Episode Book

EPISODE #6686

Fine-Tuning Voice Activity Detectors for Your Own Room

PUBLISHED	19 November 2026 at 19:11 UTC
DURATION	21:33
LISTEN	myweirdprompts.com/episode/fine-tuning-voice-activity-detectors-81133/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	omnivoice
PIPELINE	V5.36
BOOK GENERATED	19 November 2026 at 22:18 UTC

SYNOPSIS

Most popular voice activity detectors can be fine-tuned to ignore your own coughs, breaths, and room noise, while WebRTC VAD is a compiled dead end. This episode breaks down which models—including Silero, pyannote, and NVIDIA's NeMo Frame-VAD—actually ship training recipes, and why open weights on models like TEN VAD and FireRedVAD don't always mean easy retraining. It also covers threshold tuning, Silero's noise loss dial, and how to label tricky edge cases like coughs mid-sentence.

TRANSCRIPT



Corn

Daniel wants a voice activity detector that knows his voice, his microphones and his room, so it stops firing on sneezes, coughs, breaths and the odd clunk. He's asking whether you fine-tune an existing detector for that, or train a fresh one because these models are so small. And the answer to the first half is better than he expected: most of the popular detectors can be fine-tuned, a couple ship the script to do it, and exactly one is a dead end. So the real question is which ones, and what the recipe looks like.



Herman

The dead end first, because it's quick. WebRTC VAD, the old one baked into every browser. It's a statistical test per frequency band with Gaussian mixture constants compiled in. The aggressiveness setting, zero to three, just picks between tables of constants. There's no training interface at all. Only its noise estimate adapts while it runs. That's from reading the source, not from any official statement, but I'm fairly confident.



Corn

So its one adjustable thing is how rude it's allowed to be.



Herman

Pretty much. Now Silero is the other end. The repository has a tuning folder with an official fine-tuning script, a config file, a threshold search and an example data file. It's the most direct answer to Daniel's question that exists.



Corn

We did the bake-off between detectors in Choosing the Right Voice Activity Detector, so I'll skip that. Most people run Silero as a frozen file, though. I'd have guessed pyannote is out, since it's a diarizer.



Herman

You'd have guessed wrong, and so would I a year ago. Pyannote's segmentation model is trainable, with an official route, plus a Hugging Face library called diarizers that fine-tunes it. It's heavier than a pure detector, because it labels who is speaking frame by frame, so you'd be using it for speech versus silence as a side effect. NVIDIA's NeMo models,



Herman (continued)

MarbleNet and the newer Frame-VAD, are trainable too, and there's a tutorial notebook for training and tuning thresholds. I haven't worked through that notebook, so I can't tell you the file format it wants. TEN VAD and FireRedVAD are the grey zone. Both have public weights and inference code. I found no public training code for either. You could fine-tune them if you rebuilt the training loop yourself, but nobody hands you one.



Corn

So "open" and "trainable" are two different words on those.



Herman

Open weights means you can run it. It doesn't mean you can steer it. And FireRed is the one making the loudest claims, which are self-reported. On their own benchmark the false alarm rate is about two point seven percent against Silero's nine point four.



Corn

Their benchmark, their win. I'm shocked.



Herman

That's why I'd run it on my own audio before believing any of it. Which is also my first piece of advice for Daniel, by the way. Before training anything, check what the stock model already does with his sneeze.



Corn

Why would it be fine? A sneeze is loud and it has a voice-ish onset. I'd have said that's the worst case.



Herman

It might misfire for exactly that reason, but look at what NVIDIA put in. The Frame-VAD version two model is tiny, about ninety-one thousand parameters, one prediction per twenty milliseconds, and its training data deliberately includes coughs, laughter and



Herman (continued)

breathing as non-speech. They pulled cough recordings from datasets built for detecting respiratory illness, plus a corpus of vocal sounds, coughs and laughs and such. The point was to cut false positives. So that model has seen the thing Daniel is worried about.



Corn

Does it have his particular sneeze, though?



Herman

Nobody has his sneeze. But it's cheap to test, and it might be fine out of the box. Then the next rung is just settings. Threshold, minimum speech duration, padding. Higher threshold keeps only the confident frames. A minimum duration of a quarter second is a default I've seen quoted for Silero, which knocks out clicks and short bursts, though a good sneeze can outlast that.



Corn

And the padding?



