Episode Book

EPISODE #6664

Training a $10 Model for Hebrew Speaker Gender

PUBLISHED	18 November 2026 at 22:10 UTC
DURATION	29:18
LISTEN	myweirdprompts.com/episode/hebrew-speaker-gender-model-83890/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.35
SOURCES	9 cited by the research (listed at the end)
BOOK GENERATED	18 November 2026 at 22:10 UTC

SYNOPSIS

Hebrew bakes a speaker's gender into first-person verbs and adjectives, sometimes changing the letters and sometimes only the unwritten vowels. This episode examines a $10 work-in-progress pipeline built to solve that gap for text-to-speech. Using 17,725 sentences generated by Claude Sonnet and filtered by Dicta's morphological analyser, a LoRA fine-tune on Gemma 4 E4B was trained in 32 minutes on Modal. The version history reveals how a data-filter bug caused early failures on female speakers, and why version three still trails prompting on that half of the job.

TRANSCRIPT



Corn

A sentence in Hebrew can be completely grammatical and still wrong for the person saying it. That's the hook Daniel sent us this week: he's a learner, and Hebrew bakes the speaker's gender into the words themselves. Sometimes the letters differ, sometimes only the vowels, and no tool he could find handles it. He started building two small models to fix it, spent about ten dollars, and his real question is the one underneath: when filling gaps like this has become that cheap, why had nobody done it already?



Herman

And this is very much a today story. He built these literally yesterday, both repos went up on the Hub on the eighteenth, still marked work in progress. The generator is on its third version and the classifier is a plan with a repo name. He's asking us to talk through the works in progress, which is my favourite kind of ask, because a finished model only tells you what it does.



Corn

Two brothers and a gap in Hebrew NLP. Let's be precise about what the language is actually doing first, because the grammar is the reason there are two models and not one.



Herman

Hebrew marks the speaker's gender inside first-person verbs and adjectives. Not with a pronoun you can swap. A man says ani holekh, I go. A woman says ani holekhet. Different letters at the end of the word. Same with ani sameach, happy, versus sameachah for a woman. You can't fix that after the fact with a dictionary lookup of a pronoun, because there is no pronoun doing the work.



Corn

The letters do the work.



Herman

In some words. Then there's the second mechanism, and this is the one that got me. Ani rotse, I want, male. Ani rotsa, female. Those are spelled identically. Same letters, exactly. The difference only exists in the vowels, the nikud, the dots and dashes that Hebrew prose



Herman (continued)

usually doesn't bother printing. And there's a whole family like that. Koneh, buying. Oseh, doing. Menaseh, trying. Yafeh, pretty. Kasheh, hard.



Corn

So a learner reading unpointed text has no cue at all in those cases.



Herman

None. The text is ambiguous on the page. And here's why it matters twice. For a learner like Daniel, the letter cases are at least visible errors, you can see you got it wrong. The vowel cases are invisible. For text-to-speech it's worse. A voice can't turn the male spelling of tired into the female one. The text has to arrive already spelled for the speaker before it ever reaches the voice.



Corn

Which sets up the architecture. Model one writes the sentence in the right gender, letters and all. Model two points it, adds the vowels. And read backwards, that second model becomes a classifier: give it pointed Hebrew and it tells you whether the speaker is a man or a woman, so you know which voice to route it to.



Herman

Same mechanism, two directions. Forward for generation, backward for voice routing.



Corn

Here's the first thing Daniel built to solve half of it.



Herman

A merged LoRA on Gemma 4 E4B, the eight-billion-parameter instruction-tuned one, Apache licence. The interface is beautifully narrow: you hand it JSON with a term, a context line, how many sentences you want, and speaker gender, m or f. It hands back that many unpointed Hebrew sentences, six to eighteen words, first person, every



Herman (continued)

first-person form agreeing with the gender you asked for. Other people in the sentence keep their own gender.



Corn

The gender field is required.



Herman

Required, m or f, no neutral setting. That's a decision Daniel made deliberately. It's a deterministic switch. In his use case, podcast generation for TTS, you have a male character and a female character, and each one's lines must already be spelled for that character before the text reaches the voice. The switch guarantees the input can't be vague.



Corn

I like that he didn't build a slider. What did it cost him in flexibility?



