Episode Book

EPISODE #6728

Why Word Timestamps Require Forced Alignment

PUBLISHED	21 November 2026 at 17:26 UTC
DURATION	29:35
LISTEN	myweirdprompts.com/episode/word-timestamps-forced-alignment-24415/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.36
BOOK GENERATED	21 November 2026 at 19:31 UTC

SYNOPSIS

Standard speech recognition models excel at transcribing words, but pinning exact millisecond timestamps to every syllable requires a fundamentally different architecture. This breakdown covers how forced alignment uses CTC trellises and Viterbi search to snap known transcripts onto raw audio frames, where Whisper and WhisperX struggle with dropped filler words, and how transducer models like NVIDIA's Parakeet attempt to solve timing natively.

TRANSCRIPT



Corn

Anyone who's watched word-synced captions has seen it. The highlight lands half a beat behind the voice, then races to catch up, then gives up and skips two words entirely. Daniel's been watching that too, and this week he asked the question underneath it: what does it actually take to get an accurate start and end time for every word in a recording? His working picture is that alignment is its own stage, bolted into a pipeline after the transcription, and he wants to know what alignment models exist for the different ASR systems, and which of them are open source.



Herman

I love this one, because the answer is mostly yes, it's a pipeline, with one interesting exception sitting right in the middle of the tools he already uses.



Corn

We'll get to the exception. Let's start with what word alignment actually is, and why it's harder than it sounds.



Herman

So transcription gives you what was said. Alignment gives you when each piece of it was said, down to the word, ideally down to within a few tens of milliseconds. Two different jobs. And most ASR models are trained to do the first job well and give you the second one only loosely.



Corn

Because a model that's optimised to recognise words has no reason to also be precise about their edges.



Herman

Right. A typical ASR model will happily give you a segment-level timestamp. Thirty seconds of audio, here's a chunk, here's the text for that chunk, here's roughly where the chunk starts and ends. That falls out of the architecture almost for free, because most of these models process audio in windows and they know which window produced which text. But "the word responsibility starts at fourteen point two seconds and ends at fourteen



Herman (continued)

point eight" — that's a different promise. To make karaoke captions, or word-level search inside an archive, or precise cut-points for editing, you need per-word times, and per-word times have to come from somewhere.



Corn

And the somewhere is either built into the model or bolted on afterwards.



Herman

Those are the two roads, and the whole episode lives on that fork. Either the ASR architecture itself predicts durations as part of decoding, or you run a second stage that takes the audio plus the now-known transcript and hunts through the audio to find where each token actually happens.



Corn

Which is a strange job when you think about it. The second model doesn't have to understand anything. The words are already there on paper.



Herman

It's the strangest job in speech, honestly. You're given the answer sheet and asked to mark the tape. The technical name is forced alignment — "forced" because you're forcing the transcript onto the audio, whether the audio agrees or not. And the way it works is elegant, so let me lay it out, because once you see the mechanism, every tool we talk about after this is just a costume over the same skeleton.



Corn

Go.



Herman

You take a CTC model. That's Connectionist Temporal Classification, the older workhorse architecture behind wav2vec 2 — and don't worry, I won't make you sit through the acronym history, we've done episodes on that.



Corn

Generous of you.



Herman

The CTC model looks at the audio frame by frame, something like a hundred frames a second, and for every frame it emits a probability distribution over characters. Just the letter R, or a blank, or E, or a space. Now, on its own, that output is peaky and ambiguous — a character's probability might spike for one frame in the middle of a sound that lasts ten frames, and the blank token doesn't tell you whether you're looking at silence or just the gap between two identical letters.



Corn

So you can't just read timestamps off it.



Herman

You can't. That's the tempting mistake. The spike tells you the letter is in there somewhere, not where it starts and ends. So what you do instead is build a trellis — a grid where one axis is the audio frames and the other is the characters of your known transcript, including spaces — and then you run Viterbi search over it. Viterbi is a classic dynamic-programming trick: it finds the single best path through the grid, the path that assigns frames to characters in order, respecting that sounds take time and that some frames are blanks. Once you have that path, you merge the frames each word's characters landed on, and out comes a start and end time per word.



Corn

Let me make sure I've got the shape of it, because it sounds almost too tidy. Say the word is "glass" and it occupies frames nine hundred to nine hundred and twelve. The Viterbi path walks through the G column, then the L column, then the A, then the two S's with a blank between them, and each character picks up its own little stretch of frames.



Herman

Exactly that. And the blank between the two S's is the part people forget — without it, the path could collapse both S's onto the same frame or smear them across the whole word, because the model has no other way to know that "glass" has two of them. The blank is what lets the trellis say: first S lives here, second S lives there.



