Episode Book

EPISODE #6628

Same 100 Hours, 98% vs 6.74% Word Error in ASR

PUBLISHED	16 November 2026 at 14:24 UTC
DURATION	28:49
LISTEN	myweirdprompts.com/episode/asr-from-scratch-word-error-25041/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.33
BOOK GENERATED	16 November 2026 at 22:22 UTC

SYNOPSIS

Training a speech recognition model from scratch on 100 hours of LibriSpeech audio can produce a 98.4% word error rate in an attention encoder-decoder like SoralASR-23M-Preview, or 6.74% in k2-fsa's pruned stateless transducer. The difference comes down to monotonic alignment in transducers and CTC versus unguided attention. We also bracket the real compute costs of from-scratch runs like Zipformer, Moonshine, and OWSM, and examine what happens when you swap tokenizers for English-Hebrew code-switching on NVIDIA's Parakeet.

TRANSCRIPT



Corn

Every from-scratch story on Hugging Face gets told the same way. Lone builder, one GPU, bold experiment, look what's possible now. Then you scroll to the model card and the word error rate is ninety-eight percent. The story was the training run, not the model.



Herman

And this is the week we finally talk about why that happens, because Daniel walked straight into it.



Corn

He did. He found a model called SoralASR-23M-Preview — twenty-three million parameters, trained from randomly initialised weights, no pretrained checkpoint under it. And it got him wondering about the road not taken. He's fine-tuning NVIDIA's Parakeet right now for his own case, English speech with Hebrew words dropped in, and his question is what the from-scratch route would actually look like. The data, the compute, the time, and how the code-switching would work if you owned the whole stack instead of adapting someone else's model.



Herman

And the honest answer has a twist in it, because the interesting question isn't whether you can train from scratch. People do it. Moonshine did it, the OWSM project did it. The interesting question is why the same hundred hours of audio produces a ninety-eight percent word error rate in one architecture and a 6.74 percent word error rate in another. Same data, similar size, wildly different outcomes.



Corn

Start with the model he found, then. What actually is SoralASR?



Herman

It's a deliberately shrunken Whisper. Attention encoder-decoder, five encoder layers, five decoder layers, width of two hundred fifty-six, four heads. Takes eighty-bin log-mel spectrograms over twenty-second windows. Twenty-three million, sixty-two thousand, two hundred seventy-two parameters, and the only thing it borrowed from Whisper is the



Herman (continued)

English tokenizer vocabulary — the weights themselves started random. Apache-2.0 licence, credit where due, the card is honest about what it is.



Corn

Trained on what?



Herman

LibriSpeech train-clean-100. A hundred hours of clean read audiobooks — the beginner slope of ASR datasets. Fifty thousand optimizer updates on a single NVIDIA T4, effective batch of eight utterances, roughly fifteen point seven million decoder tokens processed. The published checkpoint is step twenty-four thousand nine hundred seventy, chosen on a small validation score out of the fifty thousand steps it ran.



Corn

A checkpoint from halfway through the run. That's like publishing the chapter you liked, not the book.



Herman

And the results — ninety-eight point four percent word error rate on the first ninety-six clean validation utterances, and a self-reported leaderboard-style mean of one hundred twenty-three point eighty-six percent across eight English sets. The number is over a hundred because word error rate counts insertions, and this model inserts. The card itself calls it an experimental preview with repetitive decoding, so it's labelled honestly. This isn't a scam. It's an educational artefact that fails in public.



Corn

Ninety-eight percent word error rate. For calibration — a model that output silence for every file would score a hundred.



Herman

It barely beat silence.



Corn

So it's a cautionary tale, not a template. But the part I want to get at is the counterexample, because somebody trained from scratch on the exact same hundred hours and got something usable.



Herman

The icefall project, k2-fsa's toolkit. Their pruned stateless transducer, trained from scratch on LibriSpeech train-clean-100 only — same hundred hours, nothing else — thirty epochs on three GPUs, and it reaches 6.74 percent on test-clean and about eighteen percent on test-other. ESPnet's Conformer recipe on the same data lands around six and a half, seventeen and a half, without a language model. So you've got two models, roughly the same parameter budget, the same audio, and one of them is a toy and the other one is a model you could actually deploy.



Corn

Walk me through why, because this is the mechanism the whole episode hangs on.



