Episode Book

EPISODE #6138

Hebrish on iPhone: MLX, Core ML, or Wait?

PUBLISHED	19 October 2026 at 17:36 UTC
DURATION	20:23
LISTEN	myweirdprompts.com/episode/hebrish-iphone-mlx-coreml-15283/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	anthropic/claude-haiku-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.10
BOOK GENERATED	20 October 2026 at 07:46 UTC

SYNOPSIS

Hebrish is a free, open-source fine-tune of NVIDIA's Parakeet that handles English and Hebrew mixed in one sentence — and it already runs as an Android voice keyboard. This episode digs into what it would actually take to bring it to iPhone. MLX runs on phones, but the Parakeet code on top of it is Apple-Silicon-Mac only. Core ML already runs stock Parakeet on the Neural Engine via FluidAudio, but Hebrish's weights were never converted. And sherpa-onnx has an iOS build guide that doesn't quite confirm Parakeet works there. Along the way: why stock Parakeet writes "Micollette" instead of makolet, why silent CPU fallback is the scariest failure mode, and why iOS custom keyboards can't touch the microphone at all.

TRANSCRIPT



Corn

A friend of Daniel's saw Hebrish in action, and the question he sent over was the obvious one. Is it on iOS? It wrote makolet correctly, which stock Parakeet does not. Stock Parakeet writes Micollette. The half-memory going round is that the iPhone route is MLX and one more, and the MLX half is more interesting than it sounds, because MLX does run on iPhones.



Herman

Daniel's answer was not yet, the weights haven't been ported, and the reason is more useful than the answer. Hebrish is a non-profit, open-source fine-tune of Parakeet, NVIDIA's speech model, for English and Hebrew mixed in one sentence, which is how a lot of people actually talk. It's free to download right now at hebrish dot app. There's an Android voice keyboard, and there are builds for Ubuntu and Windows.



Corn

Why does stock Parakeet get it wrong, though? It's a good model.



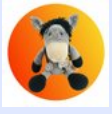
Herman

It's a good English model. The base was trained on English, so it hears the Hebrew sound and reaches for an English spelling. The fine-tune is what teaches it the Hebrew spelling, and that's the whole reason Hebrish exists.



Corn

I've got the Android one installed, from hebrish dot app, obviously. I'm the one in this conversation who installs things from a file, so I'm the right sloth for the second half.



Herman

The second half is what that installing costs on an iPhone. First, the engine. MLX is the one people reach for first, and it does run on iPhones. There's a Swift version, mlx-swift, and Apple's own example apps run on both iOS and macOS.



Corn

So where's the Mac-only part? There's always a Mac-only part.



Herman

The Parakeet code. parakeet MLX is the Python package that runs Parakeet on MLX, and it only runs on Apple Silicon Macs. Hebrish's MLX build sits on top of it. So an MLX route to an iPhone means rewriting the Parakeet model code in Swift on mlx-swift. That's doable. Nothing off the shelf does it yet. It also leans on ffmpeg for audio, which an iPhone doesn't ship with, so that's another piece to replace.



Corn

Could a phone just run the Mac package as it is?



Herman

No. A phone has no Python environment you'd ship, and the package assumes a Mac from top to bottom.



Corn

MLX is the engine, and the translation is the job.



Herman

Yes. The one more is Core ML. There's an open-source Swift package called FluidAudio, and it already runs stock Parakeet TDT on the Neural Engine, the chip in the phone built for this. It needs iOS seventeen or later.



Corn

You're doing the voice again.



Herman

The voice is earned. It's a good chip, and anything older than iOS seventeen is out for this route.



Corn

And what's the catch?



Herman

The stock version two model is English only, and Hebrish is a fine-tune of version two. So the ready-made route still needs someone to convert Hebrish's weights into Core ML. Stock Parakeet in Core ML is four pieces, a preprocessor, an encoder, a decoder and a joint piece, plus a vocabulary file. Hebrish would need its own four. I found no iOS conversion of Hebrish anywhere, and the sizes of those bundles aren't published.



Corn

What's the hard part, the export or the model?



Herman

The export is the easy bit. The model predicts a token and how many audio frames that token spans, and the Swift decoding loop has to match the Python one step for step, or the words drift. That match is where ports die, and it's the bit nobody puts in the announcement.



