



● UNDERSTANDING LOCAL SPEECH RECOGNITION

How much can you trust a local transcript?

We compared four speech-to-text engines on English and Italian recordings. Along the way, we looked at what their scores mean, why they make different mistakes, and how to choose one for your own audio.

MeetingStow engineering · October 5 · Updated October 19, 2026 · English + Italian

One of the recordings we tested contains “I couldn’t do this one.” Phonon returned “I can do this one.” The transcript reads smoothly, but it changes what the speaker meant.

That is an awkward problem if you’re choosing a transcription model. You want a useful way to compare them, but a score cannot tell you whether the word a model missed was “um,” someone’s surname, or the “not” in a decision.

We build [MeetingStow](#), which transcribes meetings on your Mac. We wanted to compare the four engines in the app: Apple SpeechAnalyzer, Qwen3-ASR, Phonon-2, and Parakeet Ultra. Ultra made the fewest word errors in our test, while Apple took the least time. Looking through the actual transcripts helped us understand what those results did and didn’t tell us.

The comparison uses 26 public clips: six English meeting excerpts and twenty Italian recordings of people reading sentences. That is enough to find useful examples and check our implementation, but it is a small sample. We’ll explain the terms first, then walk through the results. If you’d rather inspect the outputs yourself, [the recordings’ references, transcripts, scores, and reproduction tools are available below](#).

Is speech-to-text the same thing as TTS?

It’s easy to mix up the names. **Speech-to-text**, or STT, takes audio and produces written words. You’ll also see it called **automatic speech recognition**, or ASR. **Text-to-speech**, or TTS, starts with written words and generates a spoken voice. A transcription app uses the first; a tool that reads an article aloud uses the second.

This article measures speech-to-text. If you're choosing a TTS voice, you'll want to listen for pronunciation, natural pacing, and how it handles names and numbers, as well as how quickly speech starts. Our transcription scores don't measure those things. Apple has separate APIs for the two jobs: [SpeechAnalyzer for recognition](#) and [AVSpeechSynthesizer for speaking text](#).

What happens between recording a call and reading the notes?

Recognizing the words is only one part of making a transcript you can use. These are the other terms you'll encounter in the comparison:

Voice activity detection

Usually shortened to VAD. It estimates which parts of a recording contain speech. An app can use that information to skip silence or choose where to split the audio. A poorly placed cut can lose part of a word.

Diarization

Works out who spoke when, usually with labels such as "Speaker 1" and "Speaker 2." It doesn't, by itself, tell you a person's name. Recognizing someone from a saved voice profile is a separate task.

Alignment

Places recognized words on the audio timeline. That gives you timestamps you can use to replay a sentence. A transcript can have the right words and still send you to the wrong point in the recording.

Summarization

Turns the transcript into shorter notes, decisions, or action items. It is a separate model task. We didn't score recaps here, so a good transcription result is not a measurement of summary quality.

You can therefore get several different things wrong: the words, the speaker labels, the timestamps, or the summary. If you need to verify a client's deadline, having the right words attached to a useful timestamp matters more than a tidy paragraph.

Why can two apps using the same model behave differently?

The **model** contains learned patterns for recognizing speech, stored as numerical weights. The **runtime** is the software that runs the model. The app also decides how to prepare the audio, split it into chunks, and assemble the results. When we say "engine" below, we mean that implemented recognition path, not just a model name.

You'll see labels such as **FP16** and **Q8_0** in the method table. FP16 uses 16-bit floating-point numbers; Q8_0 is an 8-bit quantization format. These describe how model weights are stored. Those choices can affect memory, speed, and output. Comparing the versions an app actually uses is more informative than assuming every implementation of a model behaves alike.

"Local" means the recording is processed on the device rather than sent to a remote service for recognition. The model may still need an initial download. It doesn't tell you whether recognition is accurate, whether the app is fast on your hardware, or whether a backup service copies the resulting files.

What did we compare?

We ran the four engines on an **Apple M2 Max with 64 GB RAM, running macOS 26.5.2**. The app was a signed Debug build of MeetingStow 0.5.27 (build 35), with its native MLX package optimized. We used the app's transcription methods so that audio preparation and transcript assembly would be included in the comparison.

▼ Will these engines run on my Mac?

System requirements: MeetingStow requires an Apple Silicon Mac and macOS 15 or newer for recording, local transcription, and local recaps. The optional Apple Speech engine tested here uses SpeechAnalyzer and SpeechTranscriber, which require macOS 26 or newer. Apple-powered recaps also require macOS 26 or newer with Apple Intelligence enabled. These benchmark results were measured on macOS 26.5.2; they do not validate performance or compatibility on macOS 15.

The four implementations measured on October 19

ENGINE	RUNTIME / CHECKPOINT	PROCESSING INCLUDED
Apple	SpeechAnalyzer / SpeechTranscriber	Native recognition. Separate from Apple Intelligence recaps.
Qwen	Qwen3-ASR 1.7B · Q8_0 · transcribe.cpp	Apple evidence, speech windows, recognition, and transcript assembly.
Phonon	Phonon-2 · native Swift MLX	The app's audio preparation, windowing, recognition, and assembly.
Ultra	Parakeet Ultra · FP16 · native Swift MLX	The same native wrapper as Phonon, with the Ultra checkpoint.