Corn

And if the audio is longer than the transcript suggests — someone draws out a vowel for two seconds — the path just stretches?



Herman

It stretches, within the limits the search allows. That's the "forced" part doing its work. The path has to explain every frame with something, so a long vowel becomes a long column of frames assigned to one character. Which is fine, that's reality. The failure mode is the opposite — when the audio contains something the transcript can't explain at all.



Corn

Which is where your wildcard comes in.



Herman

Right, and there's one more piece, because audio sometimes contains speech your transcript doesn't cover — a mumbled aside, a cough that sounds like a word. The good aligners have a wildcard token that can absorb untranscribed speech, so the path doesn't get dragged off course trying to explain a noise with your clean text. Without it, the search has to assign that cough to one of your real words, and every word after it gets pushed late.



Corn

That's the mechanism. Now the question Daniel actually asked — does this have to be a separate stage?



Herman

For most models, yes, and here's why. Whisper, which is still the centre of gravity for open-source transcription, natively gives you segment-level times and nothing finer. Word times from Whisper — the ones in OpenAI's own word timestamps option, or the whisper-timestamped package — are derived, not predicted. They come from dynamic time warping over Whisper's cross-attention weights. You're reading the tea leaves of where the decoder was looking while it generated each word.



Corn

Tea leaves.



Herman

The whisper-timestamped README literally calls it experimental. And there's a nastier problem: Whisper tends to drop filler words. If it decides "um" isn't worth transcribing, the timing for the next word gets shifted back into the hole, because the attention trail has nothing anchoring it there.



Corn

And for captioning specifically that's not a corner case, is it? Fillers are everywhere in real speech. If every dropped "um" pulls the following word's timestamp early, you've got captions that lead the voice all through the conversational parts of a recording — which is most of a podcast.



Herman

It's the difference between testing on a TED talk and testing on two people arguing over who left the heating on. The TED talk has no fillers, clean pauses, and a single speaker at a fixed distance from the mic. The argument has overlaps, trailing vowels, and someone saying "I mean— no, but— look." Every one of those is a small tax on derived timestamps, and the taxes compound.



Corn

So Whisper's word times are a reconstruction of something Whisper never committed to.



Herman

Exactly why the pipeline exists. WhisperX is the classic answer, and it's the tool Daniel named. Three stages. First faster-whisper does the transcription. Then a voice-activity model finds where speech actually is, so silence and music don't drift your timings. Then a language-specific wav2vec2 CTC model does the forced alignment we just described. English, French, German, Spanish and Italian have default alignment models bundled; other languages, you pull one off Hugging Face or bring your own.



Corn

And that's been the shape of the thing for a few years now. Twenty-five thousand stars on GitHub, roughly.



Herman

Around twenty-five and a half thousand, yes, and still active — a PyPI release in May. It's the default the way duct tape is the default.



Corn

Duct tape with known limits.



Herman

Known and documented. Words with characters outside the aligner's vocabulary get no timing at all — a year written with a full stop after it, or thirteen pounds sixty written with the pound sign and the decimal point, those just come out untimed. Overlapping speech is handled poorly. And you need one alignment model per language, which is fine for English and gets thin fast for, say, Georgian.



Corn

The currency thing is my favourite. A model that can transcribe speech about money cannot time the money.



Herman

It's a vocabulary problem, not an intelligence problem, but it lands as a hole in the captions.



Corn

Okay. Now the exception, because you've been sitting on it like a hen.



Herman

Because it's the good news, and it's Parakeet. The Parakeet TDT models, the zero point six billion parameter v2 for English and v3 for twenty-five European languages, return word, character and segment timestamps natively. You run them through NeMo's transcribe with timestamps enabled and the words come out with times attached.



Corn

Not because someone bolted an aligner on. Because of the decoder.



Herman

TDT stands for Token-and-Duration Transducer. A normal transducer emits a token and moves on. The TDT decoder emits a token and, at the same time, predicts how many audio frames that token covers. It's predicting a duration with every single step of decoding. So when it emits the word "glass," it also says "that was about nine frames," and you can accumulate those durations into word and segment start and end times directly.



Corn

So alignment isn't a stage. It's a side effect of how it decodes.



Herman

That's the whole trick. And there's no external alignment model, no language-specific add-on — the timing machinery is one model, one pass.



Corn

Now hit it with the other hand.



