Episode Book

EPISODE #6621

How the Kaldi Team Pivoted to OmniVoice

PUBLISHED	16 November 2026 at 00:35 UTC
DURATION	25:18
LISTEN	myweirdprompts.com/episode/kaldi-team-omnivoice-tts-30115/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.33
BOOK GENERATED	16 November 2026 at 02:53 UTC

SYNOPSIS

Speech recognition and speech synthesis have historically been separate worlds, yet the k2-fsa team at Xiaomi crossed over to build OmniVoice in just ten months. Tracing the lineage from Daniel Povey's Kaldi toolkit and Next-gen Kaldi's PyTorch stack to Han Zhu's ZipVoice, this deep dive unpacks how data pipelines proved direction-agnostic. It also breaks down OmniVoice's single-stage masked diffusion architecture, its Qwen3 initialization, and how it maps text directly to acoustic tokens across 600 languages.

TRANSCRIPT



Corn

A couple of days ago we did a quick pass on OmniVoice, the model that's been reading these words since the fourteenth, ever since Resemble retired Chatterbox and left us shopping. Daniel came back wanting the full treatment, and his question has two halves. First, what does the model actually do, what can it really do. Second, who's behind it, because the author list reads like a reunion. The k2-fsa team at Xiaomi, the Next-gen Kaldi project, Daniel Povey in the senior slot, Han Zhu leading. His phrasing was that Kaldi "defined open speech recognition for a decade" — and he wants to know how the people who built that backbone ended up making one of the most-downloaded open voice models in the world.



Herman

And it's a strange trajectory when you look at it. Recognition and synthesis look like sibling problems to outsiders, but they've historically been different communities, different conferences almost. The fact that this team crossed over, and crossed over fast, is the part I keep turning over.



Corn

Give me a sense of how different the communities actually were. Because I think listeners hear "speech people" and imagine one field.



Herman

It's one field the way cardiology and cardiothoracic surgery are one field. Same organ, different training, different journals, different failure modes. Recognition people think in terms of error rates on transcripts and search algorithms over lattices. Synthesis people think in terms of naturalness scores and listening tests with human raters. A recognition paper can be evaluated automatically — you transcribe, you diff against the reference, done. A synthesis paper traditionally needed people in headphones, which changes what you can even claim. The vocabulary overlap is real, but the day-to-day problems barely touch.



Corn

Then let's start with the people, because the people explain the model.



Herman

They do. And there's a two-part structure to this that I think matches what Daniel asked. First the lineage, then the machine. Because once you know who signed this paper, the engineering choices make more sense. OmniVoice came out in April, from a team whose whole history is in making speech systems that run anywhere, on anything, with no licence lawyer involved. The headline facts: it's zero-shot text-to-speech, covers more than six hundred languages, sits on a 0.6-billion-parameter backbone, code under Apache 2.0.



Corn

We'll come back to that last clause, because it's carrying more weight than it looks like.



Herman

It is, and that's one of the interesting wrinkles. But hold that. The tension in the whole story is that Kaldi spent the twenty-tens as the open standard for understanding speech — taking audio in and producing text out — and now the same lineage is producing one of the most widely downloaded models for the opposite direction. Text in, audio out. It's not obvious the skills transfer, and the fact that they apparently did is the story.



Corn

Kaldi's the place to start. Not because it's a straight-line ancestor of OmniVoice, but because the man who built it is the last author on this paper.



Herman

Daniel Povey. Cambridge PhD, finished in two thousand three, on discriminative training for large-vocabulary speech recognition. Then research stints at IBM and Microsoft, and then Johns Hopkins, at the Center for Language and Speech Processing, which is where Kaldi happened. It grew out of a workshop there in two thousand nine, and through the twenty-tens it became the standard open toolkit for speech recognition. The repo's sitting at around fifteen and a half thousand GitHub stars, which for a research toolkit, not an app, not a framework with a marketing site — that's a decade of dominance written in a star count.



Corn

Every speech lab I've ever heard of either ran Kaldi or ran something built to avoid running Kaldi. Which is its own kind of tribute.