Herman

Padding is where it bites. People add a few hundred milliseconds either side so word onsets and endings don't get clipped. NVIDIA's own example comment says that padding preserves breaths and onsets. Which is the opposite of what Daniel wants. So you can turn the padding up and the breaths come right back.



Corn

So every fix for clipped words is a fix for breathing too.



Herman

At the edges, yes. Hmm, that's why the next rung up is hysteresis. Two thresholds, a higher one to start a segment and a lower one to keep it going. Silero's tuning code searches those for you by maximising accuracy on a validation set.



Corn

What's the point at which you stop fiddling and train?



Herman

When you've got a pile of recordings where the model is confidently wrong in a consistent way. If your own cough scores point nine, no threshold between point nine and the real speech will help, because real speech scores point nine too. You have to move the scores, and that means the weights.



Corn

Walk me through the Silero script. What do I actually feed it?



Herman

A data frame saved in a format called feather. Each row has a path to an audio file, sixteen kilohertz wav or opus preferred, and a list of speech segments as start and end times in seconds. The readme says to aim for labels accurate to about thirty milliseconds. Then there's a config: a switch for the eight kilohertz or sixteen kilohertz version, learning rate, batch size, epochs, a probability of augmenting each clip, and one called noise loss.



Corn

What does noise loss do?



Herman

It's the weight on the non-speech chunks. Speech chunks have weight one. So if you set it higher, the model gets punished more for firing on something that isn't speech. Which is precisely the dial for a cough problem.



Corn

One number. That's the whole hard-negative strategy?



Herman

As far as the code shows, yes. The script saves the best checkpoint by validation ROC AUC, as a TorchScript file you can drop in where the original was. And from reading the utilities file, what trains is the small recurrent decoder on top of the pretrained front end. I'll say that's my reading of the code. I haven't seen the maintainers confirm whether the encoder is frozen.



Corn

So it's less "fine-tune a neural network" and more "retune the last stage".



Herman

Which is why it's quick, and why you don't need a mountain of data. The decoder is the part that decides what counts as speech. The front end knows what sound looks like.



Corn

What about the augmentations? Does it throw coughs at the model for me?



Herman

No. That's the catch. The script applies, with some probability, between one and three transforms out of fifteen. Gaussian noise, a lossy codec, band-pass and high-pass filters, equalisation, clipping, aliasing, a room simulation, pitch shift. Those are mostly about the channel: the microphone, the codec, the room. It doesn't mix in noise corpora and it doesn't add gain changes.



Corn

So it imitates a worse microphone but not a worse person.



Herman

Fair. The coughs and sneezes have to come from your own labelled data, or from a dataset you mix in yourself. And that leads to Daniel's data question, which is the part where I'm going to give my reasoning, not a published number. I haven't seen anyone



Herman (continued)

publish hours-needed for the Silero script. My guess for a decoder-only retune is single-digit hours, with the weight on variety, not volume.



Corn

Variety of what? I only have one mouth.



Herman

Variety of conditions. Different distances from the mic, sitting and standing, a laptop mic versus a headset versus a lapel, the room with the fan on and the fan off. And then the nasty stuff, labelled as nasty. Twenty minutes of you sneezing in different postures is worth more than two hours of you talking in the same chair.



Corn

I'm not sure I can sneeze on demand. It's not a skill I've practised.



Herman

Sniff pepper. Or record for a week, and let the cough happen on its own. Whatever the detector triggers on in real life gets saved, and those clips are your negatives. That's the nice property of this problem, the errors label themselves for you.



Corn

Hold on, how many microphones, though? Do I need five?



Herman

You need the ones you'll actually use. If it's one headset in one room all day, one microphone is the right number, and the model can overfit to it on purpose, which is what he's after. If you want it to survive a switch, add each device you'll really use, and then lean on the augmentations for the rest, the filters and the room simulation. I wouldn't buy five microphones to train a detector.



Corn

What happens if you overfit to your own mic too hard? It stops working the day the headset dies.



Herman

That's the cost, and it's a real one. A cheap hedge is keeping the original file alongside your tuned one, and checking which of the two is wrong more often on a new device.



Corn

The labelling. Silero says thirty milliseconds, and I don't have a stopwatch pointed at my own voice. Who labels it?



Herman

There's a whole discussion in the project's tracker about this. Someone fine-tuning asked how to label audio, and the maintainer said to split the recording into separate chunks wherever there's a pause. Another contributor said no, mark the pause as non-speech inside the same file. The person asking used Label Studio for it.