▼ Which version of Parakeet Ultra did we use?

Parakeet Ultra is Moondream's post-trained version of NVIDIA's Parakeet TDT 0.6B v3. We use the [FP16 MLX conversion](#), pinned to revision `5acd3970b21853434afd56ed0c8f84a08a52cae8`. The conversion omits the upstream VAD head. These are measurements of our Swift implementation on a Mac, not the upstream GPU runtime or published TensorRT results.

ENGLISH / CONVERSATION

AMI meeting corpus

Six 60-second mixed-headset excerpts from ES2004a and ES2004b, starting at 60, 240, and 480 seconds. Human word annotations from all speakers are ordered by word start time.

360s · 775 normalized words

ITALIAN / READ SPEECH

Google FLEURS

The first twenty Italian test rows in converted dataset order. Some sentence content repeats. These are people reading sentences, not an Italian conversation.

275.16s · 493 normalized words

We computed speaker intervals once for each clip and supplied the same intervals to every engine. None received the reference words, a list of names, or vocabulary hints. We checked the audio files against the original fixture hashes so every run used the same recordings. See the [AMI documentation](#) and [FLEURS dataset card](#) for the source recordings' scope.

▼ How did we run and time the tests?

Word error rate (WER) = (substitutions + deletions + insertions) ÷ reference words. We sum counts across clips before dividing. English uses Whisper's English normalizer. Italian normalizes Unicode, case, punctuation, and whitespace, retaining apostrophes and numeric spelling.

Engines ran sequentially in this order: Apple, Phonon, Ultra, Qwen. Each received one warmup per language, then two measured passes per clip. Displayed time is the sum of each clip's median of those passes.

Transcription time includes audio export, resampling, segmentation, recognition, and transcript assembly. Qwen includes its Apple evidence pass. Downloads, model preparation, warmups, shared diarization, saving, voiceprint matching, and recaps are excluded. Diarization groups speech by anonymous speaker; it is timed once per fixture and shown separately below.

Two repeats show short-run consistency, not statistical confidence. Fixed order, background load, caches, and thermals can affect speed. Speaker-label quality, timestamp accuracy, peak memory, energy, and cold-start latency were not scored.

What does a word error rate of 13.94% mean?

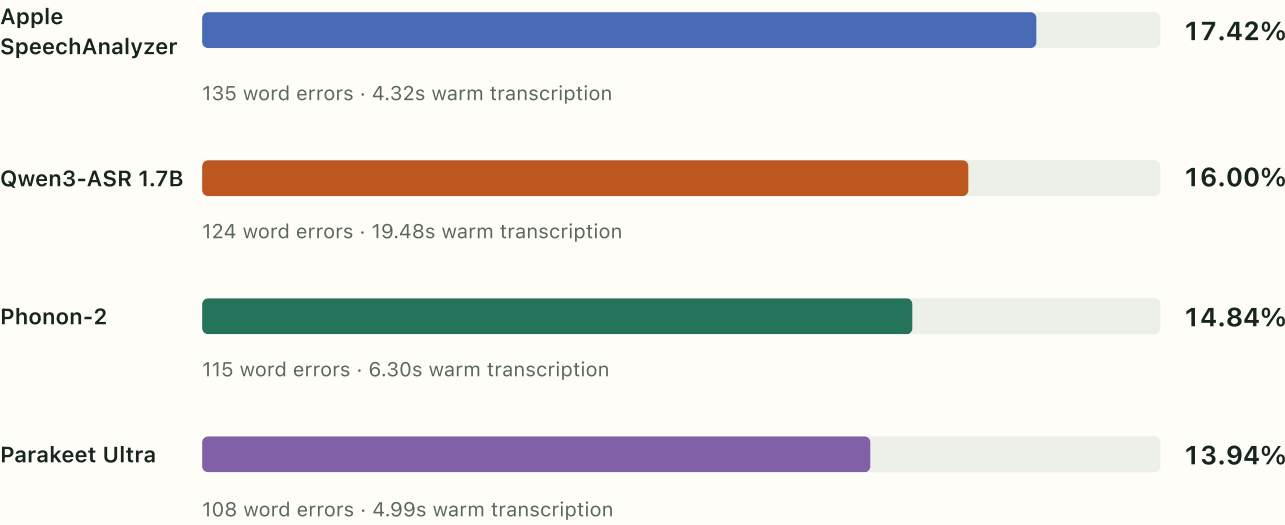
Word error rate, or **WER**, counts three kinds of mistake: substituting one word for another, leaving a word out, or adding a word that wasn't in the reference. Add those errors and divide by the number of words in the human reference. For example, one substitution and one missing word in a ten-word reference give a WER of 20%.

Lower is better, but the percentage isn't a count of sentences understood correctly. It doesn't say how serious the errors were, and it can exceed 100% if the model adds enough words. Our English result for Ultra is 108 errors across 775 reference words, which gives 13.94%.

