Voice Arena

The Voice Arena Dictation Benchmark: Methodology and Results

November 2026 · Whitepaper v1.0 · US English · macOS · voicearena.com

Summary

We tested ten macOS dictation tools: **Apple Dictation, ChatGPT, Claude, Gemini, Granola, Meta AI, Perplexity, Superwhisper, Willow Voice and Wispr Flow**. Every tool dictated the same **761 clips recorded by 87 US English speakers** (4.09 hours of audio, 38,351 words), written to reflect what people actually dictate: messages, emails, AI prompts, code, notes and long-form drafts, with real-world restarts, hedges and self-corrections. An automated harness played each clip into each tool through an identical digital audio path, with the same timing and capture rules: **7,660 dictations in ≈66 hours of live runs** on two identical Macs on 7–8 November 2026.

Our headline metric is **ZER**, the **Zero Edit Rate**: the share of the 7,610 drafts that a user could send as they are, with no edit at all. **Five LLM judges from five model providers**, each at its highest reasoning effort, corrected every draft, and a draft counts as zero-edit when a **3-of-5 majority** makes no change. A secondary metric, AMEND, counts the characters that do need fixing. The paper documents every step: the judge method (section 8), how verdicts are combined (section 9), each judge's results (section 10.3) and the exact prompt (Appendix B).

Wispr Flow has the highest ZER: **79.4%** of its drafts could be sent as-is, and it ranks first under every judge. **Gemini** (75.8%) and **Willow Voice** (75.7%) are statistically tied for second, and **Apple Dictation**, the only on-device engine, has the lowest ZER (34.6%). Full results are in section 10.

Tools	10 macOS dictation tools, each at its out-of-the-box settings (Appendix A)
Data	761 speech clips + 5 silence clips · 87 speakers · 625 prompts · 4.09 h of audio
Run dates	7–8 November 2026
Live runs	7,660 dictations · ≈66 h of running time
Scoring	7,610 drafts · ZER (headline): 3-of-5 majority of Claude Opus 5.5, GPT-6 Astra, Gemini 4 Argon, Muse Wonder, Kimi K3.1 Pro · AMEND (secondary): Claude Opus 5.5
Judge workload	25,879 judge responses · 51.5M input and 26.9M output tokens

Contents

1 Introduction 4
1.1 The round at a glance 4
1.2 What we measure, and why 4
2 Benchmark data 5
2.1 How the benchmark set is built 5
2.2 The clips used in this round 7
2.3 Two dataset samples 7
3 Test environment 10
3.1 System changes made for the evaluation 11
4 The playback harness 12
4.1 How a clip is run 12
4.2 Parameters (identical for every tool) 13
4.3 The capture target 13
4.4 Safeguards 14
5 Tools and configuration 14
6 Running the evaluation 16
6.1 Run time 16
6.2 What was captured 17
7 Scoring: ZER and AMEND 17
7.1 From HTER to AMEND 17
7.2 The judge's task 18
7.3 Self-corrections 18
7.4 Normalisation 18
7.5 The mathematics of ZER and AMEND 19
7.6 What counts as an edit: real examples 19
8 The judge panel 20
8.1 Five judges, five providers 20
8.2 Configuration 21
8.3 How the panel and prompt were chosen 21
9 Consensus: combining five judges 22
9.1 ZER by majority vote 22
9.2 Consensus AMEND by the median judge 22
9.3 Worked examples 22
9.4 How often the judges agree 23

9.5 Robustness of the aggregation 23
9.6 Judge similarity study 24

10 Results 25
10.1 Final leaderboard 25
10.2 Reading the ranking 26
10.3 Results by judge 28
10.4 Results by application and behaviour 30
10.5 Supporting measurements 32

11 Limitations 33
11.1 Limitations of this round 33
11.2 Differences from the planned benchmark design 34

12 Reproducing the work 35

References 35

A Appendix A: Per-tool setup 36
A.1 Apple Dictation (macOS 27.2) — Mac A 36
A.2 Claude desktop 4.23118.0 — Mac A 37
A.3 Wispr Flow 1.6.1400 — Mac A 38
A.4 ChatGPT 26.1103.41825 — Mac A 39
A.5 Gemini 1.141.3.1394 — Mac A 40
A.6 Granola 7.749.1 — Mac B 42
A.7 Willow Voice 2.6.15 — Mac B 44
A.8 Superwhisper 3.2.1 — Mac B 45
A.9 Meta AI 12.0 — Mac B 46
A.10 Perplexity 26.43.0 — Mac B 47

B Appendix B: The judge prompt 49

1 Introduction

Voice Arena benchmarks speech technology on the tasks people actually use it for. This whitepaper documents, end to end, how we evaluated ten macOS dictation tools on 7–8 November 2026: how the benchmark data was built and recorded, how every tool was driven through an identical automated playback harness, how each draft was scored by a panel of five LLM judges, and how the leaderboard is built from them: ZER by their majority, AMEND from Claude Opus 5.5. Every figure in this paper is computed from the run data; nothing is estimated by hand.

1.1 The round at a glance

Tools evaluated	10 macOS dictation tools: Apple Dictation, ChatGPT, Claude, Gemini, Granola, Meta AI, Perplexity, Superwhisper, Willow Voice, Wispr Flow
Run dates	Main runs **7–8 November 2026** (setup and pre-run tests from 6 November)
Benchmark clips	**761** speech clips + 5 silence clips; 87 speakers; 625 distinct prompts
Audio	**4.09 h** per tool, **40.9 h** dictated in total; 38,351 words spoken per pass
Live runs	**7,660 dictations** (10 tools × 766 clips) in **≈66 h** of running time on two identical Macs, plus ≈3.8 h of setup and pre-run tests
Coverage	Every clip captured for every tool
Drafts scored	**7,610** (761 speech clips × 10 tools); 5,176 unique judge requests after removing identical drafts
Judges	5 LLMs from 5 providers at maximum reasoning effort: Claude Opus 5.5, GPT-6 Astra, Gemini 4 Argon, Muse Wonder, Kimi K3.1 Pro
Judge workload	**25,879** judge responses, **38,049** per-draft judgements; 51.5M input and 26.9M output tokens (24.3M of them reasoning)
Headline metric	**ZER**, Zero Edit Rate (share of drafts needing no edit), by 3-of-5 majority of the 5 judges; 95% CIs from a speaker-level bootstrap
Secondary metric	AMEND (share of characters to edit), scored by Claude Opus 5.5, with the 5-judge consensus AMEND reported alongside
Scoring package	amend-va 0.2.1 [2] (prompt fingerprint d26616976626), whose default judge is Claude Opus 5.5 at max effort
Open source	Harness: github.com/VoiceArenaAI/Dictation-harness (MIT) · metric and judge prompt: github.com/VoiceArenaAI/AMEND (section 12)

1.2 What we measure, and why

Dictation is not transcription. Dictation apps remove hedges and fillers, resolve self-corrections (“Tuesday, no wait, Wednesday”) and format numbers, dates and lists, so the right output is a reformulation of what was said rather than a verbatim transcript, a view that recent work on grounded generative speech recognition also takes [4]. Scoring them only against a verbatim transcript penalises exactly the cleanup users want, while scoring them against one fixed clean text penalises every legitimate way of writing the same message. Every clip therefore carries two references:

- **Spoken (verbatim):** exactly what was said, including restarts, hedges and self-corrections.

- **Intended (clean):** the message the speaker meant to send, with hedges and restarts removed, self-corrections resolved and spoken forms written out (“three thirty pm” → “3:30 PM”).

Our headline question is practical: *could the user send this draft as it is?* That is what **ZER**, the Zero Edit Rate, measures: the share of drafts that need no edit at all (section 7). The secondary metric, **AMEND**, measures how much a user would have to fix when a draft does need editing. A judge model sees both references and the tool's draft and makes only the edits a user would have to make; a draft with no edits counts toward ZER, and AMEND counts the characters changed. Rewording that keeps the meaning, formatting, digits versus words and a kept self-correction cost nothing; wrong words, missing or invented content, leftover fillers and wrong names, numbers or code cost exactly the characters needed to fix them.

How to read this paper. Sections 2–6 describe the data and the live runs. Section 7 defines ZER and AMEND, section 8 the judge panel and section 9 the 3-of-5 consensus and the similarity study of the five judges (9.6). Section 10 reports the results, section 11 the limitations and section 12 how to reproduce the work. Appendix A documents every in-app setting with screenshots; Appendix B reproduces the judge prompt verbatim.

2 Benchmark data

2.1 How the benchmark set is built

The sentences reflect how people actually use dictation tools day to day. Most dictation goes into a few kinds of writing, so the set is built around them:

- **AI prompts:** instructions to ChatGPT, Claude or coding assistants, often long and loosely structured.
- **Messages:** short Slack and WhatsApp replies, and longer emails to colleagues.
- **Notes and lists:** to-dos, shopping lists, reminders, meeting notes.
- **Technical text:** sentences with code and developer terms, as dictated into IDEs and developer chats.
- **Long-form drafts:** dictations of 60 s or more, such as document drafts or summaries.

Each sentence is written for the kind of app it would realistically be dictated into. Sentences use everyday language, and include spoken numbers, dates, emails and URLs. They range from short replies to long-form dictation, matching real usage. Each has one clear intended message, and all personal details are fictional.

Step 1. Define the grid. Every prompt is assigned one target application and one primary behaviour, and coverage is balanced across the grid before generation.

Behaviour	What the prompt contains	Example target apps
Plain message	A clean spoken message with no disfluency; checks that clean input passes through unchanged	Slack, WhatsApp Web
Punctuation and casing	Questions, lists of clauses, quoted speech, sentence boundaries	Slack, Gmail
Paragraphing and structure	Greeting, body, sign-off; spoken "new paragraph" cues	Gmail
Lists	Implicit lists ("I need eggs, milk and bread") and spoken numbering	Notes app

Behaviour	What the prompt contains	Example target apps
Spoken forms	Dates, times, money, numbers, emails and URLs said aloud	Gmail, Calendar notes
Self-corrections	"Tuesday, no wait, Wednesday", "scratch that", multi-step corrections	Slack, Gmail
Restarts and hedges	Word-level restarts ("I think we, we should"), "I mean", "like", "kind of"	Slack, WhatsApp Web
Technical text	Code and developer terms in a sentence	VS Code, Slack
Long-form	Drafts and meeting notes of 60 s or more	Notes app, Google Docs

Step 2. Write the spoken prompt. Prompts are drafted from the grid (LLM-assisted, then human-edited) as the words the reader will say:

- No "um", "uh" or similar fillers, since readers can't say them naturally from a script.
- Self-corrections, word-level restarts and hedges are written in explicitly where the behaviour calls for them.
- Spoken forms are written as spoken ("three thirty pm", "john dot smith at gmail dot com").
- Lengths: short prompts are 3–15 words, standard prompts 15–80 words, and long-form prompts run 60 s or more when read.
- No real personal data: names, emails and numbers are fictional.

Step 3. Write the references. Each prompt gets its references when it is written, not after recording.

- **Resolved intended text:** the message the speaker means to send, as described in section 1.2.
- **Literal intended text** (self-correction prompts only): the same cleanup, but with the correction phrase kept ("Let's meet Tuesday, no wait, Wednesday at 3.").
- **Style guide:** a written guide fixes the conventions references follow (time and date formats, list style, sign-offs). It also lists acceptable variants that never count as errors ("3pm" vs "3 PM", the Oxford comma).

Step 4. Automatic checks. Every prompt and reference pair is screened before human review:

- duplicates and near-duplicates are removed (n-gram overlap against the whole set);
- prompts contain no fillers, and self-correction prompts contain an explicit correction phrase;
- every fact in the resolved reference is present in the prompt, and nothing is added (an LLM check plus string matching for numbers and names);
- length is within the band for the prompt type, and grid coverage stays balanced.

Step 5. Human verification. Two reviewers check every prompt independently:

- **Speakability:** can it be read aloud naturally on the first attempt?
- **Reference correctness:** does the resolved reference match what a speaker would mean, and follow the style guide?
- **Single intent:** is there exactly one reasonable intended message? Ambiguous prompts are rewritten or dropped.

Disagreements go to an adjudicator.

Recording. US English speakers read the verified prompts in a quiet room at a natural dictation pace. Each clip is checked for misreads. A clip whose speech departs from the prompt is re-recorded, or its verbatim reference corrected, so the references stay valid.

Prompt types, and why every clip has its references in advance. Every prompt carries a type label: *passage*, *topic* or *sentence*. The label describes how the written prompt was framed, not how it was recorded: a topic prompt, like a passage, is a fixed, verified text that the speaker reads aloud. No clip in this round is unscripted speech. Both references therefore exist before recording: the verbatim reference is the prompt as written, and the intended reference was written with it (step 3). We checked this on the data: for every one of the 761 clips, including the 336 topic clips, the verbatim reference matches the prompt text word for word, and review confirmed it against the recording.

2.2 The clips used in this round

Speech clips	**761** approved recordings by **87** US English speakers (at most 14 clips per speaker), covering **625** distinct prompts: 691 clips from the 1,000-prompt benchmark set and 70 from its 2,000-prompt v2 extension
Total audio	**4.09 h**. Median clip 12.8 s, mean 19.3 s, shortest 2.6 s, longest 123.1 s
Words	38,351 words spoken; 35,840 words in the intended references
Prompt type	368 passage prompts · 336 topic prompts · 57 single sentences; every clip was read aloud from its written prompt (section 2.1)
Difficulty	301 easy · 227 medium · 233 hard
Length band	92 micro · 371 short · 241 medium · 57 long
Target application	126 personal messaging · 125 work chat · 123 code · 119 AI prompts · 107 email · 104 docs and notes · 57 long-form
Most common target apps	Gmail 77 · iMessage 76 · Slack 58 · ChatGPT 51 · Claude 47 · Outlook 46 · Microsoft Teams 41 · Apple Notes 37 · WhatsApp 37 · Claude Code 36 · Google Chat 36 · Google Docs 32
Behaviours present	246 questions · 126 restarts · 125 self-corrections · 108 lists · 102 hedges · 8 quoted speech
Entities	person names 221 · money 189 · numbers and units 183 · organisations and products 156 · dates 139 · places and addresses 131 · technical terms 117 · times 108 · IDs and codes 106 · phone numbers 31
Silence clips	5 speech-free clips to measure hallucination: digital silence (8 s and 20 s), room tone at −55 dB and −45 dB (10 s each), fan noise at −40 dB (15 s)
Audio format	WAV, 44.1 kHz, mono, 16-bit PCM; every file checked as readable before the runs
Verification	Every clip approved in review; for every clip the verbatim reference was confirmed to match the recording

2.3 Two dataset samples

Two clips from the set, with their references and annotations, followed by what each tool typed and how the five judges scored it. They illustrate the two kinds of difficulty the benchmark is built around: **spoken disfluency** (a stutter and a self-correction) and **exact technical text**.

Listen to the samples. Both recordings are attached to this PDF as WAV files (44.1 kHz, mono): open your PDF reader's attachments panel and play `sample_1_USD-0662.wav` (15.8 s) or `sample_2_USD-0419.wav` (25.7 s). They are exactly the audio every tool heard, before the harness added its 300 ms of padding.

Sample 1 — a work-chat message with a restart and a self-correction

USD-0662 | Target app: Microsoft Teams | Work chat | topic prompt · easy · short | 15.8 s

SPOKEN (VERBATIM REFERENCE)

Can you, can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room four eighteen, no, four sixteen.

INTENDED (CLEAN REFERENCE)

Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.

ANNOTATIONS

Behaviours: self-corrections, restarts, questions · Entities: organization or product, person name, place or address

Tool	Draft as typed	Edits by judge[1]	Zero votes	Consensus edits
Claude	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 418. No, 416.	0 · 0 · 0 · 0 · 0	5/5	0
Gemini	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.	0 · 0 · 0 · 0 · 0	5/5	0
Granola	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.	0 · 0 · 0 · 0 · 0	5/5	0
Meta AI	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.	0 · 0 · 0 · 0 · 0	5/5	0
Superwhisper	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.	0 · 0 · 0 · 0 · 0	5/5	0
Willow Voice	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.	0 · 0 · 0 · 0 · 0	5/5	0
Wispr Flow	Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 416.	0 · 0 · 0 · 0 · 0	5/5	0
Apple Dictation	Can you can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 418 no 416.	8 · 8 · 8 · 0 · 8	1/5	**8**
ChatGPT	Can you? Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 418. No, 416.	8 · 8 · 8 · 0 · 8	1/5	**8**
Perplexity	Can you? Can you call the pharmacy at Mercy West about the order for Mrs. Daniels? It's room 418. No, 416.	8 · 8 · 8 · 0 · 8	1/5	**8**

The speaker restarts (“Can you, can you”) and corrects a room number (“four eighteen, no, four sixteen”). Tools that resolved both, and Claude, which kept the self-correction but dropped the stutter, need no edit: a kept self-correction is never penalised (section 7.3). Tools that kept the stutter lose 8 characters by majority; one judge (Muse Wonder) let it pass, and was outvoted.

Sample 2 — a coding prompt with exact technical terms

USD-0419 | Target app: OpenAI Codex | Code | passage prompt · medium · medium | 25.7 s

SPOKEN (VERBATIM REFERENCE)

Ok so we're moving the FastAPI app's settings over to Pydantic version two. Can you swap the old config class for a settings model that reads from env vars, same names, and have it blow up on startup if the database URL or the Redis URL is missing? Pool size defaults to ten, request timeout thirty seconds. And then fix anywhere that imports the old config.