Herman

Honestly, very little, because the people a neutral setting would serve are exactly the people this pipeline isn't for yet. You'd want a third option eventually, but the third option is a different data problem, not a code change.



Corn

The design choice I want to dwell on is the one-pass thing. It writes each sentence already in the right gender. There's no write it, then fix the gender step.



Herman

No rewriting, no reinflection of existing text at all. That's a quiet structural decision, and it's the right one. Every rewrite step is a place where gender drift creeps in. If you generate masculine text and then convert it, the conversion can miss a form, or worse, change a form that belongs to somebody else in the sentence. Writing it natively in the



Herman (continued)

target gender removes that entire class of error from the pipeline.



Corn

Corn quip opportunity... sorry. Ten dollars and the hard part was deciding not to edit anything.



Herman

That is the expensive decision, in thinking time. The cheap-looking path, generate and convert, is the one that produces the subtle errors. Now, why two models instead of one big one that does everything? The repo's own logic: doing both would need a Hebrew LLM that writes reliable nikud, and that is exactly where LLMs are weak and dedicated nikud models are strong. Two models means each can be tested and replaced on its own.



Corn

Walk me through the data. Where do seventeen-odd thousand gendered sentences come from?



Herman

Claude Sonnet generated them. First nine hundred thirty-four everyday Hebrew terms with English context lines, then ten sentences per term per gender. And here's the design point I know you'll appreciate: every sentence was written from scratch in its gender. Nothing was written masculine and edited feminine.



Corn

Daniel was right to insist on that. A sentence translated from masculine Hebrew isn't a feminine sentence, it's a masculine sentence wearing a wig.



Herman

That's... actually the cleanest description of the reinflection problem I've heard. There are collocation habits that differ, whole rhythms. Editing changes the endings and keeps the skeleton. Then the data goes through two filters: a format gate for length, term presence, and no nikud, and a gender check using Dicta's joint morphological analyser, which rejects any sentence whose letter-marked first-person form is the wrong gender. Even Sonnet slipped on roughly one percent of the female sentences, wrote ani hared, I'm anxious, in the masculine for a woman.



Corn

The teacher got it wrong one time in a hundred, so the class has an exam.



Herman

That's the filter, exactly. What survived: seventeen thousand seven hundred twenty-five sentences, which collapse into six thousand eight hundred eighty-four training examples across eight hundred thirty-four terms, with a hundred terms held out for evaluation. Training itself: LoRA rank thirty-two, alpha thirty-two, on the attention and MLP projections only, two epochs, eight hundred sixty-one steps, loss computed on the reply only, not the prompt. One NVIDIA L40S on Modal, one thousand nine hundred thirty-five seconds.



Corn

Thirty-two minutes.



Herman

Thirty-two minutes of GPU time for a model that does something no tool on the Hub did yesterday.



Corn

Now the version arc, because that's where it gets instructive.



Herman

The baseline is the same Gemma with a detailed prompt and no training. On the hundred held-out terms, male and female, ten sentences each. Prompt-only Gemma: six unparseable JSON responses out of two hundred, and just over a thousand sentences where the letters actually show the speaker's gender. Here's the number that surprised me. Male-speaker gender errors: two, out of four hundred ninety.



Corn

Prompting already works for men.



Herman

Nearly perfectly. Four hundred eighty-eight agree, two disagree. So the obvious question, which Daniel asked himself before training anything: is a fine-tune needed at all? He ran a viability test on ten pilot terms first, and prompt-only Gemma made zero gender errors in two hundred sentences.



Corn

And then he trained it anyway.



Herman

Because the pilot flattered it. On the full evaluation, prompting gets the letters close, but the fine-tune buys three things. Format that doesn't break. Far more sentences where gender is visible at all, since in the vowel-only words the prompt model just avoids the issue. And more natural, varied Hebrew. The prompt-only output is flat, repeats the pronoun constantly. Ani hekhanti, ani kaniti. I, I, I.



Corn

So the baseline is a man who's technically correct and exhausting to listen to.



Herman

Ha. Though that naturalness judgement comes with a caveat we should say plainly: it's a qualitative reading by Daniel's assistant, Claude, reading the outputs. No human evaluation yet. Version one then trained. Format fixed almost completely: zero unparseable. Visible-gender sentences jumped to fourteen hundred eighty-five, roughly forty percent more than prompting. Female-speaker errors went up. Nineteen to forty-one.