Here are the four engines under the same scoring rules. Use the language buttons to switch samples; the timings are for processing all the clips after warmup.

EXPLORE THE MEASURED RESULTS

6 meeting excerpts · 360 seconds · 775 reference words



WER ↓ lower is better

Warm transcription · excludes diarization

Word error rates for four transcription engines

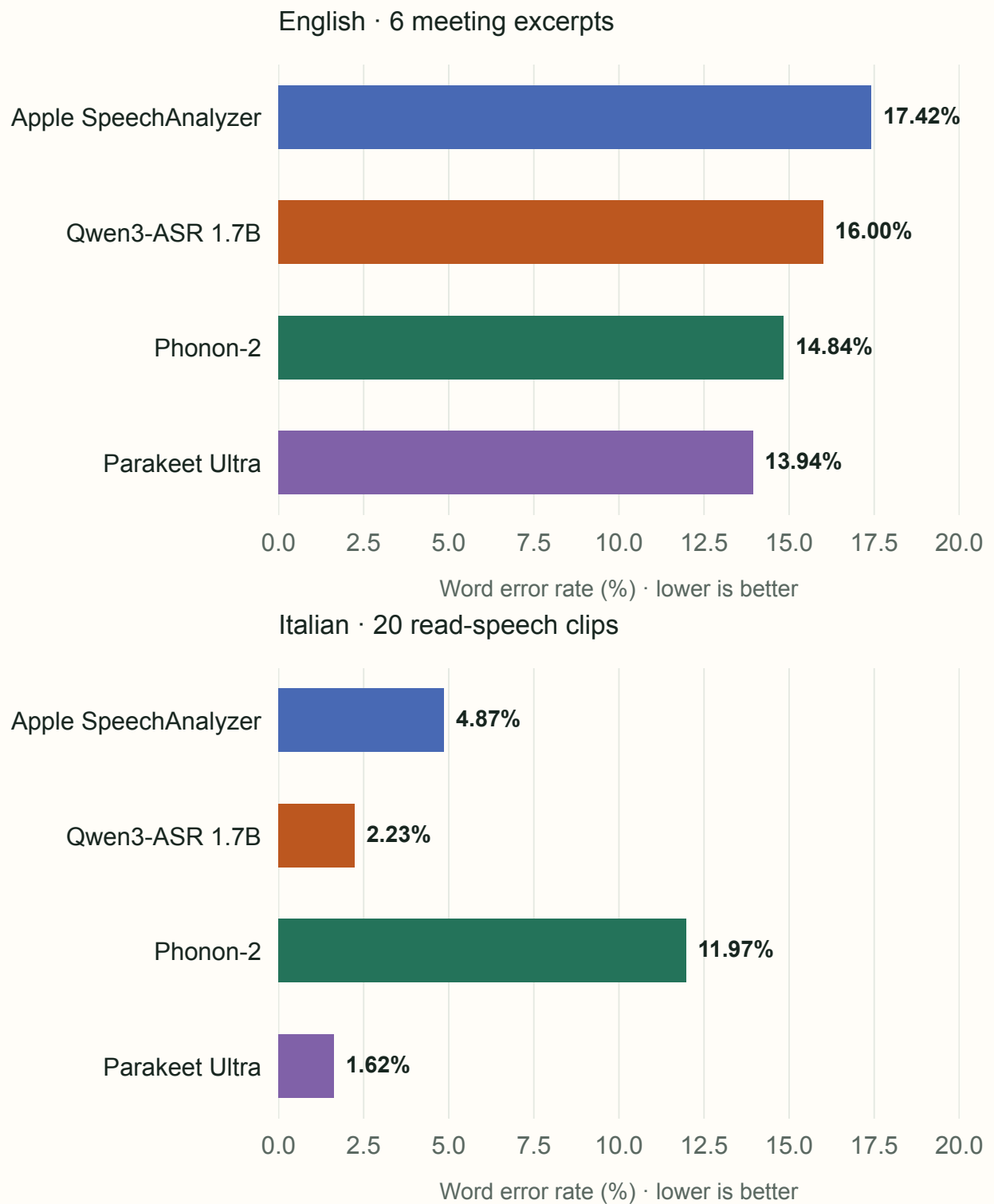


Figure 1. All four native engines, the same verified clips and scoring rules. Each language has a zero-to-twenty-percent axis. The speech conditions differ between languages. ·

SAMPLE	ENGINE	WER ↓	WORD ERRORS ↓	WARM TIME ↓
English 775 words	Apple SpeechAnalyzer	17.42%	135	4.32s
	Qwen3-ASR 1.7B	16.00%	124	19.48s
	Phonon-2	14.84%	115	6.30s
	Parakeet Ultra	13.94%	108	4.99s
Italian 493 words	Apple SpeechAnalyzer	4.87%	24	2.93s
	Qwen3-ASR 1.7B	2.23%	11	17.73s
	Phonon-2	11.97%	59	6.53s
	Parakeet Ultra	1.62%	8	4.95s

In English, Ultra makes **108 errors out of 775 words**, seven fewer than Phonon and sixteen fewer than Qwen. Its lead over Phonon is only 0.90 percentage points. Dropping one English clip at a time leaves Ultra with the lowest aggregate WER in 5 of six subsets. That tests sensitivity to these clips; it does not make six excerpts representative of meetings in general.