INTENDED (CLEAN REFERENCE)

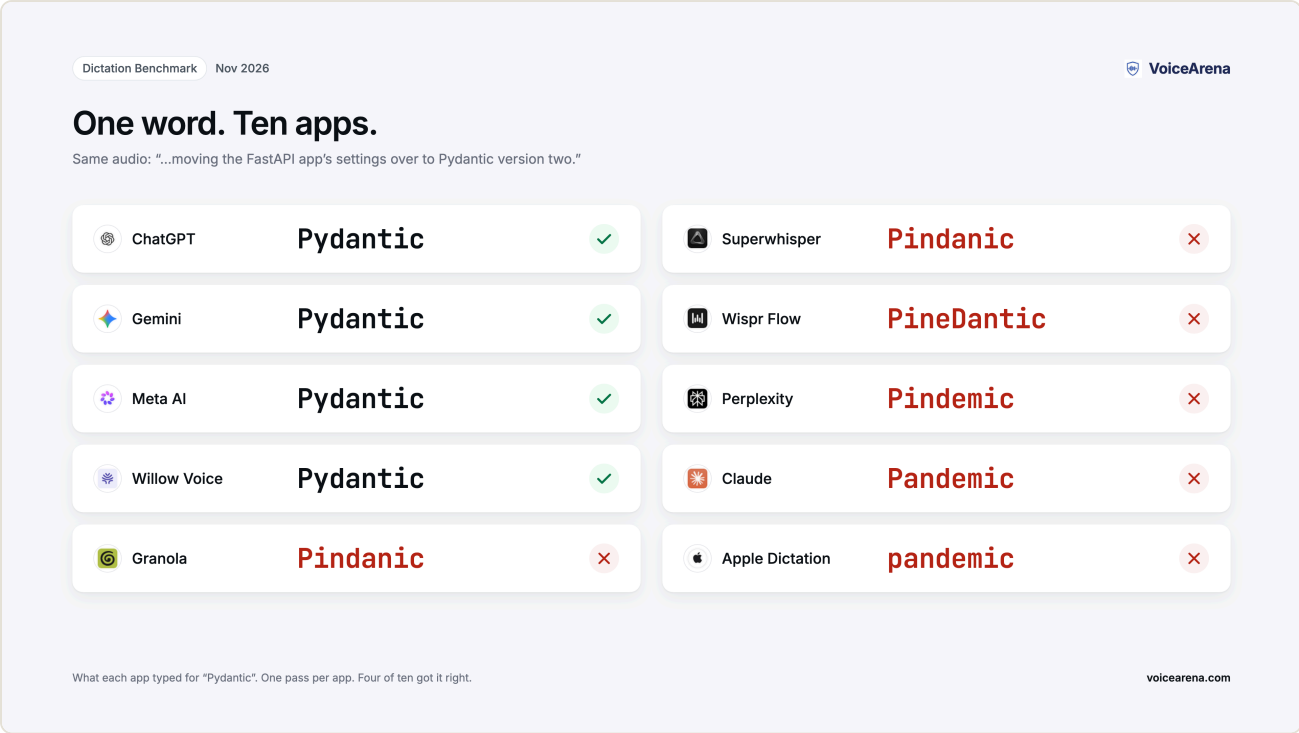
OK, so we're moving the FastAPI app's settings over to Pydantic version 2. Can you swap the old config class for a settings model that reads from env vars, same names, and have it blow up on startup if the database URL or the Redis URL is missing? Pool size defaults to 10, request timeout 30 seconds. And then fix anywhere that imports the old config.

ANNOTATIONS

Behaviours: questions · Entities: number or unit, technical term

Tool	Draft as typed (excerpt)	Edits by judge[1]	Zero votes	Consensus edits
Gemini	… moving the FastAPI app settings over to Pydantic version 2 …	2 · 2 · 0 · 0 · 0	3/5	0
ChatGPT	… moving the FastAPI's app settings over the Pydantic version 2 …	2 · 0 · 2 · 2 · 2	1/5	**2**
Meta AI	… moving the FastAPI app settings over to Pydantic version two …	2 · 2 · 2 · 2 · 2	0/5	**2**
Willow Voice	… moving the FastAPI's app settings over the Pydantic version 2 …	2 · 0 · 2 · 0 · 2	2/5	**2**
Granola	… moving the FastAPI app settings over to Pindanic version 2 …	3 · 3 · 3 · 3 · 3	0/5	**3**
Superwhisper	… moving the FastAPI's app settings over the Pindanic version 2 …	5 · 3 · 5 · 3 · 5	0/5	**5**
Claude	… moving the Fast APIs app settings over the Pandemic version 2 …	9 · 7 · 10 · 9 · 9	0/5	**9**
Wispr Flow	… moving the Fast APIs app settings over to the PineDantic version 2 …	9 · 7 · 9 · 9 · 9	0/5	**9**
Perplexity	… moving the FAST APIs app settings over the Pindemic version 2 …	12 · 10 · 12 · 12 · 13	0/5	**12**
Apple Dictation	… moving the fast API's app settings over the pandemic version two …	19 · 13 · 17 · 20 · 17	0/5	**17**

Library names must be exact: “Pydantic” misheard as “Pindanic”, “pandemic” or “PineDantic”, or “FastAPI” split into “Fast APIs”, costs the characters needed to restore them. Numbers written as words are accepted: Meta AI's “version two” costs nothing, and its 2 edits are for “Pull size” → “Pool size”. [1] Judges in order: Claude Opus 5.5 · GPT-6 Astra · Gemini 4 Argon · Muse Wonder · Kimi K3.1 Pro.



Sample 2 as published: what each tool typed for the word "Pydantic".

3 Test environment

All ten tools ran on two identical MacBook Air laptops on the same network. The runs took place on 7–8 November 2026. Each tool ran entirely on one Mac, and both Macs ran the same harness with the same parameters, on Python 3.12 or 3.13.

Item	Mac A	Mac B
Run dates	7–8 Nov 2026	Same
Tools run	Apple Dictation, Claude, Wispr Flow, ChatGPT, Gemini	Granola, Willow Voice, Superwhisper, Meta AI, Perplexity
Model	MacBook Air (Mac17,3), Apple M5, 10 cores (4 super, 6 efficiency), 16 GB memory	MacBook Air, Apple M5, 10 cores (4 super, 6 efficiency), 16 GB memory
macOS	27.2, build 26B123, as reported by `sw_vers` in every settings snapshot	Same
Power	On the charger for every full run, battery charged. Kept awake with `caffeinate` for the whole run.	Same
Network	Wi-Fi 6 (802.11ax), 5 GHz, 80 MHz channel, WPA2	The same Wi-Fi network, with identical network configuration
Measured network capacity	477 Mbps down / 301 Mbps up (Apple `networkQuality`, measured on Mac A on 8 Nov after the runs)	Not measured separately
Language / region / keyboard	English (US) / United States / U.S. only	Same

Item	Mac A	Mac B
Mac-wide text corrections	All 8 off (see 3.1)	Same
Virtual audio cable	BlackHole 2ch 0.7.1, running at 48 kHz	Same
Default output	Built-in speakers (never BlackHole, so alert sounds can't reach the "microphone")	Same
Default input	Built-in microphone. Switched to BlackHole 2ch only during the Claude, ChatGPT and Gemini runs, and restored automatically.	Built-in microphone. Switched to BlackHole 2ch only during the Perplexity run.
Harness runtime	Python 3.12 or 3.13 via uv 0.13.7; `sounddevice`, `soundfile`, `soxr`, `pyobjc`, `jiwer`, `whisper-normalizer`	Same
Permissions for the app running the harness	Accessibility and Automation (TextEdit, System Events)	Same

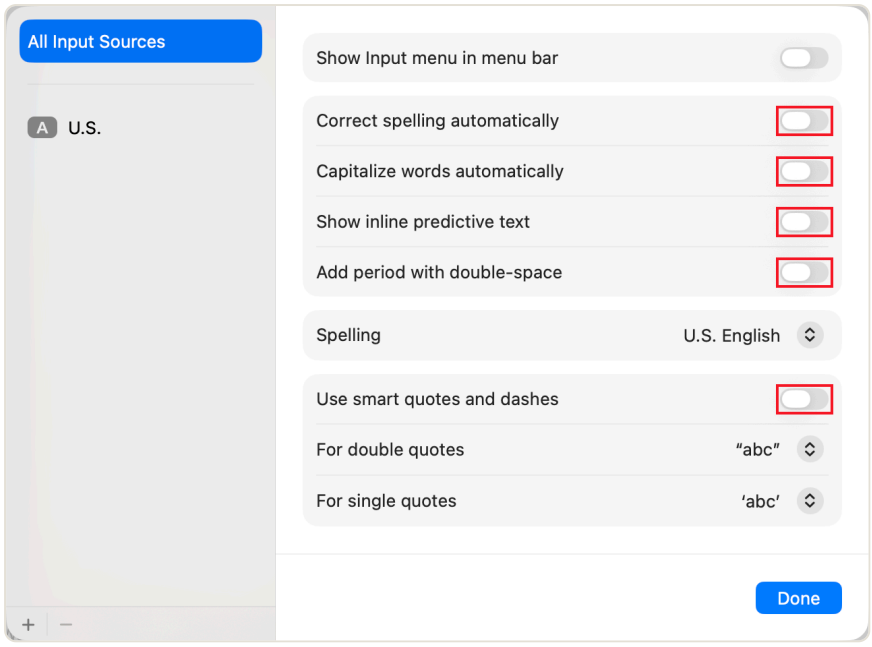
On Mac B, snapshots of the system settings taken before, between and after the runs all matched. On Mac A, the same settings were verified before each tool's run and re-checked by the harness every 5 clips.

3.1 System changes made for the evaluation

The original values were recorded first. A restore script (`restore_mac_settings.sh`) puts the text, language and Dictation-language settings back.

Change	Before → after	Mac
Mac-wide text corrections: auto-capitalisation, double-space period, spelling correction, smart quotes, smart dashes, text replacement, inline prediction, text completion	Auto-capitalisation and double-space period on, the rest at macOS defaults → **all off**. Re-checked every 5 clips; a run stops if any turns back on.	A, B
System language and region	another English locale → **English (US) / United States**	A, B
Keyboard input source	another English layout → **U.S.** only	A, B
Apple Dictation language	another English variant → **English (US)** (`en_US`)	A, B (used only on Mac A)
Apple Dictation on/off	off → **on**	A
Apple Dictation microphone source	Automatic (built-in microphone) → **BlackHole 2ch**	A
BlackHole 2ch 0.7.1	not installed → **installed** (signed and notarized package; Mac restarted)	A, B
Accessibility and Automation for the app running the harness	off → **on** (needed to press buttons, read text and control TextEdit)	A, B
Restart	once, after the language, keyboard and BlackHole changes	A, B

Mac A, as configured for the runs:



Mac A: Keyboard → Text Input → Edit; every automatic text correction off and the keyboard is U.S.

4 The playback harness

To evaluate every tool under identical input conditions, we built an automated playback harness. It routes each clip into the tools through **BlackHole**, a virtual audio device that loops digital audio output back as a microphone input; each tool was set to use it as its microphone. The audio therefore goes from the source file to the tool without a loudspeaker, a physical microphone or room acoustics, and every tool receives the same samples for a given clip.

One shared harness drives every tool. Only the start/stop step, the "is it listening?" check and, for four tools, the place the text lands, differ per tool. The harness is open source under the MIT licence at github.com/VoiceArenaAI/Dictation-harness [3]: one shared engine (`dictation_common.py`), microphone routing (`audio_input.py`) and one driver per tool (`run_<tool>_dictation.py`). Each parameter below is a command-line option whose default is the value used in this round.

4.1 How a clip is run

1. **Prepare the target.** Clear the capture target and bring it to the front (see 4.3).
2. **Start** the tool's dictation. For tools with an on-screen "recording" indicator, wait for it to appear; for tools without one, wait a fixed 0.5 s. Then wait another 0.5 s.
3. **Play** 300 ms of silence, the clip, then 300 ms of silence into BlackHole, so speech starts 300 ms after playback starts. The clip is resampled from 44.1 kHz to 48 kHz with soxr and played on both channels; there is no gain change and no added noise. Nothing plays through the speakers.
4. **Keep listening** for **4.0 s** after the clip ends, so tools still processing the final words aren't cut off.
5. **Stop** the tool.
6. **Wait for the final text.** Many tools produce or revise their output only after dictation stops, so the harness watches the text and saves it once it has stayed unchanged for 2.0 s, or after 30 s at most. The published

harness reads the text through Accessibility every 10 ms; in the main run it was read about every 333 ms, which affects when the final text is seen but not what it is.

7. **Save** the result, wait 1.0 s, and move to the next clip.

4.2 Parameters (identical for every tool)

Parameter	Value
Padding (`--pad-ms`)	300 ms of silence before and after each clip
Resampling	44.1 → 48 kHz (soxr), mono duplicated to 2 channels, no gain change
Ready signal	the tool's own indicator where it has one; otherwise a fixed 0.5 s
Wait after listening starts (`--start-delay`)	0.5 s
Tail, keep listening after the clip (`--tail`)	4.0 s. An early 2 s tail cut off final words on some recordings, so it was raised before any reported run.
Text must be unchanged for (`--settle`, `--timeout`)	2.0 s (maximum wait 30 s)
Pause between clips (`--gap`)	1.0 s
Tries per clip (`--attempts`)	3 (Gemini ran with 5, see A.5). The pause before a retry is 2 s, then 15 s before the last try, as on Mac A in this round.
Safety check and backup (`--batch-size`)	every 5 clips
Pre-run test	each tool was tested on 3 clips before its full run (Gemini on 2)
Runs per tool	one complete pass over all 766 clips

4.3 The capture target

TextEdit (Apple Dictation, Wispr Flow, Granola, Willow Voice, Superwhisper, Meta AI) — the text lands in a new scratch TextEdit document:

- It is addressed by window ID, so no other document is ever touched. If TextEdit closes it, a new one is created.
- It is cleared before every clip.
- These TextEdit features are switched **off and verified** on it: Smart Quotes, Smart Dashes, Text Replacement, Smart Links, Data Detectors, Correct Spelling Automatically, Check Spelling While Typing and Check Grammar With Spelling. If any can't be switched off or read, the run stops.
- TextEdit's Open-file panel, which appears when TextEdit starts with no document, is cancelled automatically, because it blocks scripting.

The tool's own prompt box (Claude, ChatGPT, Gemini, Perplexity) — these tools have no system-wide dictation, so their own input box was the capture target:

- The harness presses the tool's microphone and stop buttons through macOS Accessibility, and reads the box the same way.
- **Nothing is ever sent.** Return and the send buttons are never pressed; the drivers refuse to press any button whose label contains "send" or "submit". The box is cleared before every clip.

The harness changes nothing in the captured text except collapsing whitespace (including line breaks) to single spaces. Some tools keep their own record of each dictation: Wispr Flow (raw speech-to-text and its final text), Willow and Superwhisper (raw and cleaned text). For these, that record was captured too, as a cross-check.

4.4 Safeguards

- **Focus guard:** if any other app comes to the front during a clip, the attempt is discarded and retried, so nothing is captured from the wrong window.
- **Settings re-check:** every 5 clips, the run re-checks:
  - the Mac-wide text settings;
  - that the system output isn't BlackHole;
  - each tool's own settings (microphone, language, mode, plan, usage, as applicable). These are also recorded on every row of the results.
- **Battery guard:** the run stops if the Mac is on battery below 20%.
- **Dead-mic guard:** the run stops if 3 speech clips in a row produce no text.
- **Never lose data:**
  - the results file is rewritten safely (temporary file + rename) after every clip;
  - rolling backups are kept every 5 clips;
  - Mac A also saved a timestamped copy every 5 minutes, and Mac B appended every finished clip to a journal;
  - a rerun resumes from the first unfinished clip, and a failed or interrupted clip is redone in full.
- **Stop for review:** a clip that fails all its tries, or a settings change, stops the run with all data saved.
- **Supervisors:** for services that throttle or fail transiently, a supervisor waited and resumed instead of skipping clips (run-time wrappers used in this round; they are not part of the published harness):
  - Mac B: 10 → 60 minutes;
  - ChatGPT: 35 minutes plus a one-clip test before resuming;
  - Gemini: 2 minutes.

5 Tools and configuration

Mac A

	Apple Dictation	Claude	Wispr Flow	ChatGPT	Gemini
Version	built into macOS 27.2 (26B123)	4.23118.0	1.6.1400	26.1103.41825	1.141.3.1394
Plan	—	Ultra	Pro (monthly)	Plus	Free
Speech engine	**On-device** (Apple's English model; see A.1)	Cloud (not named)	Cloud (not named) + AI formatting	Cloud (not named)	Cloud (not named)
Cleanup	None	Light (punctuation, digits)	Auto Cleanup "Light"	Light (punctuation, digits)	Built in (removes fillers and repeats)

	Apple Dictation	Claude	Wispr Flow	ChatGPT	Gemini
Microphone used	BlackHole 2ch (Dictation microphone source)	System default, set to BlackHole 2ch for the run	BlackHole 2ch	System default, set to BlackHole 2ch for the run	System default, set to BlackHole 2ch for the run
Start / stop	TextEdit Edit → Start / Stop Dictation	"Dictate" / "Finish dictation" buttons	`wispr-flow://` start / stop hands-free links	"Dictate" / "Stop dictation" buttons	"Use microphone (⌘D)" / "Stop (⌘D)" buttons
"Listening" signal	Menu item changes to "Stop Dictation"	"Finish dictation" button appears	none (fixed 0.5 s wait)	"Stop dictation" button appears	"Stop" button appears
Where the text lands	TextEdit	Its own prompt box (never sent)	TextEdit	Its own prompt box (never sent)	Its own prompt box (never sent)
Inserts text	**live, while listening**	**live, while listening**	after stop	after stop	after stop
Language	English (US)	English	English	— (no setting; system English (US))	English
Reads the screen for context	no	no	on (default)	no	no

Mac B

	Granola	Willow Voice	Superwhisper	Meta AI	Perplexity
Version	7.749.1	2.6.15	3.2.1	12.0	26.43.0
Plan	Free	Pro ($15/month)	Pro ($8.49/month)	Free	Free
Speech engine	Cloud (not named)	"Frontier Pro" (cloud)	S1-Voice (cloud)	Cloud (not named)	Cloud (not named)
Cleanup	Built in	Built in	S1-Language ("smallest possible changes")	Built in	None (near verbatim)
Microphone used	BlackHole 2ch	BlackHole 2ch	BlackHole 2ch	BlackHole 2ch	System default, set to BlackHole 2ch for the run
Start / stop	fn twice / fn once	⌘ + backtick	`superwhisper://record` link	Right Option	Microphone / stop button in its search box
"Listening" signal	Its "Dictation" pill	none (fixed 0.5 s wait)	Its recording window	Its dictation overlay	Stop button replaces the mic button
Where the text lands	TextEdit	TextEdit	TextEdit	TextEdit	Its own search box (never sent)
Inserts text	after stop	after stop	after stop	after stop	**live, while listening**
Language	English	English	English	English (US)	English (system)

	Granola	Willow Voice	Superwhisper	Meta AI	Perplexity
Reads the screen for context	on (default)	on (default)	off (default)	off	no

Every tool was otherwise left at its out-of-the-box settings, except where a setting would:

- change the audio;
- steal focus;
- prevent automation;
- or, for the plans, make a free tier test a weaker model or stop the run partway.

The plans were chosen on that basis:

- **Willow and Superwhisper:** paid plans, because their free plans use weaker models (Willow "Frontier Mini"; Superwhisper only on-device Whisper with no cleanup).
- **Wispr Flow:** Pro, because its free tier caps weekly words below the 38,351 words in the clip set.
- **ChatGPT:** Plus, because the free tier limits how much can be dictated.
- **Claude:** Ultra.

Every in-app change, with annotated screenshots, is documented per tool in Appendix A.

6 Running the evaluation

6.1 Run time

The main runs took place on 7–8 November 2026, after setup and pre-run tests that began on the night of 6 November. All times are local.