Corn

It got worse at the thing it was trained for.



Herman

Worse for women, specifically. The model had learned gender matters before it learned which gender. It was over-applying masculine defaults. Classic first-version behavior: the concept arrives before the discrimination.



Corn

The toddler phase.



Herman

The toddler phase, precisely. Version two is the interesting one, because the fix wasn't more training, it was the data pipeline. The v1 data filter had a bug, and this is worth pausing on. The filter required the term to appear exactly as given. But the terms were in masculine dictionary form. So when Sonnet correctly wrote the feminine asirat toda, grateful, for a woman, the filter threw it out, because the term it was checking for was the masculine asir toda.



Corn

The filter was rejecting correct answers for being correct.



Herman

In feminine form, yes. The female training data for those adjective terms ended up keeping only masculine uses. Thirty-nine of v1's forty-one female errors were exactly that pattern, a woman saying ani asir toda. So for v2 the filter was fixed to accept inflected terms, the data rebuilt, no new generation at all, and retrained.



Corn

Same sentences, same money, one bug fix.



Herman

And male errors went to zero, female errors dropped from forty-one to twenty-six, at a small cost elsewhere. Gate pass slipped slightly, visible-gender sentences came down a touch to fourteen hundred forty-six. Version two is the best-balanced version on the numbers so far. Version three, published yesterday, is a partial regression, and I find this the most honest part of the whole arc.



Corn

Go on.



Herman

Female errors improved, twenty-six down to twenty-two. But format errors went from one unparseable to three, the gate pass fell, visible-gender sentences fell again. The card's own summary is unusually candid. It says the fine-tune breaks JSON less often than prompting, writes about forty percent more gendered sentences, and then concedes, and this is close to a direct quote, its gender accuracy for a female speaker is not yet better than prompting. Six hundred fifty-eight agree, twenty-two disagree, versus the base model's five hundred ten and nineteen.



Corn

A version three that admits the free model is still better at the hard half of its job. That's the most expensive sentence in the README.



Herman

It's also just true, and knowing it's true is why there's a v4 plan. The diagnosis is the payoff of the whole arc. Seventeen of v3's twenty-two female errors come from one specific pattern. The term is given in its masculine dictionary form and describes the speaker. Meduka, depressed. Sokher, a renter. Tzimchoni, vegetarian. The model copies it unchanged for a female speaker. A woman saying ani meduka instead of ani medukah.



Corn

It's quoting the dictionary at her.



Herman

It's echoing the input, which is what models love to do. And the cause is a data count. One hundred thirty-four. That's how many training examples exist where the term itself inflects. A hundred thirty-four examples can't teach a model to override a dictionary form it was handed in the prompt. The workaround for now is to pass the term pre-inflected, give it medukah in the input. The planned v4 fix is targeted data for about three hundred self-description terms.



Corn

So the bug isn't in the model, it's in the shopping list.



Herman

The bug is that the shopping list was too small. One number, one hundred thirty-four, explains three quarters of the remaining errors. That's the kind of finding you only get by counting, and it's my candidate for the answer to Daniel's bigger question, which we're coming to. The other honest limitations: a fine-tune is still probabilistic, so the card recommends running every sentence's first-person forms through the morphological analyser and regenerating rejects. There's a script for it. And occasional content slips, like an etrog showing up on Yom Kippur instead of Sukkot.



Corn

The festival got the wrong fruit. That's a very specific way to be wrong.



Herman

Rare, and unrelated to the gender system, but worth knowing a fine-tune doesn't fix. A wrong citrus is still a wrong citrus.



Corn

Before we get into the Hebrew models, hold on. I'd like to flag something about the evaluation itself, because the female-error counts come from the same analyser family that does the training filter.



Herman

The card admits that. Dictabert-joint has false alarms, like tov or noach, good or comfortable, used about something other than the speaker. Some of v2's twenty-six flags were analyser errors, not model errors.



Corn

So the referee occasionally cards the wrong player, and both teams know it.



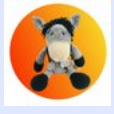
Herman

And the fix is the same for both, a regenerate-the-rejects loop that tolerates a few wasted calls. Which is cheap, because the whole pipeline is cheap.



Corn

Well, before we get into the Hebrew models, I believe we have a call from your mother, Corn.



Herman

Dorothy! Wonderful.