Does a good English score carry over to other languages?

We also checked Italian as a small test beyond English. Ultra makes **8 errors out of 493 words**, compared with Qwen’s 11 and Phonon’s 59. This is twenty read-speech clips in one additional language, not a broad multilingual benchmark. English conversation and Italian read speech are different tasks: the lower Italian percentages do not establish that any engine is “better at Italian.”

Phonon’s gap needs some context. Its [published accuracy claims concern English](#), and its [release documentation describes training on 6,000 hours of English speech](#). It derives from multilingual Parakeet, but retaining some Italian recognition does not mean that accuracy survived equally across languages. English-focused retraining is a plausible explanation for the gap; this test cannot separate that effect from compression or our runtime. Formatting accounts for only a small part of it, as the rescoring below shows.

How long would you actually wait for the transcript?

The transcription stage is quick on this machine: Ultra takes 4.99 seconds for the six minutes of English audio, and Apple takes 4.32 seconds. But we also spent 5.34 seconds separating the speakers. Adding that stage brings Ultra’s total to 10.33 seconds, before saving or creating a recap. Once recognition is this fast, work elsewhere in the app becomes a noticeable part of the wait.

Transcription time, including speaker separation

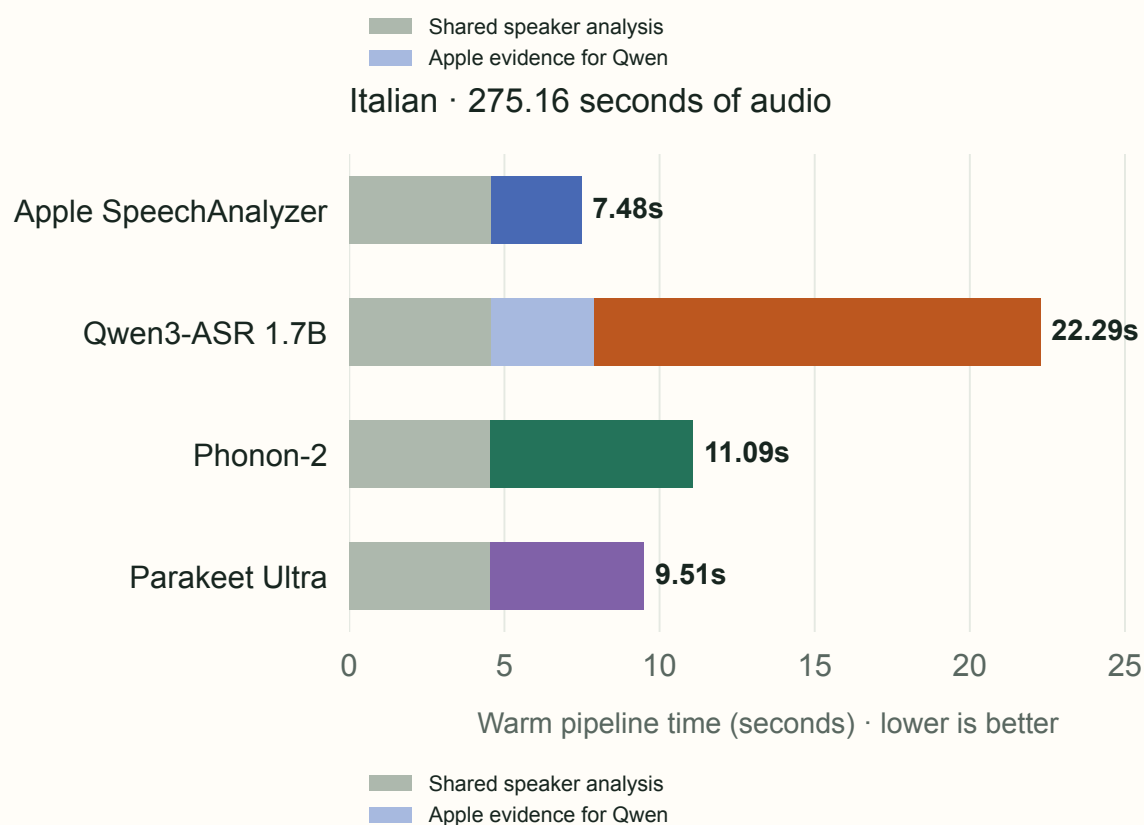
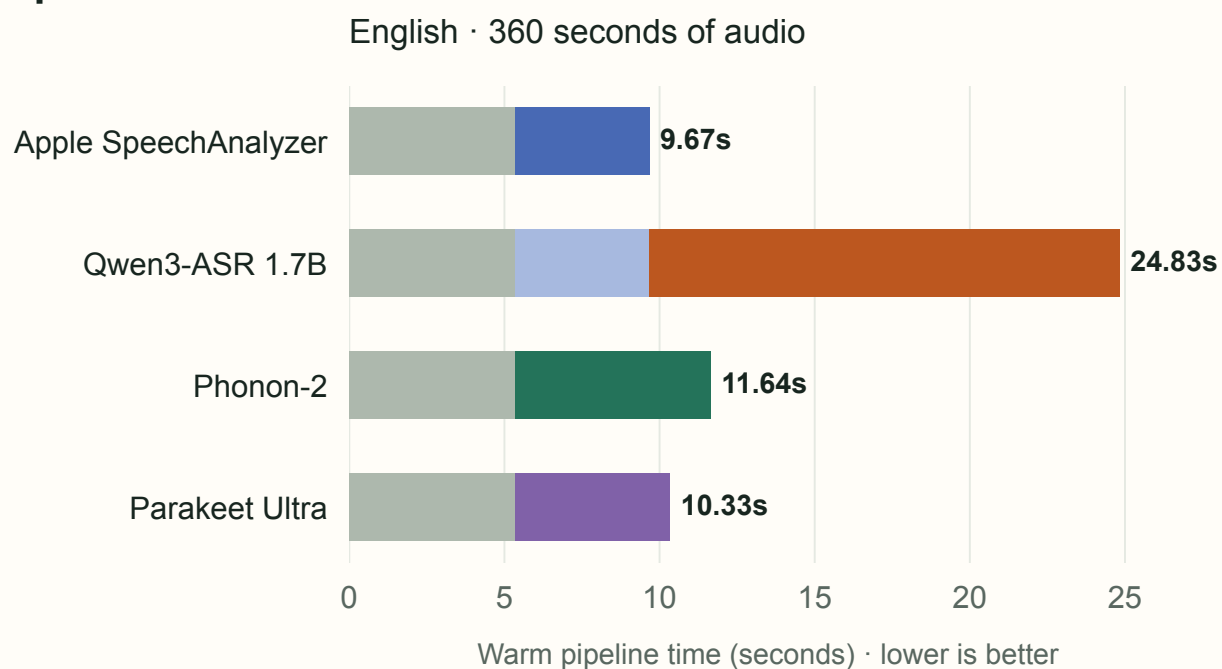