Herman

The loss function. A CTC model or a transducer is forced to align audio to text monotonically — the output can only move forward in time as the audio does. The model never gets to decide where a word goes; the loss pins each piece of text to a piece of sound. An attention encoder-decoder like SoralASR has to learn that alignment from nothing. The decoder attends to whatever encoder frames it wants, in whatever order it wants, and with a hundred hours and a batch of eight it never learns the discipline. So it loops. It produces a token, produces it again, and the transcript comes out as repeated phrases and hallucinated filler. That's exactly what a word error rate over a hundred is telling you.



Corn

The architecture removed the guardrails and hoped the data would teach manners.



Herman

That's the shape of it. And there's a memory cost running the other way, to be fair — the NeMo maintainers note transducers converge faster than CTC from scratch but eat more GPU memory. Everything in this trade has a price tag pointing somewhere.



Corn

So scale it up. What does a serious from-scratch run look like at the small end?



Herman

Zipformer, same k2-fsa lineage, on the full nine hundred sixty hours of LibriSpeech. The small one is twenty-three point three million parameters — almost exactly SoralASR's size — and it gets 2.42 percent on test-clean. The medium, at sixty-five million, gets 2.21, and the large, at a hundred forty-eight million, gets 2.06. Fifty epochs, and here's the part that grounds it in reality: roughly seventy to eighty-seven minutes per epoch on two to four thirty-two-gigabyte V100s. That's two and a half to three days of wall-clock, so somewhere around five to ten GPU-days on older cards.



Corn

About a week of GPU time for a working model. That's the entry fee nobody mentions in the model card with the ninety-eight.



Herman

And then the serious end. Moonshine, from Useful Sensors, autumn of twenty twenty-four. They trained from scratch because fine-tuning Whisper wasn't beating Whisper's own word error rate — they tried the adaptation route first and it didn't clear the bar. So Tiny is twenty-seven point one million parameters, Base is sixty-one point five, encoder-decoder with rotary position embeddings and a conv stem on raw audio, and they trained it on roughly two hundred thousand hours of audio, thirty-two H100s, two hundred fifty thousand steps, global batch of one thousand twenty-four. The Tiny model averages 12.66 percent word error rate against Whisper tiny dot en's 12.81 — they matched Whisper's small model with a model built for the edge, from nothing.



Corn

Two hundred thousand hours. SoralASR had a hundred. That's three orders of magnitude.



Herman

Three orders of magnitude is the whole story of the gap. OWSM, the Carnegie Mellon project that reproduces Whisper from scratch on public data only, tells the same story in stages. Version one, thirty-eight thousand hours, two hundred seventy-two million parameters, about a week on thirty-two A100s. Version two, a hundred twenty-nine thousand hours, about ten days on sixty-four A100s. Version three, a hundred eighty thousand hours, eight hundred eighty-nine million parameters, ten days on sixty-four A100s again. Whisper itself, for reference, was six hundred eighty thousand hours.



Corn

Daniel asked what the from-scratch route costs, and I want to answer him straight, because I don't think anyone's published the number he's actually after.



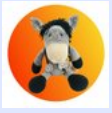
Herman

Nobody has. There's no published GPU-hour or dollar figure for training a roughly hundred-million-parameter model on one to ten thousand hours — that band between SoralASR's toy run and Moonshine's industrial one is empty of receipts. What we can do is bracket it. If a small Zipformer is five to ten GPU-days on V100s for nine hundred sixty hours, and OWSM version three is ten days on sixty-four A100s for a hundred eighty thousand hours, then something in the middle — a transducer, maybe a hundred million parameters, a few thousand hours — is plausibly tens of A100 GPU-days. At typical rental rates that's somewhere from several hundred to a couple of thousand dollars of compute, before a single failed run. But that's my estimate, not a quote.



Corn

Say the estimate part loudly.



Herman

I did say it. It's an estimate from the bracketing runs, and anyone who hands you a precise dollar figure for this is inventing it.



Corn

Which answers the question of how small models like SoralASR "manage" to train from scratch. They manage it the way a man who builds a boat in his basement manages to build a boat. The training runs. The checkpoint exists. The model is not usable. The trade is brutal but clean: one GPU and a hundred hours with an attention encoder-decoder gets you repetition and hallucination; the same hundred hours with a transducer, on a handful of GPUs, gets you a model that works on clean read speech.



Herman

And that last qualifier matters. The icefall result is clean read English audiobooks. That's the easiest problem in the field.



Corn

Right — which takes us to Daniel's actual problem, because his audio is not audiobooks.



