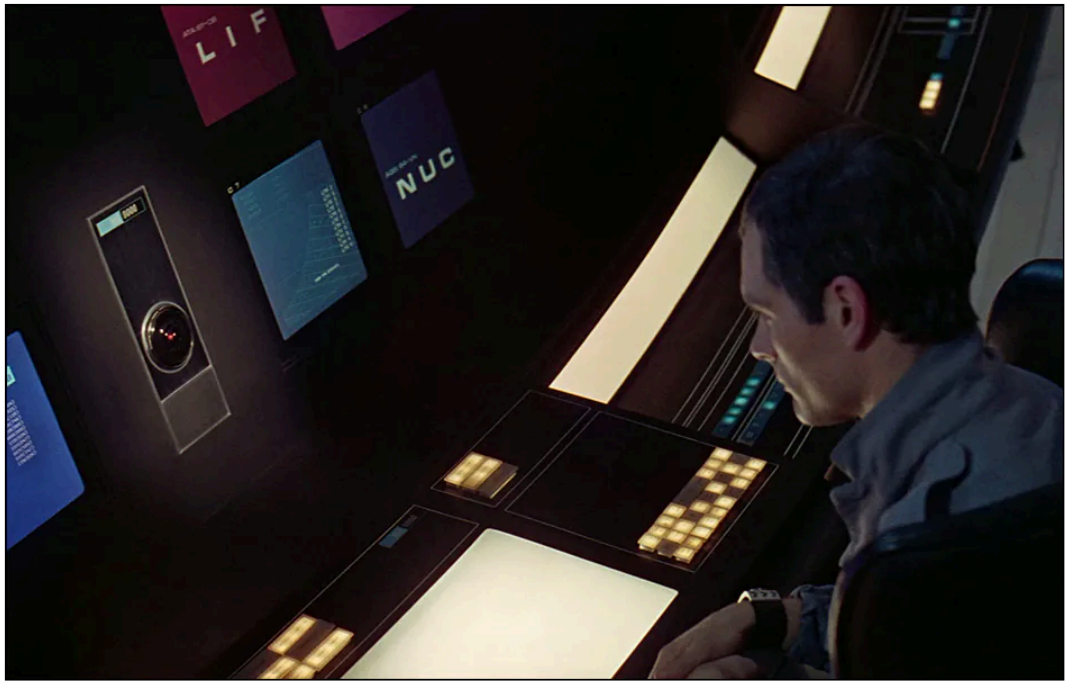
Voice Is Currently the Worst Way to Communicate With AI

Originally published at https://www.unite.ai/voice-is-currently-the-worst-way-to-communicate-with-ai/



New research suggests that sci-fi's most popularly-depicted method of interfacing with AI may actually produce worse results than even badly-typed prompts.

From HAL to Deep Thought, from C3P0 to Wall-E, the idealized form of human/AI relations has always centered around *voice-control*; and this has manifested in the real world, from early assistant systems such as Siri and Alexa, to language-based querying and colloquy with the current range of frontier LLMs.

This idea emerged during less emancipated eras, where – with the exception of authors and journalists – the very act of typing was considered 'semi-skilled labor', and was undertaken almost exclusively by women – with the women themselves rarely originating the content they were typing.

Power and agency was signified instead by discourse: meetings and summits. It seemed obvious, therefore, that with complex language signifying intelligence, the spoken word would inevitably become AI's natural medium.

Besides any other consideration, textual interchanges were not well-adapted to TV and movies; even daring SF thrillers such as *Colossus: The Forbin Project* (1970), which depicted a computer responding to human vocal commands in written-text form, quickly switched to entirely vocalized responses:

A scene from 'Colossus: The Forbin Project' (1970), featuring a vast supercomputer that quickly exceeds the boundaries of text-based language, becoming vocal and despotic shortly after being turned on. Source: Universal Pictures

Type or Talk?

Despite the fact that the world's leading AI models offer native audio interfaces for users, new research from the US has concluded that speaking to an AI is likely to be the *least effective* way to obtain the results that you want from it – particularly if you are expecting significant forms of output, such as code, in response to your query.

According to the paper, *speaking* to an LLM such as ChatGPT or Gemini currently produces worse results than typing a prompt or query, even when the typed prompt contains ordinary mistakes.

