Episode Book

EPISODE #6207

Two WERs, One Model: Inside ASR Eval Harnesses

PUBLISHED	22 October 2026 at 21:32 UTC
DURATION	22:35
LISTEN	myweirdprompts.com/episode/asr-eval-harnesses-explained-18423/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.19
BOOK GENERATED	23 October 2026 at 07:44 UTC

SYNOPSIS

You fine-tune a Whisper-style model or Parakeet on your own voice, run it twice, and get two word error rates several points apart. Neither number is a bug. This episode pulls apart the three layers of ASR evaluation — the scorer, the normaliser, and the harness — and shows why almost every WER you've ever read is really a statement about someone's regular expressions. We walk through jiwer and the Hugging Face evaluate wrapper, the Whisper English normaliser and the points it buys you, why multilingual normalisation can inflate scores by stripping vowel marks, and why code-switched Hebrew-English dictation has no canonical normaliser at all. Then it's on to the harnesses themselves: the Open ASR Leaderboard repo and its three tracks, NeMo's transcribe-and-evaluate pair, and why your RTFx will never match a published one. Plus a salesman who promises negative word error rates.

TRANSCRIPT



Corn

Same model, same audio files, two word error rates that sit several points apart, and neither number is a mistake. How does that happen? Daniel sent that one in, more or less, and it sits underneath his question about what actually drives an ASR evaluation. We covered the metrics a while back, in the episode on benchmarking custom ASR tools beyond the WER. He wants the machinery now: what are the harnesses and frameworks people run, given that he fine-tunes Whisper-style models and Parakeet on his own voice and on his Hebrew-English dictation.



Herman

The short answer to your opening question is that there are three separate layers and most people only ever see one of them. Layer one is the scorer, the code that aligns a reference sentence against a hypothesis and counts the damage. Layer two is the normaliser, which decides what the two strings look like before they get compared. Layer three is the harness, the whole apparatus that loads a dataset, runs the model, normalises, scores and writes a report.



Corn

And the two WERs differ because of layer two.



Herman

Nearly always. I'll defend that. But start at the bottom, with the scorer, because it's less glamorous than people think. Almost everything in the Python world sits on jiwer. It comes out of the Jitsi organisation, it's Apache licensed, and it computes word error rate, character error rate, match error rate and the word information measures, using a minimum edit distance with RapidFuzz underneath. The part you care about is that it gives you substitutions, insertions and deletions as separate counts, plus the alignment.



Corn

And the Hugging Face evaluate library?



Herman

A thin wrapper. The WER and CER metrics in evaluate import jiwer and call it. So a huge fraction of the WER numbers in model cards reduce to jiwer plus whatever the author did to the text beforehand.



Corn

Which is a lot of numbers riding on the phrase "whatever the author did".



Herman

That's the whole problem, yes. jiwer is a scorer, not a harness. It doesn't load audio, it doesn't run a model, and it doesn't know that the reference said "Mr." and the model said "mister". Version four changed how it treats an empty reference, by the way, which matters if your test set contains a clip that is only silence.



Corn

Which his will. Dictation has gaps in it.



Herman

It will. Hold that thought, because it comes back at the end. Now, the normaliser. Whisper ships an English text normaliser, and its steps are worth knowing because so many leaderboards inherit them. It lowercases. It deletes anything in brackets. It removes fillers, the hmm and um and uh. It closes the gap before apostrophes, expands titles so Mr becomes mister, strips commas between digits, drops periods that aren't followed by a digit, throws away symbols and diacritics apart from a few currency and percent signs, turns spelled-out numbers into digits, converts British spellings to American, and collapses whitespace.



Corn

So a model that writes "colour" and a reference that says "color" are the same word.



Herman

Under that normaliser, yes. And the Whisper paper itself says the normaliser cuts WER significantly on conversational sets like CallHome and Switchboard, because of contractions, and on WSJ because of numbers. I've only seen the next two figures second-hand, so take them loosely. A study of cockpit speech had a large Whisper going from roughly sixty-eight percent error with no normalisation down to roughly fifty-two with the basic one. And on FLEURS, the English gain was reported at around five points.



Corn

Sixteen points from tidying text. I could have done that with a find-and-replace and a nap.



Herman

Everyone could, which is the issue. Whoever wrote the harness chose the find-and-replace. The model didn't get any better.



Corn

Then the leaderboard number is partly a measurement of someone's regular expressions.



Herman

In part. And it gets worse once you leave English. The FLEURS multilingual work found the Indic-language gains were bigger and partly artefactual, because the non-English routine strips vowel signs, the matras. Strip the marks that distinguish two words and fewer errors get counted. The score improves, the model doesn't.