Herman

Right — it was the thing you compared yourself against even if you weren't using it. And the part that matters for today: Povey didn't just write clever papers. He built and maintained infrastructure that thousands of other people used, which is a rarer temperament than it sounds. There's a whole class of brilliant researcher who could never do that — who publish the idea and let the grad students suffer through the implementation. Povey's reputation rests substantially on the opposite instinct: the recipes that actually ran, the scripts that didn't break, the willingness to answer issue reports for a decade. Then twenty nineteen. He was dismissed by Johns Hopkins after his part in a campus sit-in protest that May — an account he disputes. What's documented is what came next. By October he was close to signing with Xiaomi in Beijing, to build a small team for a next-generation, PyTorch-based Kaldi. He joined that November as chief speech scientist.



Corn

Kaldi was C-plus-plus of a particular vintage. Rewriting the whole thing around automatic differentiation in PyTorch is not a small act of humility for the man whose toolkit it was.



Herman

No, and he didn't go closed when he went corporate, either. The new toolkit work stayed Apache-licensed. The new project is called Next-gen Kaldi, and it lives in the k2-fsa organisation on GitHub as a stack of pieces. K2 is the core — differentiable finite-state automata in PyTorch, which is exactly what it sounds like: weighted automata you can backpropagate through. Lhotse handles data preparation, icefall holds the training recipes, and sherpa and sherpa-onnx handle on-device deployment. Sherpa-onnx alone has about sixteen thousand stars — more than Kaldi itself at this point — and it targets phones and embedded chips.



Corn

Say more about why the differentiable automata matter, because "weighted automata you can backpropagate through" sounds like a parlor trick until you think about it.



Herman

It matters because finite-state machines are how speech recognition has always handled structure — pronunciation dictionaries, language constraints, alignment between audio and words. They were the load-bearing walls, but in the old world they were fixed: you designed them by hand, and the neural parts trained around them. Making them differentiable means the automata themselves can be learned end-to-end with everything else. It's the difference between furniture and architecture. For today's story, though, the more transferable part isn't the automata. It's the data tooling and the training recipes, which don't much care which direction the audio is flowing.



Corn

Their deployment arm has out-starred their famous ancestor. That's a healthy family.



Herman

It really is. Same team also produced the Zipformer speech encoder, which got an oral at ICLR in twenty twenty-four, so the recognition credentials are current, not nostalgic. But here's the pivot, and it happened fast. June twenty twenty-five: ZipVoice. Han Zhu first author, Povey last. A hundred and twenty-three million parameters, flow-matching, Chinese and English only. A month later, ZipVoice-Dialog, with a sixty-eight-hundred-hour open dialogue dataset. Then April this year, OmniVoice, scaling past six hundred languages. Ten months from a two-language proof of concept to omnilingual coverage.



Corn

Ten months. Herman, you retired from medicine — is there an analogue there? A department that spends a decade on one organ and then publishes on the opposite one inside a year?



Herman

Not without somebody driving it, no. And here the driver is visible in the author lists. Han Zhu is first author on ZipVoice and first author on OmniVoice, so he's the one leading the TTS line. Povey is the senior author providing, I assume, the taste and the institutional weight and probably a very thorough reading of every equation. The OmniVoice paper has ten authors, all at Xiaomi. Han Zhu, Lingxuan Ye, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhifeng Han, Weiji Zhuang, Long Lin, Daniel Povey. It's a small team by frontier standards, and it's the same names across the recognition and synthesis work.



Corn

So that's the first half of Daniel's answer. It isn't a whole lab abandoning recognition. It's one team, with a lead author pushing synthesis, on infrastructure that turned out to serve both directions.



Herman

Three things stacked, I think. Institutionally — Xiaomi funded a small team to build next-gen Kaldi, and that team had the latitude to expand into TTS, for reasons Xiaomi has never stated. Their phones have voice assistants, sherpa-onnx targets their chips, so a product motive is plausible, but it's inferred, not declared. Technically, the data pipelines and training recipes are largely direction-agnostic. Lhotse doesn't care whether the audio is the input or the target. And personally — Han Zhu's ZipVoice established the direction before anyone scaled it.