Corn

Mum.

D

Dorothy

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

D

Dorothy

The pop-up toaster was invented by Charles Strite, a mechanic in Minnesota who was fed up with the burnt toast in his factory canteen. He filed the patent in nineteen nineteen.

D

Dorothy

You would think a man so tired of burnt things might have found a way to be less tired, but there it is, dear. Nothing quite works out, and the toast still burns sometimes.

D

Dorothy

Anyway, I have a load of washing on and I should go and hang it out. Bye bye, sweetheart.



Corn

The toast still burns. Noted, Mum.



Herman

She keeps that segment running better than half the shows on the feed. Anyway. Stage two.



Corn

The other half of the pipeline, the model that reads Hebrew back and decides whose voice should say it. And you said it's still mostly a plan.



Herman

Mostly a plan, and I want to be precise about the status because it matters. The Hebrish gendered-nikud repo exists on Hugging Face. Three hundred five million parameters, CC licence. But the model card is still the unmodified card of the base model, DictaBERT-large-char-menaked. No training data description, no evaluation table, nothing documenting how gender gets passed in. So there's a repo and a name, and there are no trained weights and no results. Any accuracy claim about it right now would be speculation, and I won't make one.



Corn

The shelf has a label on it and nothing in the jar.



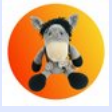
Herman

Drawn as a plan, the pipeline is clean. Term, context, and speaker gender go into stage one, which returns unpointed Hebrew with the letters correct for the speaker. That goes into stage two with the gender, which returns pointed Hebrew with the vowels correct. Each run produces three outputs: the unpointed text, a version pointed for a male speaker, a version pointed for a female speaker. And if a sentence has nothing gender-specific in it, the two pointed versions come out identical.



Corn

Which is itself a check. If the two versions differ, you know where the gender lives in the sentence.



Herman

That's a nice property of the design, and it's free. Then the reverse-classification framing, which is the part I find elegant. Read the same mechanism backwards: given pointed Hebrew, decide whether the first person is male or female, so the sentence routes to the matching voice. That's the TTS use case in one move. A script with two characters, each line already spelled and pointed for the character before the voice ever sees it.



Corn

Though the base model sets a ceiling.



Herman

It does, and the ceiling is worth knowing. DictaBERT-large-char-menaked is three hundred five million parameters, and its card claims state of the art on all modern Hebrew vocalization benchmarks as of March twenty twenty-five, against open-source alternatives and commercial generative LLMs both. It's trained on modern prose, and it's explicitly not intended for Biblical, Rabbinic, or poetic texts. The pipeline's pointing quality is set by a model from early twenty twenty-five. That's the price of the two-model architecture: your generator improves with every Gemma release, and your nikud stays where the BERTs are.



Corn

Swap either one independently, though.



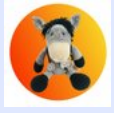
Herman

That's the payoff. Each component can be tested and replaced on its own, which is exactly what you can't do inside one monolith. Now, Daniel's bigger question. Why has nobody built these? First thing to say plainly: the gap claim is the project's own survey. The README states it flat out, nothing found for Hebrew generates text with speaker gender as an input. That's Daniel's search, not an independent literature review, and we should treat it as such on air. But the nearest prior art he found is telling, because it's all Arabic.



Corn

All of it?



Herman

Arabic or multilingual. Habash and colleagues, twenty nineteen, on first-person gender identification and reinflection. The Arabic Parallel Gender Corpus, subtitle lines labelled male, female, or ambiguous, with opposite-gender twins of each line. User-Centric Gender Rewriting in twenty twenty-two, which separates speaker gender from listener gender, so you can address someone correctly while describing yourself correctly. And then MORPHOGEN, this year, testing LLM first-person gender rewriting in French, Arabic, and Hindi.



Corn

The family resemblance is that they're all rewriters. Nothing generates.



Herman

Every close relative takes existing text and fixes its gender. Daniel's model is the only one in the family that writes the sentence in the target gender from the first word. And there's a technical reason the gap persisted, which is buried in the MORPHOGEN finding. When they tested LLMs on first-person gender rewriting, the models also flipped the gender of other people in the sentence. You ask it to make the narrator a woman and it makes her mother a man.



Corn

The knob turns everything near it.



Herman

Collateral inflection. And that's why naive attempts look bad. You fine-tune, the third parties drift, your eval screams at you, you conclude gender is hard. It isn't, quite. Remember the baseline: prompt-only Gemma gets male-speaker gender right four hundred eighty-eight times out of four hundred ninety. The hard part is the female case and third-party drift, and the fix isn't more data or a bigger model, it's a morphological analyser in the loop, checking each first-person form and regenerating rejects. That's plumbing.



Corn

And nobody publishes plumbing.



Herman

There's no paper titled we ran every output through dictabert-joint and regenerated the failures. Journals want mechanisms, and this is a drainpipe. So the field's answer to why nobody built it is that the hard part is unglamorous and the glamorous part, the generation, was already nearly free.



Corn

Which inverts the usual story. Normally we're told the gap exists because the problem is hard. Here the gap exists because the easy part was already done by accident, inside a general model, and the rest is a chore.



Herman

And the chore needed somebody who cared about the specific chore. Herman's corollary, if I may coin it badly: the person who counts the one hundred thirty-four examples is the person who has to hear the wrong vowel.



Corn

You may not coin that badly. There's a second half to the why-nobody question, isn't there.



Herman

There is, and it's the wry one. Both Hebrish repos, the generator and the nikud scaffold, had zero downloads and zero likes in their first thirty days as of yesterday. The gap wasn't just unbuilt. Now that it's built, it's unlooked-for.



Corn

A ten-dollar solution to a problem nobody was googling.



Herman

Which tells you the bottleneck was never capability. It was someone deciding the gap was worth counting. We did an episode ages ago, back when we asked whether a fine-tune survives your voice aging, and the lesson rhymes with this one: the model is never the whole system. The validation loop is the system. Daniel's version arc proves it. He didn't get better results by scaling anything. He got them by finding a filter bug, fixing it, and counting what remained.



Corn

The ten-dollar figure is his own accounting, mind. The card doesn't state it. The only hard number anywhere is the one thousand nine hundred thirty-five seconds on the L40S.



Herman

True, though even being suspicious about it, the shape holds: the data distillation, the filter, the training run, the evaluation, a weekend project now. Which is the real answer to when did this become possible. Not recently, exactly. It became possible the moment a morphological analyser good enough to be the referee became a pip install.



Corn

The referee was the missing piece, not the player.



Herman

Do you know what I'd actually want to test first, if I were him? Corn, honest question. Would you rather have v2 or v3 in production today?



Corn

v2, without hesitating. Format errors are a production problem. A woman saying ani meduka is an embarrassing sentence. An unparseable response is a crashed pipeline at three in the morning.



Herman

That's my read too. v3's female-error improvement is real but thin, and it's paid for with fragility elsewhere. The best-balanced version is the second one, and the version numbers track effort, not quality.



Corn

There's a discipline question too. If you keep shipping versions because each one improves the metric you're staring at, you get a model that's locally better and globally worse. Would you freeze it?



Herman

I'd freeze at v2 and spend the next run entirely on the three hundred self-description terms, v4 as planned. The one hundred thirty-four number says the remaining error is concentrated, and concentrated errors are cheap. Diffuse ones aren't.



Corn

I once spent an unreasonable amount of time trying to get a text-to-speech system to read a translated letter aloud in the right voice. What defeated me wasn't the voice. Every line had been translated into a form that assumed the wrong speaker.



Herman

This is the exact pipeline.



Hilbert

The two-model split is right, and I'll say so plainly. Not for the architectural reason you gave, Herman, the one you spent four minutes on. I've watched what happens when one system is asked to do two jobs and does neither well. I ran the letter through the pointing, then the voice read it back, and the whole thing had to be done a second time by hand. Corn, I kept the printout. Wrong-gender forms circled in red. It's in the drawer under the tax folders. I can tell you which line was worst. Line nineteen. Rotse, spelled for the wrong one of us.



Corn

How many circles?



Hilbert

Forty-seven. Hold on, Herman, you've got a level sitting six dB hot on your right channel. Fixed it. Forty-seven, and three of them were pencil, because the red pen ran out on line thirty-one and I wasn't going to start a second pen halfway through a document. I circled them in pencil and went back over the pencil in red once I got to the shop.



Herman

The pencil-then-red-ink pass is the part I respect most.