Corn

And Hebrew?



Herman

Hebrew is Daniel's problem, and the honest position is that Whisper's English normaliser is simply wrong for it. The spelled-number rules, the contractions, the British-to-American table, none of it applies. The basic multilingual one imports a symbol-and-diacritic



Herman (continued)

remover, and I haven't verified exactly what it does to a Hebrew string, so I wouldn't trust it blind. Then there's his real case, Hebrew with English words dropped into the middle, a product name, a repo name, an acronym. A normaliser built for one language mangles the other half of the same sentence.



Corn

Is there anything built for that?



Herman

There's a newer library called OpenWER, out of a paper from this summer, that does language-specific normalisation and compound-word handling. Against jiwer it reported an average reduction of about seven and a half points across fifty-two Common Voice languages, and about four against Whisper's normaliser. I can't tell you whether Hebrew was among the fifty-two, so I wouldn't lean on it for him. The takeaway is that the normaliser moves scores by points, and nobody's published a canonical one for code-switched Hebrew.



Corn

Which means whatever he writes will be as good as any.



Herman

As good as any, and better than any if it's the same for every model he tests. Consistency beats cleverness.



Corn

That's the most expensive sentence in speech recognition, and it's free. I'd rather he just never touched the text at all. Raw output against raw reference. Cleaner, no choices.



Herman

Try it. Your reference says "twenty-three", the model writes "twenty three" or the digits, and you've scored a perfectly correct transcript as wrong. The raw comparison is not neutral either, it just hides its choices in the formatting habits of whoever typed the reference. You can't skip layer two, you can only decide whether you're doing it on purpose.



Corn

That's a fair hit. I'll take the on purpose version. It's what lets him tell his own fine-tune from the base model, I suppose, rather than which one happens to write numbers the way his reference does. And that is why you should always build your own harness instead of trusting a framework you do not understand.

L

Larry

Folks, let me tell you about a satisfied customer of mine, a gentleman by the name of Gerald, no wait, it was Harold, Harold from a town so small the leaderboard only had one rung. Harold ran his own little evaluation harness on his kitchen table, and his word error rate came out at minus forty percent. Not forty. Minus forty. He was so far below zero the words started correcting him. Then he swapped in the Sclite framework and his fine-tune got so accurate it transcribed conversations from next Tuesday. That product is the Harness Buddy Two Thousand. Order one and your WER will go so low it files for unemployment.



Herman

Word error rate cannot be negative. It can exceed a hundred percent, because insertions count as errors and a model that babbles can produce more errors than the reference has words, but there is no way to land below zero. The floor is a perfect transcript. And sclite doesn't run a model, it only scores one, so you cannot swap it in as though it were a fine-tune.



Corn

Let it go, Herman. We were at the point where a harness is more than a scorer, so let's go and see what the big ones do.



Herman

Fine. The one everybody has heard of is the Open ASR Leaderboard, and the useful thing is that it's a repository, not only a webpage. It's Apache licensed, from Hugging Face, and the paper behind it compares eighty-six systems across twelve datasets in its latest revision. Earlier versions said sixty-plus systems and ten datasets, so the numbers move with the version you read. There are three tracks: English short-form, English long-form, and multilingual short-form.



Corn

And what does running it look like?



Herman

Each model family has its own run_eval script, and a shell script drives it, so there's a Whisper one, there are ones for the NeMo models, and so on. They load the benchmark datasets from Hugging Face, run inference, write predictions to a JSONL file, and print two numbers. WER, and a speed figure called RTFx, how many seconds of audio you get through per second of compute. There's a normalizer folder in the repo, and I haven't read what's in it, so I can't tell you its rules.



Corn

Two numbers. One of them I'm guessing depends on the machine.



Herman

Very much so. The official runs go through Hugging Face Jobs on a single H two hundred with a hundred and forty-one gigabytes of memory, in a fixed Docker image. That's how they keep the speed column comparable. It's also why Daniel's rig will never reproduce a published RTFx. Same code, different GPU, different batch size, and the figure is just a different figure. His WER should be reproducible if he matches the text handling. His RTFx is a statement about his hardware.



Corn

One detail I liked: even the leaderboard cheats a little to make long audio fit.



Herman

The earnings call set is chunked, and the predictions get concatenated back together per call before scoring, so the chunking doesn't distort the count. Small thing, but it's the sort of choice that's invisible unless you read the harness.