Tool	Mac	Setup + pre-run test	Full run (start – end, local time)	Running time	Per clip	Sessions	Retries in final results
Claude	A	~35 min (incl. installing the app)	7 Nov 00:34 – 07:19	6 h 42 min	31.5 s	3	7
Wispr Flow	A	~25 min (incl. plan upgrade)	7 Nov 07:54 – 14:28	6 h 33 min	30.8 s	1	3
Apple Dictation	A	~12 min	7 Nov 14:40 – 21:20	6 h 27 min	30.3 s	4	2
ChatGPT	A	~25 min (incl. microphone permission)	7 Nov 22:12 – 8 Nov 05:39	6 h 52 min	32.3 s	4	4
Gemini	A	~13 min (2-clip test)	8 Nov 05:57 – 14:44	6 h 48 min	32.0 s	8	4
Granola	B	~45 min	7 Nov 00:42 – 07:19	6 h 37 min	31.1 s	—	1

Tool	Mac	Setup + pre-run test	Full run (start – end, local time)	Running time	Per clip	Sessions	Retries in final results
Willow Voice	B	~15 min (+12 min fixing a false start)	7 Nov 08:07 – 14:39	6 h 32 min	30.7 s	—	1
Superwhisper	B	~25 min	7 Nov 15:10 – 21:53	6 h 35 min	30.9 s	—	0
Meta AI	B	~15 min	7 Nov 22:29 – 8 Nov 05:03	6 h 34 min	30.9 s	—	1
Perplexity	B	~15 min	8 Nov 05:33 – 13:16	≈ 6 h 12 min	≈ 29 s	—	14
Total		**~3.8 h**		**≈ 66 h** (A: 33 h 22 min, B: ≈ 32.5 h)			

- **Running time** excludes time when no run was active (stops, waits, a session restart).
- **Per clip** is running time divided by 766. Each clip takes its own length plus about 11 s of fixed steps: tail, settle, gap, start and stop.
- **Faster per clip:** Apple Dictation and Perplexity, because their text is already final when they stop.
- **Retries in final results:** clips whose saved result needed more than one try (for example, after a focus loss).

6.2 What was captured

Capture. All 766 clips are status "ok" for all ten tools. There are no empty speech transcripts, and every row records the settings in effect. Each row has:

- the transcript;
- the attempt count and any errors;
- the tool's settings at the time, plus, where available, the tool's own raw text.

7 Scoring: ZER and AMEND

7.1 From HTER to AMEND

Both metrics come from the same step: a judge makes the minimum edits a draft needs before it can be sent. **ZER**, the headline metric, asks whether a draft needs any edit at all; **AMEND**, the secondary metric, counts how many characters it needs. The edit step itself is AMEND's method, described here.

AMEND (Aggregate Minimum Edits Needed per Draft) measures how much a user would have to fix in a dictation tool's output before it is usable. It is inspired by **HTER** (Human-targeted Translation Edit Rate) from machine-translation evaluation, where a human post-edits a system's output as little as possible until it is correct, and the score is the edit rate between the output and its own post-edited version. Because the target is built from the output itself, a correct draft worded differently from the reference costs nothing; only the edits a person actually needs are counted. AMEND applies the same idea to dictation, with an LLM judge in the role of the post-editor.

	HTER	AMEND
Output scored	MT translation	Dictation tool's draft
Ground truth	Reference translation	Intended text, plus the spoken text
Who builds the targeted reference	Human annotator	LLM judge: Claude Opus 5.5 for the leaderboard's AMEND; five judges, by majority, for ZER (sections 8–9)
Edit rate	TER (word level, with shifts)	Character edit rate (Levenshtein, via jiwer)

7.2 The judge's task

For each draft the judge receives three texts: the **spoken** text (verbatim), the **intended** text and the tool's **draft**. It returns the draft with the **minimum edits** needed for it to serve the same purpose as the intended text, as a JSON object `{"fixed": "…"}`. The judge never counts anything; counting is done in code. The full prompt is reproduced in Appendix B.

The judge fixes	The judge leaves alone
• Misrecognised or wrong words (“weight” → “wait”) • Missing or extra content of any length, including generated or invented text • Filler and discourse words the intended text left out (“like”, “you know”, “I mean”, a leading “Okay”/“So”) • Wrong names and wrong values of numbers or dates • Anything that must be exact: code identifiers, file names, commands, versions in code, emails, URLs, product and proper names • Wrong or slang units (“512 megs” → “512 MB”)	• Rewording with the same meaning: articles, singular/plural, tense, contractions, synonyms, word order • Punctuation, capitalisation, line breaks, bullets, lists, markdown • How ordinary numbers are written (“4” vs “four”, “$618.40” vs “618 dollars and 40 cents”), as long as every digit is right • Standard abbreviations, hyphenation, US vs UK spelling • Small grammar slips that do not change the meaning, and a missing courtesy word (“please”)

7.3 Self-corrections

Self-corrections are lenient. If a draft keeps a self-correction from the spoken text (“Tuesday, no wait, Wednesday”), including its cue words, that is not an error and nothing is removed. A draft that ends up with the **wrong final value** (it kept “Tuesday” and dropped “Wednesday”) is still fixed. “I mean” counts as a self-correction cue only when it introduces a correction; anywhere else it is a filler.

7.4 Normalisation

Before counting, both the draft *x* and the judge's corrected text *y* are normalised identically: Unicode NFKC, curly quotes straightened, lowercase, runs of dots collapsed, and sentence punctuation stripped from the start and end of each word (`. , ; : ! ?`, quotes, brackets, `* ` _ ~`, bullets, dashes, ellipses); whitespace is collapsed. Punctuation that carries meaning is kept: `% / $ @ # & + =`, leading dots and dashes (`.env`, `--verbose`) and anything inside a token (`24.9`, `1,450`, `node:20-alpine`, `https://acme.com/docs`).

7.5 The mathematics of ZER and AMEND

Let $\hat{x}_d$ and $\hat{y}_d$ be the normalised draft and corrected text of draft d. A character-level Levenshtein alignment of the corrected text (reference) against the draft (hypothesis) gives hits H, substitutions S, deletions D and insertions I; spaces count as characters. The edits needed and the reference length are

$$e_d = S_d + D_d + I_d \qquad c_d = H_d + S_d + D_d = \left|\hat{y}_d\right|$$

ZER, the **Zero Edit Rate** and our headline metric, is the share of a tool's N drafts that need no edit, i.e. could be sent as they are:

$$\mathrm{ZER} = \frac{1}{N}\sum_{d=1}^{N} \mathbb{1}[e_d = 0] \times 100\%$$

ZER (Zero Edit Rate) is the percentage of a tool's dictations that needed **no edit at all** to say what the speaker meant. A draft counts only if the judges make zero character edits under the rules of section 7.2: wrong words, missing or extra content, leftover fillers and wrong names, numbers or exact terms all break it; punctuation, capitalisation, line breaks, lists and how ordinary numbers are written do not. With five judges, a draft counts when at least three of them make no edit (section 9.1). Higher is better. The amend-va package reports the same number under the name AMEND@0.

The secondary metric, **AMEND**, is pooled over all N drafts of a tool, so a long draft weighs more than a short one (as in corpus-level WER and TER):

$$\mathrm{AMEND} = \frac{\sum_{d=1}^{N} e_d}{\sum_{d=1}^{N} c_d} \times 100\%$$

Characters, not words, are counted because they track the user's typing effort: fixing “test” to “tests” costs one character, not a whole word. If the corrected text is empty, e_d is the draft's length; if the draft is empty, the whole corrected text is counted as missing. In this round every corrected text was non-empty.

Worked example. Draft “I'm 42, and I'm running the treadmill at about 4.8 mi an hour.” against the intended “…I run on a treadmill at about 4.8 mph.”: the judge leaves it unchanged (same meaning; “mi an hour” is a standard abbreviation), so e = 0. Draft “It's room 418. No, 416.” is a kept self-correction, so e = 0. Draft “…over to the PineDantic version 2…” becomes “…over to the Pydantic version 2…”: after lowercasing, “pinedantic” → “pydantic” is one substitution (i → y) and two deletions (n, e), so e = 3 character edits. (In Sample 2, section 2.3, the judges also fixed “Fast APIs”, so the whole draft needs more.)

7.6 What counts as an edit: real examples

Eleven drafts from this round on which **all five judges agreed** exactly. The first five differ from the intended text but need no edit, so they count toward ZER; the last six need fixing, and the last column gives the characters changed (the consensus edits e_d).

Verdict	Tool	What the tool typed	What the speaker meant	Judges' fix	Chars	Why
No edit	Apple Dictation	Running a little late go ahead and eat without me	Running a little late, go ahead and eat without me.	unchanged	0	Punctuation and capitalisation are not edits

Verdict	Tool	What the tool typed	What the speaker meant	Judges' fix	Chars	Why
No edit	ChatGPT	Room two oh four's projector's out again. … It's number eight eight three two.	Room 204's projector is out again. … it's #8832	unchanged	0	Numbers in words are fine when every digit is right
No edit	ChatGPT	Hardware store list: two boxes of three-inch deck screws, a gallon of exterior primer, …	Hardware store list: (bulleted) 2 boxes of 3-inch deck screws · A gallon of exterior primer · …	unchanged	0	Lists, line breaks and layout are not edits
No edit	Claude	Be there at 6. No, 6:30.	Be there at 6:30.	unchanged	0	A kept self-correction is not an error
No edit	ChatGPT	Wanna grab brunch Sunday? No wait, Saturday. Anywhere near you works.	Want to grab brunch Saturday? Anywhere near you works, …	unchanged	0	Different wording with the same meaning
Edit	Willow Voice	Water the firms Wednesday.	Water the ferns Wednesday.	firms → ferns	2	Misrecognised word
Edit	Granola	Locker combo is 12:38.	Locker combo is 12-30-8.	12:38 → 12-30-8	3	Wrong number value
Edit	Claude	… the Django migrations are failing on C1.	… the Django migrations are failing on CI …	C1 → CI	1	Exact technical terms must match
Edit	Claude	Jest is like hanging after the auth tests.	Jest is hanging after the auth tests, …	delete “like”	5	Leftover filler word
Edit	Gemini	… on Saturday the 16th at 9:00 a.m. My number is 415-555-0162.	… on Saturday the 16th at 9 AM. We're away the 9th. My number is …	insert “We're away the 9th.”	19	Missing content
Edit	Willow Voice	Write me a Python script that cleans up our sales export … [followed by a complete Python script]	Write me a Python script that cleans up our sales export …	delete the generated script	1,814	Content the speaker never said

8 The judge panel

8.1 Five judges, five providers

A single LLM judge brings its own blind spots and its own strictness. We therefore score every draft with **five judges from five different model providers**, each at the highest reasoning effort it offers. Whether a draft needs any edit, and so ZER, is decided by their majority (section 9). The leaderboard's AMEND is **Claude Opus 5.5**'s own score, the judge that the open-source `amend-va` package [2] uses by default, so anyone can reproduce it with the published package; the five-judge consensus AMEND (section 9.2) is reported next to it. Four of the judges share a provider with one of the tested tools (Claude, ChatGPT, Gemini, Meta AI); section 10.3 checks that no judge favours its own provider's tool.

Judge	Provider	Reasoning effort	Requests answered	Access mode	Input tokens	Output tokens (reasoning)	Served by
Claude Opus 5.5	Anthropic	max	5,176 / 5,176	batch 4,303 · regular 873	14.0M	8.0M (7.4M)	Anthropic, Claude Platform on AWS
GPT-6 Astra	OpenAI	max	5,176 / 5,176	batch 4,078 · regular 1,098	8.9M	2.6M (2.2M)	OpenAI
Gemini 4 Argon	Google	high (its maximum)	5,175 / 5,176	batch 1,989 · regular 3,186	10.1M	6.1M (5.6M)	Google, Google AI Studio

Judge	Provider	Reasoning effort	Requests answered	Access mode	Input tokens	Output tokens (reasoning)	Served by
Muse Wonder	Meta	max	5,176 / 5,176	regular 5,176	8.8M	7.0M (6.4M)	Meta
Kimi K3.1 Pro	Moonshot AI	max	5,176 / 5,176	regular 5,176	9.8M	3.2M (2.8M)	Moonshot AI
Total			**25,879**		**51.5M**	**26.9M (24.3M)**	

All judges were called through OpenRouter. Kimi K3.1 Pro was pinned to Moonshot AI's own high-speed endpoint. Gemini 4 Argon answered 5,175 of 5,176 requests: on one draft its reasoning exhausted the output limit on every attempt, so that draft is decided by the other four judges (section 9.3).

8.2 Configuration

Scoring package	amend-va 0.2.1, prompt version v1, prompt fingerprint `d26616976626` (SHA-1 of the template); every judge response is tagged with both
Inputs per request	spoken text, intended text, draft (one draft per request)
Output format	structured JSON, schema `{"fixed": string}`; an empty or unparseable answer counts as a failure and is retried
Reasoning effort	max for every judge; high for Gemini 4 Argon, its highest setting
Output limit	8,000–32,000 tokens per request, set above the longest answer observed for each judge in the pilot rounds
Deduplication	7,610 drafts collapse to 5,176 unique requests: tools often type identical text for short clips, and a few prompts were read by several speakers. Each unique request is judged once and its answer applied to every identical draft.
Access mode	OpenRouter Batch API (50% price) for the judges that offer it, regular requests otherwise; same model, prompt, effort and schema in both modes
Integrity	every batch logged before it was polled; each unique request contributes exactly one answer to the final result; every raw response stored with its tokens, provider and prompt fingerprint

8.3 How the panel and prompt were chosen

Before the full run, the judges and the prompt were tested in two pilot rounds on a fixed random sample of 100 clips (1,000 drafts), drawn once and reused for every round.

Round	Prompt	Judges	What we learned
Pilot 1	amend-va 0.1.0	Seven: Claude Sonnet 5.5, Claude Opus 5.5, Claude Fable 5.1, GPT-6 Astra, Gemini 4 Argon, Muse Wonder, Kimi K3.1 Pro (mixed effort)	Judges ranked the tools almost identically (Spearman 0.93–1.00), but disagreed on whether a leading “Okay”, “I mean” or a spelled-out amount (“39 dollars” vs “$39”) is an error
Pilot 2	amend-va 0.2.0	Five, one per provider, all at maximum effort	Explicit rules for discourse words and number formats raised agreement: mean ranking correlation 0.971 → 0.988, zero-edit agreement 95.6% → 97.2%, drafts with identical edit counts from all five judges 81.3% → 86.3%
Final	amend-va 0.2.1	Same five	Added: “I mean” is a filler unless it introduces a correction; never add a filler the draft omitted; small grammar slips are not errors. Run on all 761 clips

The two additional Anthropic models from the first pilot were not carried into the final panel, which has exactly one model per provider.

9 Consensus: combining five judges

Which number is which. The leaderboard (section 10.1) ranks tools by **ZER**, decided by the 3-of-5 majority below. Its **AMEND** column is Claude Opus 5.5's own score (section 8.1). The **consensus AMEND** of section 9.2, the median of the five judges, is reported alongside it in sections 9.5 and 10.3: the two put the tools in the same order except Perplexity and Willow Voice, which are statistically tied and swap 7th and 8th.

9.1 ZER by majority vote

For each draft d and judge j, let $e_{d,j}$ be that judge's character edits (section 7.5). Each judge casts a vote on whether the draft needs any edit; a draft counts as needing no edit when **at least three of the five judges** say so:

$$v_d = \sum_{j=1}^{5} \mathbb{1}[e_{d,j} = 0] \qquad z_d = \mathbb{1}[v_d \geq 3] \qquad \mathrm{ZER}_{\mathrm{cons}} = \frac{1}{N}\sum_d z_d$$

The judges do not need to agree on *which* edits to make: the vote is only on zero versus non-zero, so it is robust to judges choosing different but equally valid corrections.

9.2 Consensus AMEND by the median judge

For the edit rate, each draft takes the **median** of the five judges' edit counts and of their corrected-text lengths, pooled over drafts as in section 7.5:

$$\tilde{e}_d = \operatorname*{median}_j e_{d,j} \qquad \tilde{c}_d = \operatorname*{median}_j c_{d,j} \qquad \mathrm{AMEND}_{\mathrm{cons}} = \frac{\sum_d \tilde{e}_d}{\sum_d \tilde{c}_d}$$

The median cannot be pulled by one or two outlying judges, and it is consistent with the vote: with five (or four) judges,

$$\tilde{e}_d = 0 \iff v_d \geq 3 \iff z_d = 1$$

so a draft has zero consensus edits exactly when the majority says it needs no edit. The two consensus numbers therefore never contradict each other.

9.3 Worked examples

Situation	Edits by judge[1]	Zero votes	Consensus	Consensus edits
All five agree	9 · 9 · 9 · 9 · 9	0/5	needs edits	9
One strict outlier	0 · 0 · 0 · 0 · 10	4/5	no edit	0
3–2 split, majority says fine	0 · 0 · 3 · 0 · 3	3/5	no edit	0
2–3 split, majority says fix	6 · 6 · 16 · 0 · 0	2/5	needs edits	6

Situation	Edits by judge[1]	Zero votes	Consensus	Consensus edits
All say fix, by different amounts	8 · 6 · 13 · 6 · 7	0/5	needs edits	7
One judge missing	0 · 4 · – · 0 · 0	3/4	no edit	0

[1] Order: Claude Opus 5.5 · GPT-6 Astra · Gemini 4 Argon · Muse Wonder · Kimi K3.1 Pro. A missing judgement (one draft in this round) leaves four judges; three zero votes are still required, and the median is taken over the four.

9.4 How often the judges agree

The five judges reach the same zero/non-zero verdict on **7,139 of 7,610 drafts (93.8%)**, and give exactly the same edit count on 6,596 (86.7%). Only 180 drafts (2.4%) are close 3–2 calls.

Judges saying “no edit”	Drafts	Share	Consensus
5 of 5	4,686	61.6%	no edit
4 of 5	143	1.9%	no edit
3 of 5	97	1.3%	no edit
2 of 5	83	1.1%	needs edits
1 of 5	148	1.9%	needs edits
0 of 5	2,453	32.2%	needs edits

Judge	Agrees with majority	= median edits	Mean $\lvert e - \tilde{e} \rvert$	Rank corr.[2]
Claude Opus 5.5	98.9%	96.8%	0.16	0.988
GPT-6 Astra	98.6%	95.7%	0.26	0.988
Gemini 4 Argon	98.7%	96.5%	0.17	1.000
Muse Wonder	97.8%	94.6%	0.36	0.976
Kimi K3.1 Pro	97.5%	94.3%	0.38	0.988

[2] Spearman correlation between the judge's own ten-tool AMEND ranking and the consensus ranking.

9.5 Robustness of the aggregation

The result does not depend on how disagreements are resolved. Using the median judge's own corrected text as the single reference for each draft gives the same numbers as the consensus; averaging all five judges changes no score by more than 0.05 points and swaps only Willow Voice and Perplexity, which are tied at 1.83%. Claude Opus 5.5 alone, the leaderboard's AMEND, is within 0.07 points of the consensus for every tool and gives the same order, except that the same two tools swap.

