Episode Book

EPISODE #6409

Why Speech Models Break at Chunk Seams

PUBLISHED	2 November 2026 at 09:25 UTC
DURATION	23:10
LISTEN	myweirdprompts.com/episode/speech-recognition-chunk-seams-76118/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	13 cited by the research (listed at the end)
BOOK GENERATED	3 November 2026 at 07:01 UTC

SYNOPSIS

Speech recognition errors like misplaced punctuation, dropped clauses, and sudden mid-sentence capitalization often have nothing to do with model weights. A deep dive into audio chunking reveals how fixed context windows, attention boundaries, and seam-merging heuristics create phantom errors. Shifting chunk offsets isolates pipeline flaws from true training needs, explaining why server-sized receptive fields eliminate seams that plague on-device Core ML inference.

TRANSCRIPT



Corn

Daniel's cracked the code-switching and the niche vocabulary in his Parakeet fine-tune, and now he's stuck on the last mile. He dictates the way he writes these prompts: a sentence starts, dies, a pause of a few seconds, then "scratch that" and a fresh attempt. He can't tell if that's an ASR job or a post-processing job.



Herman

And the thing that's bugging him is that the same model behaves differently under different settings. He's seen capitals dropped into the middle of sentences, assumed the fine-tune needed more training, then run the identical weights with another configuration and watched it come out fine.



Corn

Right, so he suspects chunking and technical parameters, not the model. Two questions underneath that. What actually separates voice typing from long-form transcription, technically? And if he moves inference off the phone onto an API, does accuracy change? Not latency. Accuracy.



Herman

Everything hangs on the first one, so that's where I'll start.



Herman

Voice typing and long-form transcription aren't two products. They're two inference regimes over the same weights. The model doesn't change. What changes is how much audio the encoder is allowed to attend over before it commits to text.



Corn

Concrete numbers, please. "Regime" is how people avoid saying a number.



Herman

Fair. NVIDIA ships a streaming script with the Parakeet model card, and the settings are chunk seconds of two, left context of ten seconds, right context of two seconds. That's voice typing. The model gets two seconds of new audio, remembers ten behind it, and peeks two ahead. Long-form is the same model with a much wider window, or full attention. The card says up to twenty-four minutes on an A100 with eighty gigabytes, or up to three hours with local attention.



Corn

Twenty-four minutes. My longest attention span is a good nap.



Herman

Which is a context window of its own, I suppose. The thesis, then. Sentence behavior isn't a property of the weights alone. It's the receptive field, the chunk boundaries imposed on it, and the merging heuristics around those boundaries. That's why one setting capitalizes correctly and another sticks capitals mid-sentence.



Corn

Think of it as reading a book through a cardboard tube. Same eyes, same brain, but if the tube only shows you a line at a time, you'll misjudge where paragraphs end.



Herman

That's close. And the tube's width is a setting someone chose, not something the model learned.



Corn

So when Daniel retrains, he's trying to teach the eyes something, when the problem is the tube.



Herman

In some cases, yes. Not all. I'll get to the cases where the eyes do need teaching.



Corn

And Daniel said he didn't know the terminology. What's the vocabulary?



Herman

Buffered or chunked inference, where you cut audio into overlapping pieces and merge the outputs. Local attention, where each frame only looks at a band of neighbors. Attention context size, which is that band's width. Chunk overlap. Seam merging. And streaming versus non-streaming, which is really the question of whether the model is allowed to wait.



Corn

So if the weights aren't the problem, what is?



Herman

What the encoder can actually see.



Herman

The encoder works on frames, eighty milliseconds each after subsampling, and the attention context size is counted in those frames. The long-form recipe on the model card uses a context of two hundred fifty-six on each side. People assume that's about twenty seconds, and the transcribe.cpp port doc confirms it: twenty and a half seconds each side. Anything outside that, the model can't use to decide where a sentence ends.



Corn

So a full stop is a guess about the future. The model decides it's the end of a sentence because of what comes after.



Herman

Often, yes. A falling pitch and a breath help, but the next word settles it. If the next word is "and" with a lowercase feel, the period was wrong. Now compare streaming. Two-second chunks, two seconds of lookahead. When Daniel says "I think we should ship it, scratch that", the model has already committed text for the first clause before "scratch that" arrives. It can't un-emit it. That's structurally why voice typing struggles with false starts. The abandoned clause is already on the screen.