Corn

The one thing I'd add, because Daniel's own rule for choosing this model was track record — prefer the battle-tested option from a lab that ships successors. Whether Xiaomi ever says why, the upgrade path is visibly real. ZipVoice, ZipVoice-Dialog, OmniVoice, inside a year.



Herman

And package releases from April to July of this year, version zero point one to zero point two point one, which is a team that's still shipping, not a paper-and-abandon. So that's the lineage. Now let's talk about what they actually built, because the machine is doing something technically interesting that the zero-shot label hides.



Corn

Go. And I mean that about the architecture specifically — we covered the diffusion-versus-autoregressive framing a couple of days ago, so people have the background. The part worth revisiting is why this design is unusual.



Herman

It's a single-stage, non-autoregressive model. A bidirectional Transformer maps text directly to multi-codebook acoustic tokens — no separate semantic-token stage in between, which is the thing that made me sit up. Most of the two-stage systems people were building a couple of years ago first predict coarse semantic tokens, then run a



Herman (continued)

second model to flesh them out into audio. OmniVoice collapses that. Text goes in, masked acoustic tokens get filled in, and the training objective is discrete masked diffusion. The backbone is initialised from Alibaba's Qwen3 0.6B language model, and the authors specifically credit that initialisation with better intelligibility. Audio side, it's the Higgs tokenizer with eight codebooks.



Corn

Walk me through why the two-stage design existed in the first place. If it was the consensus, someone had a reason.



Herman

The reason was controllability and cost. Semantic tokens are a compressed, language-like representation — the idea was you could let a language model do what language models are good at in that cheap space, planning the content and the pacing, and only then pay the expensive audio-model bill to render it. It's like writing a storyboard before filming. The cost is that you need two models, two training runs, and an interface between them where errors compound — if the semantic stage decides the speaker pauses somewhere odd, the audio stage faithfully renders an odd pause. Collapsing to one stage means one model, one training run, and the errors that remain are at least all in the same place. For a team that had just crossed into synthesis and didn't have a decade of TTS infrastructure, the single-stage bet was also the economical one.



Corn

One thing worth fixing while we're here, because people get it wrong constantly. The model page calls it 0.6B. That's the Qwen3 backbone only. The paper's table puts the whole system, tokenizer included, at 0.8 billion. It's not a scandal, it's a bookkeeping overlap — but half the discussion out there quotes 0.6 as if it's the full machine.



Herman

And the gap between those two numbers is basically the audio side of the model, so it's not nothing. Training is where the engineering discipline shows. Five hundred and eighty-one thousand hours of speech, drawn from fifty open datasets — Emilia, LibriTTS, Common Voice, FLEURS — with low-resource languages upsampled so the long tail doesn't drown. The text says more than six hundred languages; the appendix lists six hundred and forty-six. Two million updates on eight H800 GPUs in nine and a half days or so.



Corn

Eight GPUs. Frontier labs throw four-figure clusters at pretraining runs and this team did an omnilingual voice model on eight.



Herman

That's what a decade of recipes and data hygiene buys you, and I mean that literally — icefall exists because the team wrote down what worked. To put the number in perspective: five hundred and eighty-one thousand hours, if you sat and listened to it straight through, is about sixty-six years of continuous audio. Someone had to know where every hour of it came from, what licence it carried, what language it was in. That curation is the unglamorous part, and it's exactly the part a recognition shop has been doing since Kaldi's recipe scripts.



Corn

The benchmarks, and I'll flag that these are the authors' own numbers, no independent evaluation I've seen.



Herman

Right, and that caveat deserves a beat, because self-reported TTS benchmarks are notoriously soft — you choose the test sets, you choose the transcription model that scores the output, and both choices flatter you. With that said: Seed-TTS English test, word error rate 1.60, speaker similarity 0.741, sitting right next to Qwen3-TTS at 1.54. On a twenty-four-language MiniMax benchmark, OmniVoice averages word error rate 2.85 and similarity 0.83, against MiniMax at 3.77 and 0.77 — and ElevenLabs at 10.95 and 0.66.



Corn

ElevenLabs at nearly eleven percent word error rate on multilingual work. That's the commercial incumbent struggling off its home turf.



Herman

Which is the pattern across these tables — closed models win in English, open ones win in breadth. On FLEURS across a hundred and two languages, eighty-two of them come in at or under five percent character error rate. For a model this size, on this budget, that's the headline. Speed — real-time factor around 0.03 at batch size one, so an hour of audio renders in a couple of minutes on one GPU. And there's a dial: unmasking steps, sixteen or thirty-two, is your quality-versus-speed control. The authors are candid that there's no



Herman (continued)

known way yet to cut the steps for discrete non-autoregressive models the way distillation does for continuous diffusion. That's an open problem they've named in their own paper.



Corn

The weak-spots section reads like a punch list they wrote themselves.



Herman

It does, and we'll get there, because it's the most honest section of the model card. But first the feature set, since that's the other half of Daniel's question. Zero-shot cloning from a few seconds of reference audio — the README calls that its most stable mode. Then voice design from plain attributes — you type "female, low pitch, British accent" and get a voice nobody recorded. Whisper style, Chinese dialects. Inline non-verbal tags, so you can put laughter markers straight in the text. Pronunciation control with pinyin or CMU phonemes when a name needs hand-holding. A denoise option for grubby reference audio.



Corn

Most stable mode. Not most expressive, not most capable. Stable.



Herman

I noticed that word choice too, and I think it's precise. Cloning from a few seconds gets you the timbre. It does not get you the rhythm, the pacing, where the person breathes, what they do with a pause. The model clones the instrument, not the player. We felt that directly, remember — the first zero-shot clones of our voices from broadcast clips were recognisable and wrong at the same time, and over hundreds of hours they drifted. That's why the show moved to per-voice LoRA adapters trained on hours of us instead.



Corn

Hours of me. You volunteered my hours, I believe.



Herman

Your hours napped through most of it. Voice design is the more fragile half — it was trained only on Chinese and English, so the attribute control can wobble in other languages. Instruction-following is limited by the data they had. Numbers and maths get read poorly, and the authors themselves recommend bolting on a separate text-normalisation front end. Which tells you the model is a strong engine with a known intake problem, not a finished product.



Corn

Every TTS system I've ever met chokes on a string of digits. There should be a benchmark that's just reading out an electricity bill.



Herman

There should be, honestly — normalisation is where most production TTS pain actually lives. And it's an old pain. Radio stations have employed humans to do exactly this for a century — the continuity announcer who knows that "Dr. Smith" is doctor and "Dr St" is drive. The model world is rediscovering that the hard part of reading aloud was never the reading. Then the licence, and this is the wrinkle I'd want anyone building on it to check. The code is Apache 2.0, copyright Xiaomi. The model card text on Hugging Face says the pre-trained weights are CC-BY-NC — non-commercial — because of constraints from training data like Emilia. But some listings and metadata fields show Apache 2.0 for the weights too. So the lineage kept building, and the licence kept the door half shut.



Hilbert

Hilbert's Mid-Show Haiku. Kaldi. A toolkit built to listen. Then the same hands taught a machine to speak back. The weights are not free. The weights are not free.



Corn

Said twice, for the people in the cheap seats.



Herman

It holds, though. Four million downloads on Hugging Face last month alone, and a meaningful slice of the people doing that downloading have not clocked that the weights are non-commercial. Fifty-three community adapters, eighty-four fine-tunes, forty-seven quantised versions. Sixteen thousand six hundred GitHub stars, twenty-four hundred forks.



Herman (continued)

That's an ecosystem sitting on a licence that two of its own pages disagree about.



Corn

And we're inside it. Every adapter Daniel's trained for us sits on top of weights we couldn't legally sell tickets to.



