Podcast Transcript

EPISODE #332

Who’s Talking? The Tech of Speaker Identification

Published January 28, 2026 • Runtime: 27:07

https://myweirdprompts.com/episode/speaker-identification-diarization-tech/

EPISODE SYNOPSIS

Tired of manually labeling who said what in your meeting transcripts? In this episode, Herman and Corn explore the technical bridge between speaker diarization and true speaker identification, diving into cutting-edge tools like Pyannote and Picovoice. They discuss how mathematical voice embeddings and "digital fingerprints" are revolutionizing how we process audio, making it easier than ever to programmatically identify known speakers even in noisy environments.

DANIEL'S PROMPT

Daniel

I'd like to talk about voice identification and diarization. My wife and I record our weekly apartment meetings and I’ve been using Gemini to transcribe them, but I’m looking for more advanced tools. What are the current tools available for reliable and programmatic diarization based on actual audio samples of known speakers, especially for recurring voices in a conversation?

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts. I am Corn, and I am sitting here in our living room in Jerusalem with my brother.

Herman

Herman Poppleberry, ready to dive into the deep end of the audio pool. And we have a really practical one today. Our housemate Daniel sent us a voice note about something he and his wife Hannah are doing. They have been recording their weekly apartment meetings, which sounds incredibly organized by the way, and they are looking for a better way to handle the transcription and specifically the voice identification.

Corn

Right, Daniel mentioned he is currently using Gemini to transcribe the meetings, but he is running into that classic wall where the AI can tell different people are talking, but it does not necessarily know who is who without being told every single time. He is looking for a programmatic way to use actual audio samples of known speakers to identify them automatically.

Herman

It is the difference between diarization and speaker identification, and it is a fascinating technical challenge. Most people use those terms interchangeably, but in the world of audio engineering and machine learning, they are distinct processes that often get bundled together.

Corn

I think that is a great place to start. Before we get into the specific tools like Pyannote or Picovoice, let us break down that distinction. Because if you are searching for a solution, you need to know what you are actually asking the machine to do.

Herman

Exactly. So, speaker diarization is basically the process of partitioning an audio stream into homogeneous segments according to the speaker identity. In simpler terms, it is the machine asking, who spoke when? It marks speaker zero, speaker one, speaker two. It does not know that speaker zero is Daniel. It just knows it is the same voice that spoke at the two minute mark and the five minute mark.

Corn

And then speaker identification, or speaker recognition, is the next layer. That is where you take those segments and compare them against a known profile or a voice print to say, okay, this segment matches the mathematical representation of Daniel Poppleberry.

Herman

Right. And what Daniel is looking for is a system where he can provide a few seconds of his voice and a few seconds of Hannah's voice as a reference, and then have the software automatically label the transcript with their names every week without manual intervention.

Corn

It sounds like a simple request, but historically, this has been one of the hardest problems in speech processing, especially when you have what they call the cocktail party effect, people talking over each other or background noise in an apartment. But the technology has really leaped forward in the last year or two.

Herman

It really has. If we look at how this works under the hood, it is all about embeddings. When a model hears a voice, it converts that audio into a high dimensional vector, basically a long list of numbers that represents the unique characteristics of that voice, the pitch, the resonance, the cadence.

Corn

So it is like a digital fingerprint, but for sound.

Herman

Very much so. And the goal of these modern models is to make sure that two different clips of Daniel's voice produce vectors that are very close to each other in that mathematical space, while a clip of Hannah's voice produces a vector that is far away.

Corn

So, Herman, if Daniel wants to build something programmatic for his apartment meetings, what are the actual heavy hitters in the tool space right now? I know you have been tracking the open source world pretty closely.

Herman

If you want to go the open source route, and you are comfortable with a bit of Python, Pyannote audio is the absolute gold standard right now. It is an open source toolkit built on top of PyTorch, and it is incredibly modular. As of late 2024 into 2025, their 3.1 version remains a benchmark that most researchers use.[1]

Corn

I have seen Pyannote mentioned in a few of the technical forums. What makes it the leader? Is it just the accuracy, or is it the way it handles the pipeline?

Herman

It is both. Pyannote does not just do one thing. It has separate models for voice activity detection, which is just identifying if anyone is talking at all, then speaker change detection, and finally the embedding extraction. The cool thing for Daniel's use case is that you can use Pyannote to extract an embedding from a clean ten second clip of his voice and save that as a reference file. Then, when he processes a new meeting, the system can compare every speaker it finds against those saved embeddings.

Corn

So he could basically create a folder called known speakers, put a wave file of himself and a wave file of Hannah in there, and the script could just loop through and match them up?