Figure 2. Shared diarization adds 5.34s for English and 4.55s for Italian to every engine. Qwen's Apple evidence time is separated from its other processing.

These are **warm** runs: the model is already loaded and has processed a clip in each language. A **cold start** would include loading and initialization; a first-time setup may also include downloading model files. We haven't combined those into a startup benchmark. And processing a finished recording faster than its duration doesn't tell you how soon live captions would appear while someone is speaking.

We made two measured passes per clip. Every repeated word-error count matched, although one Apple Italian name changed spelling without changing the score. Both outputs and their timings are in the download. Two passes are useful as a consistency check, but they don't tell us how speed varies over many runs.

What does a word error sound like?

The counts become more useful when you listen to the recording beside the output. Here is the sentence from the opening, with its surrounding words. Phonon changes the negation; the other three engines preserve it.

ENGLISH · ES2004B_240_300

Negation reversed

Human reference

“I ran out of time so I **couldn't** do this one as fun as the last one”

Phonon

“I ran out of time so I **can** do this one as fun as the last one”

Apple + Qwen + Ultra

All three preserve “**couldn't**” in this sentence.

Original 60-second benchmark excerpt. AMI Consortium / University of Edinburgh, CC BY 4.0; extracted from ES2004b mixed-headset audio.

If you were writing notes from the Phonon version, you could come away with a different understanding of what the speaker could do. We didn't run this through a summarizer, so we can't say how a recap model would handle it. The transcription error is already enough to warrant checking the audio.

A description becomes an action

Human reference

“so all the components that I’ve put forward are uh **low in cost**”

Phonon + Ultra

“So all the components that I’ve put forward are uh **lowering costs**”

Qwen

“So all the components that I’ve put forward are **low in cost**”

Original 60-second benchmark excerpt. AMI Consortium / University of Edinburgh, CC BY 4.0; extracted from ES2004b mixed-headset audio.

“Low in cost” describes the components; “lowering costs” says they are reducing costs. Ultra makes this mistake too, despite having the lowest overall WER. Qwen preserves the phrase, while Apple writes “low in costs.” A model that wins the aggregate comparison can still lose on a sentence you care about.

What happens to a company slogan?