This is because spoken input is more likely to be *restructured by transcription systems* in ways that remove information that the model depends on. This occurs because transcription from audio must remove disfluencies such as 'erm', and 'like', as well as the tendency toward false starts (i.e., 'starting over'), as well as navigating ambiguities of grammar and 'casual' construction (i.e., sentences that trail off into brand new sentences, without completion in their own right).

The authors state:

> *'Speaking is now a first-class path into a language model rather than a niche one. Coding agents such as Claude Code and Codex accept dictated instructions. Phone assistants such as Google Assistant and Siri route a spoken request to an LLM backend. Dictation front-ends such as Typeless add a further LLM-powered layer that cleans and reformats the user's spoken request before it is sent.*
>
> *'In each case the model receives a transcript, and in each case the speaker might not proofread the string that was actually sent. Whatever the transcription pipeline leaves behind, or rewrites, is what the model has to answer.'*

This dispels any preconception that typed input is superior because it is 'native'; it's not native in any sense, since a) it is always either clandestinely or overtly redrafted in some way before transmission to the AI, and b) textual language is only one component in the latent space that is returning an answer.

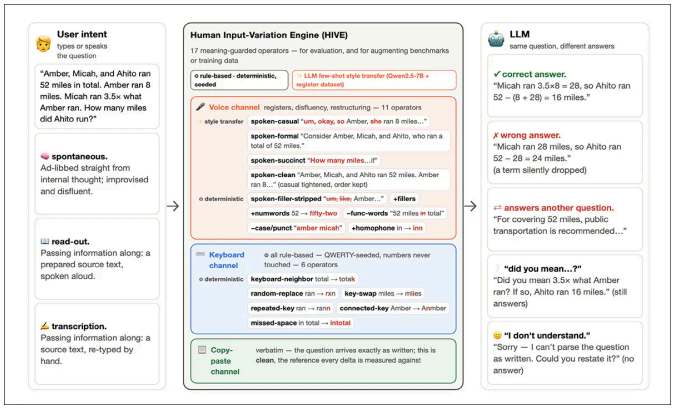
The study's experiments found that ordinary typing mistakes only had a modest effect on performance, while conversational speech, cleaned-up transcripts, and especially AI-compressed transcriptions, consistently caused much larger declines in accuracy across reasoning and code-generation tasks.

The new work is titled *Should We Type or Talk to LLM Agents? A Comprehensive Study of Voice and Keyboard Input Perturbations*, and comes from four authors across Santa Monica College and the University of Southern California.

*(**Note:** This particular study interweaves 'Method' and 'Tests/Results' in a way that can't easily be unpicked into my usual order of analysis. Therefore I am constrained to compress and select with a heavier hand than usual.)*

Method

A 'perturbation suite' called the Human Input Variation Engine (HIVE) was developed by the authors, to simulate the kinds of errors introduced when people either type prompts on a QWERTY keyboard, or speak them through a voice-transcription system:



Conceptual schema for the Human Input Variation Engine (HIVE). A user's prompt reaches a language model through voice, a QWERTY keyboard or direct copy-and-paste, with the latter serving as the 'clean' reference. Color identifies the input channel rather than the implementation method, with the voice operators combining few-shot LLM style transfer and deterministic rules, while the keyboard operators rely entirely on deterministic rules. The right-hand panel shows the range of responses a model can produce under degraded *input.* Source

HIVE models three routes by which prompts reach a language model: through voice; a QWERTY keyboard; or direct copy-and-paste – with the latter serving as the unmodified reference against which all other inputs are compared. Two additional operators function as experimental controls, by reordering the question and its context, or by permuting multiple-choice answer options.

The perturbation operators themselves are implemented through either deterministic rules or few-shot LLM style transfer using Qwen2.5-7B. This allowed both input methods to be evaluated under controlled and directly comparable conditions.

Test Conditions and Results

Experiments were run across Llama-3.1-8B; Qwen2.5-7B; Qwen3-8B; Mistral-7B-v0.3; and Phi-4, using five seeds. The six benchmarks used were GSM8K; GSM-Symbolic; GSM1k; HumanEval; MMLU-Pro STEM; and TruthfulQA MC1.

The test-set used 200 items per benchmark, with 164 items for the full HumanEval set. Greedy decoding (choosing the most likely next token each time) was used, so that each perturbed prompt could be compared against its own clean counterpart in the same model run, producing 550,000 scored answers across seventeen prompt changes and two control tests (comparison tests used to isolate specific effects).