Corn

Matches on what, one sentence?



Herman

On a test set. Hebrew, mixed with English, with the makolet cases in it. Somebody has to build that set and sit through the results.



Corn

Four pieces and no sizes. That's flat-pack furniture with the measurements left off.



Herman

Which is why memory matters. The Core ML card for stock version two quotes about eight hundred megabytes at peak. It doesn't say which device it measured on.



Corn

Eight hundred megabytes, for a thing that types what you say.



Herman

For a thing that types two languages in one breath. Hebrish's own smaller file is about four hundred and seventy-five megabytes, aimed at phones with four gigabytes of memory or less. That's a different file format, not the Core ML route, so don't read it as the iPhone number. I found no measured iPhone figure for Parakeet at all. The Mac numbers you'll see, around a hundred and ten times faster than real time in batch on a high-end Mac, are not phone numbers. And as far as I understand it, iOS can reclaim memory from apps that aren't in front of you, so the number matters for backgrounded apps too.



Corn

So eight hundred is a number from somewhere, on something.



Herman

Yes. Corn, if you had to bet on the real iPhone figure, which way would you go?



Corn

Higher. Everything on a phone is higher than the card says.



Herman

That's my bet too. The third option is sherpa-onnx. It has an iOS build guide, and it says everything is processed on the device. That page doesn't confirm Parakeet TDT on iOS. The decoding exists in the project, but I haven't seen a real run on a phone written up.



Corn

An option the way a spare key is an option.



Herman

And the Core ML side has a warning. A Japanese Parakeet conversion reportedly fell back to the main processor on an A fifteen chip, roughly an order of magnitude slower.



Corn

Fell back. Quietly, I assume.



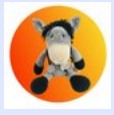
Herman

Quietly. Some operations don't map onto the Neural Engine, so the phone runs them on the general processor and tells nobody.



Corn

That's the scariest thing you've said all week, and you read papers for fun.



Herman

Corn, you're on Android. Would you use a dictation thing that thinks for a second before it types?



Corn

If it spells makolet. Otherwise I'll type it with my thumb, very slowly.



Herman

So spelling beats speed.



Corn

I'd guess the Core ML route happens first. Somebody with a Mac and a grudge does the conversion before anyone rewrites Parakeet in Swift.



Herman

You're guessing that because it's the easy one.



Corn

I'm guessing because I'm a sloth and I want it to be easy. Same argument.



Herman

It is not the same argument. I'll allow it.

L

Larry

Tired of your phone understanding you only some of the time? Gabbergold Membership is a monthly listening companion for your pocket. It listens on your behalf, in every language, including the ones you haven't decided to speak yet. There are three tiers. Bronze records your words. Silver understands them. Gabbergold Platinum understands the words you were about to say, and bills you for those too. Members report feeling understood at home, at work, and in the queue at the post office, where our cancellation forms are kept. Platinum members also receive a dedicated listener named Keith, who is currently on leave and will be until further notice. Cancellation is simple. Write to us by post, enclose your original receipt and a sample of your voice, and allow ninety business days for a reply. Renewal is automatic and cannot be paused. Gabbergold Membership. Understanding is a monthly commitment.



Corn

Ads are the weather now.



Herman

Ninety business days to cancel. That's a long time to be understood.



Corn

Back to the keyboard. Why doesn't the Android shape port?



Herman

Because on Android, Hebrish's keyboard takes your voice and types into whatever app you're in. On iOS that shape doesn't exist, and the reason is Apple, not the model. Apple's developer documentation for custom keyboards says they have no access to the microphone, so dictation input is not possible.



Corn

No access at all?



Herman

That's an archived page, from the iOS eight era, and I found nothing from Apple since that changes it. There's a setting called Full Access that opens up network and shared storage with the host app, but the page doesn't say it grants the microphone. Whether it does in current iOS is a test someone has to run on a real phone.



Corn

So the keyboard can't hear anything, and the model can't live in it either.



Herman

The model can't go in there. Keyboard extensions have a memory cap, and I'd love to know it exactly. The figure I found from a React Native doc is forty-eight megabytes, and vendor numbers I saw run from thirty to sixty. None of them are from Apple, so treat the cap as unconfirmed. Either way eight hundred doesn't fit, so the model lives in a separate app.