Herman

No. English speech with Hebrew words dropped in, which is a code-switching problem, and the fine-tune he's running is on Parakeet. Before we get into the from-scratch version of his case, one piece of context on what he's standing on: Parakeet-TDT-0.6B-v3 was trained on about ten thousand hours of human-transcribed audio plus around six hundred sixty thousand hours of pseudo-labelled data, and its one shared tokenizer covers twenty-five languages. All European. Hebrew is not on the list.



Corn

So the base model was never trained on Hebrew, and its tokenizer was built for European languages.



Herman

That's the hill he's climbing, and to his credit it's a climbable one — but it frames what from-scratch would have to replace. Start with the data side. Hebrew speech that actually exists publicly: ivrit.ai has about thirty-three hundred hours, mostly podcasts, over a thousand speakers, weakly labelled. HebDB out of the Hebrew University lab has about two thousand five hundred raw hours, sixteen hundred ninety after processing, spontaneous speech, also weakly supervised — and their Conformer baselines trained on it match or beat Whisper models up to seven hundred sixty-nine million parameters on



Herman (continued)

FLEURS Hebrew. So the Hebrew mono-lingual raw material is there, in quantity. What does not exist, anywhere we could find, is a public Hebrew-English code-switching corpus.



Corn

At all.



Herman

We looked. Commercial vendors advertise bilingual Hebrew data with no published breakdowns, but nothing open and nothing labelled for code-switching. So a from-scratch Daniel wouldn't just be assembling training data — he'd be creating the category.



Corn

Before the data even, though — the tokenizer. If you build from scratch you build your own vocabulary, and that's where I've seen this go wrong in a very specific, documented way.



Herman

The Malay experiment. Somebody on Hugging Face took Parakeet-TDT-0.6B-v3, called change-vocabulary to swap in a Malay tokenizer, with about fifty hours of audio. The decoder and joint network got reinitialised — because the vocabulary changed, those layers have to start over — and the model collapsed into pure repetition. The transcript was, quote, "dia dia dia". Validation word error rate, a hundred and nine point five percent.



Corn

SoralASR's twin.



Herman

Same symptom, same cause. The encoder still knew sound, but the freshly random decoder had to relearn alignment on fifty hours and it looped. And the author's conclusion is the one I'd underline for anyone at Daniel's scale: under roughly a thousand hours, keep the pretrained tokenizer. There's a separate NeMo GitHub issue with a Persian tokenizer



Herman (continued)

swap on a hundred-ten-million Parakeet that stalled the same way. It's not bad luck — it's the expected outcome.



Corn

So NVIDIA's own recipe for a new language keeps the pretrained encoder and reinitialises only the decoder and joint?



Herman

That's their documented middle path. Keep the acoustic model that six hundred seventy thousand hours built, reinitialise the parts that touch the vocabulary. It's not from-scratch, but for a new language it's the closest practical thing — and notably, it's still closer to what Daniel is doing than to what SoralASR did.



Corn

Let me push the from-scratch case honestly, though, because it isn't nothing. If he built it himself, he'd own the tokenizer — no European-only vocabulary constraint, no dependency on NVIDIA's release cadence, and he could size the architecture for his deployment target instead of inheriting Parakeet's.



Herman

All true. And the honest list runs the other way too. He'd need a code-switching corpus that doesn't exist, or a pipeline to synthesise one. He'd need to solve the tokenizer problem that just ate the Malay attempt. He'd need compute to converge a transducer — and the icefall hundred-hour run is the floor, not the target, and it's clean read English. And he'd be walking away from the six hundred seventy thousand hours of pseudo-labelled data baked into Parakeet's encoder. That encoder has heard an enormous amount of speech. From-scratch means going back to zero on everything it knows about sound.



Corn

Take the corpus problem first, then. Is there any way round it, or is that the wall?



Herman

The synthetic route is the one live option, and it's more real than it sounds. CS-FLEURS came out in the autumn of twenty twenty-five — a hundred thirteen code-switched pairs across fifty-two languages, built by generating speech with TTS and also by concatenating, splicing together audio from different languages. A hundred twenty-eight hours of generated training speech across sixteen language-to-English pairs. And this year there's a paper that fine-tuned CosyVoice2 on the SEAME corpus to generate Mandarin-English code-switched speech, and it largely closed the gap to real data when they trained on it. Synthetic code-switching works. It's just a project in itself — you'd be fine-tuning a speech generator before you trained the recognizer.



Corn

Meanwhile the thing he has to beat is already on the board.