SOURCE

RECOGNIZED PHRASE

Human reference

we put fashion in electronics

Apple

report fashion in electronics

Phonon + Ultra

we put fashion in electronics

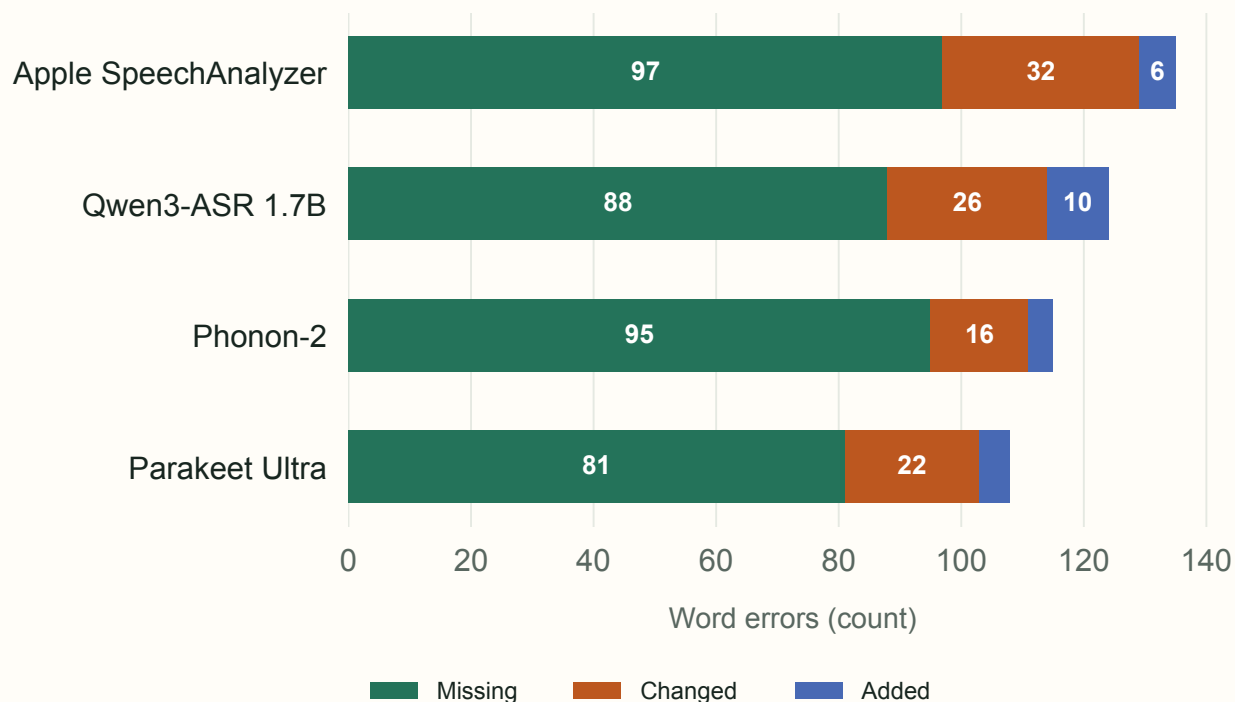
Qwen

Rupert fashion **and** electronics

The speaker is describing the company’s image. Qwen turns “we put” into the name “Rupert,” while Apple turns it into “report.” This is why a plausible word in a transcript still needs checking against the recording. None of the engines received the company slogan as a vocabulary hint; we did not test whether supplying it would fix this passage.