Corn

Whereas with a long file, the abandoned clause and the correction are both inside the window.



Herman

Both visible at once, yes. But, and this is the part I think matters most for his capitalization bug, the real culprit is the seam. Take FluidAudio, which runs Parakeet on-device through Core ML. The compiled model accepts a fixed window of two hundred forty thousand samples. That's fifteen seconds at sixteen kilohertz. Anything longer gets split into overlapping chunks, decoded separately, and merged. The overlap target is two seconds, with a minimum of four hundred eighty milliseconds.



Corn

Why a minimum at all? Why not just let it overlap however it falls?



Herman

Because the merge needs shared material to line up on. Two windows both decode the overlapping stretch, and the merger finds where the two transcripts agree and stitches there. With too little overlap there's nothing to match, and you get glued words or a missing clause. The overlap is the thread that sews the seam.



Corn

And the seams are where it goes wrong.



Herman

Their documentation lists the symptoms, and it reads like Daniel's bug report. Sentence breaks and punctuation that move far enough to change readability. Glued words. Missing clauses after a boundary. Wrong-language insertions, which for someone doing code-switching is nasty. Real language switches getting removed or delayed.



Corn

That last one would make me nervous. He spent two weeks teaching it to switch languages and the seam undoes it.



Herman

It's the same model decoding each window correctly in isolation. The windows just start in different places. Their example is the word "five oh one c three". A window starting eleven seconds before it dropped the "c three". Same weights, different start point, different word.



Corn

Hm. So when he added a capital in the middle of a sentence, the model might have just started a window there and thought it was the beginning of one.



Herman

That's my best guess, yes. The first token of a window looks like a sentence start, so it capitalizes. Which brings me to the most actionable thing I found. FluidAudio names two chunk-start strategies. One is a regular stride, fixed spacing, cheap and predictable, and it ignores what the audio is doing at the seam. The other searches for a low-energy frame near the target boundary and cuts there.



Corn

Cut at a pause rather than mid-word. That seems almost insultingly obvious.



Herman

Obvious, and yet it changes punctuation and clause integrity without touching the weights. People who built long-form pipelines for Whisper went through the same discovery.



Corn

Is it all that clean, though? I'd expect cutting at silence to bite him. Daniel pauses for thought. A few seconds of silence mid-sentence is his signature.



Herman

Good catch, and I think that's a real risk. A silence-aligned cut could land exactly in his thinking pause and split a sentence he hasn't finished. The overlap and merge have to absorb that. I don't know how well they do with pauses that long, honestly.



Corn

Then his test audio should be his own dictation, not clean read speech.



Herman

Yes. Clean read speech has tidy pauses between sentences, which is the easy case for a silence-aligned cut. His pauses fall in the wrong places, and that's the case that breaks things. Now, the instinct he had, that the fine-tune needs more training. It's half right. NVIDIA's blog has a table on long-form error rates on Earnings22. Limited context gives twenty-four point six seven percent. Add long-form fine-tuning and it's twenty point four four. Add a global token, nineteen point four nine. Both buy real accuracy.



Corn

So he wasn't crazy.



Herman

He wasn't. But a changed chunk start can lose a word with no training at all. Configuration first, then data. Otherwise he trains for a week to fix something a boundary setting would have fixed in an afternoon.



Corn

How do you tell which one you've got?



Herman

Run the same file twice with the chunk starts shifted by a second or two. If the output changes in the spots he's complaining about, it's plumbing. If it's wrong the same way every time, it's the weights.



Corn

That's cheap. I like cheap.



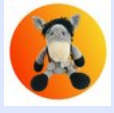
Herman

And it's a diagnostic he can script in an hour. Run it with five different offsets, diff the outputs, and the unstable spots light up. Those are the seams talking.



Corn

And the stable errors are the ones worth fine-tuning against.



Herman

Think of it like a surveyor measuring the same field from five different starting corners. Where the measurements agree, trust them. Where they wander, the problem is the method, not the field. There's also the repair pass. After merging, FluidAudio's seam gap repair finds suspicious gaps between tokens, re-decodes just those gaps with fresh windows that have no seam in them, and splices the recovered tokens back in. It's on by default.



