

I Built a Mac You Can Just Talk To. It Runs on Jev.

SayOpen hears what you say, works out what you mean **while you're still talking**, and does it: opens apps, clicks buttons inside them, scrolls, picks files, types with Wispr Flow.

Built on Jev by TypeSafe AI

~400 ms per decision

macOS · speech runs on-device

First, what Jev is

Jev is a **System One model** from TypeSafe AI. You don't chat with it. You hand it a sentence and a list of options, and it hands back **a pick and a probability** for each option. That's it.

That sounds small. It's the missing piece for voice. An LLM writes you a paragraph in two seconds. Jev answers **"which of these 32 actions did he mean, and how sure are you?"** in about 0.4 seconds. Fast enough to act mid-sentence.

PART 1 What you can say to it

AREA	SAY SOMETHING LIKE
Apps	"open Slack" · "open the app I make 3D models in" → Blender · "Claude kholo"
Inside Claude	"change model to Opus 5.5" · "set effort to high" · "open the tab that needs input"
Browser	"open the MANIK 2.0 Google Doc" · "switch to the Usage tab" · "click the first sign-in"
Moving around	"scroll down" · "scroll to the top" · "go down two" · "press escape" · "go to tab 3"
Terminal	"yes, trust this folder" · "option two" → picks from Claude Code's list
Files	"open built with Jev PNG" (heard as "build with Jeff PNG", still found)
Typing	"open Wispr Flow" ... talk ... "close Wispr Flow" → stop phrase removed
Mac	"night mode" · "volume to half" · "skip this song" · "take a screenshot"

THE CHAINING BIT

"Open terminal, add a new tab and let me type" is three commands. SayOpen splits on **and / then / phir / aur** and fires each one the moment it's decided, so the terminal is already open before you finish the sentence.

PART 2

Why Jev, not hand-written rules

We tested it the fair way

We wrote the best rule engine we could in a day (patterns, a nickname list, spelling-distance matching), then **ran both on the same 74 real sentences**. Same executor, same actions. Only the deciding differs.

TEST	JEV	RULES
All 74 commands	63 / 74	43 / 74
Tabs and windows	14 / 16	9 / 16
System (volume, dark mode...)	15 / 17	8 / 17
Hinglish	6 / 6	2 / 6
Ignoring chatter	8 / 8	5 / 8
Open-an-app, 50 phrasings	96%	70%
"Describe the app" phrasings	13 / 14	4 / 14
Time per decision	~400 ms	~8 ms

READ THIS HONESTLY

Rules only win when you say the **exact name**. That's why SayOpen uses both: code takes the exact, fixed phrases instantly ("scroll down", "press escape"), and **Jev takes everything humans actually say**.

PART 3

Seven jobs Jev does in one app

1 · Which action?

Picks from 32 actions: open, click, scroll, search, note, volume... Routing.

2 · Which app?

Picks from every app installed (115 on this Mac), even when you only describe it. Ranking.

3 · Which words are the content?

"remind me to call Ankit": it picks the span, so the reminder holds only your words. Extraction.

4 · Is this even a command?

"I love music" shouldn't open Music. It judges intent before anything fires. Verification.

5 · What should I click?

Reads the buttons on screen and picks the one that gets you closer to the goal. Agentic UI.

6 · Which file did he mean?

Several files match "my PPT": it picks the one you meant. Disambiguation.

7 · How sure am I?

Every answer comes with a probability. Above 0.5 it acts when you finish the clause; to act mid-sentence it needs 0.8. Below that, it waits for more words. Closing a tab also needs a real "close" word, because wrong is worse than slow.

1 Hear it on the Mac

Apple's on-device speech recognition streams words as you speak. We feed it every app name as vocabulary, otherwise it writes "Claude" as "cloud". A wake phrase ("Hey Mac") stops a YouTube video from giving commands.

2 Cut the sentence into clauses

Code finds the seams (and, then, commas). Each clause is judged on its own. A clause is "closed" once you've moved on, so an open one like "open Claude..." doesn't fire when you're really saying "open Claude Code".

3 Ask Jev everything in one request

One call asks all the questions at once: action, app, content words, setting, is-this-a-command. Most answers go unused. A second round trip would cost ~400 ms, so asking speculatively is cheaper.

```
# one clause, one request
"open a new tab in claude code"  (real log, 1.2 s)
  action → new_tab    0.87
  app    → Claude     0.98
  command → yes       0.95
# code then turns "tab" into a Claude session: Go > Code, File > New Session
```

4 Do it like a person would

Menus first ("Shell > New Tab" beats pressing ⌘T). Then real buttons, read through macOS Accessibility. Apps that fake their buttons (Dia's tabs) get a real mouse click. Terminal lists are read as text and answered with arrow keys and Enter.

5 Teach it the places people name

"Cowork", "the tab that needs input", "the model": a small map of where those live in Claude. Your browser history finds docs by title, so "the MANIK 2.0 doc" opens the right sheet, or jumps to its tab if it's already open.

6 Hand typing to Wispr Flow

Say "open Wispr Flow" and SayOpen presses Wispr's hands-free shortcut, then stops treating speech as commands. Say "close Wispr Flow" and it stops Wispr and deletes the stop phrase, checking the text box after every keystroke so it never eats your words.



PART 5

What broke, and the fix

"Open the tab that needs input" opened a new tab

A new tab now needs a "new" word. Without one, it finds the existing tab on screen. In Claude it jumps to the session marked **Awaiting input**.

"Close Wispr Flow" deleted too much, then nothing

It counted characters from Wispr's copy. It now reads the actual text box and deletes back to exactly where your sentence ended, full stop kept.

"Product" opened Wispr Flow. "Make" opened Terminal.

A one or two word fragment must actually name the app now. Describing an app in a full sentence still works.

"Clear this from the text from the tab" closed a tab

Closing and quitting wait for the end of the sentence and need a real close word.

Dia's tabs ignored every click

They claim to be pressable and then refuse. SayOpen now falls back to a real mouse click and puts your pointer back.

Claude's model menu was invisible

It sits 28 levels deep in the window, exactly where our scan stopped. Deeper scan, plus clicking through Claude's "Switch model?" confirm.

WHERE IT STILL FALLS SHORT

Speech recognition is the weak link, not Jev: it heard "Monday standup" as "midday stand". It's Mac only, it needs Accessibility permission to click and type for you, and it isn't a public download yet.

PART 6

Use Jev in your own product

IF YOUR APP HAS...

JEV CAN...

A support inbox

route each ticket to the right team, with a confidence to auto-send or escalate

A lead list

rank leads against your ideal customer, in plain English criteria

Forms and messages

pull out the exact words that matter, never paraphrased

Any "if this then that"

replace a brittle prompt-and-parse step with a typed, scored decision

Everyone's building chatbots. The unlock is small, fast decisions you can wire into code.

Pick one "if" in your product and give it to Jev this week.

@manikk.ai

75 Days of AI · SayOpen, built with Jev