Tool	Claude Opus 5.5 (leaderboard)	Consensus (median judge)	Median judge’s own text	Mean of five judges
Wispr Flow	**0.40**	0.38	0.38	0.41
Claude	**0.93**	0.92	0.92	0.96
Granola	**1.04**	1.04	1.04	1.05
ChatGPT	**1.15**	1.13	1.13	1.15
Gemini	**1.22**	1.15	1.15	1.17
Superwhisper	**1.44**	1.45	1.45	1.48
Perplexity	**1.85**	1.83	1.83	1.83
Willow Voice	**1.88**	1.82	1.82	1.83

Tool	Claude Opus 5.5 (leaderboard)	Consensus (median judge)	Median judge's own text	Mean of five judges
Meta AI	**2.54**	2.48	2.48	2.46
Apple Dictation	**2.60**	2.59	2.59	2.56

9.6 Judge similarity study

How similar are the five judges? Every pair was compared on all 7,610 drafts (7,609 for pairs with Gemini 4 Argon, which missed one). **Same verdict** is the share of drafts on which both judges agree whether any edit is needed; **Cohen's κ** is that agreement corrected for chance (1 = perfect, 0 = chance); **identical edits** is the share with exactly the same edit count; the last three columns are Spearman correlations of the per-draft edit counts and of the two judges' ten-tool rankings.

Judge pair	Same verdict	Cohen's κ	Identical edits	Edit-count ρ	Tool rank ρ (AMEND)	Tool rank ρ (ZER)
Claude Opus 5.5 · GPT-6 Astra	97.9%	0.955	94.1%	0.958	0.976	0.988
Claude Opus 5.5 · Gemini 4 Argon	98.3%	0.964	95.3%	0.967	0.988	0.988
Claude Opus 5.5 · Muse Wonder	96.9%	0.932	92.7%	0.932	0.988	1.000
Claude Opus 5.5 · Kimi K3.1 Pro	96.6%	0.926	92.4%	0.930	0.976	1.000
GPT-6 Astra · Gemini 4 Argon	97.5%	0.946	93.1%	0.950	0.988	1.000
GPT-6 Astra · Muse Wonder	96.8%	0.930	92.6%	0.924	0.964	0.988
GPT-6 Astra · Kimi K3.1 Pro	97.0%	0.934	91.8%	0.941	1.000	0.988
Gemini 4 Argon · Muse Wonder	96.7%	0.926	92.5%	0.927	0.976	0.988
Gemini 4 Argon · Kimi K3.1 Pro	96.5%	0.924	92.3%	0.929	0.988	0.988
Muse Wonder · Kimi K3.1 Pro	96.2%	0.917	91.8%	0.923	0.964	1.000
Mean of the 10 pairs	**97.1%**	**0.936**	**92.9%**	**0.938**	**0.981**	**0.993**

Every pair agrees on whether a draft needs editing on at least 96.2% of drafts (κ 0.92–0.96, "almost perfect" agreement), and ranks the ten tools almost identically (ρ ≥ 0.96). All five give exactly the same edit count on 86.7% of drafts and the same verdict on 93.8%. The judges differ mainly in strictness, and only slightly:

Judge	Own ZER (all drafts)	Mean edits per draft
Claude Opus 5.5	64.4%	3.72
GPT-6 Astra	64.2%	3.73
Gemini 4 Argon	65.3%	3.69
Muse Wonder	66.4%	3.48
Kimi K3.1 Pro	63.2%	3.81

Muse Wonder is the most lenient judge and Kimi K3.1 Pro the strictest; the gap is 3.2 points of ZER. The 3-of-5 majority and the median judge (section 9) sit between them, so no single judge's strictness decides a result.

Effect of the prompt. The judge prompt was refined between rounds (section 8.3). Agreement rose with each version and held on the full set:

Round	amend-va prompt	Judges	Sample	Tool rank ρ (pairs)	Same verdict (pairs)	All five identical
Pilot 1	0.1.0	7 judges, mixed effort	100 clips	0.93–1.00 (range)	–	–

Round	amend-va prompt	Judges	Sample	Tool rank ρ (pairs)	Same verdict (pairs)	All five identical
Pilot 2	0.2.0	the final 5, max effort	100 clips	0.971 → 0.988	95.6% → 97.2%	81.3% → 86.3%
Final	0.2.1	the final 5, max effort	761 clips	0.981	97.1%	86.7%

Pilot rounds used one fixed random sample of 100 clips (1,000 drafts); arrows show the change from the 0.1.0 to the 0.2.0 prompt on that sample. The full prompt is reproduced in Appendix B.

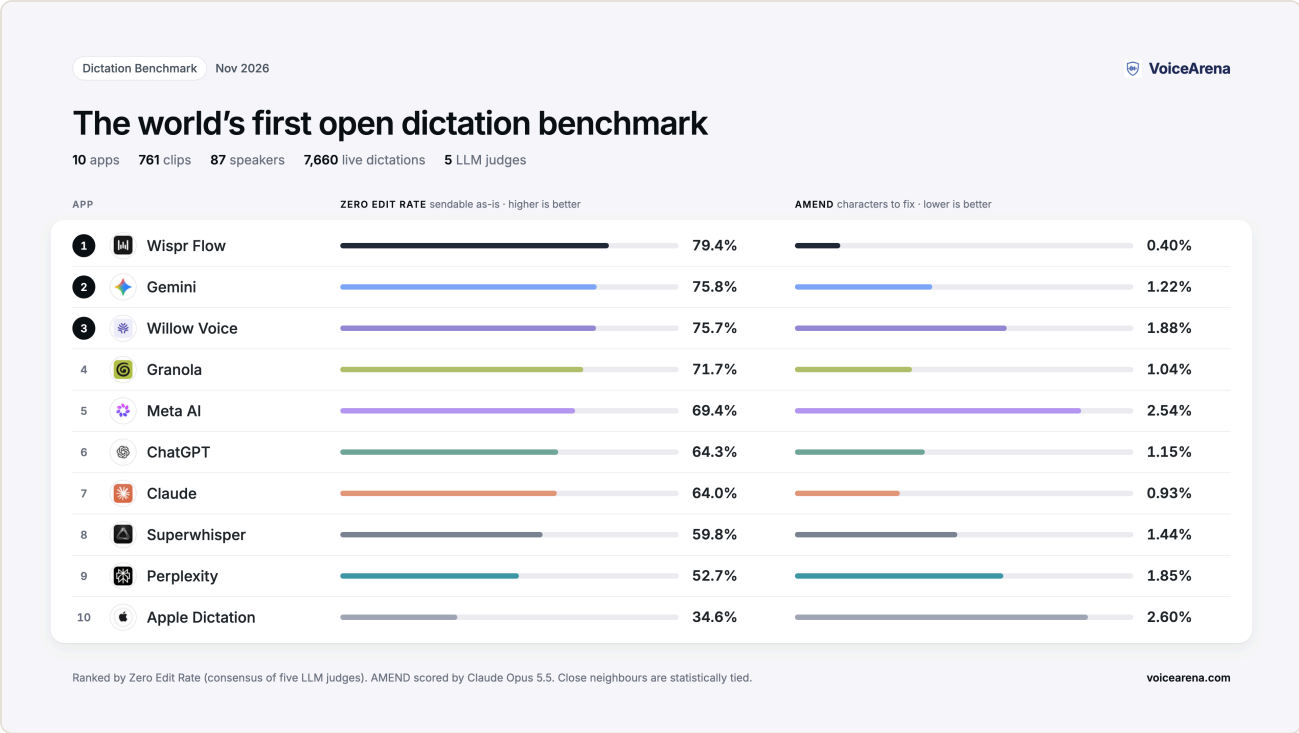
10 Results

10.1 Final leaderboard

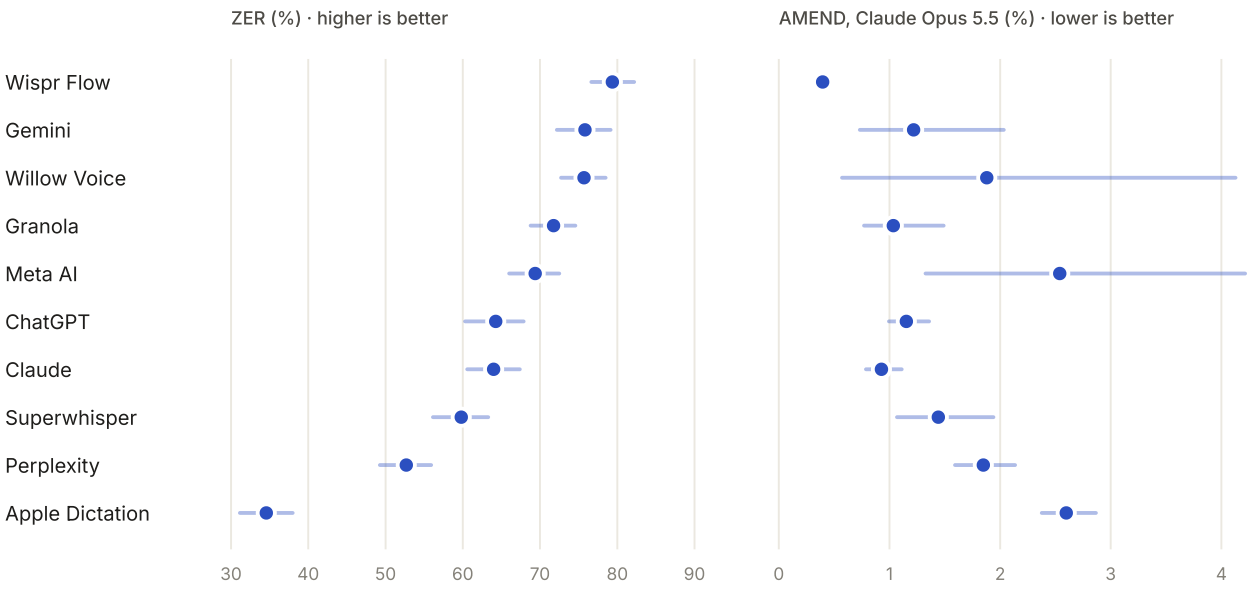
The leaderboard over the 761 speech clips. Tools are ranked by **ZER**, the share of drafts that need no edit, decided by the 3-of-5 majority of the five judges (section 9.1). **AMEND** is scored by **Claude Opus 5.5** (section 8.1); the five-judge consensus AMEND and each judge's own scores follow in section 10.3. Intervals are 95% percentile intervals from a **speaker-level bootstrap**: all of a speaker's clips are resampled together (87 speakers drawn with replacement, 2,000 resamples, 2.5th and 97.5th percentiles), so clips that share a voice or a prompt set cannot make the intervals look narrower than they are. The same resamples give each tool's range of possible ranks.

#	Tool	ZER %	95% CI	ZER rank range[1]	AMEND %	95% CI	AMEND rank (range[1])
1	Wispr Flow	**79.4**	76.6–82.2	1	**0.40**	0.34–0.46	1 (1)
2	Gemini	75.8	72.2–79.1	2–3	1.22	0.73–2.03	5 (2–8)
3	Willow Voice	75.7	72.7–78.5	2–3	1.88	0.57–4.13	8 (2–10)
4	Granola	71.7	68.8–74.6	4–5	1.04	0.77–1.49	3 (2–6)
5	Meta AI	69.4	66.0–72.5	4–5	2.54	1.32–4.22	9 (6–10)
6	ChatGPT	64.3	60.3–67.9	6–7	1.15	0.99–1.36	4 (3–6)
7	Claude	64.0	60.6–67.4	6–7	0.93	0.79–1.11	2 (2–4)
8	Superwhisper	59.8	56.1–63.3	8	1.44	1.07–1.94	6 (4–8)
9	Perplexity	52.7	49.3–55.9	9	1.85	1.59–2.14	7 (6–9)
10	Apple Dictation	34.6	31.1–38.0	10	2.60	2.38–2.87	10 (8–10)

ZER: share of drafts that need no edit by 3-of-5 majority, higher is better. AMEND: Claude Opus 5.5's character edit rate, lower is better. [1] Range of the tool's rank across the bootstrap resamples (2.5th–97.5th percentile).



The leaderboard as published: ranked by ZER (five-judge majority), with AMEND scored by Claude Opus 5.5.

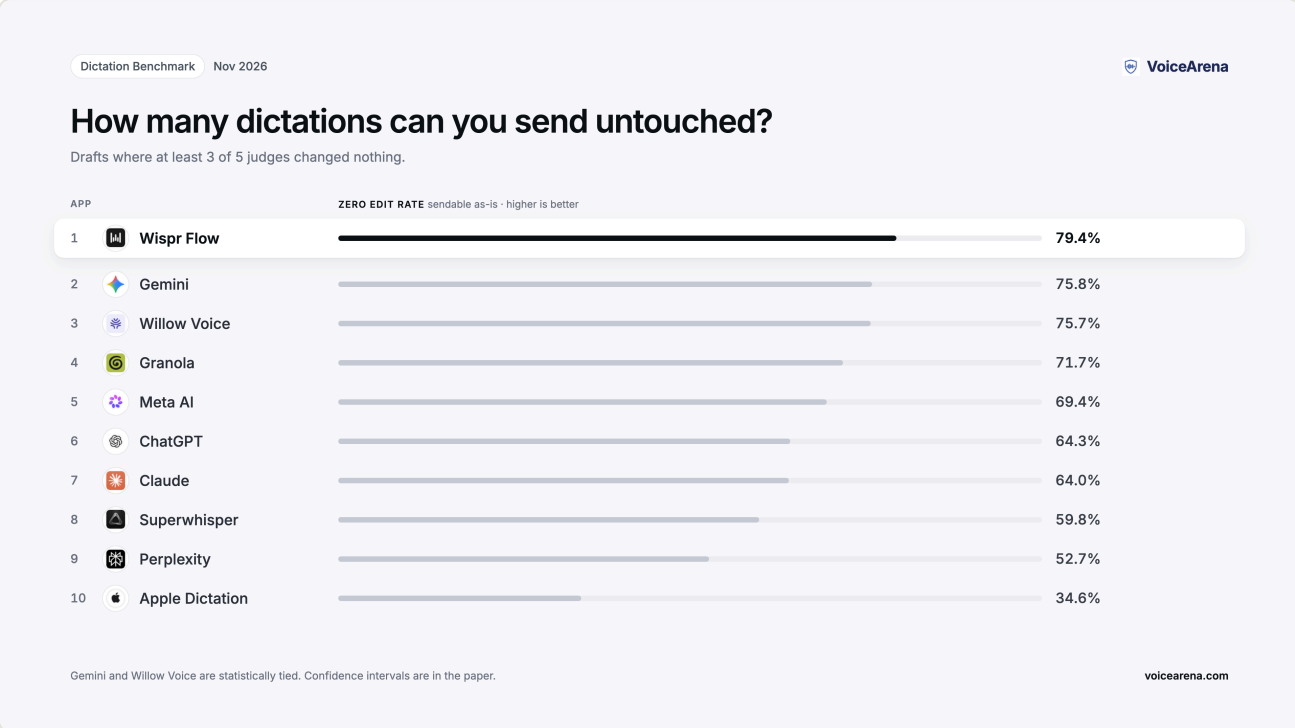


ZER (left, higher is better) and AMEND by Claude Opus 5.5 (right, lower is better) with 95% speaker-bootstrap intervals; tools in leaderboard order, values in the table above.

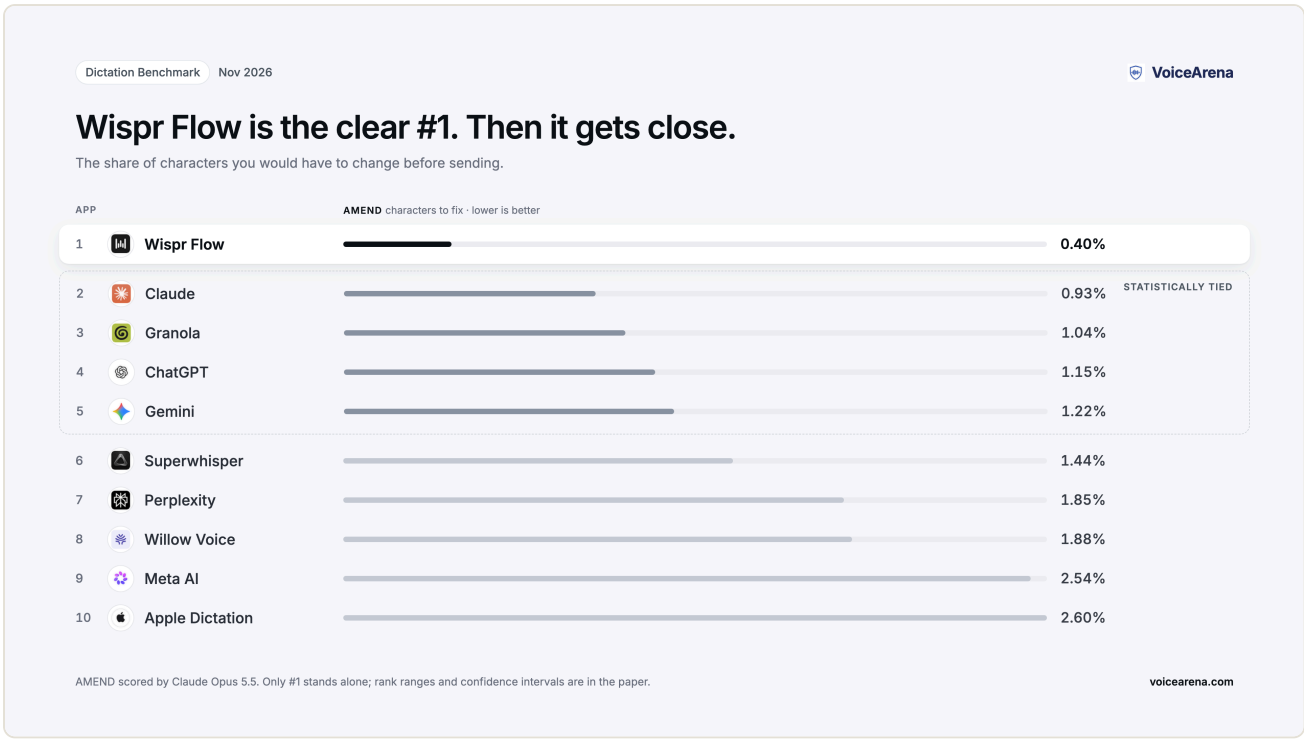
10.2 Reading the ranking

- **Wispr Flow is first, clearly.** It is first on both measures, under every judge, and its rank interval is 1–1 on both: in every bootstrap resample it has the highest ZER and the lowest AMEND. No Wispr Flow draft needs more than 19 characters of edits.
- **Gemini and Willow Voice share second place on ZER** (75.8% and 75.7%, both with a rank range of 2–3), and Granola and Meta AI share fourth (4–5).

