Episode Book

EPISODE #6681

Teaching Hebrew Pronunciation to Many Synthetic Voices

PUBLISHED	19 November 2026 at 13:10 UTC
DURATION	24:26
LISTEN	myweirdprompts.com/episode/hebrew-pronunciation-synthetic-voices-38228/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.36
BOOK GENERATED	19 November 2026 at 13:34 UTC

SYNOPSIS

Recording an hour of Hebrew-English speech daily creates a rich long-tail corpus, but it entangles one speaker's voice with the language's pronunciation. Testing a proposed pipeline—converting speech to IPA and rendering it across synthetic voices—reveals why universal phone recognizers like ZIPA and POWSM fail by erasing stress and normalizing to dictionaries. Instead, pairing text-based G2P tools like Phonikud with Hebrew-tuned TTS models like Kokoro, or using accent-preserving voice conversion with HuBERT, offers a cleaner path.

TRANSCRIPT



Corn

An hour a day. That's Daniel's number for his Hebrew-English corpus — an hour of recorded speech, every day, working through what he calls the long tail. Unusual Israeli names, street names, the words that make a recognition model quietly give up.



Herman

The long tail is very long.



Corn

His words. Daniel is fine-tuning a Parakeet-family model on his own read speech, open-sourcing the corpus as he goes, and somewhere in the middle of all that recording he noticed something. He says he's teaching the model two things at once. One is selfish, his word — his voice, every nuance of it, because he wants excellent recognition for himself. The other is the part everyone else gets: how Hebrew names are actually pronounced, including the stress. And those two things are stuck together in one man's vocal tract.



Herman

I like that framing. It's an honest audit of what a single-speaker corpus contains.



Corn

So yesterday he had an idea. Record the sentence, convert it to something symbolic — IPA, phonemes — where his recording acts as the pronunciation teacher, and then render that intermediate form out in many different synthetic voices. Many speakers, one pronunciation lesson. He wants to know if there's an established methodology for that, and if not, what the best model is for each end of the pipeline. Let's take it apart.



Herman

The teacher-student framing is the part I keep turning over. Daniel has actually been down a road adjacent to this before — he built Meorav, the English-with-Hebrew TTS model, and in that project the phoneme route is exactly what failed him. He wrote ARPAbet to Israeli pronunciation rules, he hand-wrote IPA into Zonos, and the gutturals never came out. What worked was masked multi-speaker fine-tuning with a swapped-in



Herman (continued)

x-vector.



Corn

Which is a hint we should probably keep in our pocket, because his new idea runs straight back through the thing that already bit him.



Herman

It does, and the bite marks are useful data.



Herman

Start with the long tail, because it explains the whole project shape. Recognition models are trained on distributions — the most common words get seen millions of times, and the further you go from the center, the thinner the coverage. English first names are fine. Common Hebrew names, mostly fine. But an Israeli street name in Rehavia or a surname from Yemeni Hebrew that doesn't follow the usual spelling patterns? That's the far end of the distribution, and it's not a small region. There are thousands of street names in Jerusalem alone.



Corn

And you can't record your way across it in one heroic weekend.



Herman

You can't. That's why the hour-a-day approach is right, structurally — incremental fine-tuning where the corpus grows in whatever order the tail presents itself. The model doesn't need every name; it needs enough examples that it generalizes the pattern, that Hebrew names inside English sentences are normal and have recoverable structure.



Corn

Now the two things being taught. Walk me through the difference, because on the surface "teaching it my voice" and "teaching it the words" sound like the same activity.



Herman

They're separable targets that happen to share a microphone. Voice adaptation is acoustic — his formants, his timing, his glottal quirks. A model tuned on one speaker learns to expect that speaker's vocal tract, and it gets very good at that tract. Pronunciation encoding is symbolic — which phone sequence the name is, and where the stress falls. Milra or milel, final stress or penultimate. That's knowledge about the language, not about the man.



Corn

And the corpus teaches both, but it only demonstrates the second through the first.



Herman

That's the trap. A public dataset that is one voice teaches "how Daniel says these words" more than "how these words are said." Other speakers don't have his tract, his accent — Irish-accented Hebrew, which is its own beautiful thing — his speaking rate. So the open-sourced artifact is less useful to other people than the work that went into it deserves.



Corn

His fix is synthetic speakers. Mine would be too, if I had one. So does the pipeline he's imagining exist? Does it have a name I can blame?



Herman

It doesn't. I looked at this from every angle I could, and there is no established method that goes teacher recording, phonetic representation, many TTS voices, into an ASR corpus. Nobody has blessed that exact chain with an acronym. The nearest things are three. TTS speaker-diversity augmentation, where you generate many voices from text. Accent-preserving voice conversion, where you change the voice and keep the accent. And speech editing, where you splice named entities into real recordings.



Corn

Notice that none of those three has an IPA step in the middle.



Herman

None of them. And I think that absence is the interesting part, because the IPA bottleneck is appealing on paper — symbolic, interpretable, and it separates pronunciation from voice by construction. You can read the intermediate form and check it. It's sheet music.



Corn

Sheet music that the musicians can't necessarily play.



Herman

Let's do the teacher end first. Best current option for audio-to-phones is ZIPA, a universal phone recognizer out of the Zhu and Mortensen groups, trained on eighty-eight languages, about seventeen thousand hours. Solid piece of work. Hebrew is not one of the eighty-eight.



Corn

Zero-shot, then.



Herman

Zero-shot, which is not automatically disqualifying — universal phone recognizers are built for that. But there's a worse problem. ZIPA's training labels come from pronunciation dictionaries and grapheme-to-phoneme conversion, and the authors are candid about the consequence: the model smooths out phonetic variation. It matches the dictionary form better than it matches what was actually said.



Corn

Herman.



Herman

Yes.



Corn

He wants to teach his pronunciation, and the best teacher-side model normalizes toward the dictionary.



Herman

That's the crux of the whole idea. The recognizer is a very polite student. It's heard how the word is supposed to go, and when you say it your way, it quietly writes down the standard way.



Corn

The one thing he's the sole source of, and the tool's entire training regime is built to discard it.



Herman

There's a newer one, POWSM, from the ESPnet crowd, that does phone recognition, ASR, and audio-guided G2P in a single model. Marginally better numbers on unseen languages — phonetic feature error rate around eighteen point seven versus ZIPA's nineteen. But it drops tones, it doesn't model stress at all, and it wasn't evaluated on code-switching. So on the exact axis Daniel cares about, stress in Hebrew names, it's silent.



Corn

You said the error rate numbers like they're close. Is eighteen point seven good?



Herman

In context, yes — the older wav2vec2 phoneme models sit around twenty-one, and in-domain, on languages it trained on, POWSM gets down to about two point six. So the unseen-language penalty is enormous. These are general-purpose tools, and general-purpose is the wrong shape for this job.



Corn

Here's what strikes me. Daniel already knows the text of every sentence he records. He's not transcribing unknown speech. He's reading prompts.



Herman

And that changes everything.



Corn

So don't run a recognizer over the audio at all.



Herman

That's the move. Generate the IPA from the text with a Hebrew G2P — the one worth naming is Phonikud, open source, and it's the only tool I found that writes stress down explicitly, a stress mark before the stressed vowel. It outputs the sounds that actually matter: the uvular fricative for het and khaf, the uvular resh, the glottal stop. Then your recording does a different job — it validates and corrects the generated transcription, and it supplies the timing. The audio becomes the reference standard, not the source.



Corn

Which is honestly a better role for it. He is the pronunciation teacher still — but he's teaching the symbol layer directly, with his ears and his mouth, rather than hoping a recognizer can hear the lesson through its dictionary accent.



Herman

And one point of order people get wrong constantly — Modern Hebrew as spoken doesn't need pharyngeals. People assume the model needs ancient gutturals. It doesn't. Het has merged with khaf, ayin has mostly merged with aleph into a glottal stop or nothing, and resh is usually uvular, the French-style r. The sounds an English model lacks are three, and Phonikud covers all three.