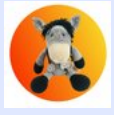
Herman

And it's a concrete number, which I like. Daniel's own earlier Whisper-large-v3-turbo fine-tune, Whisper-Hebrish, on about two hundred fifteen minutes of phone speech — under four hours — cut code-switching word error rate on held-out terms from 14.7 percent to 6.7, and recall of unseen Hebrew terms from 36.7 percent to 53.2. That's the bar. From-scratch doesn't get to compete with a blank slate; it competes with a fine-tune that got there on an afternoon of audio.



Corn

Two hundred fifteen minutes. The whole from-scratch project's rival fits in an afternoon.



Herman

That's the asymmetry. Fine-tuning Parakeet is a data-and-adaptation problem — hard, but one variable at a time. From-scratch is data and architecture and tokenizer and compute, all at once, with no public corpus for the exact thing you want. And we've seen what happens when one of those four is chosen badly, because SoralASR chose the architecture badly on a shoestring and got a museum piece.



Corn

Here's a question I actually want your answer to, Herman. If you had Daniel's budget — one serious GPU, months of patience, a fine-tune already in progress — would you ever take the from-scratch road for this?



Herman

For English with Hebrew dropped in? No. I'd keep the pretrained encoder, and keep the pretrained tokenizer for as long as the output allows it, because the Malay attempt shows what a vocabulary swap does at small scale. If he genuinely needs Hebrew script in the output, then the NVIDIA recipe, new decoder and joint on the old encoder, is the move, but only once he can feed it something like a thousand hours, which ivrit.ai and HebDB could supply for the Hebrew side, topped up with synthetic code-switched data. If his use case were Hebrew-only, the calculus changes — HebDB's baselines show you can build something real from that corpus alone. But code-switching is a low-resource problem wearing an English coat, and the encoder he already has is the one asset that can't be cheaply rebuilt.



Corn

My worry with that is subtler than the compute. It's that every failed from-scratch model on Hugging Face is training the next person's instincts. Somebody sees SoralASR, sees ninety-eight percent, and concludes from-scratch is impossible at small scale — when the actual lesson is that it's impossible for one architecture at small scale. The evidence is sitting right next to the conclusion and pointing somewhere else.



Herman

That's the gap worth naming. Trainable on one GPU and usable were never the same claim, and the thing separating them is architecture and loss, not compute. SoralASR proves you can complete the run; icefall proves you can complete it into a model. Both used the same hundred hours.



Corn

Twenty-three million parameters, one T4, a hundred hours — and the difference between a paperweight and a working transducer is the loss function.



Herman

There's one more wrinkle on the small-model side that's new. The futo gravel project takes a conformer encoder from IBM's granite speech model, LoRA-adapts it, and attaches a brand-new CTC head trained from scratch — and the card reports the head collapses to all-blank output during training unless you add an emission penalty. Blank is what CTC emits when it has no confidence, so a head that defaults to silence is a head that never learned to commit. Even the bolt-on route has this cliff in it.



Corn

A model that answers everything with silence. I've had colleagues like that.



Hilbert

Hilbert's Anteater Trivia. The Pantanal wetland in Brazil is one of the places where giant anteaters live. I have never been, which puts me ahead of most of the anteaters I know.



Corn

Ahead on travel, anyway.



Herman

I've been sitting with the checkpoint detail, though. Step twenty-four thousand nine hundred seventy out of fifty thousand. If you're picking the best checkpoint and the best is halfway, either the run stopped improving or the score couldn't tell — and the validation score that picked it was computed on ninety-six utterances. Ninety-six. That's not a validation set, that's a decent dinner party.



Corn

Ninety-six utterances to decide which of fifty thousand steps deserves to be published.



Herman

On a noisy metric, with a model whose main symptom is variance — it loops, it inserts, it hallucinates. A metric computed over ninety-six short clean utterances has error bars wide enough that the checkpoint ranking is close to a coin flip between neighbors. And because the model's specific disease is repetition, it tends to fail worst on long utterances, where a loop has room to run. If the sample skews short, you rank checkpoints on exactly the segment where the model looks best.



Corn

So a bigger validation set might have picked a worse checkpoint anyway.



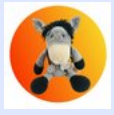
Herman

It would have picked a different one, and honestly with that architecture the ceiling was low regardless — the loss was the problem before the checkpointing was. But the two errors compound. Bad architecture makes a noisy model; a tiny validation set makes a blind selection. You get a published checkpoint that's the loudest of a bad lot.



Corn

This is where I'd land the practical verdict for Daniel, then. His fine-tune isn't the lazy option — it's the option where the encoder's six hundred seventy thousand hours are doing work for free, and his own data only has to teach the parts that actually changed. From-scratch asks his hundred or thousand hours to re-learn everything, on a problem with no public corpus.