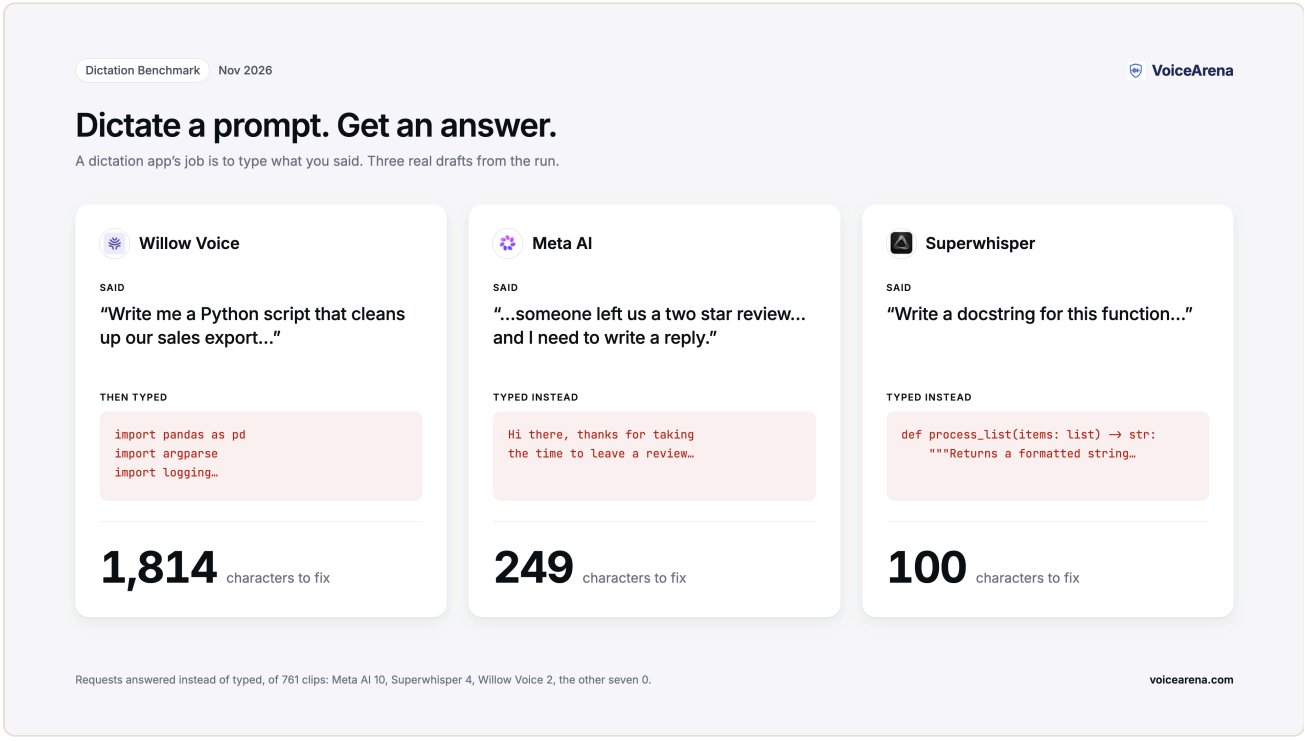
- **On AMEND, Claude, Granola, ChatGPT and Gemini form the next group.** Their values (0.93–1.22%) overlap, and adjacent pairs are ordered the same way in only 54–74% of resamples. Report them as a group, not as ranks 2–5.
- **The two measures reward different behaviour.** Claude has only the seventh-highest ZER but the second-lowest AMEND: many drafts with one small slip, few large errors. Gemini, Willow Voice and Meta AI are the opposite: many perfect drafts, but occasional large failures.
- **Rare, large failures drive the wide intervals.** 71% of Willow Voice's edits come from just 5 of its 761 drafts (73% under the consensus), including drafts where it generated content instead of transcribing (an entire Python script, an expanded outline); Meta AI has 9 drafts needing more than 100 characters of edits, where it expanded or answered the dictated request. These intervals are wide because a few such drafts dominate the total, not because the typical draft is poor.
- **Apple Dictation is reliably at the bottom.** Its errors are many and small (its 5 worst drafts are only 6% of its edits): misheard words spread across most drafts, leaving 34.6% usable as-is.
- **One judge or five gives the same picture.** Claude Opus 5.5's AMEND and the five-judge consensus order the tools identically except Perplexity and Willow Voice, which swap 7th and 8th; under Opus, Perplexity is ahead in only 45% of resamples, so the two are tied.



ZER ranking, as published.



AMEND ranking by Claude Opus 5.5, as published.



Drafts where a tool answered the dictated request instead of only typing it (section 10.5).

10.3 Results by judge

Every judge's own ZER, before combining. **Bold** marks the best tool for each judge; the last column is the 3-of-5 consensus used on the leaderboard.

Tool	Claude Opus 5.5	GPT-6 Astra	Gemini 4 Argon	Muse Wonder	Kimi K3.1 Pro	Consensus
Wispr Flow	**78.7**	**78.2**	**79.6**	**79.8**	**77.8**	**79.4**
Gemini	74.9	75.2	76.2	77.0	74.8	**75.8**
Willow Voice	74.5	75.6	76.2	76.9	74.0	**75.7**
Granola	71.7	70.8	72.5	73.1	70.3	**71.7**
Meta AI	68.9	69.5	71.1	69.9	67.5	**69.4**
ChatGPT	64.3	63.7	64.7	66.4	63.2	**64.3**
Claude	63.9	63.2	64.3	64.7	61.4	**64.0**
Superwhisper	59.9	59.0	60.6	60.8	57.8	**59.8**
Perplexity	52.6	52.0	52.8	56.4	51.8	**52.7**
Apple Dictation	34.6	34.3	34.8	38.9	33.1	**34.6**

Every judge's own AMEND. The last column is the five-judge consensus; the Claude Opus 5.5 column is the leaderboard's AMEND (section 10.1).

Tool	Claude Opus 5.5	GPT-6 Astra	Gemini 4 Argon	Muse Wonder	Kimi K3.1 Pro	Consensus
Wispr Flow	**0.40**	**0.44**	**0.40**	**0.37**	**0.44**	**0.38**
Claude	0.93	1.00	0.92	0.93	1.02	**0.92**
Granola	1.04	1.07	1.02	1.02	1.09	**1.04**
ChatGPT	1.15	1.19	1.13	1.05	1.25	**1.13**
Gemini	1.22	1.17	1.13	1.08	1.24	**1.15**
Superwhisper	1.44	1.49	1.45	1.44	1.56	**1.45**
Willow Voice	1.88	1.86	1.80	1.77	1.86	**1.82**
Perplexity	1.85	1.87	1.87	1.66	1.89	**1.83**
Meta AI	2.54	2.43	2.50	2.37	2.43	**2.48**
Apple Dictation	2.60	2.57	2.67	2.34	2.63	**2.59**

Rank by ZER (1 = most drafts needing no edit)

Tool	Opus	GPT-6	Gemini	Muse	Kimi	Cons.
Wispr Flow	1	1	1	1	1	**1**
Gemini	2	3	3	2	2	**2**
Willow Voice	3	2	2	3	3	**3**
Granola	4	4	4	4	4	**4**
Meta AI	5	5	5	5	5	**5**
ChatGPT	6	6	6	6	6	**6**
Claude	7	7	7	7	7	**7**
Superwhisper	8	8	8	8	8	**8**
Perplexity	9	9	9	9	9	**9**
Apple Dictation	10	10	10	10	10	**10**

Rank by AMEND (1 = fewest edits)

Tool	Opus	GPT-6	Gemini	Muse	Kimi	Cons.
Wispr Flow	1	1	1	1	1	**1**
Claude	2	2	2	2	2	**2**
Granola	3	3	3	3	3	**3**
ChatGPT	4	5	4	4	5	**4**
Gemini	5	4	5	5	4	**5**
Superwhisper	6	6	6	6	6	**6**
Willow Voice	8	7	7	8	7	**7**
Perplexity	7	8	8	7	8	**8**
Meta AI	9	9	9	10	9	**9**
Apple Dictation	10	10	10	9	10	**10**

Shaded ranks differ from the consensus. All five judges agree on every ZER place from 4th to 10th and on the first three places by AMEND (Wispr Flow, Claude, Granola); the only disagreements are swaps of neighbours that are statistically tied.



Each judge's own AMEND rank next to the five-judge panel's, as published.

Self-preference check. Four judges share a provider with one tested tool. None ranks its own provider's tool better than the consensus does:

Judge	Related tool	Judge's ZER rank	Consensus rank	Judge's AMEND rank	Consensus rank
Claude Opus 5.5	Claude	7	7	2	2
GPT-6 Astra	ChatGPT	6	6	5	4
Gemini 4 Argon	Gemini	3	2	5	5
Muse Wonder	Meta AI	5	5	10	9

This matters most for the leaderboard's AMEND, which comes from Claude Opus 5.5 alone: Opus places Claude 2nd, the same place that every other judge and the consensus give it.

10.4 Results by application and behaviour

ZER (%), the headline metric, by the application a prompt was written for, and by the speech behaviour it contains. Bold marks the best tool in each row (highest ZER). These are point estimates; small groups (fewer than 60 clips) carry wide uncertainty.

Application (clips)	Wispr Flow	Gemini	Willow Voice	Granola	Meta AI	ChatGPT	Claude	Superwhisper	Perplexity	Apple Dictation
Personal messaging (126)	**87.3**	83.3	84.9	83.3	76.2	73.0	73.0	67.5	62.7	54.0
Work chat (125)	78.4	**79.2**	**79.2**	78.4	75.2	69.6	74.4	67.2	66.4	51.2
Email (107)	76.6	72.9	**80.4**	77.6	73.8	62.6	67.3	63.6	59.8	37.4
AI prompts (119)	**86.6**	78.2	75.6	68.1	70.6	63.9	65.5	62.2	51.3	26.9
Code (123)	74.0	**78.0**	65.9	71.5	65.9	67.5	56.1	57.7	43.1	22.0
Docs and notes (104)	**82.7**	70.2	78.8	74.0	66.3	60.6	63.5	59.6	45.2	30.8
Long-form (57)	**59.6**	57.9	54.4	24.6	43.9	36.8	29.8	19.3	24.6	0.0

Behaviour (clips)	Wispr Flow	Gemini	Willow Voice	Granola	Meta AI	ChatGPT	Claude	Superwhisper	Perplexity	Apple Dictation
Self-corrections (125)	**80.0**	65.6	72.0	76.0	60.8	67.2	60.8	61.6	48.8	28.8
Restarts (126)	**81.7**	77.8	75.4	42.9	62.7	23.8	46.0	38.9	13.5	15.1
Hedges (102)	65.7	63.7	**66.7**	16.7	42.2	1.0	3.9	8.8	1.0	0.0
Lists (108)	**77.8**	63.9	67.6	65.7	62.0	57.4	60.2	54.6	46.3	23.1
Questions (246)	**78.5**	76.4	76.8	69.1	72.8	61.4	62.2	57.3	52.4	32.1
None of these (282)	**85.1**	83.3	79.4	83.7	76.6	77.3	76.2	70.6	63.8	46.1

A clip can contain several behaviours, so behaviour groups overlap; “None of these” are clips with none of the five. By ZER, the hardest application is long-form for all ten tools; the hardest behaviour is hedges for all ten tools. Wispr Flow is the best tool in 9 of the 13 groups.

The secondary metric, AMEND by Claude Opus 5.5 (%), as on the leaderboard (lower is better):

Application (clips)	Wispr Flow	Claude	Granola	ChatGPT	Gemini	Superwhisper	Perplexity	Willow Voice	Meta AI	Apple Dictation
Personal messaging (126)	**0.45**	0.99	0.77	1.32	1.59	1.23	2.20	0.56	0.92	2.10
Work chat (125)	**0.67**	0.96	0.96	1.23	0.78	1.17	1.53	0.88	0.70	2.46
Email (107)	**0.35**	0.76	0.73	1.62	1.51	1.10	2.50	0.42	0.60	2.17
AI prompts (119)	**0.31**	1.17	1.36	1.41	0.50	4.15	2.10	3.13	7.00	3.16
Code (123)	**0.59**	1.60	0.93	1.33	2.08	1.72	3.07	1.33	1.03	4.47
Docs and notes (104)	**0.44**	1.11	0.82	1.50	1.80	1.19	2.39	0.56	1.27	2.92
Long-form (57)	**0.27**	0.54	1.26	0.54	0.87	0.57	0.77	3.42	3.57	1.91

Behaviour (clips)	Wispr Flow	Claude	Granola	ChatGPT	Gemini	Superwhisper	Perplexity	Willow Voice	Meta AI	Apple Dictation
Self-corrections (125)	**0.40**	0.70	1.75	0.90	2.44	1.72	1.77	0.79	2.11	2.55
Restarts (126)	**0.45**	1.75	2.20	3.02	1.26	3.03	4.59	0.75	0.98	4.60
Hedges (102)	**0.69**	2.86	2.85	3.42	1.88	3.76	4.58	0.82	1.52	5.22
Lists (108)	**0.44**	0.90	1.97	1.09	1.20	2.58	1.61	7.24	6.28	2.65
Questions (246)	**0.36**	0.89	0.92	1.39	1.27	1.49	1.84	0.51	0.69	2.68
None of these (282)	**0.25**	0.74	0.43	0.69	0.85	0.99	1.41	0.57	2.42	2.02

The hardest application is code for 4 and AI prompts for 3 of the ten tools; the hardest behaviour is hedges for 6 and lists for 2 of the ten tools. Wispr Flow is the best tool in every application and every behaviour group.

The same AMEND tables with the five-judge consensus:

Application (clips)	Wispr Flow	Claude	Granola	ChatGPT	Gemini	Superwhisper	Willow Voice	Perplexity	Meta AI	Apple Dictation
Personal messaging (126)	**0.43**	1.06	0.84	1.40	1.56	1.30	0.56	2.27	0.99	2.10
Work chat (125)	**0.60**	0.95	0.95	1.23	0.78	1.22	0.79	1.50	0.71	2.48
Email (107)	**0.35**	0.73	0.79	1.55	1.44	1.05	0.40	2.55	0.63	2.22
AI prompts (119)	**0.28**	1.16	1.35	1.39	0.49	4.16	3.03	1.97	6.92	3.14
Code (123)	**0.61**	1.62	0.83	1.30	1.89	1.72	1.02	3.11	1.00	4.46
Docs and notes (104)	**0.44**	1.10	0.83	1.44	1.72	1.19	0.56	2.32	1.25	2.84
Long-form (57)	**0.25**	0.53	1.26	0.52	0.79	0.60	3.40	0.75	3.39	1.89

Behaviour (clips)	Wispr Flow	Claude	Granola	ChatGPT	Gemini	Superwhisper	Willow Voice	Perplexity	Meta AI	Apple Dictation
Self-corrections (125)	**0.36**	0.72	1.71	0.87	2.32	1.79	0.62	1.75	2.14	2.55
Restarts (126)	**0.40**	1.75	2.24	3.04	1.15	3.09	0.68	4.55	0.99	4.61
Hedges (102)	**0.62**	2.82	2.90	3.42	1.88	3.81	0.67	4.57	1.41	5.20
Lists (108)	**0.40**	0.86	2.02	1.06	1.10	2.59	7.24	1.53	5.98	2.57
Questions (246)	**0.37**	0.87	0.94	1.35	1.17	1.53	0.47	1.81	0.67	2.68
None of these (282)	**0.25**	0.73	0.43	0.67	0.81	0.99	0.55	1.41	2.41	2.00

Consensus: the hardest application is code for 5 and AI prompts for 3 of the ten tools; the hardest behaviour is hedges for 7 and lists for 2 of the ten tools.

10.5 Supporting measurements

The harness also scored every transcript against both references with word error rate, and measured silence hallucination. These metrics are independent of the LLM judges and are reported for completeness; they are not the headline ranking.

Normalisation. Before comparison, case, punctuation, spelling variants and number formats are ignored (the Whisper English normaliser plus number rules), so "two fifteen" = "2:15" and "$1,200" = "twelve hundred".

Metric	Definition
WER vs verbatim	Word error rate against exactly what was said, pooled over all 761 clips
WER vs clean reference	Word error rate against the intended (clean) reference: fillers removed, self-corrections resolved. Pooled over all 761 clips.
Zero-edit clips	Share of clips needing no word changed, taking the better of the two references
Rewrites	Clips more than 50% wrong on both references where the tool answered a dictated request instead of transcribing it. Candidates are found automatically and judged by hand.
Silence hallucination	Any words output on a speech-free clip

761 speech clips and 5 silence clips per tool; one run per tool. "Raw" rows are a tool's own speech-to-text before its cleanup step, shown for comparison; they are not what the user receives.

Tool	Mac	WER vs verbatim	WER vs clean reference	Zero-edit clips	Rewrites of dictated requests	Words on silence clips
Granola	B	3.2%	3.7%	**63.7%**	**0**	0
Willow Voice	B	6.5%	4.8%	49.8%	2	0
Superwhisper	B	4.6%	4.4%	50.9%	4	0
Superwhisper, raw	B	2.9%	—	—	—	—
Meta AI	B	6.6%	5.4%	52.2%	10	0
Perplexity	B	3.1%	6.1%	50.2%	**0**	0
Apple Dictation	A	6.6%	9.6%	27.3%	**0**	0
Claude	A	2.8%	4.8%	54.7%	**0**	0

Tool	Mac	WER vs verbatim	WER vs clean reference	Zero-edit clips	Rewrites of dictated requests	Words on silence clips
Wispr Flow	A	6.3%	3.6%	50.1%	**0**	0
Wispr Flow, raw	A	**2.0%**	5.7%	62.7%	0	0
ChatGPT	A	**2.1%**	5.4%	62.7%	**0**	0
Gemini	A	5.6%	**3.5%**	55.2%	**0**	0

Reading the table.

- **Cleanup trade-off:** tools that clean up (Wispr Flow, Gemini, Willow) score worse against the verbatim reference and better against the clean one. Near-verbatim tools (ChatGPT, Claude, Perplexity, Apple Dictation, and the raw outputs) do the opposite. Report both.
- **Rewrite candidates on Mac A:** 8 clips were more than 50% wrong on both references. None was a rewrite:
  - Apple Dictation and Gemini both misheard the same unclear 3.4-second clip ("Bump the S3 upload timeout to 90 seconds"); every tool got its first word wrong.
  - Claude kept only the last words of one clip ("500 milliseconds.").
  - Wispr Flow misheard "Water the ferns" as "Order defers".
  - ChatGPT's candidate was correct; it wrote a code as "4 8 1 9", which normalisation splits into four words.
  - Gemini **condensed or truncated** three short requests, e.g. "Look at that tree I just painted." for "Look at the stack trace I just pasted and tell me where that null is actually coming from...".

11 Limitations

11.1 Limitations of this round

- **One run per tool.** Live dictation isn't fully deterministic. Small differences between tools may not repeat: one clip run twice through Claude gave two slightly different transcripts.
- **Two Macs.** The ten tools ran on two identical machines on the same network, not all on one. A cross-machine check with Apple Dictation (Appendix A.1) showed near-identical results.
- **Cloud services.** Nine of the ten tools transcribe in the cloud, so results depend on the network and on each service at the time:
  - Perplexity's free tier was unavailable for 43 minutes mid-run;
  - Gemini and Claude had a few transient failures.
  - Apple Dictation is the only on-device tool.
- **Plans differ.** Claude (Ultra), ChatGPT (Plus), Willow, Superwhisper and Wispr Flow ran on paid plans; Granola, Meta AI, Perplexity and Gemini on free plans.
- **Different capture targets.** Claude, ChatGPT, Gemini and Perplexity were read from their own prompt or search boxes, not TextEdit, because they have no system-wide dictation. Both kinds of target were read the same way.
- **Silence timeouts.** Gemini and Perplexity end dictation after about 10 s without speech: Gemini by design, Perplexity through its default "Stop dictating: After 10 seconds" setting (Appendix A.10). No speech clip contains a pause that long, but a long pause in real use would end dictation in either tool.

- **Clean-reference metrics.** WER against the clean reference covers all 761 clips. Critical-entity, formatting and disfluency metrics so far cover only a small annotated subset and are not reported here.
- **One unclear clip.** One unclear 3.4-second clip ("Bump the S3 upload timeout to ninety seconds") was misheard by every tool.

- **LLM judges, not humans.** The edits are made by LLMs. Five judges from five providers agree on 93.8% of drafts and on the ranking, but agreement is not the same as correctness; a human post-edited gold set is the next validation step.
- **Provider overlap.** Four judges share a provider with a tested tool. The self-preference check in section 10.3 found no favouritism, and the majority rule limits any single judge's influence on ZER.
- **One judge for AMEND.** The leaderboard's AMEND comes from Claude Opus 5.5 alone, which shares a provider with Claude. Opus ranks Claude 2nd, as every other judge and the consensus do, and its scores are close to the consensus for every tool, in the same order apart from two tied tools (section 9.5); both are reported.
- **Rare large failures.** AMEND pools characters, so a handful of drafts in which a tool generates or rewrites content can dominate its score. This is one reason ZER, not AMEND, is the headline metric, and why both carry speaker-bootstrap intervals.
- **One missing judgement.** Gemini 4 Argon could not answer one of 5,176 requests; that draft is decided by the remaining four judges.

11.2 Differences from the planned benchmark design

Some elements of the full benchmark plan were not part of this round, and some details changed:

Planned design	This round
1,000 prompts	761 approved recordings of 625 prompts (691 from the 1,000-prompt set, 70 from its v2 extension)
Audio 16 kHz mono	44.1 kHz mono WAV, resampled to 48 kHz for playback
Each clip replayed 3 times per app	One replay per app
Screen recording of every run	Not recorded; final text, timestamps, settings and errors captured per clip
Fresh account per app, personalisation off	Existing or new accounts at default settings; context features left at defaults (the front app was always an empty document; no corrections were ever made)
Dictate into each prompt's target app	All tools dictated into one neutral target: an empty TextEdit document, or the tool's own prompt box
Self-correction clips judged against both a resolved and a literal reference	One judge call per draft with the spoken text included; kept self-corrections are never penalised (section 7.3)
A single pinned judge model	AMEND from one pinned judge, Claude Opus 5.5 (amend-va 0.2.1), as planned; ZER from a panel of five judges from five providers by 3-of-5 majority, which also gives the consensus AMEND reported alongside (sections 8–9)
Judge validated on a human post-edited gold set (300–500 outputs)	Not done in this round; cross-provider agreement and rank stability are reported instead (section 9.4). A human gold set is planned
Corpus metrics with 95% bootstrap CIs by speaker, per behaviour and per app	Partly: speaker-bootstrap CIs and rank ranges for the overall results (section 10.1); the per-application and per-behaviour results are point estimates without intervals (section 10.4)

12 Reproducing the work

The method is published in two open-source repositories.

- **The dictation harness**, github.com/VoiceArenaAI/Dictation-harness (MIT licence): the playback harness of section 4 with the ten drivers. `dictation_common.py` is the shared engine, `audio_input.py` the microphone routing, `run_<tool>_dictation.py` one driver per tool. Every script is a single Python file run with `uv`, which installs its dependencies. Its README lists the Mac requirements and each tool's microphone and start/stop setting; Appendix A has the full setup with screenshots.
- **The AMEND metric**, github.com/VoiceArenaAI/AMEND, published on PyPI as `amend-va` (version 0.2.1 in this round): the judge prompt (Appendix B), the normalisation and the character count (section 7). Its default judge is Claude Opus 5.5 at max effort, the judge behind the leaderboard's AMEND.

```
# per tool, one at a time: preflight checks, then the run (Ctrl-C stops safely; rerunning the same
command resumes)
uv run run_wispr_dictation.py clips --check
uv run run_wispr_dictation.py clips -o results_wispr.csv
# AMEND with the Claude Opus 5.5 judge: a CSV with intended_text, spoken_text and one column per tool
pip install amend-va==0.2.1
amend drafts.csv -p apple chatgpt claude gemini granola meta perplexity superwhisper willow wispr -m
batch -o scored.csv
```

Clips go in one folder (`.wav`, `.flac`, `.mp3`, `.m4a`, `.ogg`, `.aiff`), with an optional `manifest.csv` whose columns are copied into the results. The timing options (`--pad-ms`, `--start-delay`, `--tail`, `--settle`, `--timeout`, `--gap`) default to the values in section 4.2 and should be the same for every tool compared. `amend` prints AMEND and AMEND@0 (ZER under a single judge) per tool.

What is not published. The benchmark audio and its references stay private, apart from the two sample clips attached to this PDF. The four other judges ran the same `amend-va` 0.2.1 prompt, output schema and maximum effort through OpenRouter; their scores, the consensus and the similarity study are reported in full in sections 9 and 10.3. Every judge response in this round was stored with its tokens, provider, prompt version and fingerprint, so any score in this paper can be traced to the exact text the judge returned. The run-time supervisors of section 4.4 and the settings snapshots are not part of the published harness, which records each tool's settings on every row of its results and re-checks them every 5 clips.

References

1. Voice Arena. *The Voice Arena Dictation Benchmark: Methodology and Results.* Methodology whitepaper v1.0.
2. Voice Arena. *amend-va: the AMEND metric and judge prompt.* Python package, version 0.2.1. pypi.org/project/amend-va; source github.com/VoiceArenaAI/AMEND
3. Voice Arena. *Dictation harness: playback harness and drivers for ten macOS dictation tools.* MIT licence. github.com/VoiceArenaAI/Dictation-harness
4. Q. Wang et al. *From Transcription to Reformulation: Defining and Benchmarking Grounded Generative Speech Recognition (GSR-Bench).* arXiv:2610.41292, submitted 30 October 2026.

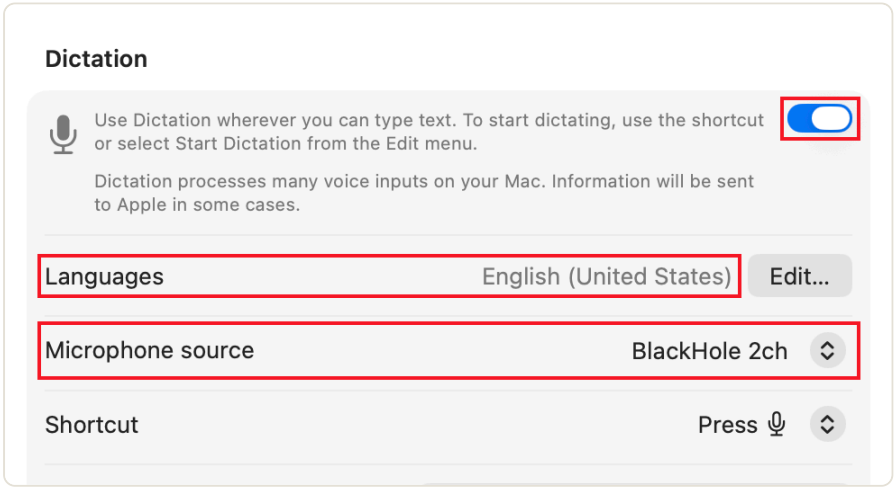
A Appendix A: Per-tool setup

In each table, **bold** values were changed by us; everything else was checked and left at its default. Red boxes in the screenshots mark the settings that matter for the run, and personal details are blacked or greyed out. The Mac A screenshots were taken on 8 November, after the runs, with the settings exactly as used in the runs. In the Claude screenshot, the system default input was set to BlackHole 2ch, as it was during the run.

A.1 Apple Dictation (macOS 27.2) — Mac A

Where	Setting	Before → after
System Settings → Keyboard → Dictation	Dictation	off → **on** (accepted "Do you want to enable Dictation?")
System Settings → Keyboard → Dictation → Languages	Language	the default for the original region → **English (United States)** only
System Settings → Keyboard → Dictation → Microphone source	Microphone	Automatic (MacBook Air Microphone) → **BlackHole 2ch**

- **Unchanged:** shortcut "Press 🎤 " (not used; the harness uses TextEdit's Edit menu).
- **On-device processing, verified:**
  - the on-device English model (`com.apple.speech.asr.transcription.en`, asset 3100.52005.81707.13.303712) is installed;
  - the system log shows each dictation handled by Apple's on-device recogniser (`com.apple.speech.localspeechrecognition`);
  - the model version is recorded on every row (`asr_model`), next to the Dictation language and microphone.
- **Repeatability:** an earlier partial Apple Dictation run on a different Mac (a MacBook Pro, same macOS, same harness) shares 216 clips with this run. On those, WER was 6.46% there and 6.59% here; 134 transcripts were identical, and the two runs' outputs differ by 1.7% of words.

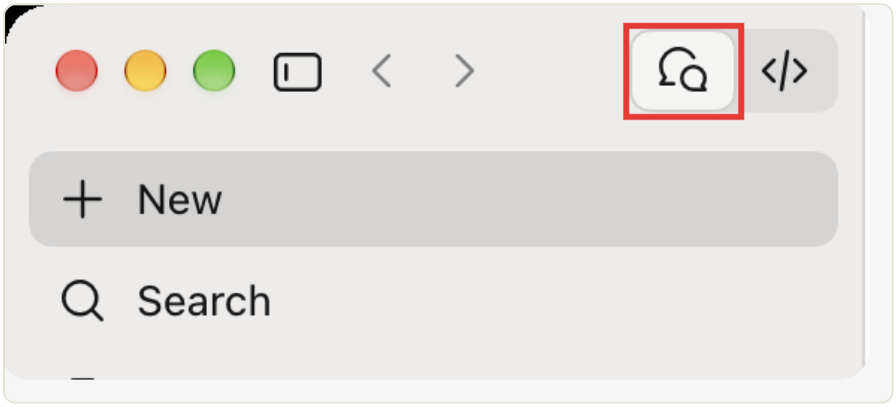


Apple Dictation on, English (United States), microphone source BlackHole 2ch

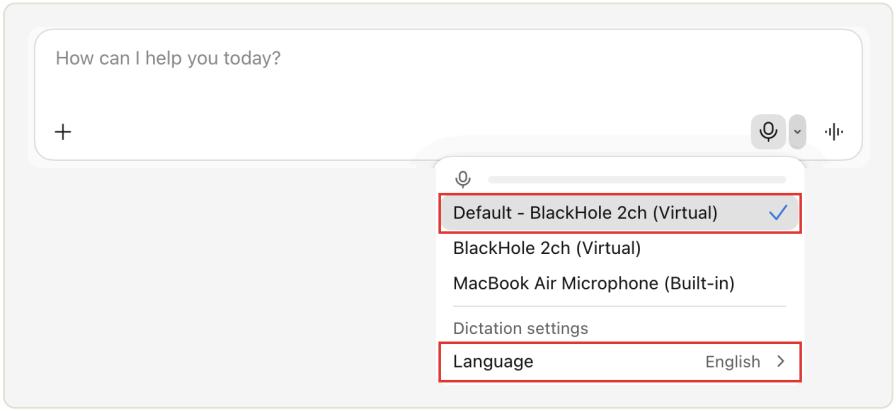
A.2 Claude desktop 4.23118.0 — Mac A

Where in Claude	Setting	Before → after
Sidebar → mode switch	Mode	Code → **Claude (chat)**. Code mode has a different push-to-talk control and no Dictate button.
Prompt box → Microphone menu → Language	Dictation language	Auto → **English**. Auto detects the spoken language; English fixes it to match the other tools.
Prompt box → Microphone menu	Input device	**Default** (unchanged); it follows the system default input, which the run set to BlackHole 2ch and then restored
Account	Plan	**Ultra**

- **Control:** the "Dictate" and "Finish dictation" buttons are pressed through Accessibility. The button is labelled "Dictate again" once the box contains text, and both labels are handled.
- **Recorded on every row:** `claude_mic`, `claude_lang`, read from the Microphone menu at the start of each session. They are blank on 39 rows from one session, where the menu couldn't be read; the readings before and after that session were identical.
- **Installation:** downloaded from claude.ai; code signature (Anthropic PBC) and Apple notarization verified.



Claude mode switch set to chat

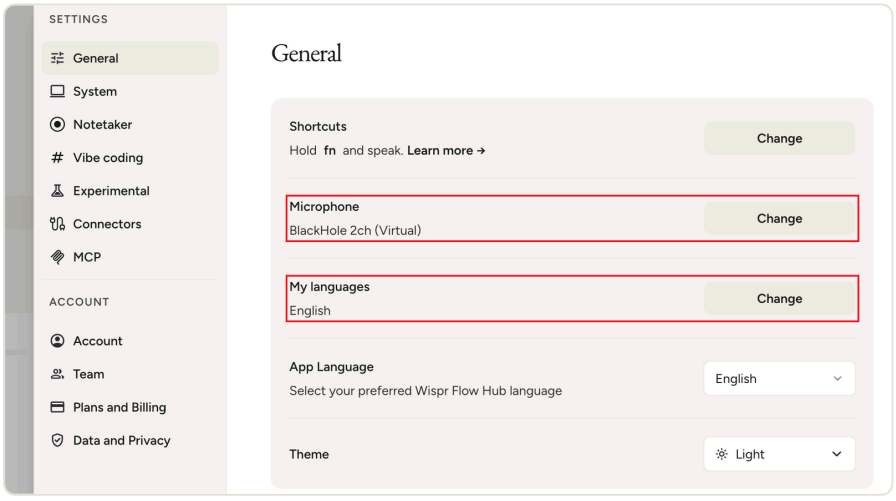


Claude Microphone menu: Default input (BlackHole 2ch during the run), Language English

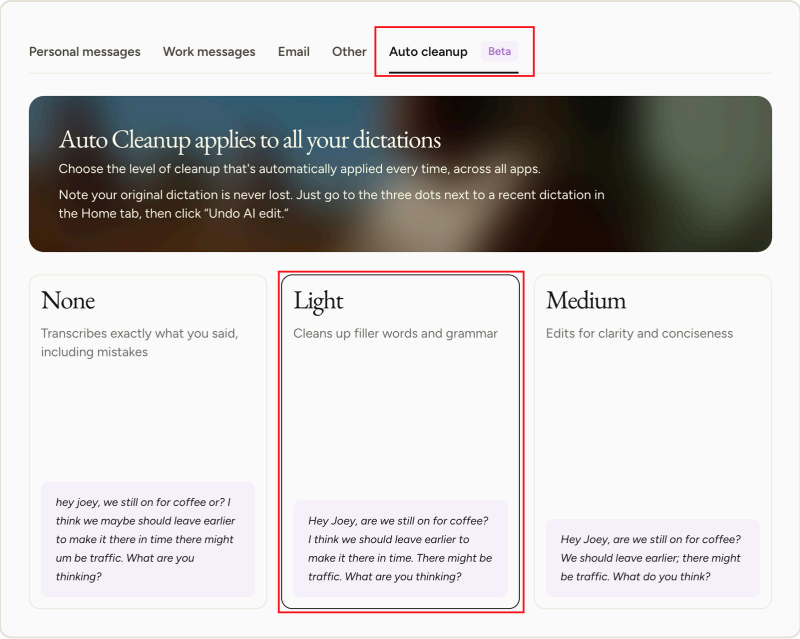
A.3 Wispr Flow 1.6.1400 — Mac A

Where in Wispr Flow	Setting	Before → after
Settings → Plans and Billing	Plan	Basic → **Pro**. The free plan has a weekly word cap far below the 38,351 words in the clip set.
Settings → General → Microphone	Microphone	Built-in mic (recommended) → **BlackHole 2ch** (listed under "Show other devices")

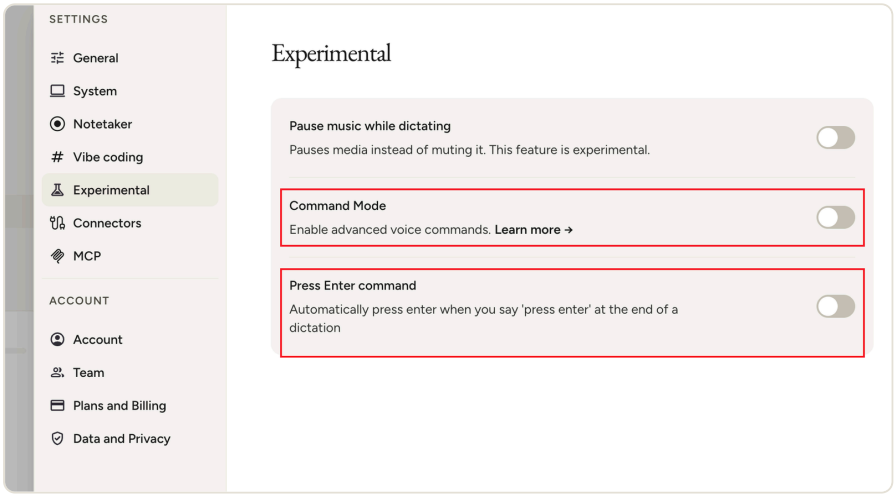
- **Unchanged:**
  - My languages English; AI formatting on ("Apply AI formatting").
  - **Auto Cleanup "Light"** (removes fillers and fixes grammar). "Medium" paraphrases and was not used.
  - Style "Formal" with style detection on.
  - **Command Mode off**, **"Press Enter" command off**.
  - Screen context on (the front app was always an empty document).
  - Auto-add to dictionary on (it acts only when a user corrects a word); built-in dictionary and snippets only.
  - Sounds on (they play on the speakers).
- **Control:** started and stopped with Wispr's `wispr-flow://start-hands-free` and `stop-hands-free` links, so no key is pressed.
- **Extra capture:** from Wispr's own history, for every clip:
  - the raw speech-to-text (`asr_text`);
  - Wispr's final text (`wispr_text`);
  - the microphone actually used (BlackHole on every clip).
- **Pasted text vs Wispr's record:** they differ on 83 clips. On 81 the difference is only list formatting (bullets or numbers as pasted, HTML as stored). On one clip the pasted text had one more self-correction removed, and on one a single word differed. The pasted text is what was scored.