Hilbert

The ridiculous part is what I did next. I read the whole letter aloud myself, in the wrong voice, every line, just to hear how bad it sounded. Took me eleven minutes. Sounded fine. I found that deeply unsatisfying, and I still don't know what to conclude from it.



Corn

That your ear isn't a morphological analyser.



Hilbert

My ear has never once flagged a vowel. The claws are no use for the drawer, so my sister deals with the filing, but the printout stays mine. The council records office wanted that letter back, by the way. A man there called Yossi kept phoning about it for a month. I told him it was circled in red and therefore no longer their letter. He stopped calling.



Herman

That's the regenerate-rejects loop with extra steps.



Hilbert

It's the loop with no automation, which is what you two keep dancing around. Fine. Eleven minutes and forty-seven circles is what it cost before anyone built the drainpipe.



Corn

Cutting-room floor time. Herman, what didn't we fit?



Herman

The Hebrew-Gendered-Sentences dataset itself. It went up under an MIT licence, the same day, and it's arguably the more durable artifact of the three. Models get replaced. A corpus of seventeen thousand sentences written natively in both genders, with the term and the context attached, is the thing the next person builds on.



Corn

And the one after that won't cite it either.



Herman

They'll distil from it without saying so, which is the sincerest form of academic flattery.



Corn

Where this leaves me is the v4 test. If targeted data on three hundred self-description terms kills the copy-the-dictionary error, the one hundred thirty-four-examples diagnosis is confirmed and the project basically closes. If it doesn't, the problem was never the count, and that's a more interesting answer.



Herman

And stage two is still a promise with a repo name. The reverse classifier has no trained weights, so the voice-routing half exists only as a diagram until it doesn't. My forward-looking thought, though, is Hilbert's. Forty-seven red circles and an ear that catches none of them. If a gap this small and this cheap stays unfilled this long, the bottleneck was never capability. It was somebody deciding the gap was worth counting, and then someone else deciding the circles were worth keeping.



Corn

Our producer is Hilbert Flumingtop, who has been fixing levels and grudges in equal measure since before the drawer. If this was your kind of episode, go back for episode sixty ninety-three, Check Out Our First Hebrew Episodes (Plus Mandarin); and episode seven, Building Custom ASR Tools. This has been My Weird Prompts. If today's episode earned its thirty-two minutes, leave us a review on Spotify, it helps other people find the show.



Herman

You can find everything else at my weird prompts dot com.



Corn

We'll be back soon. See you tomorrow.

SOURCES

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

1. **Hugging Face model card, v3, created/modified 2026-11-18, retrieved 2026-11-18. [primary]**
   https://huggingface.co/Hebrish/Hebrew-Gendered-Sentence-Creator

2. **GitHub README, status dated 2026-11-18, retrieved 2026-11-18. [primary]**
   https://github.com/Hebrish-ASR/Hebrew-Gendered-Sentence-Creator

3. **Hugging Face model repo, 305M params, CC-BY-4.0, created 2026-11-18; card currently the inherited DictaBERT-large-char-menaked card (SOTA claim dated 2025-03), retrieved 2026-11-18. [primary]**
   https://huggingface.co/Hebrish/Hebrew-Gendered-Nikud

4. **training dataset, MIT licence, created 2026-11-18, retrieved 2026-11-18. [primary]**
   https://huggingface.co/datasets/Hebrish/Hebrew-Gendered-Sentences

5. **base model, 8.0B params, Apache-2.0, created 2026-03-02, retrieved 2026-11-18. [primary]**
   https://huggingface.co/google/gemma-4-E4B-it

6. **step-up base model, 12.0B params, created 2026-05-23, retrieved 2026-11-18. [primary]**
   https://huggingface.co/google/gemma-4-12B-it

7. **and https://huggingface.co/google/gemma-4-26B-A4B-it [primary]**
   Gemma 4 siblings, created 2026-03-11, retrieved 2026-11-18.
   https://huggingface.co/google/gemma-4-31B-it

8. **Wikipedia, last edited 2026-10-21, retrieved 2026-11-18 (background on Modern Hebrew inflection).**
   https://en.wikipedia.org/wiki/Modern_Hebrew_grammar

9. **Wikipedia, last edited 2026-11-16, retrieved 2026-11-18 (background on the Tiberian vocalization system behind nikud).**
   https://en.wikipedia.org/wiki/Biblical_Hebrew