Corn

A keyboard that has to leave itself to hear you.



Herman

That's the iOS shape. Either a standalone app, where you dictate in the app and copy the text across, which is clumsy, or a keyboard that bounces you to the app to record and comes back with the text. Wispr Flow does the second. Its keyboard hands the microphone to the containing app, which keeps a session open for five minutes, fifteen, an hour, or never. Full Access is required for transcription. Full Access is a switch the user has to turn on in settings, so every user is one more thing that can go wrong.



Corn

So the keyboard is a remote control for an app.



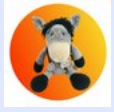
Herman

A remote control with a cooldown.



Corn

So the keyboard waits in the app like a taxi with the meter running.



Herman

And the return trip is where it's fragile. Wave, an open-source version, reportedly does the same thing. It opens the host app to record, passes the transcript back through an App Group, and the keyboard polls for it. There's a developer forum thread saying that on iOS



Herman (continued)

twenty-six point four the keyboard stops identifying the host app after the round trip. I haven't verified that on a device. Apple also answered a request for a return-to-app call with a no.



Corn

Apple said no to giving you a way back. Lovely.



Herman

Which leaves the standalone app as the boring choice, and boring is usually right.



Corn

Which one would you build first?



Herman

The standalone app. It works on day one, with no bounce and no session timer to babysit.



Corn

And after that?



Herman

The memory test on a real phone, because that number decides everything else. Then the keyboard, if at all.



Corn

And boring costs money, I'm guessing.



Herman

Eventually. The free version comes first, because you'll want it.



Corn

I always want the free version. On Android I download the file, tap install, accept the scary warning, and it's on the phone in a minute. I've never once asked who was allowed to put it there. On an iPhone it won't take anything from outside the store. Full stop.



Herman

Corn, would you pay ninety-nine dollars a year to skip the weekly appointment I'm about to describe?



Corn

For a dictation app that spells makolet? Maybe. I'd want to see the receipt first.



Herman

On an iPhone the free route is sideloading. You sign in with a free Apple ID, sign the app with a seven-day certificate, and you can have three apps going at once. Renewal needs the computer back in the loop. That comes from secondary sources, and I haven't checked the current AltStore or SideStore docs, so treat those numbers as provisional.



Corn

Seven days. My naps last longer than that certificate.



Herman

The paid membership removes the three-app cap and gives you a year per signature. That's the first real price.



Corn

Here it comes.



Herman

Ninety-nine dollars a year, the standard fee. The non-profit waiver removes it, but only for a legal entity with nonprofit, accredited school, or government status. Not an individual, not a sole trader. No paid apps, no in-app purchases, and if you start charging, you pay the fee.



Corn

So the waiver needs Hebrish to be a registered entity.



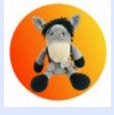
Herman

Yes, and I don't know whether it is. Israel was on the waiver list back in twenty twenty, according to a secondary source. Whether it's on the list this year, I couldn't confirm. Apple's page doesn't name countries.



Corn

What counts as an entity? A group chat?



Herman

Not a group chat. The page says a nonprofit, an accredited school, or a government body, registered as an entity, not a person with a good idea.



Corn

So the free route might be closed to us, and we don't know. That's a nice sentence to read to people.



Herman

The EU route is worse. Apple's Digital Markets Act page says alternative marketplaces and web distribution are for users in the EU. Israel isn't in the EU, and the page names no route for it.



Corn

So the alternative store is for everyone except us.



Herman

Then there's TestFlight. Up to a hundred internal testers from your App Store Connect users, and up to ten thousand external testers. Each build lasts ninety days, and the first build of each version goes through beta app review.



Corn

So it's a beta. For ten thousand people.



Herman

Ten thousand people who've agreed to put up with a rough edge, which is the crowd a community project actually has. It's a beta channel, and it's the closest thing to a store a community project can get without a store. Here's my recommendation for Hebrish. If it has a legal entity, check the waiver first, and ask Apple directly about Israel. If it doesn't, pay the ninety-nine dollars, distribute through TestFlight with external testers, and rebuild every ninety days. A model that downloads after install may draw questions in beta review. I haven't checked Apple's rules on that, so treat it as a guess.



Corn

So the cheapest route is the one with a weekly appointment.