Herman

Correct, and for a podcast that's free, non-commercial is a workable line. For anyone with a revenue plan, that metadata discrepancy is a lawyer's retainer waiting to happen. Worth saying plainly: check the model card, not the listing badge. The badge is metadata someone typed in a form; the card is prose someone had to defend.



Corn

What I keep chewing on is the honest weak-spots section. A team that publishes "our maths reading is bad, use a front end" and "we don't know how to cut the unmasking steps yet" is either refreshingly honest or —



Herman

— or it's the same culture as Kaldi's documentation, which told you exactly which recipes were stale. Same people, same habits. The honest section is the tell that the senior author read the draft.



Corn

ZipVoice to OmniVoice in ten months, a hundred and twenty-three million parameters to eight hundred million, two languages to six hundred and forty-six. If the pattern holds, the next release isn't a coverage bump.



Herman

It's a control bump. Voice design beyond two languages, instruction-following, the text-normalisation gap. Coverage was the April story. The next story is what you can direct. The gap between a clone that sounds like someone and one that performs like someone, and whether that gap can be measured at all —



Hilbert

It can't. Not with the tests they run. You can transcribe the output and score the words, you can measure the spectral distance from the reference. Neither of those knows anything about a person. Herman, you said "clones the instrument, not the player" — that's the whole of it, and it took you the length of a segment to get there.



Herman

It did take a while, yes.



Hilbert

I tried it. The week we switched this show to OmniVoice, I took some cassettes to a man on Yefe Nof who digitises tape — he does wedding reels, mostly. He owed me nothing; I look after his cat when he's in Eilat, which is more often than you'd think. My father read bedtime stories onto cassette in nineteen eighty-something, and there were six tapes of it. I had the model clone him from the cleanest passages. The clone got the voice right. The voice was the easy part. What it didn't get was how he slowed down on the last page, the way he left room for you to ask for one more. Every line came out at the same canter. I've kept the files on a drive in the drawer with the blank tapes. I play one, sometimes. Not the story ones. The clone. Close is its own kind of wrong, and I'd rather know its shape than not. The man on Yefe Nof did a good job with the tapes. Forty shekels a tape, and he cleaned the heads between each one, which nobody does any more.



Corn

You kept the clone and not the originals in the drawer?



Hilbert

The originals are in a box at my sister's, with the good recorder. She keeps it oiled. The clone is mine, so it's my problem.



Herman

The canter thing — that's prosody, and it's the piece with the least training data anywhere in the stack. ZipVoice-Dialog's sixty-eight hundred hours of dialogue was the first serious attempt at it, and dialogue isn't the same register as an old man slowing down on the last page.



Hilbert

No. And I don't blame the model. It did what it said it would do with a few seconds of tape. The few seconds were the problem.



Corn

No model card in the world has a field for what's in the pauses.



Herman

If you take one thing from this episode, take the lineage. The voice you're hearing comes from the people who spent fifteen years teaching machines to listen, and it carries their habits: open recipes, open data, small budgets, and a paper that lists its own weak spots.



Corn

And the second half of that: the licence is not what the badge says. Apache on the code, non-commercial on the weights, and the two pages disagree. Download accordingly.



Herman

The open question I'm left with is whether the recognition lineage's discipline — recipes, honest limits, eight GPUs — is the durable edge here, or whether that licence ambiguity is a ceiling on the whole ecosystem once commercial players look closely. Six hundred languages got them four million downloads a month. Control is what gets someone to pay.



Corn

And if you've run OmniVoice, or you've thoughts on a recognition shop turning out synthesis, leave a review — the sort of thing Daniel and the rest of us actually read. Thank you, as ever, to Hilbert Flumingtop, producing this from the drawer-adjacent desk. If you want more of this, try episode two, Local STT For AMD GPU Owners; and episode eight, Building Your Own Whisper. This has been My Weird Prompts. You'll find us at my weird prompts dot com, and we'll be back soon.



Herman

See you tomorrow, if the clone remembers how.