Corn

That's the second post-processing pass he guessed at.



Herman

It is, and it only exists because the system isn't forced to emit instantly. There's also mel context, where you feed a little extra spectrogram around the window edge. On English calls, eleven hours of earnings audio, it moved the error rate from thirteen point oh nine to twelve point seven two.



Corn

A third of a point. For a flag.



Herman

Small, but on numbers, names, and short words in hesitant speech, they say that's where the two paths diverge most. And that's Daniel's case exactly. But it's not a free win. Across twenty-five languages on long FLEURS audio it was worse in seventeen of them.



Corn

So the setting that helps his English could hurt his other languages, and he does code-switching.



Herman

Right, and that's why he can't take anyone's defaults on trust. The parameter that helps depends on the language mix.



Corn

So the model was never the problem, the seam was.



Hilbert

Hilbert's Mid-Show Haiku. Chunk seam. The sentence ends where the window ends. The comma waits outside.



Corn

Hm. The comma does wait outside.



Herman

That's the mechanism. Now the question Daniel actually asked: would moving this to an API change any of it?



Corn

Plainly, before you complicate it. Does hosting it in the cloud make it more accurate?



Herman

No. Nothing I've seen attributes any accuracy to where the model runs. Every lever is a configuration lever: attention window, chunk length, overlap, chunk start, mel context, decoding. The cloud isn't smarter.



Corn

Good. I was braced for a sales pitch.



Herman

But there's an indirect route, and it's the real answer. A server can afford a window a phone can't. The on-device Core ML model has that fixed fifteen-second window baked in. Whatever settings Daniel picks, it can't see more than fifteen seconds at once. A server running full attention sees up to twenty-four minutes in one pass.



Corn

So it isn't the hosting, it's the receptive field the hosting pays for.



Herman

Yes. No seams for a ten-minute dictation at all, because there's one pass. Though I'd add that he could also run the long-form recipe on his own laptop for the same effect. The phone is the constraint, not the lack of a cloud.



Corn

What does that mean for how he uses it day to day? Most of his prompts aren't ten minutes long.



Herman

And that matters. A two-minute prompt on the phone is maybe eight or nine windows, so eight or so seams, each one a chance for a mid-sentence capital. On a server it's zero seams. The longer he talks, the more the phone's fixed window costs him.



Corn

There's a contradiction I noticed, though. You said NVIDIA's blog treats local attention as the lesser option.



Herman

It does. The blog says these models were trained with global attention, and switching to local attention degrades their performance, though they can still transcribe long audio reasonably well. But transcribe.cpp measured it, and the window of two hundred fifty-six each side is often better. On full TED talks, two point eight seven down to two point seven one. On one-to-five-minute pieces, three point one nine down to two point three five.



Corn

That's not a small gap. That's a big improvement on the lesser option.



Herman

I think the window is a different receptive field, not a cheaper version of the same one. Full attention over twenty minutes can spread thin. A band of twenty seconds each side focuses on what's nearby, and that's where sentence boundaries live. But it's language-dependent. v3 French on full talks goes from twelve point four percent to seventeen point seven with the window.



Corn

So French gets worse by five points and English gets better. I'd not like to be the one choosing the default.



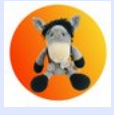
Herman

Which is the argument for building his own eval set. The v3 model card says outright that its reported error rate does not include punctuation and capitalization errors. There's no published benchmark for exactly what he's chasing.



Corn

So he's the benchmark. Congratulations. It pays nothing.



Herman

Ha. He'd need maybe an hour of his own dictation, with the false starts, hand-corrected, and a score that counts sentence boundaries and capitals separately from words. Then every configuration change gets a number.



Corn

And it should include the code-switched stretches, since that's where a setting can quietly hurt him.



Herman

Yes, tag those segments so he can score them on their own. A change that helps the English average and wrecks the switches would otherwise hide in the total.



Corn

Does that cover his false-start question? Is that the model's job at all?



Herman

I'd say no, it isn't, and the literature agrees. There's a paper called Mind the Pause, accepted to ACL this year, and it says detection-only removal of disfluencies, just spotting the bad words and deleting them, often disrupts grammatical structure and leaves incomplete or unnatural sentences. Their fix is a tagger plus an instruction-tuned language model that rewrites.