Herman

The other hand is that NVIDIA publishes word error rates for Parakeet and nothing else. No timestamp-error figures at all. So we know the timestamps exist, and we do not know, from the model cards, how good they are. And here's the detail that I found telling: in the Canary one B v2 and Parakeet TDT v3 paper they published in September twenty twenty-five, when NVIDIA needed timestamps for their own evaluation, they generated them with the NeMo Forced Aligner and an auxiliary CTC model. Not with the native ones.



Corn

Their own paper didn't trust their own timestamps.



Herman

I wouldn't put it that strongly — they also recommend segment-level timestamps for translation output, which suggests they know word-level is the shakier tier. But yes. When the accuracy mattered, they reached for the separate aligner, same as everyone else. Native doesn't mean caption-grade.



Corn

Native means the demo works.



Herman

Native means nobody can accuse you of not having them.



Hilbert

Hilbert's Anteater Trivia. Giant anteaters can swim across rivers. I have crossed a few rivers myself, mostly to get away from people who wanted to talk about anteaters.



Corn

That's the correct number of rivers.



Herman

Right — so Parakeet is the exception, and even there the separate stage keeps coming back. For almost everything else, alignment is its own stage. Now, what does the landscape of that stage look like if you're the one building the pipeline? Let's organise it by what the aligner attaches to, because that's the practical question — you don't pick an aligner in a vacuum, you pick one that fits your ASR model.



Corn

Start with the Whisper family, since that's where most people start.



Herman

WhisperX, obviously. But there are two others worth knowing. CrisperWhisper two point zero, from an outfit called Nyra Health, takes Whisper and adds attention heads that were specifically trained to find word boundaries — the idea being to make the derived timestamps less tea-leaf and more measured. It also has an optional English forced-aligner timing mode for when you want the full treatment. And then there's stable-ts, which is a Whisper wrapper with an align function for when you already know the text and just need the timings, and it'll write karaoke-style ASS subtitles directly.



Corn

Past tense.



Herman

Ah — yes, fair. stable-ts was archived and made read-only at the end of May. It still works, plenty of people still use it, but it's frozen. And that matters for exactly the reason you'd expect: ASR models move, and a frozen wrapper drifts out of true with them. A pure aligner ages better — alignment is a settled enough problem that a pre-deep-learning tool still holds its own, and we'll get to that, because it's the funniest fact in this whole episode.



Corn

You've been holding that fact since before the show.



Herman

Since Tuesday. Let's do the CTC side. The NeMo Forced Aligner, NFA, comes from NVIDIA and works with CTC models or hybrid CTC-transducer models in CTC mode. Pure transducers, no — the TDT trick we described is unusual, and a standard transducer has no per-frame probabilities to search over. NFA outputs the standard formats: CTM files at token, word and segment level, and it'll also write ASS subtitle files with word highlighting, karaoke style, straight out of the box. And it handles long audio, an hour or more, in one go.



Corn

Licence?



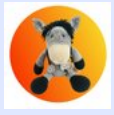
Herman

Apache 2.0 for NeMo, which is about as comfortable as it gets.



Corn

The man reads model cards the way other people read menus.



Herman

You've checked the licence on every appliance in our kitchen.



Corn

The toaster is BSD-licensed, it can stay.



Herman

Then the language-agnostic tier, and this is where it gets wide. Montreal Forced Aligner, MFA, is the grand old name — Kaldi-based, GMM-HMM, the pre-deep-learning approach that uses a pronunciation dictionary and an acoustic model rather than a neural network. MIT licence, and here's the thing: it's still actively maintained. A push a week ago. Version three point four. This thing predates half our listeners and it is not dead.



Corn

And it's about to beat everybody, isn't it.



Herman

It is about to beat everybody, but let me finish the survey first. On the neural side there's ctc-forced-aligner, by Mahmoud Ashraf — the code is BSD, and the default model is Meta's MMS, three hundred million parameters, covering roughly eleven hundred and thirty languages, with romanisation for scripts that aren't Latin. It is the most-downloaded forced aligner on Hugging Face, about two point four million downloads.



Corn

Eleven hundred languages.



Herman

Meta's MMS project really did cover that much of the world.



Corn

A model trained on eleven hundred languages and it still won't time a pound sign.



Herman

Different model, same shape of problem. And notice the licence split: the code is BSD, but the weights, both Ashraf's aligner and Meta's underlying MMS one B model, are CC-BY-NC, non-commercial. If you're building a product rather than a hobby, that distinction is the whole due-diligence exercise. CrisperWhisper has the same split — MIT



Herman (continued)

code, non-commercial research weights.



Corn

So the free code can be carrying unfree weights. It's a lot of packaging.