Herman

Precisely. There is a bit of a learning curve because you have to handle the clustering logic yourself if you want it to be perfectly programmatic, but the community has built some great wrappers around it. There is a project called WeSpeaker that is also worth looking at. It is a research toolkit that focuses specifically on speaker embedding and speaker recognition. It is trained on massive datasets like Vox Celeb, which contains thousands of different voices, so it is very robust to different recording conditions.

Corn

That is interesting. You mentioned recording conditions. Daniel and Hannah are recording in an apartment. I imagine there is some reverb, maybe a refrigerator humming in the background. Does that mess with the embeddings?

Herman

It can. This is where the pre processing comes in. Most modern pipelines will include a de reverberation step or a noise suppression step before the voice even hits the embedding model. But one of the reasons Pyannote is so popular is that it is quite resilient. It uses what they call an end to end neural diarization approach, or E E N D. Instead of doing everything in separate steps, the newer models try to learn the whole process at once, which helps them handle things like overlapping speech much better.

Corn

Overlapping speech is the big one. In a natural conversation, people do not wait for a perfect three second pause to start talking. They jump in, they laugh, they say yeah or right while the other person is still finishing. How does a programmatic tool handle two voices at the exact same time?

Herman

That is actually the cutting edge of the field right now. Older systems would just fail or assign that segment to whoever was louder. But newer architectures can actually output multiple speaker labels for the same time stamp. They use something called power set encoding. Instead of just saying is it speaker A or speaker B, the model can say it is speaker A and speaker B simultaneously.

Corn

That is impressive. But let us say Daniel does not want to manage a full Python environment and keep up with PyTorch updates. What about the commercial API side? I know we have talked about Assembly AI and Deepgram in the past.

Herman

Yeah, if you want something that just works via an API call, the landscape is very competitive. Deepgram, for instance, has their Nova-3 model, which is incredibly fast. They have a feature specifically for diarization where you just pass a flag in your request, and it returns the transcript with speaker labels.[2]

Corn

But does Deepgram allow for that specific speaker identification Daniel asked about? Can he tell Deepgram, hey, this is Daniel, and then have it remember him for the next meeting?

Herman

That is the catch with most cloud APIs. They are usually doing blind diarization. They are very good at telling you there are two people, but they do not typically store a voice profile for you unless you are using their enterprise level speaker search features. However, Assembly AI has been moving in that direction. They offer a speaker diarization model that is very accurate, and while it is primarily designed to distinguish between voices in a single file, you can take the output and match it to your own database of voice prints on your end.

Corn

So he would still have to do a little bit of the legwork himself. He gets the transcript back with speaker one and speaker two, and then he has to run a small local script to identify which one is him based on the audio segments.

Herman

Right. But there is another tool that I think might be exactly what Daniel is looking for, especially if he wants to build a dedicated app for his apartment. It is called Picovoice, specifically their Eagle engine.

Corn

Oh, I remember hearing about Picovoice. They do a lot of on device stuff, right?

Herman

Yes, and that is the key. Picovoice Eagle is specifically designed for speaker recognition. It is not just a general transcriber. It is built to recognize specific people. The workflow is very clean. You do an enrollment phase where the person speaks for about twenty or thirty seconds. The engine creates a very small, highly compressed speaker profile. Then, during the actual conversation, it can identify those enrolled speakers in real time with very low latency.

Corn

And because it is on device, he would not have to worry about sending his and Hannah's private apartment discussions to a cloud server every week.

Herman

Exactly. It runs on Windows, Mac, Linux, and even mobile or Raspberry Pi. For a housemate who is tech savvy like Daniel, he could set up a dedicated recording device in the kitchen or living room that automatically recognizes who is talking and logs the meeting notes. It is very programmatic and very reliable because it is looking for those specific enrolled profiles rather than trying to guess the speakers from scratch every time.

Corn

That sounds like a winner for his specific use case. But I want to push on the reliability bit for a second. We have talked about how voices are like fingerprints, but voices change. If Daniel has a cold, or if he is tired, or if he is just really excited about a new project and his pitch goes up, do these models hold up?

Herman

That is a great question, and it is something researchers call intra speaker variability. Your voice is not a static thing. It changes based on your health, your mood, and even the time of day. Most high quality models, like the ones used in Picovoice or the latest Pyannote embeddings, are trained to ignore those temporary fluctuations and focus on the physiological characteristics of your vocal tract, the things that do not change.

Corn

Like the physical shape of your throat and mouth?

Herman

Exactly. The length of your vocal folds, the shape of your nasal cavity, these things create resonant frequencies called formants that are relatively stable. A good model can see through a stuffy nose because the underlying structure is still the same. However, if you are doing a very high stakes identification, like for a bank, they usually require a longer enrollment period to capture those variations. For an apartment meeting, a thirty second enrollment is usually plenty.

Corn

You know, it is funny we are talking about this because it reminds me of that episode we did a while back, I think it was episode one hundred ninety six, where we talked about voice cloning. Back then, we were looking at how easy it is to mimic someone's voice. Does that pose a problem for these identification tools? If Daniel's voice is being cloned by an AI, would Picovoice Eagle be able to tell the difference between the real Daniel and a high quality clone?

Herman

That is a massive area of research right now, often called anti spoofing or liveness detection. Most standard speaker identification tools are not inherently designed to detect clones. They are looking for the voice print. If the clone is good enough to perfectly replicate those formants we talked about, it might fool a basic identification system. But the high end systems are starting to incorporate artifacts that are present in synthetic speech but absent in human speech. It is a bit of an arms race. But for Daniel's apartment meetings, I do not think he has to worry about a deepfake Hannah trying to sabotage their property ladder discussions.

Corn

Hopefully not! Unless their apartment meetings get a lot more intense than I imagine. But going back to the practical side, if Daniel wants to implement this, he mentioned he is currently using Gemini. If he wants to keep using a large language model for the actual summary or the property ladder analysis, how does he bridge the gap between the raw audio and the final identified transcript?

Herman

That is the perfect workflow. Step one is the audio processing. He uses a tool like Pyannote or Picovoice to get a diarized transcript. So he has a file that says, Daniel said this, Hannah said that. Then, he feeds that structured text into Gemini or GPT four. By giving the model the names up front, he saves a ton of tokens and prevents the AI from hallucinating who said what.

Corn

Right, because Gemini is great at summarizing, but it is guessing at the speakers based on context. If someone says, I will take the trash out, Gemini might guess it is Daniel because of some internal bias, even if it was actually Hannah. By giving it the verified labels from a tool like Picovoice, the summary becomes much more accurate.

Herman

Exactly. And if he wants to be really fancy, he can use a tool called Whisper from OpenAI for the actual speech to text part. Whisper is arguably the best general purpose transcription model out there. There is a version called Faster Whisper that is optimized for speed. He can run Faster Whisper for the text, Pyannote for the speaker labels, and then combine them.

Corn

Wait, so he would be running two different models at the same time? One for the words and one for the voices?

Herman

Yes, that is actually how most professional transcription services work behind the scenes. They have a speech to text engine and a diarization engine running in parallel. Then they align the time stamps. If the transcription says the word property started at ten point five seconds and ended at eleven seconds, and the diarization engine says speaker A was talking from ten seconds to twelve seconds, the system knows that Daniel said property.

Corn

That sounds like it could get complicated if the time stamps do not align perfectly. Is there anything that combines them into one neat package?

Herman

There are some integrated pipelines. There is a popular GitHub repository called WhisperX that does exactly this. It takes Whisper for the transcription, uses a process called forced alignment to get word level time stamps, and then integrates Pyannote for the diarization. It is probably the most powerful open source tool Daniel could use right now if he wants a one stop shop for a highly accurate, identified transcript.

Corn

WhisperX. I will have to check that out. It sounds like exactly the kind of thing Daniel would enjoy tinkering with. But what about the recurring voice aspect? He mentioned that these are people he knows and the voices do not change. Is there any advantage to having months of data? Can the system learn and get better over time?

Herman

Absolutely. This is where you move into the territory of speaker adaptation. If Daniel has fifty recordings of himself, he can create a much more robust average embedding. Instead of just one ten second clip, he can use a representative sample of his voice from different days. This makes the system much more reliable against those variations we talked about earlier, like being tired or having a cold.

Corn

It is like building a more detailed three dimensional map of his voice.

Herman

Exactly. And for developers, this is usually handled by a vector database. You store the embeddings for Daniel and Hannah, and then for every new segment of audio, you do a similarity search. If the new segment is ninety eight percent similar to the Daniel vector, you label it Daniel. If it is only seventy percent similar, maybe you flag it for manual review or label it as an unknown guest.

Corn

That brings up a good point. What if they have a guest over for one of these meetings? Maybe a financial advisor or a friend. How does a system that is tuned for Daniel and Hannah handle a third, unknown voice?

Herman

Most of these tools have a threshold setting. You can set it so that if a voice does not match a known profile by at least eighty five percent, it gets labeled as speaker unknown or guest. Pyannote is particularly good at this because it will still diarize the guest. It will say, okay, I do not know who this is, but it is definitely a third person who is not Daniel or Hannah.

Corn

That is really useful. It prevents the system from accidentally misattributing the guest's words to one of the residents.

Herman

Right. Now, I do want to touch on one thing Daniel mentioned about the current state of things. He is using Gemini, which is a massive model. We are seeing a trend where these large multimodal models are starting to handle audio directly. In the future, we might not need separate diarization tools. You might just be able to upload an audio file to a model like Gemini one point five Pro or the latest GPT models and say, here is a sample of Daniel, here is a sample of Hannah, now transcribe the whole thing.

Corn

We are already seeing some of that, aren't we? With the ability to upload files directly into the chat interface.

Herman

We are, but the reliability for long files is still a bit hit or miss. The specialized tools like Picovoice or Pyannote are still significantly more accurate for speaker identification because that is their entire purpose. The big models are generalists. They are amazing at understanding the content, but they can still get confused about the fine grained details of who is talking when the voices are similar.

Corn

That makes sense. It is the old specialist versus generalist debate. If you want the absolute best results for a specific task, you use a tool built for that task.

Herman

Exactly. And for Daniel's apartment meetings, where accuracy probably matters for things like budget decisions or action items, I would definitely lean toward a specialized pipeline.

Corn

So, to summarize the options for him, if he wants open source and is okay with Python, WhisperX or Pyannote audio are the way to go. If he wants something on device and programmatic for an app, Picovoice Eagle is a fantastic choice. And if he just wants an easy API, Deepgram or Assembly AI are the commercial leaders, though he might have to do a little extra work to map the speaker labels to names.

Herman

That is a perfect summary. And honestly, for a weekly meeting, once he sets up a script using something like WhisperX, he could probably automate the whole thing. He could have a folder on his computer where he drops the audio file, and ten minutes later, a perfectly formatted, identified transcript appears in his inbox.

Corn

That sounds like the dream. No more manual labeling. I can see why he is excited about this. It is one of those small chores that technology is finally ready to take off our plates.

Herman

It really is. And it has implications far beyond apartment meetings. Think about journalism, or legal proceedings, or even just family history. Being able to automatically and accurately attribute speech in large archives of audio is a massive unlock for human knowledge.

Corn

You know, it also makes me think about the accessibility angle. For someone who is hard of hearing, having a real time transcript that clearly identifies who is speaking in a room can be life changing.

Herman

Absolutely. We talked about that a bit in episode three hundred twenty one when we were looking at AI animation and character voices, but the real world application for accessibility is even more profound. If you are in a group setting and you can see a screen that says, Herman is talking right now, it removes so much of the exhaustion of trying to follow a conversation.

Corn

It is amazing how these niche technical questions, like the one Daniel sent us, often lead to these much bigger conversations about how we interact with the world and each other.

Herman

That is why I love this show. There is always a deeper layer to the onion.

Corn

Speaking of deeper layers, I want to ask about one more technical detail before we wrap up. What about the sampling rate of the audio? Daniel is probably just recording on a phone or a laptop. Does he need high fidelity equipment to make these identification tools work?

Herman

Surprisingly, no. Most of these models are trained on telephone data or standard sixteen kilohertz audio. While a better microphone will always help, especially with noise reduction, you do not need a studio grade setup. As long as the voice is clear and not overly distorted, the embedding models are very good at extracting those unique features. In fact, some models are specifically designed to be robust to the kind of compression you get on a standard mobile phone recording.

Corn

That is good to know. So he does not need to go out and buy a bunch of professional gear just to get his apartment meetings transcribed.

Herman

No, his current setup is likely fine. The magic is in the math, not the microphone.

Corn

I love that phrase. The magic is in the math. It sounds like a Herman Poppleberry original.

Herman

I might have to put that on a t shirt.

Corn

Well, I think we have given Daniel a lot to chew on. From Pyannote to Picovoice, there are some really powerful tools out there that can do exactly what he is looking for. It is just a matter of how much he wants to get his hands dirty with the code versus using a ready made API.

Herman

And I hope he shares the results with us! I would love to see how their property ladder discussions look once they are perfectly diarized.

Corn

Maybe we can get him to give us a demo in a future episode.

Herman

That would be great.

Corn

Before we sign off, I just want to say a quick thank you to everyone who has been listening and supporting the show. We have been doing this for over three hundred episodes now[1], and the community that has grown around My Weird Prompts is just incredible.

Herman

It really is. We love hearing from you. Whether it is a technical question like Daniel's or just a weird thought you had at three in the morning, send it our way.

Corn

And if you are enjoying the show, we would really appreciate it if you could leave us a review on Spotify or whatever podcast app you use. It genuinely helps other curious people find us and join the conversation.

Herman

Yeah, it makes a huge difference. We read all of them, even the ones that tell us we talk about vectors too much.

Corn

Hey, there is no such thing as talking about vectors too much in this house.

Herman

Fair point.

Corn

You can find all our past episodes and a contact form at our website, myweirdprompts.com. We are also on Spotify, so make sure to follow us there for new episodes every week.

Herman

Thanks for joining us today in Jerusalem. It has been a blast.

Corn

Definitely. Thanks to Daniel for the prompt, and thanks to all of you for listening. This has been My Weird Prompts.

Herman

Until next time, keep staying curious.

Corn

Bye everyone!