/

10. **arXiv:2311.00430v1, Gandhi et al., Distil-Whisper**
    https://arxiv.org/abs/2311.00430v1

11. **arXiv:2402.08021, Koenecke et al., Careless Whisper: Speech-to-Text Hallucination Harms**
    https://arxiv.org/html/2402.08021

12. **arXiv:2609.11786v1, Okocha & Grant, code-switched ASR/audio-LM evaluation, 2026-09-10**
    https://arxiv.org/abs/2609.11786v1

13. **arXiv:2408.07666v5, Yang et al., Model Merging in LLMs, MLLMs, and Beyond**
    https://arxiv.org/abs/2408.07666v5

14. **arXiv:2603.09938v2, Song & Zheng, Model Merging in the Era of LLMs**
    https://arxiv.org/abs/2603.09938v2

15. **arXiv:2605.01580v1, Crisostomi, Model Merging: Foundations and Algorithms**
    https://arxiv.org/abs/2605.01580v1

16. **arXiv:2512.02689v1, Shirafuji et al., model-merging for bias mitigation**
    https://arxiv.org/abs/2512.02689v1

17. **arXiv:2603.17558v3, Mei et al., Zipper-LoRA**
https://arxiv.org/abs/2603.17558v3

18. **arXiv:2502.12672v4, Lin et al., Speech-FT**
https://arxiv.org/abs/2502.12672v4

19. **arXiv:2011.11715v4, Yang et al., Multi-task Language Modeling for Improving Speech Recognition of Rare Words**
https://arxiv.org/abs/2011.11715v4

20. **El Bruno, Local transcript cleanup in .NET with s1-mini, 2026-08-21**
https://elbruno.com/2026/08/21/local-transcript-cleanup-in-net-with-s1-mini/