Herman

Now, a brief word from Dorothy with her History of Toast. Mum, take it away.

D

Dorothy

Hello dear, it's your mother, with Dorothy's History of Toast.

D

Dorothy

In January nineteen forty-three, the United States government banned sliced bread, as a wartime economy measure.

D

Dorothy

People were so upset that the ban was lifted within two months. Nothing lasts, sweetheart, not even a nice loaf.

D

Dorothy

Anyway. I've got a stew on and I've lost the lid again, so I'll let you go. Bye bye.



Corn

The government banned sliced bread and held out two months.



Herman

Against the entire American public. Alright — the rendering end, and this is where the pipeline lives or dies. You have your IPA with stress marks. Now you need many voices saying it. The immediate problem: a stock English multi-speaker TTS will not render Hebrew phones no matter how correct your symbols are.



Corn

That's not a hypothesis for Daniel, that's a finding. Meorav proved it.



Herman

His own experiment, twice over. And the literature agrees with him. Papercup ran a study on zero-shot multilingual TTS and found that unseen phonemes given random embeddings were mostly mispronounced or just skipped — a trill collapsed to g when the English phone set had no trill. Staib's group showed phoneme-based models cannot synthesize unseen phonemes without manual mapping or retraining. Unknown symbol, nearest English sound. Every time.



Corn

So the IPA string sits there, correct and beautiful, and the renderer reads it with an accent so thick it erases the content.



Herman

Zonos, for instance — the model Daniel actually hand-fed IPA — phonemizes internally with eSpeak and documents English, Japanese, Chinese, French, German. No raw IPA input, no Hebrew. The fix exists, though. You fine-tune the renderer on Hebrew phones. There's a Kokoro Hebrew fine-tune that takes Phonikud IPA and maps it onto Kokoro's one hundred seventy-eight token vocabulary, and the Phonikud project itself says an English phoneme TTS can be fine-tuned on about an hour of Hebrew. One hour.



Corn

That's roughly one day of Daniel's recording schedule.



Herman

I noticed that too. He accidentally already has the bootstrap data.



Corn

So the symbolic route works, conditionally. Renderer must know the phones, G2P must be the same one at training and inference — no swapping the ruler halfway through the measurement — and then many voices.



Herman

There are Piper and StyleTTS2-based Hebrew models with stress markers too, so the ecology exists. But here's the road that skips the bottleneck entirely, and I want your instinct on it. Voice conversion. Speech resynthesis from discrete units — the Polyak work from Meta — takes HuBERT units for content, plus pitch and a speaker embedding, feeds a decoder, and you can swap the speaker embedding while keeping the content. And there's a recent Edinburgh paper that found accent lives most strongly in the mid-early HuBERT layers, while tokens trained with ASR supervision strip accent entirely. They explicitly name "accent-preserving" conversion as a distinct thing from accent conversion.



Corn

Distinct and opposite. Accent conversion changes how something sounds toward a target accent. Accent-preserving keeps the source accent and changes the person.



Herman

Which is Daniel's goal, named by someone else. Keep how Daniel says the name, change who's saying it. The catch is that mid-layer tokens carry speaker and accent together — you don't get a clean separation, you get a bundle, and the converter has to not drop the part you want.



Corn

That's the kNN-VC family, right?



Herman

kNN-VC swaps in nearest frames from a target speaker's WavLM features. And there's a paper from August last year, scalable controllable accented TTS, that did almost exactly Daniel's operation — took accented training utterances and re-voiced them into random LibriTTS speakers, deliberately, because conversion keeps the source accent. It was for TTS training data, not ASR, but the mechanism is the same. One accented speaker in, many voices out, accent intact.



Corn

But.



Herman

Rhythm leaks. Prosody leaks through every content-preserving conversion route unless you explicitly resample duration. Daniel already watched this happen with Seed-VC — his accent's rhythm came across whether he invited it or not. So the converted voices are all speaking with his cadence wearing different bodies.