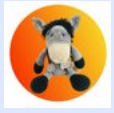
Herman

It's the shipping container question, and the weights are always the cargo. Now the newest entrant, which is different in kind: Qwen put out a ForcedAligner in January, zero point six billion parameters, alongside their ASR model. It's a speech-LLM — non-autoregressive, predicts timestamps as slot-filled tokens, so the model literally generates "one four two zero" as text in a timestamp slot. Eleven languages, Apache 2.0 for code and weights, which is the cleanest licence in the field. Limits: five minutes of audio per call.



Corn

Timestamps generated as tokens. The model reads out the clock.



Herman

That's exactly what it does. And there's an older generation that still gets mentioned — Gentle, which is Kaldi with a REST server, and aeneas, which did alignment by synthesising the transcript with text-to-speech and dynamic time warping the synthesis against the audio. Clever, clever, and aeneas's own documentation admits word-level results may be worse than ASR-based approaches. Its last release was 2017.



Corn

Align by comparing the real voice to a robot reading the same words. That's the eighties detective of alignment methods.



Herman

It works surprisingly well for subtitle-level timing. Anyway — the numbers. This is the part I've been waiting to say, because the 2024 independent comparison is the punchline to the entire deep-learning era of this field. They measured the share of word boundaries landing within twenty-five milliseconds of hand-labelled truth.



Corn

On studio speech or real speech?



Herman

Both. On TIMIT, which is read speech in a quiet room, MFA got about seventy-three percent of boundaries within twenty-five milliseconds, WhisperX fifty-three, the MMS-based aligner about forty-four. On Buckeye, which is conversational speech, same ordering — MFA seventy percent, MMS fifty-three, WhisperX forty-three. Median shift on TIMIT: MFA twelve and a half milliseconds, WhisperX twenty-three, MMS twenty-nine.



Corn

Hold on. The Kaldi GMM-HMM dictionary aligner from before deep learning beat the neural end-to-end systems.



Herman

Beat all of them. The authors' own conclusion — the classic aligner still wins on precision. And there's a sting in the setup: the end-to-end systems were only scored on words they recognised correctly, so the comparison actually flatters the neural side.



Corn

Twenty years of architecture progress and the podium hasn't moved.



Herman

Now, to be fair to the modern side, there's the Qwen vendor table — their own numbers, so salt accordingly. Accumulated average shift, milliseconds, lower is better. On MFA-labelled data: Qwen's aligner forty-three, NeMo's one hundred thirty, WhisperX one hundred thirty-three. On human-labelled data Qwen thirty-two, NFA one hundred one. And then the row that made me sit up: on three hundred seconds of concatenated audio, Qwen fifty-three, NFA two hundred forty-seven, WhisperX two thousand seven hundred.



Corn

Two thousand seven hundred milliseconds.



Herman

Two point seven seconds of accumulated drift. WhisperX's timings wander on long audio — the segment-based errors compound. That's the row I'd want anyone building a caption pipeline to see, because it's the difference between a demo on a thirty-second clip and a two-hour podcast.



Corn

And it explains something anyone who's used these tools has felt but maybe couldn't name. You test on a clip, everything's tight, you run the two-hour file and by the end the highlight is a full beat ahead of the voice. It wasn't your imagination — it's two point seven seconds of compounding.



Herman

And the compounding is structural, not a bug you can patch. Segment boundaries carry small errors, each segment's alignment is anchored to those boundaries, and errors at the anchor propagate forward. Chunk the audio and re-anchor, and you reset the accumulation — which is why the drift number and the chunking advice are really the same finding stated twice.



Corn

So the choice is between the antique that's precise and the modern thing that doesn't fall apart at length.



Herman

The antique, the new LLM, or you chunk the audio and re-anchor. Which, incidentally, is what we talked about when we covered chunk seams breaking speech models — same disease, different symptom.



Corn

And Parakeet users sit in the middle of all this. They've got the timestamps coming out natively and no published error figures to tell him how far to trust them.



Herman

Which means the benchmark they actually need doesn't exist yet, and we did a whole episode once on benchmarking custom ASR beyond word error rate — this is precisely that problem again. Timestamp error is a different metric than WER, and almost nobody publishes it. If you want to know whether your word times are caption-grade, you have to build a little evaluation yourself, and MFA's output on the same audio is a reasonable stand-in for ground truth if you can't hand-label.



Corn

He'd hand-label thirty minutes of audio before he'd trust a vendor table.



Herman

He absolutely would, and honestly that's the right instinct. Qwen grading its own homework, and leaving MFA itself out of the table, is exactly the kind of result you want a second opinion on.



Corn

Let's do the boring tier before we run out of room. The paid APIs.