Corn

Herman, I want to ask you about the headline finding in that paper. It's the one I half-remember: the best average accuracy goes to one kind of model, and the fastest goes to another.



Herman

The paper says Conformer encoders paired with transformer decoders get the best average WER, and the CTC and TDT decoders offer the best RTFx. Parakeet is a TDT model, the token-and-duration transducer, so Daniel is sitting on the fast side of that trade. Whisper-style models with the autoregressive decoder sit on the accuracy side, and pay for it in speed. If you were choosing for dictation, which side would you take?



Corn

Speed, and it's not close. If I'm talking and waiting for text, a point of WER is cheaper than a second of lag. My worry is the other direction, whether the slower model is quietly better on his accent.



Herman

That's what the leaderboard can't tell him. It's an average across other people's audio. The gap between the two families could be bigger or smaller on a single speaker with a mixed vocabulary, and the only way to find out is to run both on his files. Which leads neatly to how NeMo does it, since that's the toolchain for Parakeet.



Corn

Go on, then. What's the fiddly bit?



Herman

Before that, a question for you, Corn. When you test a fine-tune, do you trust the first run, or do you run it twice?



Corn

Twice, and I'd want a nap in between to see if it still says the same thing. Greedy decoding should repeat, anyway.



Herman

It should, and that's a fair test of whether he's pinned the settings. Back to NeMo. It's two scripts that work as a pair. One transcribes, one evaluates, and the evaluator depends on the transcriber in the same folder. Everything runs off a manifest, a JSONL file where each line points to an audio file and carries its duration and reference text. The transcribe script runs your checkpoint over the manifest and writes predictions back as another manifest, with a predicted-text field added. The evaluator can then score it. There's an option to score only, so if you already have a manifest with predictions in it, from any model at all, you can skip inference and just compute errors. There's another to score characters instead of words.



Corn

Characters would matter for Hebrew.



Herman

It might. Hebrew words attach prefixes, so a single missing letter turns one word into a wrong word, and word error rate gets harsh. Reporting both is cheap. The decoding strategy is also a setting you pass, greedy batch for the transducer models, with beam-style options as well, and it changes the score. Which is why a number without its decoding settings next to it is half a number.



Corn

So the harness in NeMo is basically a manifest with opinions.



Herman

A manifest with a scorer bolted on, yes. It's a good one to borrow from, because the manifest idea is the right shape for a personal setup.



Corn

What about the commercial lot? Everyone's got a leaderboard these days.



Herman

Artificial Analysis is the one worth knowing, because it's a deliberately different philosophy. Their second version of the AA-WER metric is a weighted blend of about eight hours of audio. Half the weight is a proprietary set of voice-agent speech they recorded themselves, a few hundred samples. A quarter is cleaned European Parliament audio, and a quarter is cleaned earnings calls. They call the vendors' hosted APIs directly and weight by time.



Corn

A private test set that nobody can train on.



Herman

That's the appeal and the cost. Nobody can leak it into training, and nobody can inspect it. You trust their normalisation, their cleaning, their weights. And it only evaluates what has an API, so a checkpoint on your own disk is outside it. There's also a figure for the top of the open leaderboard floating around from a third-party aggregator, a Granite speech model at about five and a third percent mean WER, but it's second-hand and I haven't verified it, so I'd leave it alone.



Corn

Then there are the old ones.



Herman

Sclite, the NIST scoring toolkit. It's still the reference scorer in the field, it's been around for decades, and it produces the alignment reports people used to print out. Newer packaging of it keeps appearing in Linux distributions, so it's alive. And MeetEval, from the group at Paderborn, for when more than one person is talking. It does concatenated minimum-permutation WER, the cpWER, and a time-constrained version that cares whether the words land in the right stretch of time, with a collar. It reads the standard transcript formats, and it makes an HTML view of the alignments. Daniel only needs it if he ever scores multi-speaker material.



Corn

He mostly dictates alone, so that one's a spectator sport for him.



Herman

For now. Which gets us to the part he can actually use. A minimal harness, built on the three layers. Corn, I want to try it as a sequence and have you tell me where I've overbuilt it.



Corn

Go on. I'll heckle.



Herman

One. A held-out manifest made from his own recordings, audio that never appeared in training, tuning or validation of any model he's testing. Two. One normaliser, written once, with Hebrew handling and a rule for English words inside Hebrew sentences, applied identically to every model's output and to the references. Three. jiwer for the score, but not only the aggregate: per utterance, with substitutions, insertions and deletions saved as separate counts to a JSONL file, one line per recording. Four. Run the same recipe against the base model, the fine-tune, and one off-the-shelf competitor.