Corn

A hundred people who all pause where he pauses.



Herman

For an ASR corpus, is that a real problem or a tolerable one? You're not asking the synthetic voices to be expressive. You're asking them to vary.



Corn

Tolerable, probably — but it means you get voice diversity without getting speaking-style diversity. Those are the two axes the single-speaker corpus is poor on, and this only fixes one.



Herman

That's a sharper distinction than most of the papers make. Now the question underneath everything — does any of this synthetic multi-speaker data actually help ASR? Because if it doesn't, the whole teacher-student scaffolding is decoration.



Corn

And the answer is yes, with a list of conditions attached.



Herman

A January paper out of NTU and A*STAR is the most on-point study I've seen. Mandarin-English code-switching, SEAME corpus, so structurally the same problem as Daniel's. They fine-tuned CosyVoice 2's language model on the code-switched text, generated synthetic speech, and fine-tuned Whisper on the mix. Real data only, a hundred hours: mixed error rate around twelve point one on one dev set. Add a hundred hours of synthetic speech reusing the original speakers' voices — modest gain, eleven point one. Add a hundred hours of synthetic speech with random speaker embeddings — ten point one. The authors said it flat: the crucial ingredient is speaker diversity.



Corn

Same volume of synthetic data, the only change is whose voices — and the gain nearly doubles.



Herman

And the negative control holds: two hundred hours of synthetic only, no real data at all, lands at twelve point two — worse than the hundred hours of real speech. Synthetic complements real. It does not replace it.



Corn

Which I suspect is the sentence Daniel most needs to tattoo somewhere.



Herman

The street-name precedent is the one that made me sit up, though I've only seen it in summary, not the full paper. Singapore road names and addresses — a recognition task that's almost entirely the long tail — and they report error rate from fifty-seven point six percent down to sixteen point five using synthetic speech alone.



Corn

Fifty-seven to sixteen.



Herman

With an interesting wrinkle: using several different TTS systems gave more useful diversity than one system with many speakers. Different engines make different mistakes, and the spread of mistakes is itself informative for training.



Corn

And there's a lighter option nobody ever mentions — the contextual biasing work from September. Instead of retraining anything, they synthesized each rare word in eighteen variants to harvest pronunciations.



Herman

Three TTS engines, three voices, two templates per hotword, and biased-word error rate dropped forty-three percent with no retraining at all. For a model that already mostly works and just needs the names, that's a serious contender. Daniel is doing incremental fine-tuning anyway, so the full route suits him — but if he ever wants a quick win on a batch of new street names, that's it.



Corn

Tempting for a man recording an hour a day to have a method that requires zero hours a day.



Herman

Though hold on — the limits matter as much as the wins. Edinburgh again, Minixhofer and colleagues: training ASR on synthetic speech leaves a real gap, best case about one point four six times the error rate of real data. Diffusion-based TTS used extra speakers better than the older MSE-trained systems. And a twenty twenty-five evaluation of augmentation tricks for low-resource ASR found voice-conversion speaker augmentation was ineffective in their setup, while other attribute augmentation gave eleven to thirty-five percent relative gains. Speaker augmentation is not a free win in every regime.



Corn

So the honest stack-up is: synthetic multi-speaker data helps, when it's diverse, when it sits on top of real data, when the renderer actually pronounces the words correctly. Fail any of those and you've added a hundred hours of noise with a filename scheme.



Herman

And that last condition loops all the way back to the beginning. However you get the diversity, the pronunciation has to survive the trip. If the IPA bottleneck normalizes the stress, or the renderer maps the uvular fricative to the nearest English sound, you've built a machine for teaching many voices the wrong name.



Corn

Better forty hours of correct synthetic speech than four hundred of confident errors. Wrong training audio is worse than no training audio, because it arrives with a straight face.



Herman

Which points to the sequencing. He already knows the pronunciation — that's the one input a recognizer can't give him. The Phonikud route gives him a checkable symbol layer, his own ears do the validation, an hour of Hebrew fine-tuning makes the renderer capable, and the speaker embeddings supply the diversity that the Singapore and SEAME results both say is the active ingredient. It's not a named methodology. It's a sensible assembly of named parts.