Missing, changed, and added words

English · 775 reference words



Italian · 493 reference words

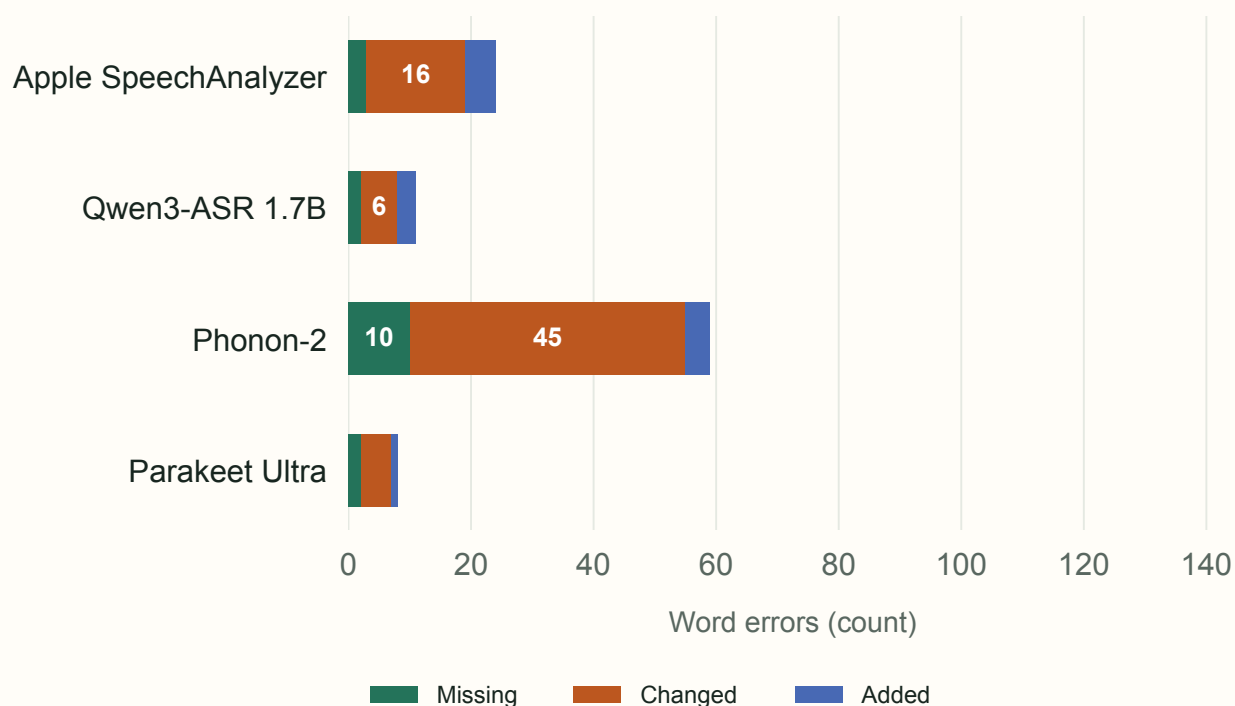


Figure 3. Missing words dominate the English score. Ordering overlapping speakers into one reference sequence is a difficult target for a mixed audio track; these counts do not establish which omissions came from overlap. ·

Other production problems remain outside this test: clipping, quiet microphones, music, long silence, echo, and sentences that switch languages. Plausible but unsupported text is also a failure to investigate. We did not run a silence or hallucination stress test, so this article assigns no hallucination rate to any engine.

Why can the same model get a different score?

Should “twenty five” and “25” count as different words?

In the first English clip, the reference says “we’ve got twenty five minutes.” Phonon also writes “twenty five,” while Apple, Qwen, and Ultra write “25.” The meaning is the same. Our English normalizer already treats those forms alike, so this difference does not penalize any engine.

Italian gives us a separate check because our original Italian scorer keeps numeric spelling. We rescored it with the same limited rules on both sides: split letter–digit boundaries, expand integer digits to Italian cardinal words, and expand numeric `m` to `metri`. For example, `15 metri` and `quindici metri` both mean “fifteen meters.” These rules forgive that difference without forgiving Phonon’s change from 2017 to 207 in the same recording.

Italian scores with and without number normalization

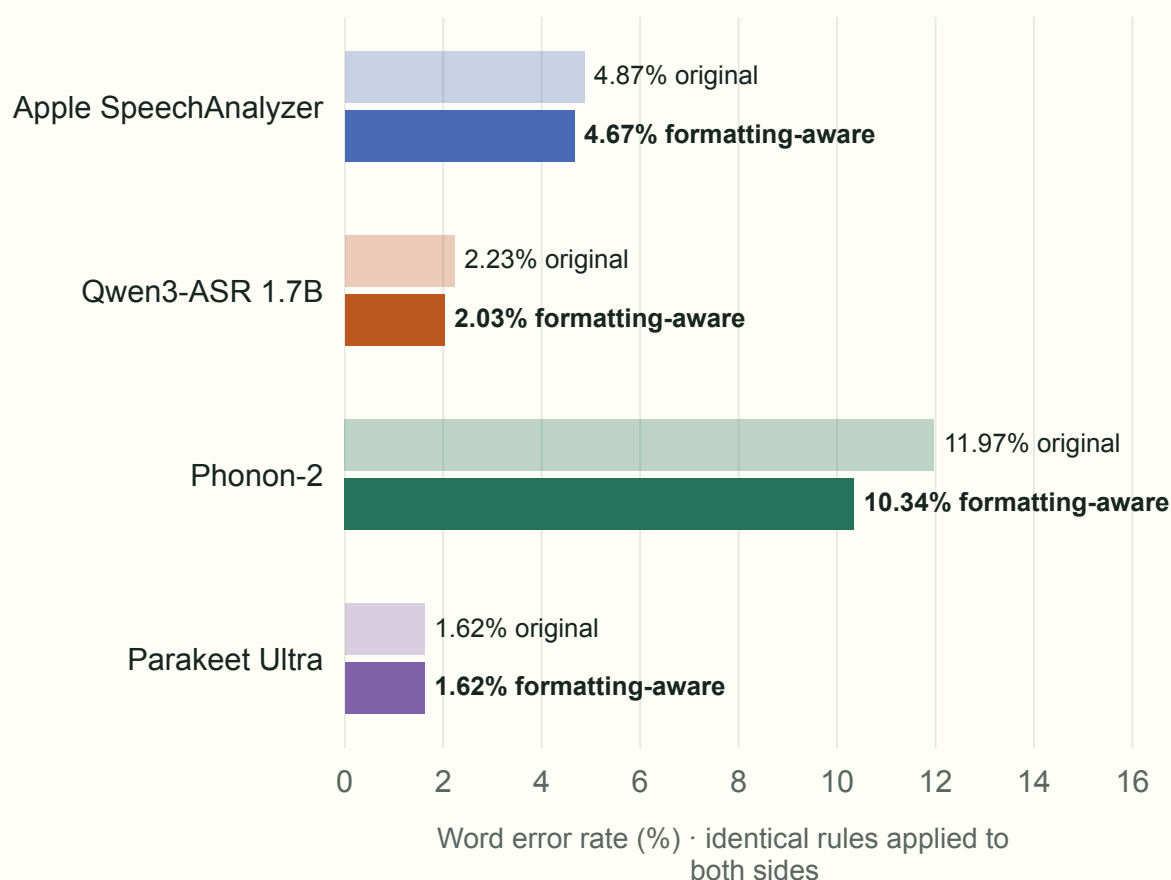


Figure 4. Post-hoc sensitivity analysis of the saved outputs, with identical rules for each reference and hypothesis. These rules do not implement complete Italian number normalization. ·

Apple falls from 24 to 23 errors, Qwen from 11 to 10, and Phonon from 59 to 51. Ultra stays at 8. Formatting explains some errors, but it does not change the ordering on these recordings.