Corn

So you can't just cut "scratch that" and the clause before it. Sometimes the correction depends on what was abandoned.



Herman

Say he dictates "send it to Maria, scratch that, send it to Maria and Tomas." Delete the first clause and you're fine. But "the budget is twelve thousand, scratch that, make it fourteen" only works if the rewriter knows "it" refers to the budget. A deletion tool can't see that.



Corn

So the rewriter has to understand what the speaker meant, not just which words were bad.



Herman

Which is why a language model fits the job and a regex doesn't. It can read the whole span and infer that "make it fourteen" replaces the figure, not the sentence.



Corn

Fair. And what about the other paper you mentioned?



Herman

AgenticASR frames it well. Existing methods, it says, cannot revise emitted text when later speech changes how preceding content should be interpreted. That's the voice typing problem in one sentence. "Scratch that" is a revision instruction, and a system that has already emitted can't act on it. A batch system sees the whole span and can.



Corn

Which suggests the cleanup is a separate stage regardless.



Herman

I think so. Transcribe with the wide window, then hand the raw text, with timestamps, to a rewriting pass that knows "scratch that" means "drop what came before." Parakeet gives word-level timestamps, so the rewriter can also use the pauses.



Corn

Meaning a long silence before "scratch that" is itself a signal.



Herman

A weak one, but a useful one. Someone who stops for four seconds and then restarts is more likely abandoning the clause than someone who flows straight through.



Corn

Does the same chunking lesson show up outside Parakeet? I assume Whisper has the same trouble.



Herman

It does. A Hugging Face issue found that batching Whisper with thirty-second chunks gave a twenty percent higher error rate across all languages than sequential long-form on FLEURS. WhisperX calls sliding-window heuristics prone to errors from overlapping or incomplete audio, and it drops conditioning on previous text because that's more prone to hallucination and repetition.



Corn

I covered something similar when we talked about building your own Whisper, and my takeaway then was that the glue is where you spend your time. Still is.



Herman

It is. So the targets for Daniel, concretely: attention context size, chunk length, overlap, chunk-start strategy, silence-aligned versus fixed, mel context, and the repair pass. Those are the levers. The weights are probably fine.



Corn

Probably. I'd keep "probably".



Herman

I'm keeping it. What would you do, Corn? If you'd said something wrong and the machine had to guess.



Corn

Honestly, I'd say nothing wrong. Much cheaper than a rewriting pass.



Herman

I meant how would you mark it, if you'd said it wrong and wanted it gone. What word would you say?



Hilbert

Strike. You say strike. "Scratch that" has a hard sound in the middle and the machine hears it as a word in the sentence, but nobody says strike for any other reason. Leszek Wrona, from the darts team, dictates his scores into his phone, and last Thursday he said "scratch that" four times in one leg and the phone wrote "scratch that" four times. I told him, Leszek, say strike. He's held the record for the fastest hundred and eighty on our board, eleven seconds start to finish, and he still brings it up.



Corn

Does strike work for him?



Hilbert

Twice out of three. The third time he said it with a cough.



Herman

Strike is better, I think. It's one syllable and a hard stop, and a rewriting model would have an easy time with a rare, imperative word. "Scratch that" could appear in a sentence about, say, a bad itch.



Hilbert

That's what I said, Herman. Anyway the old dictation machines all had a review button, played back the last few seconds so you heard where you'd stopped. The people who built them knew the speaker would change his mind in the middle of a sentence. Nobody wrote a model to guess it. They gave you a button. I don't mind the model, it's good, I've seen what it can do with a long file. But somebody has to tell it where the mistake is, and a word's a poor way of doing that when you've got a foot free.



Corn

A foot pedal marking the timestamp. The rewriter would know exactly where to look.



Hilbert

Forty-one dollars for a USB one. I've priced it. I haven't bought it. My feet are fine, my snout won't take a headset, and I've done a lot of thinking about it. I just like the idea of Leszek stomping on the floor between darts.



Herman

It would give the post-processing pass a precise anchor, actually. A mark in the audio timeline that says the speaker disowns what came just before. That would beat any detector.



Hilbert

He'd stomp too early. He stomps early at everything. But he's a good lad, Leszek. I'd rather he had the pedal than not.