Hilbert

Do either of you know what happens to a voice when the man stops being able to do it? I've been sitting here thinking about my cousin. He did the prompts for a phone menu system, back when those were recorded by a person in a booth. Hundreds of them. Press one, please stay on the line, that kind of work. He had the rasp even then, just a bit of it, and it suited him. Years later the company wanted to update the system and they couldn't find anybody who could re-record to match him, because by then his voice had gone rough, and the booth was gone too. They spliced old recordings to cover the new prompts. Corn, you'd have heard the seams from across the street.



Corn

I expect I would.



Hilbert

Daniel's got the right of it, writing the pronunciation down. My cousin never did. He had a notebook of phonetic spellings, his own alphabet, nobody else could read a word of it. When the voice went, that's where the knowledge was, and it wasn't knowledge anymore. I've got the notebook here, in a drawer with thirty-one cassette tapes of his outtakes. I keep meaning to digitize the tapes. The notebook I don't touch. It's not my voice, it's his, and he's gone now. I said I'd bring it in to show you, Herman, but I won't. It's a private thing. Anyway. Thirty-one takes of please hold. He could never get the hold how they wanted it.



Corn

Some man in a suit, deciding how "hold" should sound.



Herman

You know, the notebook thing is technically a solved problem now — that's exactly what Phonikud does, writes the pronunciation in a form anyone can read. His alphabet just came a few decades early. Nobody around him could parse the format.



Corn

And the tapes outliving the notebook's usefulness — that's the single-speaker corpus with no intermediate layer. The audio is all there is, and when the tract that made it is gone, so is the lesson.



Herman

Daniel's whole idea is the notebook that other people can read. The IPA string is the part you hand forward. The voice doesn't travel anyway, but the pronunciation, if it's written in something shared, does.



Corn

Although write it in something a stock renderer can't pronounce, and you've built a beautiful notebook nobody's machine can read either. The renderer has to learn the letters before the book is worth anything.



Herman

The misconception, then, for anyone tuning in with the same idea Daniel had. People assume an audio-to-IPA recognizer will capture exactly how a person pronounces the Hebrew.



Corn

It won't. The recognizers are trained on dictionary pronunciations and they smooth toward the standard form — they're built to normalize away the very speaker-specific pronunciation you recorded the audio to preserve. The teacher step works backwards. Go from the text with a G2P and use your own ears as the corrector.



Herman

And the companion belief — that any good multi-speaker TTS can render IPA. It maps unknown symbols to the nearest English sound. Fine-tune the renderer on the Hebrew phones first.



Corn

One hour of Hebrew, apparently.



Herman

One hour. I keep coming back to that number — it's smaller than the thing it unlocks.



Corn

Forward-looking, then. The bottleneck question is really a bet on direction. Right now, symbolic control costs you a Hebrew-aware renderer and acoustic fidelity costs you a leak of rhythm. If zero-shot multilingual phoneme rendering keeps improving, the IPA route gets cheaper every year, and Daniel's intermediate layer becomes the part everyone standardizes on. The pronunciation, written down, outliving every voice that ever said it. Including Hilbert's cousin's.



Herman

The notebook that reads itself. Long tail's still long, but an hour a day adds up.



Corn

Thanks as always to Hilbert Flumingtop at the desk, producing the show out of thirty-one takes of this very conversation.



Herman

He got the hold how he wanted it, eventually.



Corn

For more along these lines, there's episode eight, Building Your Own Whisper; episode seven, Building Custom ASR Tools; and episode nine, Benchmarking Custom ASR Tools - Beyond The WER. This has been My Weird Prompts, the human-AI collaboration podcast. No question too obscure, no rabbit hole too deep. If you enjoyed this one, the website is my weird prompts dot com, and a review wherever you listen helps more than you'd think. See you tomorrow, Herman.



Herman

Tomorrow, Corn.