What happens if Qwen runs without Apple’s help?

Our Qwen path uses Apple evidence to influence speech windows and text decisions, as well as to place text on the timeline. We reran Qwen on all 26 clips with that evidence disabled, using the app’s voice-activity-based fallback. This changes several pipeline decisions together, so it cannot isolate the effect of alignment alone.

Qwen scores with and without Apple evidence

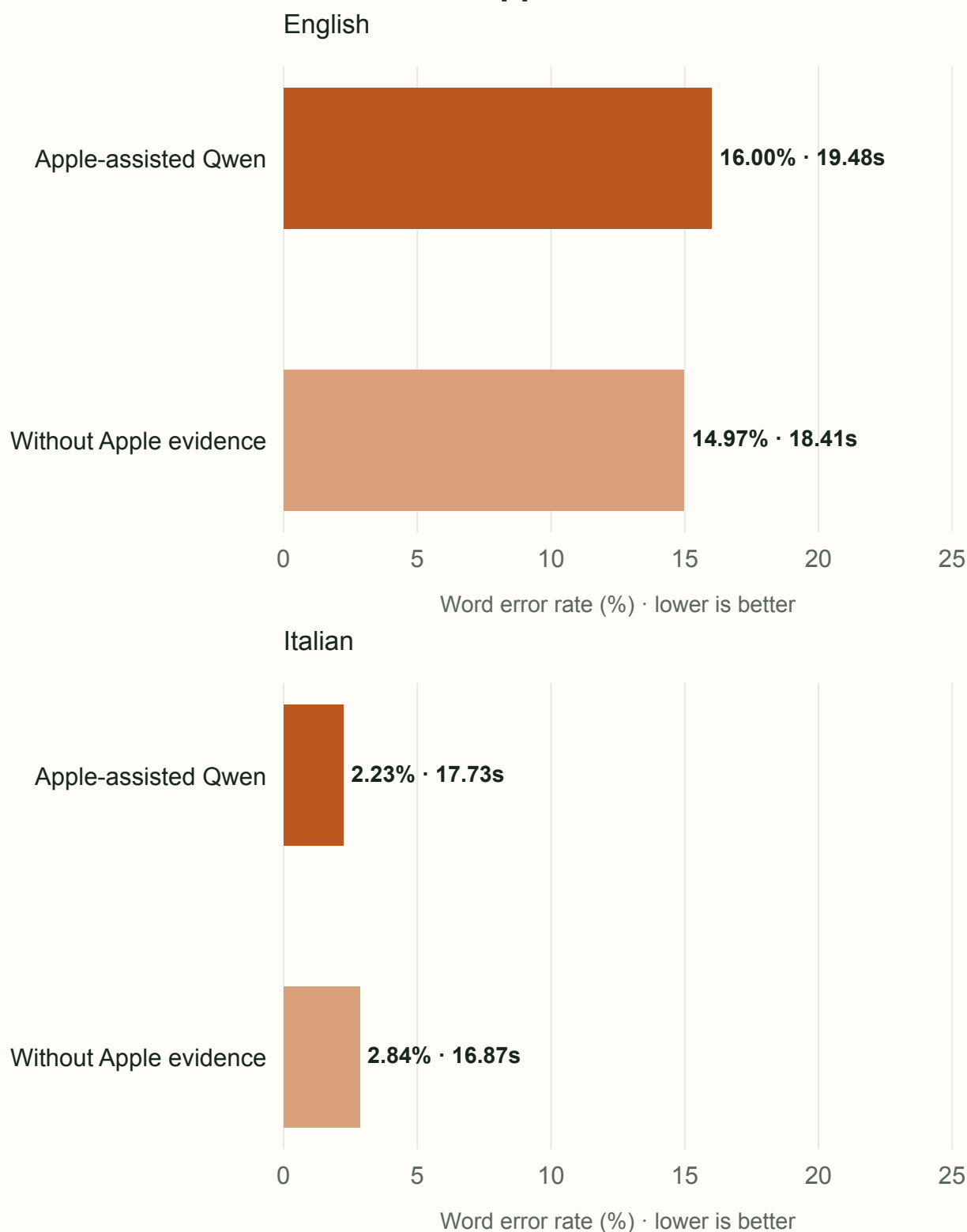


Figure 5. Same build, audio hashes, scorer, and two warm passes. The alternative ran afterward in a separate process, so speed differences remain directional. Shared diarization is excluded. ·

Without Apple evidence, English changes from 124 to 116 errors; Italian changes from 11 to 14. The same pipeline change helps one sample and hurts the other.