Herman

Brief, because it's the same answer four times. OpenAI's transcription API returns word timestamps — you set the timestamp granularity to word and ask for verbose JSON. ElevenLabs Scribe, AssemblyAI, and Deepgram's Nova-3 all return word timestamps natively in one call. No pipeline, no aligner, no wildcard tokens. You pay per audio and you don't own the gears.



Corn

For someone who ships open source, that's less a shortcut and more a dead end.



Herman

Daniel's stack — Parakeet, local inference — is built on owning the gears. And the captions step at the end is open regardless: word timings go into SRT or WebVTT cues, or ASS subtitles with per-word highlighting, which NFA and stable-ts both write directly. Though I'll tell you where people actually get stuck, having watched this pipeline built in our own kitchen, metaphorically. Not the model. The text normalisation.



Corn

Numbers, currency, punctuation.



Herman

Before alignment can happen, the transcript and the audio have to agree on what's in them. Thirteen pounds sixty, spoken, against a pound sign and a decimal point written, or a full stop after a year that nobody says out loud. The aligner can't time a character sequence the audio never produced, and the model card never warns you. That's the practical failure point, and it's boring in the way most real failure points are boring.



Corn

So the pipeline is transcription, alignment, normalisation, formatting. Two of those are hard science and two are plumbing.



Herman

And the plumbing is where the captions visibly break.



Hilbert

Normalisation, yes. You're right about that and I'll say so, since nobody else will. I did karaoke captions for a community theatre run of Pirates of Penzance a few years back — that's the one thing you two have said today that I've actually lived through, and Herman, you described the pipeline like a man reading a manual, which I mean as a compliment. Three weeks. The aligner skipped a price — thirteen pounds sixty, typed with the pound sign — in one of the patter songs, left a hole in the captions the length of a held note, so the whole house thought the tenor dried. It wasn't his fault. I still have the CTM file — the output, with the gap where the currency should be. I use it as a bookmark. What I never got answered was whether the model or the actor was at fault, because the man had a cold and kept stopping mid-word on "responsibility," which no trellis is going to survive. I've been meaning to write to the theatre about their microphone placement, which was the real culprit, but the stage manager moved to Swindon and I don't chase people.



Corn

You kept the failure.



Hilbert

It was the clearest thing the model produced all night, Corn. A perfect rectangle of nothing where the money went.



Herman

The blank span is actually diagnostic, you know — an untimed word in WhisperX output usually means the word was dropped for its characters before the search ran, not during it. Your aligner wasn't confused, it was refusing to look.



Hilbert

Then I blamed the wrong end of it for three weeks. That tracks. Cold's on the singer, that part I stand by.



Corn

Worth saying what that hands Daniel, though — a benchmark he can run for free. If the native Parakeet timestamps leave blanks or drift on the currency lines in his test audio, he'll see it in an afternoon, no vendor table required.



Herman

And the comparison costs him nothing but an evening. Run the same clip through Parakeet's native timestamps, through NFA with an auxiliary CTC model, and through MFA, lay the three CTM files side by side, and look at where they disagree. The disagreements are the finding. If native and NFA agree to within twenty milliseconds everywhere, he can caption with confidence. If they diverge on the fillers and the numbers, he knows exactly which lines to hand-check.



Corn

Where I land is that the pipeline picture — alignment as a separate stage you choose, rather than a property of the model — describes the whole field except one model family, and that family's own authors reached for the separate stage when accuracy mattered.



Herman

And that one family is worth knowing about. Parakeet isn't like Whisper — the TDT decoder really does hand you word times in one pass. But one pass and a trustworthy pass are different claims, and nobody's published the second one yet. The interesting question going forward is whether it stays two jobs at all. Qwen's aligner already predicts timestamps as tokens in a single model, and if the speech-LLM crowd closes the accuracy gap with the old Kaldi machinery, the separate stage stops being a stage and becomes a fallback. Though if MFA is still winning benchmarks in 2024 with a pronunciation dictionary and a Markov assumption, I wouldn't retire it on anyone's say-so.



Corn

Which is the pattern we keep hitting. The old tool outlives its third successor.



Herman

The dictionary people didn't need a revolution, they needed boundaries. Speaking of which — thanks to Hilbert Flumingtop for producing the show, as ever. If this was your kind of episode, go back for episode nine, Benchmarking Custom ASR Tools - Beyond The



Herman (continued)

WER; and episode ten, How ASR Went From Frustration To ... Whisper Magic. This has been My Weird Prompts, the human-AI collaboration podcast. If word-synced captions have ever fought you, you now know which stage was losing. Rate us on Spotify, it helps. See you tomorrow, Corn.



Corn

Tomorrow, Herman.