Corn

Where have you overbuilt? Honestly, nowhere, but I'd cut step three if he only wants a single number.



Herman

He doesn't want a single number, though. The aggregate tells him a fine-tune improved. The per-utterance file tells him why. If the deletions went up, the model is dropping words. If the insertions went up, it's inventing them. A bare WER hides which of those happened, and they mean very different things for dictation. Now I have one for you. If you had to pick which failure is worse in dictation, a dropped word or an invented one, which is it?



Corn

Invented. A dropped word makes me reread the sentence. An invented one goes out in an email with my name on it.



Herman

That's why the split matters, then. A single WER weighs them the same.



Corn

Fine, I'm persuaded that the file's worth it. Though I'd still like to see him defend the manifest, because "held-out" is doing a lot of work in your first step. He'll record on the same microphone, in the same room, reading from the same sorts of notes.



Hilbert

Sclite is one program in the toolkit. The toolkit is SCTK. Herman, you said hidden variable more times than the normaliser deserves. How held out is held out? If the sample is out of the building but your cousin holds the key, is it out?



Herman

Out means the model never saw it and nobody tuned the settings by looking at it. Cousins are fine, as long as the cousin doesn't edit the training set.



Hilbert

Borough Weights and Measures had a ruling. Test material had to be kept out of the premises where it was tested. I have the letter, signed by a Mr. Alder Pomfret, Deputy Examiner of Counted Goods. He required four hundred tapes. Three hundred and ninety-one. He didn't count the ones with a cough on them, that was his rule and it's in paragraph six. The tapes went into a locker at the coach station and Mr. Pomfret held the key. He wasn't permitted to listen to them either, so he never did.



Herman

That's a clean separation. Whoever holds the test set can't peek at it, and the people tuning can't reach it. The usual leak is someone glancing at the examples to see why the score looks wrong, and then adjusting the model to suit.



Corn

The obvious difficulty is that nobody could check the tapes were still in the locker.



Hilbert

Two pounds ten a week for the locker. The letter allowed it as an expense.



Herman

Which is cheap for a test set that stays honest. Anyway, the rest of the recipe is where the dictation case gets particular. Five. The silence and hallucination cases, because a score on speech tells him nothing about what the model does when nobody is talking. Put in clips of silence, room noise, a breath, a keyboard. Whisper-style decoders are known to produce fluent text out of nothing, and I believe a TDT decoder is less prone to that since it works frame by frame, but I haven't seen the leaderboard measure it.



Corn

That's the jiwer wrinkle you mentioned. The reference is empty and every word the model says is an insertion.



Herman

That's the case, and in the newer version it's handled differently from the old one, so he should check what his installed version does and not assume. Six. Record the decoding settings next to the result. Beam size, greedy or not, chunk length, any temperature fallback on the Whisper side, the model revision, the normaliser version. Seed is a setting too. The number is the output of all of it, and in three months he won't remember which run was which.



Corn

That's the step everybody skips.



Herman

Because it's the only one that feels like paperwork. Last piece, and this is the answer to why bother. The leaderboard is mostly English, mostly read or broadcast speech, with an earnings-call set and a parliamentary one for variety. His Hebrew-English dictation sits outside anything it contains. A model eight places higher on the leaderboard could be worse on his voice, and there's no way to know except to run it.



Corn

So the leaderboard tells you what's worth trying and your own set tells you what to keep.



Herman

Close enough that I won't argue. The one thing I'd add is that his own set also isn't a fixed object. As his vocabulary shifts, new repo names and new Hebrew slang, the held-out set should pick up a few fresh recordings, and he should freeze a copy each time so the old results stay comparable.



Corn

Misconception time. I think the one I'd name is that the leaderboard number is the model's accuracy, as if it were a property of the model, like its weight.



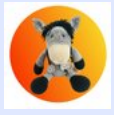
Herman

It's a property of the model, the data, the text handling and the decoding, all at once. Change any of them and the number moves, and the normaliser alone can move it by points. A published WER is a measurement of a setup, and it only transfers to someone else's setup if they copy the setup.



Corn

Which in practice means that Daniel's best comparison isn't against a table, it's against last month's run of his own harness. A frozen test set kept somewhere you can't casually reach, even if it costs you a locker, and every model run through the same recipe.



Herman

Fair enough, and it ends up as the thing that makes the next fine-tune worth doing at all.



Corn

Thanks to Hilbert Flumingtop for producing the show, as always.



Herman

This has been My Weird Prompts. You can find every episode at my weird prompts dot com.



Corn

See you tomorrow.