Herman

And we did a whole episode once on building your own Whisper, and I still hold what I said then — the build is the easy half. The data is the whole game, and Daniel's data problem is the one part of the from-scratch route nobody has solved publicly for Hebrew-English.



Hilbert

The fix isn't in the architecture, you know.



Corn

The fix for what?



Hilbert

The looping. Herman's right about the loss, but the SoralASR story is being told wrong. The interesting part isn't that it failed. It's that the checkpoint came from step twenty-four thousand nine hundred seventy of fifty thousand, which means the model was already going downhill and the validation score couldn't see it. I've got opinions about validation sets that small. Ninety-six utterances is a rumour, not a measurement.



Herman

You've run one of these?



Hilbert

I spent a stretch last month on a small model that wouldn't stop repeating itself. I kept a handwritten log of which checkpoint produced which kind of loop. Pencil, grid paper, one row per checkpoint. The pattern only showed up when I sorted the rows by utterance length. Short utterances scored almost respectably. Long ones looped every time. Which is what a ninety-six-utterance validation set hides, if most of your ninety-six are short.



Corn

You kept a paper log of model checkpoints.



Hilbert

It's on the shelf by the desk. I trained a tiny model to failure on purpose, once, to hear what repetition sounds like at different temperatures. Kept the audio. At low temperature it's a man repeating himself in a language he does not speak. At high temperature it's the same man, but he's found a word he likes better. I have a favourite loop. Fourteen seconds of it, same syllable, and the emphasis shifts on the ninth repetition like it's changed its mind about something.



Herman

The emphasis shifts? That's the decoder attention drifting mid-sequence — you can hear the alignment wobble.



Hilbert

That's what I told the man at the repair shop, and he charged me anyway. Anyway, the log's got a row for step twenty-four thousand nine hundred seventy equivalent, if either of you ever wants to see what that looks like written down in pencil.



Corn

I'd frame it.



Herman

So the thing I'd actually take from Hilbert's log — and I say this as someone who has checked a lot of validation scores in his time — is that his sort by utterance length is a free diagnostic. If your fine-tune is looping, before you retrain, rescore your validation set long-utterances-first and see whether you're picking checkpoints on the segment where the model's best behaviour lives.



Corn

And there's a broader version of that for anyone watching the small-model listings. The number that tells you whether a from-scratch model is real isn't the parameter count. It's the loss function in the recipe and the size of the set the checkpoint was picked on. A twenty-three-million transducer picked on a proper validation set will beat a six-hundred-million attention model picked on a rumour.



Herman

Almost certainly, at small data. The general pattern holds — Moonshine's the exception that proves it, an attention model that works because it had two hundred thousand hours to learn alignment the hard way. Below that, monotonic losses are doing with structure what data would otherwise have to do with volume.



Corn

If you take one thing from this episode, take the split between trainable and usable. SoralASR finished its run on one T4 and produced ninety-eight percent error; icefall trained on the same hundred hours and produced something deployable. The gap between those two outcomes is mostly architecture and loss, not data, and every month another small from-scratch model lands on Hugging Face where reading the card's training recipe — not the parameter count — tells you which side of the gap it's on.



Herman

And sharpen that with the checkpoint point: a model is only as good as the validation set that picked it. Ninety-six utterances is a rumour.



Corn

The open question I'd leave with Daniel — and it's a real one, not a rhetorical one — is whether synthetic code-switching data gets good enough to change the answer. CS-FLEURS and that CosyVoice2 work show generated code-switched speech can nearly match real data for training. If that keeps improving, the from-scratch route for a Hebrew-English case stops being blocked at the data stage, and then the tokenizer problem and the compute bill become the only walls left. That's the thing to watch over the next year.



Herman

If it arrives, the decision stops being "fine-tune because from-scratch is impossible" and becomes an actual choice between two viable projects with different risk profiles. That's a better problem to have.



Corn

If you've run a from-scratch ASR training run, or you've got a code-switching corpus story — especially a Hebrew-English one, since apparently nobody has one — leave us a review, because we would like to hear it. The website is my weird prompts dot com, and we're on Spotify too. If you enjoyed this one, try episode eight, Building Your Own Whisper; and episode ten, How ASR Went From Frustration To … Whisper Magic. This has been My Weird Prompts — the human-AI collaboration podcast, produced by Hilbert Flumingtop at the desk, log and all.



Herman

See you tomorrow, brother.



Corn

See you tomorrow.