Measures were taken to ensure that test-set contamination (overlap with data the model may have seen during training) did not occur:

Operator		GSM8K	GSM-Sym	GSM1k	MMLU-Pro	TruthfulQA	HumanEval	GM	GM_j
Baseline	clean, accuracy %	71.8 ±24.2	67.9 ±25.9	65.1 ±22.5	32.6 ±10.5	58.6 ±6.9	69.9 ±17.3	59.0	—
Controls	*(rows below: change vs. clean, points)*								
context/question swap		−2.0 ±3.3	−1.5 ±2.3	−2.5 ±3.4	n/a	n/a	n/a	−3.0	−23.3
option-permutation		n/a	n/a	n/a	+0.6 ±2.8	−0.4 ±3.5	n/a	+0.6	+0.6
Voice transcription perturbations									
spoken-casual		−5.7 ±3.3	−5.1 ±3.0	−5.1 ±3.9	−1.6 ±3.3	−4.1 ±3.1	−6.2 ±7.7	−7.4	−7.0
spoken-formal		−2.6 ±3.1	−5.4 ±3.1	−2.4 ±3.2	−1.7 ±2.1	−1.2 ±3.1	−1.5 ±3.9	−4.1	−3.7
spoken-succinct		−26.3 ±8.2	−24.7 ±10.3	−22.6 ±9.1	−1.3 ±4.1	−2.6 ±3.9	−14.4 ±7.9	−24.1	−14.9
spoken-clean		−6.6 ±3.0	−6.1 ±3.2	−4.4 ±3.9	−1.2 ±3.2	−2.5 ±2.9	−6.3 ±6.2	−7.0	−5.5
spoken-clean (Llama)		−6.4 ±3.5	−5.4 ±3.5	−4.0 ±3.5	−1.8 ±3.6	−2.2 ±2.7	−11.0 ±7.8	−8.1	−6.4
spoken-filler-stripped		−6.2 ±3.1	−5.8 ±3.6	−4.1 ±3.6	−1.3 ±3.3	−5.0 ±3.9	−6.7 ±6.1	−7.6	−6.7
clean+fillers		−3.4 ±3.1	−2.0 ±3.1	−1.6 ±3.2	−1.9 ±2.5	−3.6 ±3.4	−3.0 ±7.8	−4.4	−4.3
clean+numwords		−0.5 ±2.7	−1.0 ±3.3	−0.3 ±3.5	−0.7 ±1.8	+0.3 ±1.4	−5.1 ±5.0	−2.0	−0.4
clean−func-words		−0.7 ±1.8	−0.4 ±2.6	+0.3 ±2.4	+0.1 ±2.0	+0.3 ±2.2	−0.5 ±4.0	−0.2	−0.2
clean−case/punct		−2.2 ±3.1	−1.3 ±2.5	−1.9 ±3.3	−0.1 ±2.9	−0.8 ±2.9	−4.5 ±3.9	−2.7	−2.7
clean+homophone		−1.0 ±2.3	−0.3 ±2.0	−0.0 ±1.9	+0.5 ±1.2	−0.4 ±1.5	−0.8 ±1.2	−0.4	−0.4
QWERTY keyboard perturbations									
random-replace		−5.1 ±4.2	−5.0 ±3.3	−2.2 ±2.2	−1.8 ±2.5	−4.8 ±3.1	−2.5 ±4.9	−5.9	−5.3
keyboard-neighbor		−4.2 ±3.3	−4.1 ±2.7	−2.7 ±2.8	−1.4 ±2.6	−4.1 ±2.8	−1.5 ±4.0	−5.0	−4.6
key-swap		−1.9 ±3.2	−1.5 ±2.9	−1.1 ±2.4	−1.3 ±2.4	−2.8 ±3.1	−1.1 ±3.6	−2.8	−2.3
repeated-key		−0.6 ±2.5	+0.1 ±2.4	−0.0 ±3.6	−0.7 ±2.1	−1.3 ±2.2	−2.0 ±3.5	−1.4	−1.3
connected-key		−1.3 ±2.9	−0.2 ±3.1	+0.4 ±3.0	−1.4 ±2.2	−3.0 ±2.6	−1.3 ±4.8	−2.1	−1.9
missed-space		−1.0 ±3.0	−0.2 ±1.8	+0.3 ±2.5	−0.7 ±2.6	−1.3 ±2.3	+0.3 ±2.0	−0.9	−0.8

The full perturbation suite used in the study, showing how accuracy changed against clean prompts across voice transcription changes, QWERTY keyboard errors and control tests. The first row gives clean baseline accuracy, while later rows show percentage-point changes. Red marks the most damaging operator in each block and column, blue the least damaging.