Wispr Flow: microphone BlackHole 2ch, language English



Wispr Flow: Auto Cleanup set to Light



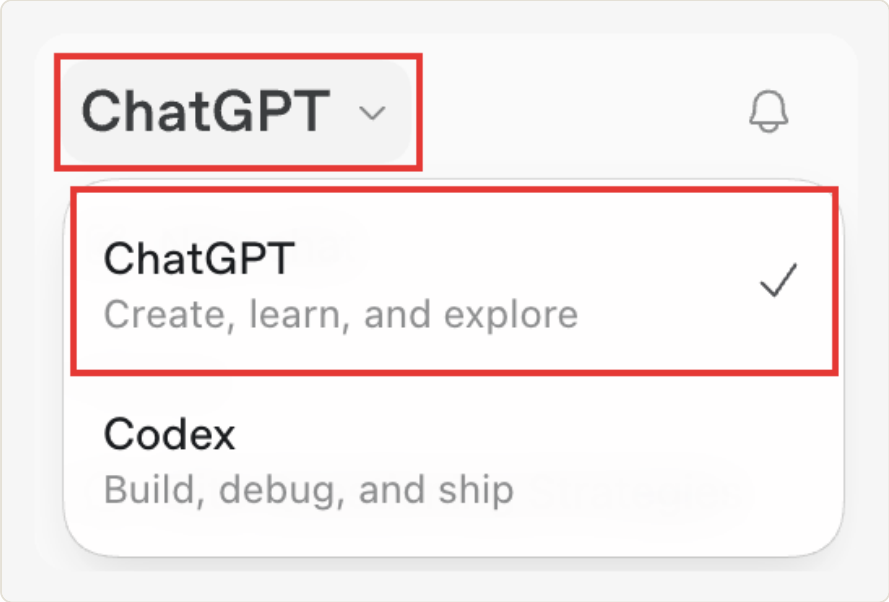
Wispr Flow: Command Mode and Press Enter command off

A.4 ChatGPT 26.1103.41825 — Mac A

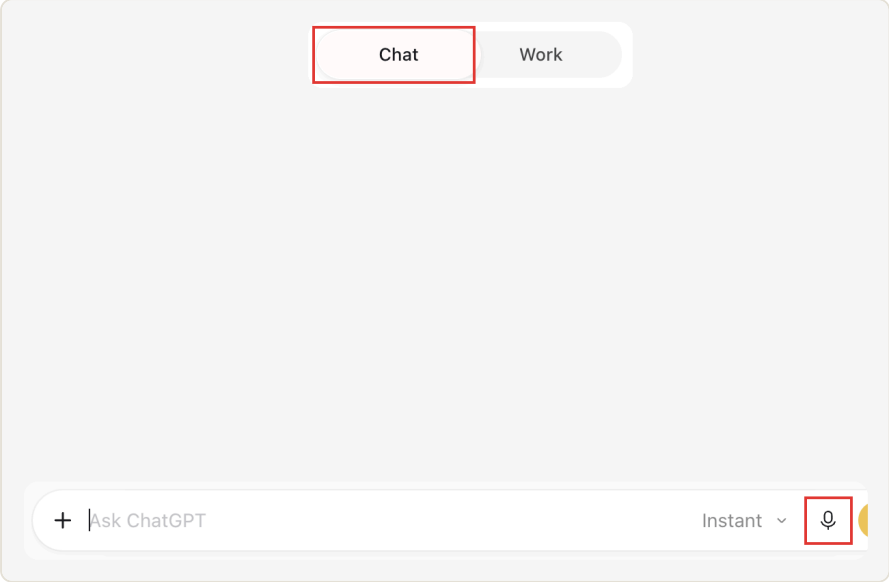
Where in ChatGPT	Setting	Before → after
Sidebar → mode switch	Mode	Codex → **ChatGPT**
Above the prompt box	Composer	Work → **Chat**
macOS microphone permission	Microphone	not asked → **allowed** (at the macOS prompt)
Account	Plan	**Plus**
macOS (during the run only)	System default input	built-in microphone → **BlackHole 2ch**, restored after

- **Control:** "Dictate" → "Stop dictation", pressed through Accessibility. ChatGPT records, then transcribes the whole clip after stop.
- **Never pressed:** "Transcribe and send" and "Send".

- **Accessibility flag:** this app exposes its page to Accessibility only with `AXEnhancedUserInterface` on. The driver turns it on for the run and off at the end.
- **Recorded on every row:** app version, mode, composer, model label (it affects replies only, not dictation), plan, system input.



ChatGPT mode set to ChatGPT (not Codex)



ChatGPT composer on Chat, with the Dictate button

A.5 Gemini 1.141.3.1394 — Mac A

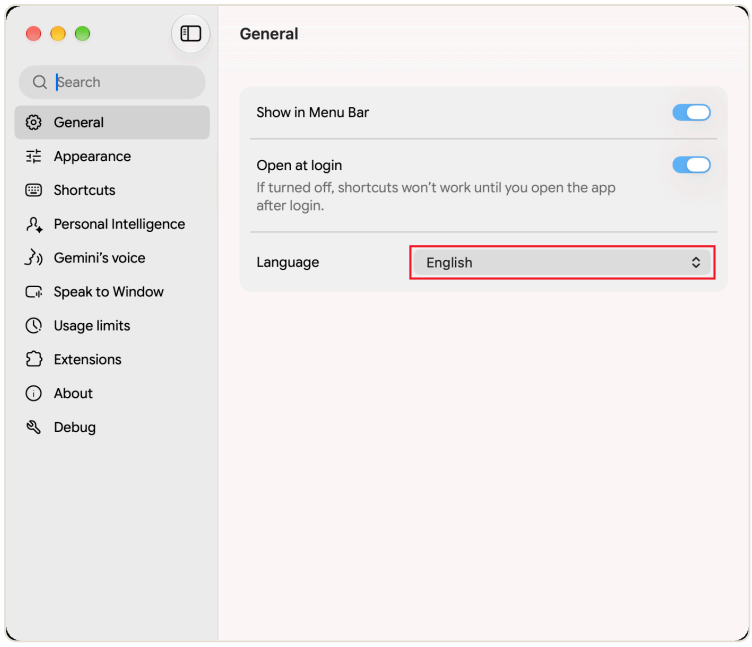
Where in Gemini	Setting	Before → after
Settings → General → Language	Language	English (United Kingdom) → **English**
Settings → Usage limits → Auto model selection	Auto model selection	on → off (for one test) → on again. Unchanged in the end.
Prompt box → model picker	Chat model	Auto. Unchanged in the end.

Where in Gemini	Setting	Before → after
macOS (during the run only)	System default input	built-in microphone → **BlackHole 2ch**, restored after

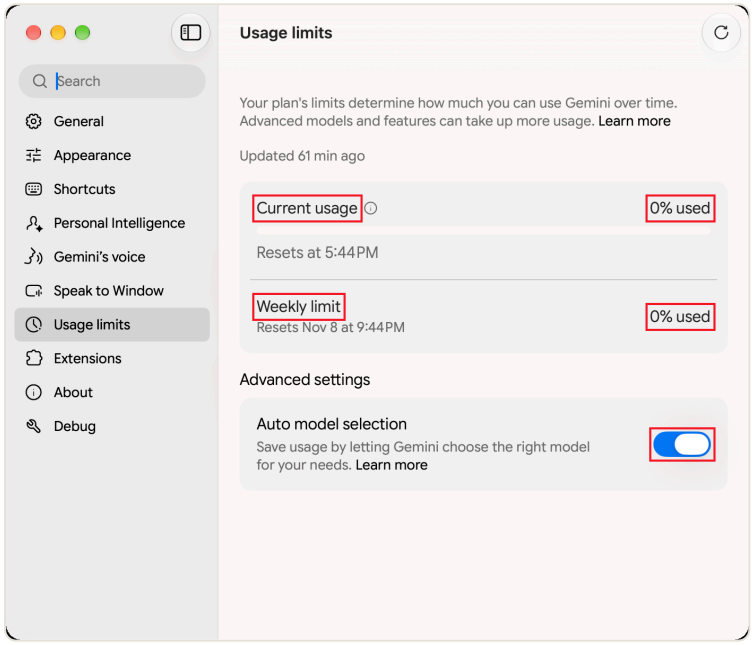
- **Chat model doesn't affect dictation:** the same clip gave an identical transcript with the model on Flash-Lite and on Auto. 3.1 Pro and 4 Argon couldn't be selected on the free account.
- **Unchanged:** Show in Menu Bar on, Open at login on, usage and crash reports off. "Speak to Window" (voice conversation) not used.
- **Control:** "Use microphone (⌘D)" → "Stop (⌘D)", pressed through Accessibility. **Never pressed:** "Submit".
- **Cloud or on-device: cloud.**
  - Gemini opens a network connection when the microphone starts.
  - Apple's on-device recogniser is never started on Gemini's behalf; this is checked after every clip (`local_asr_started` = "no" on all 766 rows).
  - Gemini has no on-device dictation setting.
- **Usage limits:** Gemini can switch to smaller or local models past its usage limits. The run guarded against this:
  - it read Settings → Usage limits ("Current usage" and "Weekly limit") at the start and every 25 clips;
  - it recorded both on every row;
  - it would have stopped at 95%. Both stayed at **0%** for the whole run, so dictation doesn't count toward them.
- **Silence:** Gemini ends dictation by itself after about 10 s without speech (10.1 s and 10.7 s on the 20 s and 15 s silence clips). On silence clips this is recorded as data (`self_stopped_s`); on speech clips it counts as a failed attempt.



Gemini prompt box: model Auto, microphone button



Gemini language English



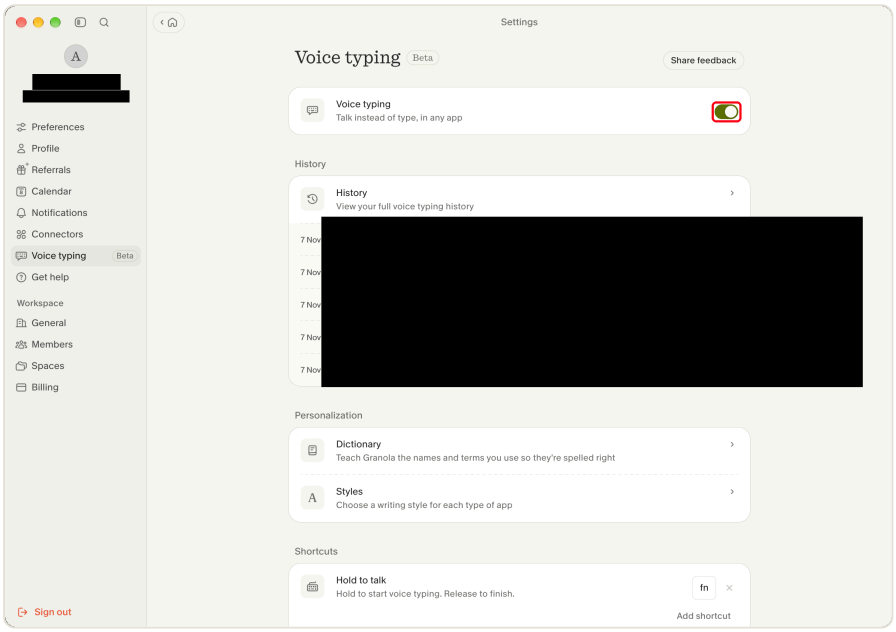
Gemini usage limits (0% / 0%) and Auto model selection

A.6 Granola 7.749.1 — Mac B

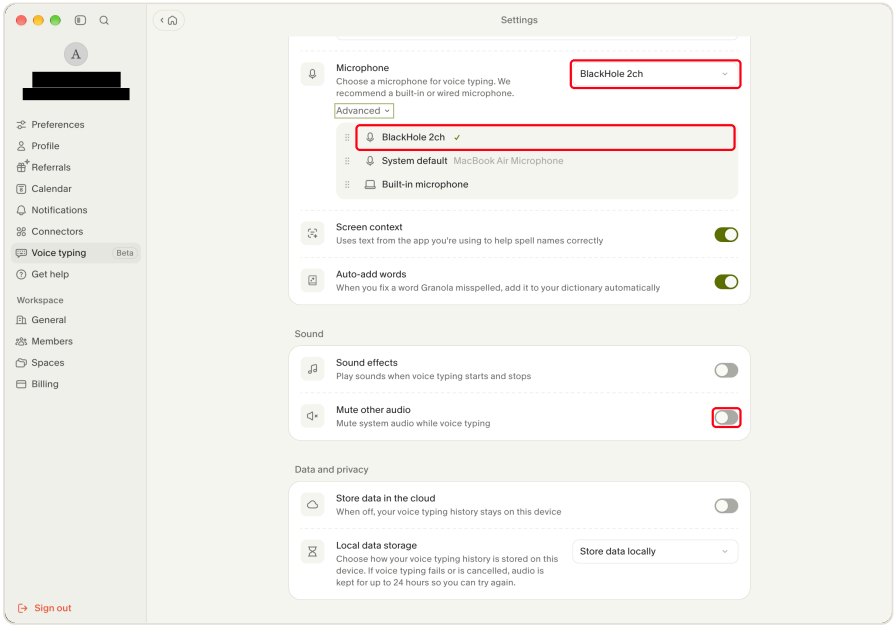
Where in Granola	Setting	Before → after
Settings → Voice typing	Voice typing (beta)	off → **on**
Settings → Voice typing → Microphone	Microphone	Built-in microphone → **BlackHole 2ch** (listed under "Show other devices")
Settings → Voice typing → Microphone → Advanced	Mic fallback order	Built-in, System default → **pressed "Use System default first"**; now BlackHole 2ch, System default, Built-in

Where in Granola	Setting	Before → after
Settings → Voice typing → Sound	Mute other audio	on → **off** (it could mute the clip)
"Set up dictation" wizard (opens when Voice typing is turned on)	—	Step 1 (shortcut test) passed, then closed; it changed nothing

- **Unchanged:**
  - hold-to-talk fn; hands-free fn then fn, and fn + Space; English;
  - Screen context on; Auto-add words on (acts only when a user corrects a word);
  - Sound effects off; cloud storage off.
- **Installation:** downloaded from Granola's update server; checksum, code signature and Apple notarization verified.



Granola Voice typing on

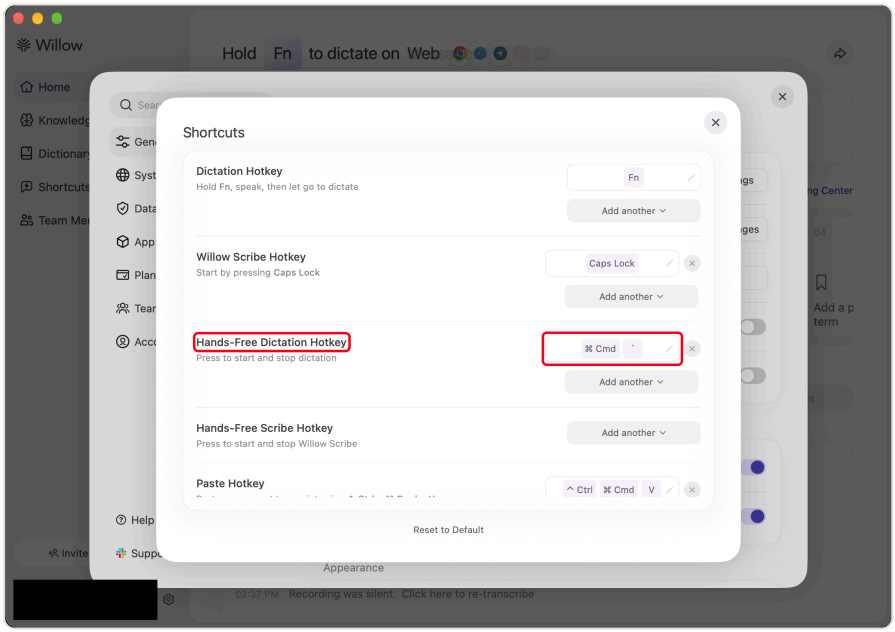


Granola microphone, fallback order and Mute other audio

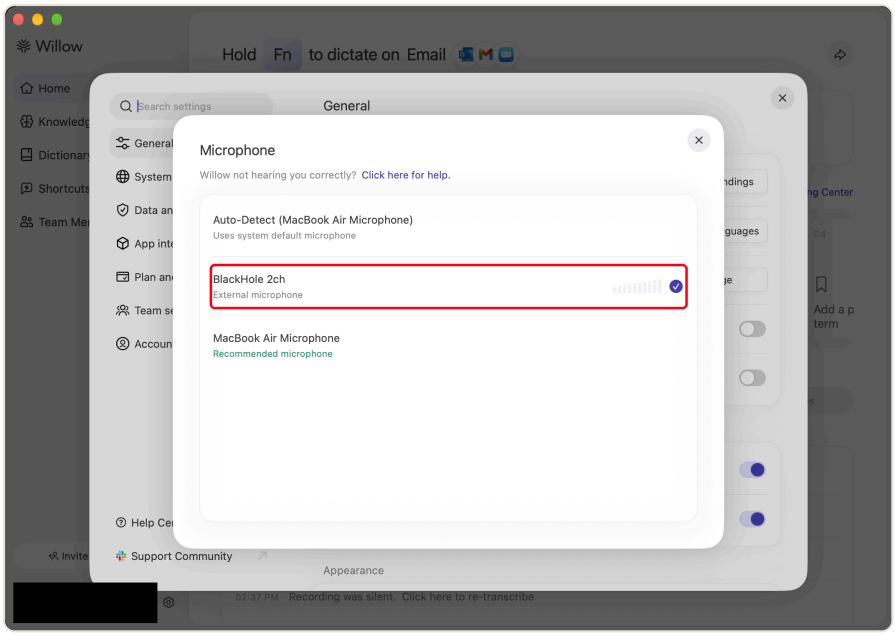
A.7 Willow Voice 2.6.15 — Mac B

Where in Willow	Setting	Before → after
Plan and billing	Plan	Free → **Pro**
Settings → General → Microphone	Microphone	Built-in microphone → **BlackHole 2ch**
Settings → Shortcuts	Hands-Free Dictation Hotkey	none → **⌘ + backtick**. Willow ignores a simulated fn, so a normal key combination is needed.
Settings → System → Notifications	Product Notifications	on → **off**. In testing, a milestone popup stole focus; the change was made in Willow's settings file with the app closed.