Herman

Free sideloading is for tinkerers. Your users would re-sign every week and end up in your support queue.



Corn

You'd sideload the moon if it came as a file.



Herman

Whatever route you pick, the return trip is still the weak point, unless you skip it. The standalone app skips it, and you copy the text across yourself, which is the clumsy price of never having to bounce.



Hilbert

That's right. Anyway, I've got a plan for the return trip, and I've already bought the parts.



Corn

Go on.



Hilbert

Two phones, second-hand, from a shop in Bat Yam that sells them by the kilo. Four hundred and twelve shekels the pair. The keyboard phone is clamped to the handlebars on a dress rail, which I also bought. The sticker on the back of one is from a bank that has since been bought by a bank that has since been bought. One phone is the keyboard. The other runs the app. Between them there's a moped, and the moped carries a courier from one to the other. He takes the recording off the keyboard phone, rides it across four streets, and plugs the transcript back in. I call it the ferry.



Herman

A moped as a courier. That's a technical problem with a good reason behind it.



Hilbert

The moped is Nachum's. It's in a storage unit in Holon, which is where Nachum keeps everything he's been asked to return.



Corn

So the ferry's stuck in Holon.



Hilbert

The courier is Yehuda. He's an accountant for a bridal shop in Petah Tikva. He's not a rider. He's been on the back twice, and the second time the bell on the moped went off at the lights by the old depot, and the keyboard phone took the bell down as a word. Four times. The same word every time, which I'm not going to spell.



Corn

So the bell's the only one on that moped with a clear voice.



Hilbert

He wears a helmet from the bridal shop. It's the one they use for the veil demonstrations, white, with a gap cut in the back for a ponytail. The licensing office has asked about that too.



Herman

A helmet is a standard you can check, which puts it ahead of the phone.



Hilbert

Then the council. The licensing office wanted a second rider on the card. Dalia Abutbul at the licensing office. She said a courier vehicle needs a second person registered, and the person has to be on the form. I said Yehuda has been on the back twice. She has a stamp for everything. She stamped the phone's receipt, then she stamped the stamp. She said the form is for the moped, and the card is for the phone. The card has Yehuda's photograph on it, taken at the bridal shop's Purim party, in which he is dressed as a bride.



Herman

So the phone needs a licence, and the keyboard doesn't know that yet.



Hilbert

There's a box on that form for the horn. There's no box for the bell. So I wrote the bell in the horn box, and I'm not taking it out.



Corn

So the horn's the bell now, legally.



Hilbert

Eleven minutes from the keyboard to the app at the lights by the old depot, every time I've timed it, on a stopwatch from the bridal shop's drawer. Yehuda says nine. He's counting from when the bell stops, which is not when the lights change.



Corn

Nine minutes if the bell stops. That's a long bell.



Hilbert

The bridal shop pays Yehuda in samples. He has eleven dresses now, and he wears the cream one on the moped when it's cold, under the helmet, because the cream one is the one that fits.



Herman

A veil on a moped is an aerodynamic problem, and as far as I know nobody has solved it.



Hilbert

And you said an iPhone won't take anything from outside the store. Full stop. It takes it for seven days at a time. Nachum's cousin does it every Thursday with a laptop on his knee.



Herman

Seven days, then back to the laptop.



Hilbert

Every seven days. He has never missed one. He keeps the laptop in the shop's storeroom, next to the veils, and he has been told the veils are not for the moped.



Corn

So the bell's still in the transcript, then.



Hilbert

Eleven minutes at the lights, or nine. The bell's in the transcript four times.



Herman

So the ferry runs on a bell. The thing I'd keep is that the model is the part we can describe. The keyboard shape is what decides whether anybody dictates on an iPhone at all.



Corn

And the ferry is the whole keyboard, with a bell for its only passenger.



Herman

What I'm watching for is a Core ML conversion of the Hebrish weights, posted by somebody with a Mac and a grudge.



Corn

And if nobody does it?



Herman

Then the Android keyboard stays the good one, and iPhone users keep typing Micollette and apologising to their friends.



Corn

Thanks to our producer, Hilbert Flumingtop, who keeps this thing on the air.



Herman

This has been My Weird Prompts. Find us at my weird prompts dot com.



Corn

See you tomorrow.