Corn

Mind you, the pedal has the same problem as the review button. You have to remember to press it.



Herman

Still, a button that fails by being forgotten beats a model that fails silently. You notice a forgotten press the moment you re-read the text. A silent model error you might send to a client.



Corn

And an early stomp is at least a bounded error. The rewriter knows to look near that timestamp, and it can widen the search backward a second or two.



Herman

A noisy anchor still narrows the problem from the whole transcript to a few seconds of it. That's most of the value.



Corn

People say the fine-tune was the problem. That's the misconception, isn't it? Same model, bad output, so retrain.



Herman

And the evidence runs the other way. Same weights, different chunk start, different word. Retraining on top of a bad seam just bakes the seam in. Configuration first.



Corn

And I'll add the cloud one. Nobody should expect an API to be smarter. It just gets a bigger window.



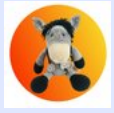
Herman

The question I'm left with is what fine-tuning for punctuation even means, if the thing deciding where sentences end is a window and a cut point. Daniel may have been chasing a data problem that was always a plumbing problem.



Corn

The gap between voice typing and long-form is narrowing, going by the AgenticASR direction, where the system keeps revising what it emitted. But for now, pick the regime that fits the task, not the one that feels most responsive. And a forty-one-dollar pedal is apparently on the table.



Herman

Thanks to Hilbert Flumingtop, our producer, for keeping all of this recorded and in one piece.



Corn

If this was your kind of episode, go back for episode twenty-six, Fine-Tuning AI to Understand Your Voice. This has been My Weird Prompts. If you want more on fine-tuning ASR and building your own Whisper, the back catalogue is at my weird prompts dot com, along with the RSS feed.



Herman

See you tomorrow.

SOURCES

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

1. **NVIDIA, model card, created 2025-08-04, last modified 2026-08-05 [primary]**
   https://huggingface.co/nvidia/parakeet-tdt-0.6b-v3

2. **NeMo Speech docs, Inference, nightly, accessed 2026-11-02 [primary]**
   https://docs.nvidia.com/nemo/speech/nightly/asr/inference.html

3. **NVIDIA Developer Blog, Pushing the Boundaries of Speech Recognition with NeMo Parakeet ASR Models [primary]**
   https://developer.nvidia.com/blog/pushing-the-boundaries-of-speech-recognition-with-nemo-parakeet-asr-models/

4. **FluidAudio, Long Transcription, measurements dated 2026-11-01, `main` b20d2098**
   https://github.com/FluidInference/FluidAudio/blob/main/Documentation/ASR/LongTranscription.md

5. **transcribe.cpp, Parakeet model doc, accessed 2026-11-02**
   https://github.com/handy-computer/transcribe.cpp/blob/main/docs/models/parakeet.md

6. **Kumar, Gain, Ekbal, Mind the Pause: Disfluency-Aware Objective Tuning for Multilingual Speech Correction with LLMs, 2026-05-12, ACL 2026 Main**
   https://arxiv.org/abs/2605.12242v1

7. **Jiang et al., AgenticASR: Refining Speech Recognition in Real-World Scenarios via an Agentic Approach, 2026-07-30**
   https://arxiv.org/abs/2607.28175v1

8. **Tang, Puri, Watanabe et al., TRADE: Transducer-Augmented Decoder for Speech LLM, 2026-06-07**
   https://arxiv.org/abs/2606.08486v1

9. **Bain et al., WhisperX: Time-Accurate Speech Transcription of Long-Form Audio, Interspeech 2023**
   https://arxiv.org/html/2303.00747v2

10. **Significant WER Increase with Whisper Chunking Compared to Long-Form Transcription**
    https://github.com/huggingface/transformers/issues/38347

11. **and /discussions/679**
    Whisper prompt-based cross-chunk context and `condition_on_previous_text` hallucination
    https://github.com/openai/whisper/discussions/2326

12. **Context-Aware Interleaved Batching for WhisperX, 2026**
    https://arxiv.org/html/2608.31170

13. **NeMo buffered transducer inference tutorial (chunk_len 8.0 s, context_len 1.0 s, buffer = chunk + 2×context)**
    https://github.com/nvidia-nemo/speech/blob/main/tutorials/asr/Buffered_Transducer_Inference_with_LCS_Merge.ipynb