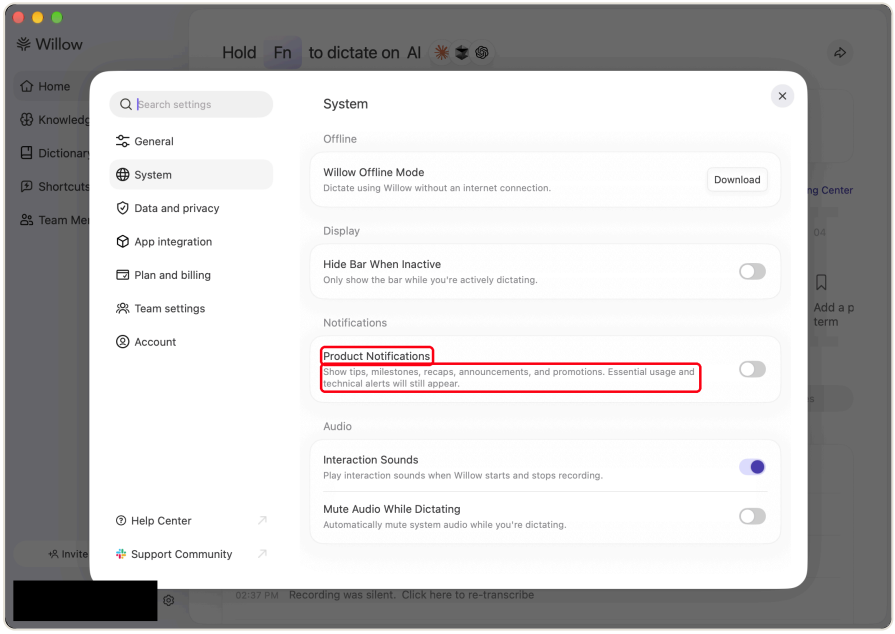
- **Unchanged:**
  - language English, auto-detect off; hold Fn to dictate; Caps Lock for Scribe;
  - Context Awareness on;
  - voice commands "press enter" and "paste clipboard" on (neither phrase occurs in any clip);
  - Interaction Sounds on (played on the speakers); Mute Audio While Dictating off;
  - Privacy Mode on;
  - offline model not downloaded, so transcription is in the cloud.
- **Verified:** Willow's settings file differs from the original in exactly these keys.
- **Other actions:** the harness presses Dismiss on Willow's "We couldn't hear you" popup before each attempt, because the popup steals focus.



Willow hands-free hotkey ⌘ + backtick



Willow microphone set to BlackHole 2ch



Willow product notifications off

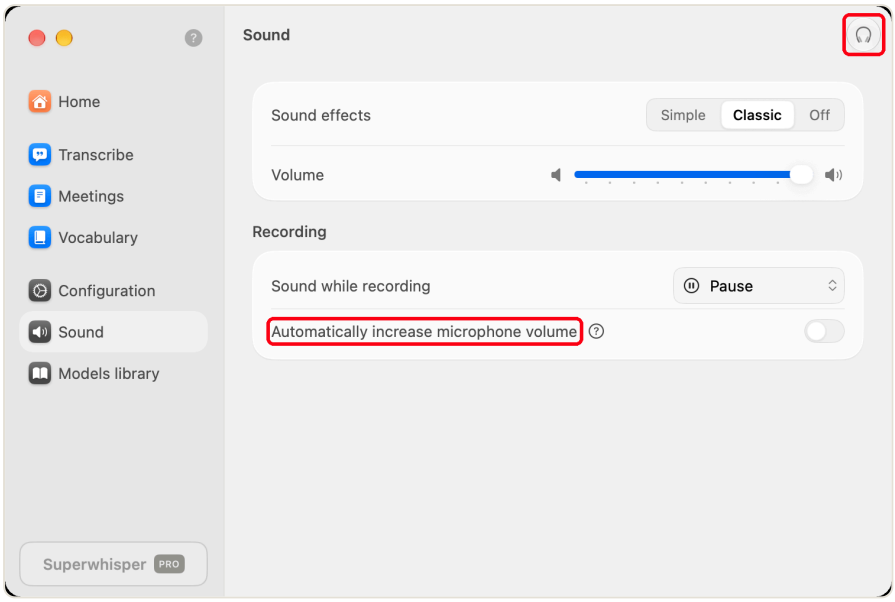
A.8 Superwhisper 3.2.1 — Mac B

Where in Superwhisper	Setting	Before → after
Account	Plan	trial → **Pro**
Microphone (headphones icon / menu bar)	Microphone	MacBook Air Microphone (default) → **BlackHole 2ch**
Settings → Sound → Recording	Automatically increase microphone volume	on → **off** (it would change the input level)

- **Unchanged:**
  - mode "Default" (S1-Voice + S1-Language cleanup); English;

  - context from app, clipboard and selection off; realtime off; literal punctuation off; auto-capitalise on;
  - vocabulary and replacements at defaults; shortcuts at defaults (not used);
  - sound effects "Classic", sound while recording "Pause"; recordings kept.

- **Control:** started and stopped through Superwhisper's `superwhisper://record` link, so no key is pressed.
- **Extra capture:** both its raw speech-to-text and its cleaned result were saved for every clip.

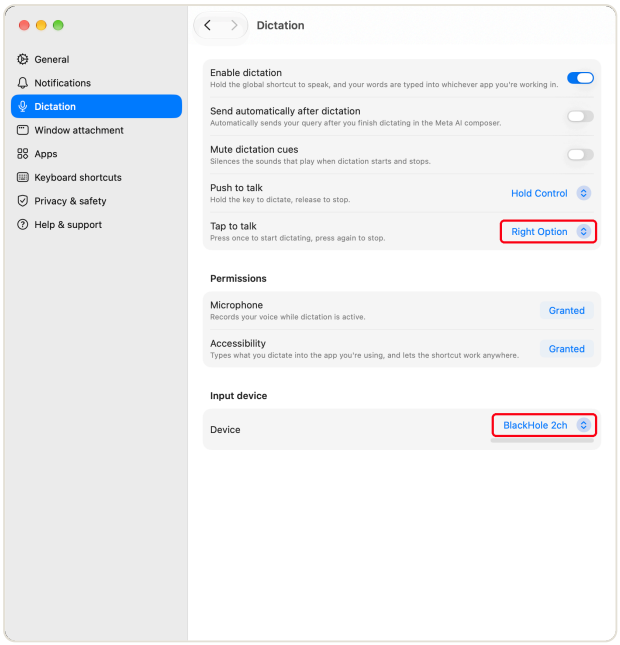


Superwhisper microphone and automatic volume off

A.9 Meta AI 12.0 — Mac B

Where in Meta AI	Setting	Before → after
Settings → Dictation → Tap to talk	Shortcut	No shortcut → **Right Option** (a start/stop toggle no other tool uses)
Settings → Dictation → Input device	Device	System default → **BlackHole 2ch**

- **Unchanged:**
  - Enable dictation on; push to talk Hold Control (not used);
  - Send automatically after dictation off (it applies only to Meta AI's own chat box, which wasn't used);
  - dictation cues on; language English (US); window attachment off.
- **Which dictation was used:** Meta AI's system-wide dictation, which types into the front app (TextEdit), not its chat box.



Meta AI tap to talk and input device

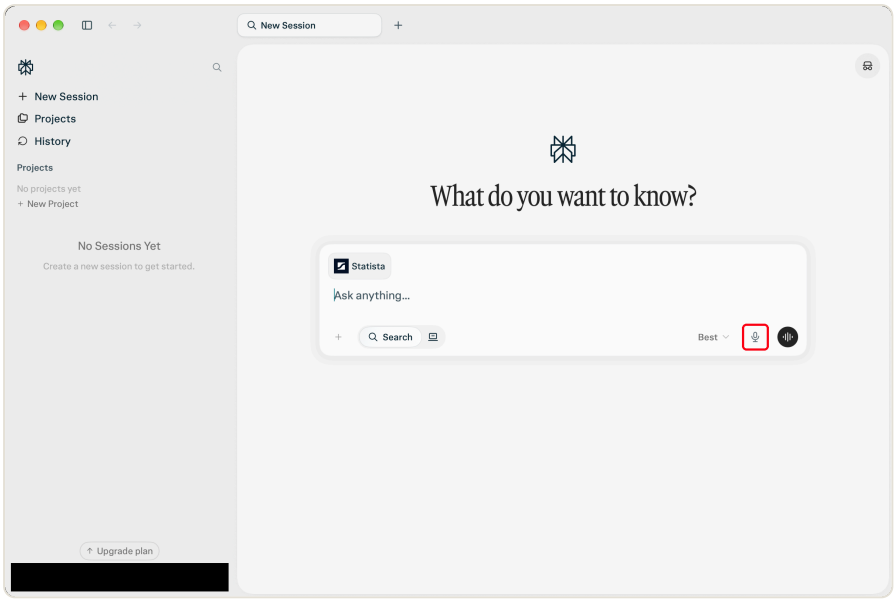
A.10 Perplexity 26.43.0 — Mac B

Perplexity has no system-wide dictation; its "Start voice" shortcut opens a spoken conversation. Dictation exists only in its own search box, so that box was the capture target:

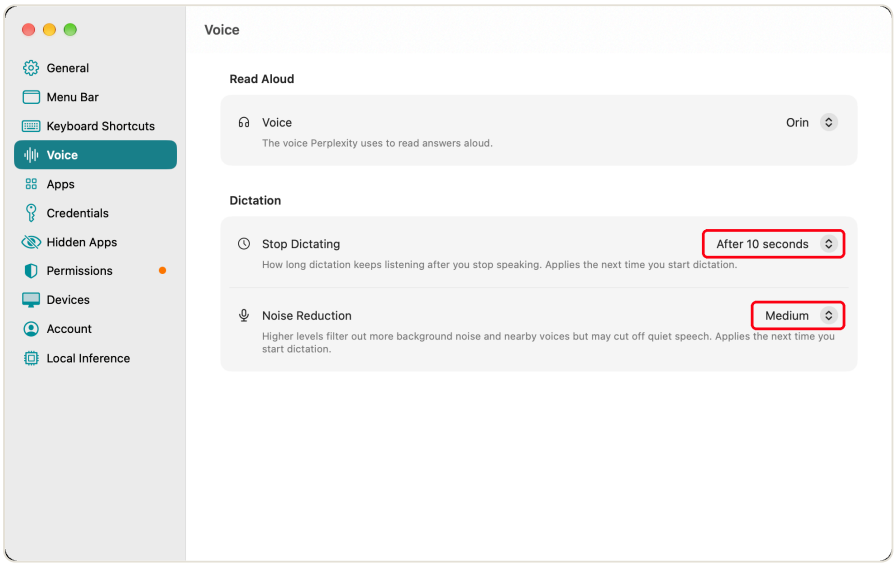
- The harness presses its microphone and stop buttons through Accessibility, and reads the box the same way.
- The query is **never sent**: Return and the send button are never pressed, and the box is cleared before every clip. After the run the app still showed no sessions.
- Perplexity has no microphone setting; it records from the Mac's default input. The harness therefore switched the **system default input to BlackHole 2ch** for the run, and restored it automatically at the end, including after Ctrl-C or a crash.

Where in Perplexity	Setting	Before → after
Settings → Permissions → Microphone	Microphone permission	not granted → **granted**
macOS (during the run only)	System default input	built-in microphone → **BlackHole 2ch**, restored after

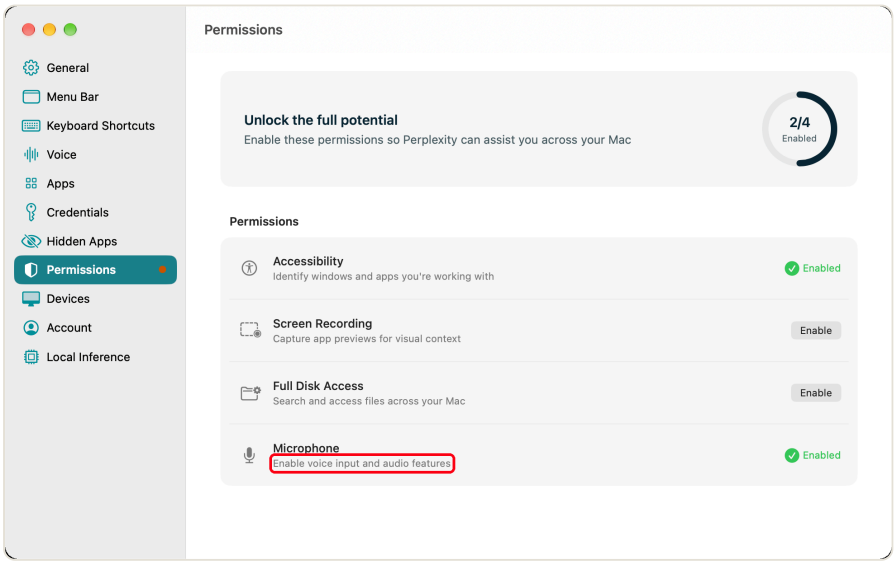
- **Unchanged:** both dictation settings were left at their defaults and logged on every row:
  - Stop dictating: "After 10 seconds" (of silence);
  - Noise reduction: "Medium" (higher levels "may cut off quiet speech").
  - Search mode and model were also left at defaults (never used).
- **Accessibility permission:** shown as enabled at the end of the evaluation but not at setup; the harness doesn't rely on it.



Perplexity search box and microphone button



Perplexity dictation settings (defaults)



Perplexity microphone permission

B Appendix B: The judge prompt

The exact prompt sent to every judge for every draft (amend-va 0.2.1, prompt version v1). `{spoken}`, `{intended}` and `{draft}` are replaced by the three texts; the response must match the JSON schema `{"fixed": string}`.

```
You are evaluating the output of a dictation tool. You will be given the INTENDED text
(what the user wanted to end up with) and the DRAFT (what the dictation tool produced).

Your job: find what the user would actually have to fix in the DRAFT so that it serves
the same purpose as the INTENDED text, and fix only that, using the fewest possible
word edits. The DRAFT does not need the same wording as the INTENDED text. If a reader
would understand the same thing and act the same way, leave it as it is.

Fix these (they change the meaning, or the user would have to retype them):
- Misrecognized or wrong words ("weight" -> "wait", "offensive" -> "defensive",
  "chronic" -> "cron")
- Missing or extra content of any length. Add anything the INTENDED text says that the
  DRAFT is missing, and delete anything the tool added that is not in the INTENDED text
  (for example an answer it generated, invented content, or repeated text)
- Filler words ("like", "you know", "um", "I mean") that the INTENDED text left out,
  including discourse words at the start of a sentence ("Okay", "OK", "So", "Oh",
  "Well", "Oh, and") when the INTENDED text does not start that sentence with them.
  Only remove fillers: never add a filler or discourse word, even if the INTENDED
  text has one that the DRAFT left out
- Wrong names, including misspelled names, and numbers or dates with the wrong value
  (a different number or date, not the same one written differently)
- Anything that must be written exactly: code identifiers, file names, commands, field
  and config values, versions in code, emails, URLs, and product and proper names.
  Make them match the INTENDED text exactly, including any digits in them
  ("calculate prorated amount" -> "calculateProratedAmount",
  "start date" -> "start_date", "billing ts" -> "billing.ts",
  "PG Bouncer" -> "PgBouncer", "node twenty alpine" -> "node:20-alpine")
- A wrong unit, or an informal/slang unit ("512 megs" -> "512 MB")

Do NOT fix these (same meaning, or only formatting):
- Different wording with the same meaning: articles, singular/plural, tense or
  contraction changes that keep the meaning ("a" vs "the", "note" vs "notes",
  "that handles" vs "that'd handle"), synonyms, paraphrases and word-order changes
  ("I'll mention" vs "I want to mention")
- Small grammar slips and spelling variants that do not change what the reader
  understands: plural vs possessive ("league" vs "league's", "deadlines" vs
  "deadline's"), "they" vs "they'd", "we" vs "we'd", "logs" vs "log", "holiday" vs
  "holidays", and a missing or extra courtesy word ("please")
- Punctuation, capitalization, line breaks, bullets, numbered lists, markdown
- How ordinary numbers are written. Some dictation tools cannot write digits, so
  numbers in words are acceptable, as are digits, decimals vs fractions, and spoken
  separators ("4" vs "four", "$1,450" vs "fourteen fifty", "1.5" vs "1 1/2",
  "$618.40" vs "618 dollars and 40 cents", "10:30 AM" vs "ten thirty AM"). This covers
  amounts, quantities, dates, times, phone numbers, and order, account and ticket
  numbers ("415-555-0147" vs "four one five, five five five, zero one four seven",
  "#ER-3312" vs "ER dash three three one two"), as long as every digit and letter is
  right. A wrong digit is still an error. It does not cover the exact-text items
  above (code, file names, commands, versions in code, emails, URLs, product and
  proper names), where digits must match.
- Standard abbreviations ("tbsp" vs "tablespoons", "lb" vs "pound",
  "mph" vs "miles an hour", "Dr." vs "Doctor")
- Hyphenation and compounding of ordinary words ("short-staffed" vs "short staffed")
- Contractions vs full forms ("we're" vs "we are")
- US vs UK spelling

Rules:
- Leave every part of the DRAFT that does not need fixing exactly as it is.
```

```
- When you insert or replace words, match the formatting style of the surrounding
  DRAFT text, including writing numbers as words if the DRAFT does.
- Make each fix as small as possible: replace one word rather than rewriting the
  sentence around it. But fix a missing or extra passage in full.
- If the DRAFT is empty, "fixed" is the INTENDED text.

You are also given SPOKEN: a verbatim transcript of what the user actually said,
including any self-corrections. A self-correction can replace something ("Tuesday,
no wait, Wednesday", "ten AM, sorry, ten thirty", "five times, no, make it three")
or retract something ("get milk and eggs, oh wait, we have eggs, take that off").

Self-corrections are lenient. If a difference between the DRAFT and the INTENDED
text exists only because the DRAFT kept a self-correction from SPOKEN (for example
it kept both the original and the corrected version, kept a retracted item together
with the retraction, or kept cue words such as "no wait", "sorry", "actually no",
"no, make it", "scratch that", "take that off"), leave it unchanged.
It is not an error.

"I mean" is a self-correction cue only when it introduces a correction ("Tuesday,
I mean Wednesday"). Anywhere else ("and I mean, fix every import") it is a filler
and follows the filler rule above.

This takes priority over the rule about extra content. Keep a kept self-correction
whole: the original, the cue words and the correction. This applies even if the cue
words were misheard ("wait now" for "wait no", "no way" for "no wait"). Never delete
only part of it.

But if the DRAFT ends up with the wrong final value, fix it. For example, it kept
"Tuesday" and dropped the correction to "Wednesday", or it mixed the two up.
Everything else follows the normal rules above.

<spoken>
{spoken}
</spoken>

<intended>
{intended}
</intended>

<draft>
{draft}
</draft>

Respond with JSON whose "fixed" field holds only the corrected draft text, with no
explanation, notes or reasoning in it. If no changes are needed, "fixed" is the DRAFT
exactly as given.
```