Corn

Two schools, then. Cut the audio or mark the audio.



Herman

And I'd go with marking, because cutting changes what the model hears at a boundary. It never sees a sentence ending naturally and trailing off into room tone. But either way, hand labelling at that precision is slow, so the cheap route is alignment. You take speech where the words are known, run a forced aligner, which gives you word times, and everything outside the word times is non-speech. One recent dataset, LibriVAD, built its frame labels exactly like that from LibriSpeech with the Montreal Forced Aligner.



Corn

Which only works when you have a transcript.



Herman

Or a good enough transcript from a speech recognition model, which for your own voice is probably very good. My own suggestion, and I want to flag this as mine and not something I've seen published, is that you could also let a bigger detector pre-label your recordings, and then you go through and fix only the clips where it argues with itself. I haven't seen anyone report doing it.



Corn

The labeller labels the labeller's mistakes.



Herman

Only the ones you check. Anyway. LibriVAD is useful for another reason. It mixes in nine kinds of noise at six levels of signal to noise, from minus five to twenty decibels, and the finding that stuck with me is that more training data helped the out-of-distribution test more than the in-domain test.



Corn

So training on more stuff made it better on stuff it hadn't heard, and not much better on stuff it had.



Herman

Which is a point for variety again. And here's where I'd slow down on the labelling, because the labels are the part people rush. If your negatives include a cough that sits directly against the last word of a sentence, and you mark the whole thing as speech because the word and the cough run together, you've taught the model that coughs are speech when they're near speech.



Corn

Which is most coughs. People cough while talking.



Herman

Yes, and that's the real difficulty, not the isolated sneeze in a quiet room. The isolated one is easy. The hard ones are the ones that happen in the middle of a sentence you want kept, or at the edge of one. So you decide the policy, label to the policy, and stay consistent. A model can learn nearly any rule you like, but it can't learn a rule you applied differently on Tuesday.



Corn

I'd have said consistent labelling is the one thing I'm bad at. Slowly, but badly.



Herman

Then take a smaller set and be strict about it. Two hours labelled the same way beats ten hours labelled by three moods.



Corn

Variety, variety. You've said it three times and I still own one mouth.



Herman

Then here's the trick for owning one mouth. You've been assuming Daniel builds the dataset from his own voice alone, but there's mixing. You take clean speech, whether it's his or a public set, and mix in cough and laughter clips at random positions, and the labels come free because you know where you put the cough.



Corn

Synthetic sneezes.



Hilbert

Hilbert's Anteater Trivia. Giant anteaters spend only about a minute at each nest, leaving before the soldier ants or termites arrive in force. I have spent longer than that deciding whether to attend a meeting.



Corn

Moving on.



Herman

There are public sources of those negatives: the vocal-sound collection and the two cough corpora I mentioned are the kind of thing NVIDIA mixed into Frame-VAD. The sound-event research goes further. There's a line of work where a big model trained on the five hundred and twenty-seven event classes of AudioSet produces soft frame labels that a tiny binary model then learns from. Coughs and laughs are classes in there already, so it's a ready-made source of hard negatives. I'm not sure of the details on that one, so treat it as a direction, not a recipe.



Corn

Okay, so say I've got my recordings, my cough clips, my labels. How long does the training itself take? Am I renting a GPU for a week?



Herman

No. Not even close. The only comparison I have is the pyannote side. The diarizers library fine-tuned a segmentation model on a Japanese call dataset in about five minutes on one twenty-four gigabyte card, and the diarization error fell from twenty-five percent to eighteen. That's a heavier model than a pure detector, and it's a different task, but it tells you the order of magnitude. The decoder in Silero is smaller than that. It should be minutes to an hour, not days, and I'd be surprised if it needed more than a consumer card.



Corn

Then why does everyone act like training is a project?



Herman

Because the model isn't the project. The data is. Would you label your own sneezes, Corn, or would you rather never hear them again?



Corn

I'd label them slowly and resent every one. But I'd sooner do that than run two models. Why not leave the detector alone and put a tiny classifier behind it, cough or not cough, on any chunk it flags?



Herman

That works, and plenty of people do it. The catch is latency and plumbing: a second model to ship and a decision that waits on it. Fine-tuning folds the knowledge into the thing already running. But if you're stuck with one of the grey-zone models, a second-stage filter is the escape hatch.



Corn

So that's where the filter earns its place.



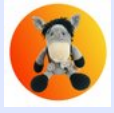
Herman

Or you switch to a trainable one, and size might decide it. Silero's file is about two megabytes. TEN VAD's library is three hundred and six kilobytes, and its licence is Apache with additional conditions, which I'd read before building anything on it.



Corn

Training from scratch, then, since that was the other half of the question. When does that make sense?



Herman

When you need something none of the existing ones give you. A different frame rate, a licence you can live with, or a size smaller than what exists. Otherwise you're re-learning what sound looks like, and the pretrained front end already knows that. Training from



Herman (continued)

scratch means you need the thousands of hours the big ones used. Frame-VAD's data was about twenty-six hundred hours of real speech, a thousand synthetic and three hundred and thirty of noise.



Corn

And Daniel has a single voice and a spare weekend.



Herman

And the model can be small precisely because it was trained on a lot. The smallness is the end product, not the starting point. There's one place where small and from scratch does work, though, and it goes the other direction from what Daniel asked.



Corn

Go on.



Herman

Personal VAD, from Google. It doesn't just say speech or not speech. It labels each frame as non-speech, speech from the target person, or speech from someone else, conditioned on an embedding of the target speaker's voice. The network is a small two-layer recurrent model, about a hundred and thirty thousand parameters, and the paper says it quantises down to about a hundred kilobytes.



Corn

So that's the "only my voice" answer.



Herman

Yes, but it's built for the other person in the room, not for the sneeze. It rejects somebody else talking. It doesn't claim to reject a cough, and I'd be careful to say it isn't a cough filter. Though it's trained from scratch, on a big pile of multi-speaker and noisy data, not on one person's voice.



Hilbert

A hundred and thirty. Eight-bit, not a hundred. Herman, you keep calling them rungs, and every one is the same height. Anyway, this man on the train from Crewe to Euston, across from me with a handheld recorder, the voice-operated kind, a switch on the side marked VOX. It dropped the front off every sentence. So he coughed before each one. Little cough, then dictate. The recorder woke on the cough and caught the sentence whole.



Herman

So the cough was the wake word.



Hilbert

Corn, the second channel is peaking, pull it down two. Two hours to Euston and not one sentence without it. You teach that recorder his cough is not speech, you've fixed the sneezes and stopped the man dictating. He'd have sat there coughing at it. I still cough before I hit record, to find where the file starts.



Corn

He'd need a different cough.



Hilbert

I bought it off him on the platform. Forty pounds. He wanted fifty.



Herman

A noise that's meant to mean something can't go in the negatives, and that goes for a click or a throat-clear as much as a cough. So write down what each noise is for before you label anything. Then the evaluation changes too. You don't want one number. You want a held-out set, recorded in your actual room, with the sneezes and coughs and the real speech in separate piles, and you look at two things on it.



Corn

Which are?



Herman

How often it fires on the negative pile, and how often it misses on the speech pile, both at the threshold you'll deploy. NVIDIA's models and Silero's tuning both report ROC AUC, which is nice for picking a checkpoint, but it doesn't tell you what happens at your threshold. The FireRed card reports false alarm and miss rate at a fixed threshold, which is closer to what you'll live with.



Corn

And you do the threshold search last, after tuning.



Herman

After. Silero's script ships the search for that. The sequence is the ladder: first test the stock model on your sneeze, then settings, then hysteresis, then the decoder retune on your recordings, and from scratch only if one of those three needs, a frame rate or a licence or a size, can't be met.



Corn

And the holdout has to be recorded on a different day from the training clips.



Herman

Yes. If the cough you test on is from the same afternoon as the cough you trained on, you've mostly tested whether it remembers the afternoon.



Corn

Anything that nearly got cut?



Herman

There's a package on the Python index called open voice activity detection, from last May, which is a re-implementation of the Silero-style inference, and it's licensed under a share-alike Creative Commons licence. Silero itself is labelled MIT, though the readme's licence badge and its visible text didn't quite agree when I looked. So before you wrap a community port into a product, check which of the two you're actually holding.



Corn

Daniel's version is short. Run the stock model on a recording of a bad morning, and decide what each noise should mean before you train any of them to mean nothing.



Herman

Thanks to Hilbert Flumingtop for producing, as always.



Corn

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



Herman

See you tomorrow.