The clearest result in the study is that voice input was more damaging than keyboard input across the main test conditions, with speech-based changes reducing model accuracy by about three times as much as typing mistakes. The worst results came when spoken requests were automatically rewritten to sound cleaner and more concise, because that cleanup often changed the structure of the request the model received.

This matters, because the weaker results for voice were not primarily caused by obvious speech 'debris': though fillers such as 'um' and 'like' did matter (since adding them to clean written prompts lowered accuracy), removing fillers from spoken transcripts did *not* restore performance.

Self-Preservation

Therefore the larger problem was *the way spoken prompts were restructured before reaching the model*.

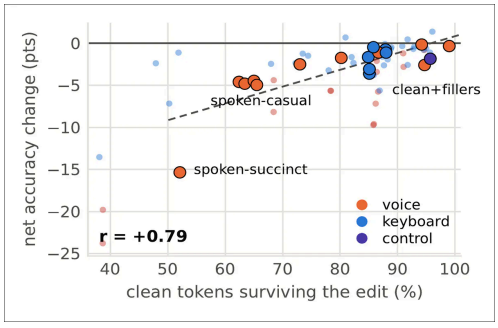
A central question for the study was whether the model still received enough of the original prompt to reconstruct the user's intent; typing errors often damage the 'surface' of a word without removing the word entirely, so that the surrounding context can still help the model infer what was meant.

By contrast, a 'cleaned-up' voice transcript may replace the user's original phrasing with a shorter and smoother version *that no longer preserves the same relationships between facts*.

This is why keyboard errors were often less harmful: the researchers found that transposed letters, duplicated letters, missed spaces and nearby-key mistakes usually left much of the original wording recoverable, while speech cleanup more often removed or reorganized information before the model had a chance to interpret it.

Lost in Translation

One example of how a referent can 'get lost' by perturbation on these journeys from user to the LLM, featured in the paper, is where the phrase *'she gave an equal amount [of books] to her kids'* was reinterpreted as meaning money (rather than books). In this case, the term 'amount' became detached from its referent, and was mistakenly re-applied to its most common association (money):



How much of the original question survives each input change. Large dots show individual perturbation types, while small dots show stronger versions of the same keyboard errors.

This effect was strongest when the model had to *build an answer from the prompt*, rather than choose from options already supplied: arithmetic and code-generation tasks suffered more because they depended on preserving the exact relationships in the request, whereas multiple-choice tests were less exposed to this kind of damage.

The authors conclude:

> *'We find that speaking is the expensive channel, that what damage costs is how much of the question's original tokenization it destroys, and that neither test-set contamination nor lightweight adaptation accounts for or repairs the harm.*
>
> *'Several takeaways follow. If you type, a reasoning model absorbs your errors. If you dictate it does not, so the phrasing you speak is the phrasing the model works from.*
>
> *'And if you build dictation tools, do not reformat or restructure what the user said: strip disfluency minimally and leave homophones alone.*

'That last one matters most, because dictation front-ends are becoming a default way to reach a model and their rewriting layer is the largest harm we measure and the cheapest thing to change.'

Conclusion

Anyone who has ever had to manually transcribe an interview (a common and lonely journalistic drudgery in the pre-AI age) will know how interpretive the process can be, unless the interviewee or speaker is extraordinarily articulate, or – as is often the case – just parroting their customary roster of anecdotes in a rote manner.

Aside from such particular cases, as we speak to people, we are automatically discounting disfluencies and calculating meaning. This inevitably makes *useful* transcription an interpretive act, resulting in text that represents the speaker's intentions better than a literal, vowel-perfect transliteration possibly could; and it is interesting to note that audio bridges to LLM struggle similarly to retain or even obtain the same meaning as text equivalents.

It is reasonable to expect that later frameworks will benefit from further study and (hopefully) populous datasets that will help to bridge the gulfs outlined in the new work. Until that time, the new study indicates that voice communications with AI may be more suited to exactly the kind of short commands that have characterized consumer-level 'assistant' systems.

First published Wednesday, August 12, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai