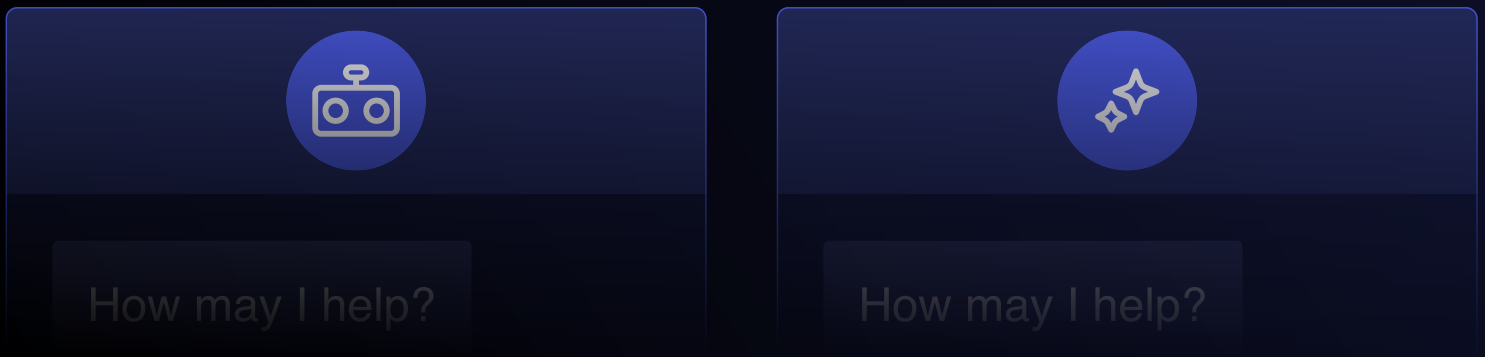
The Lure of the Chatbot

Invene

Alex Melton | July 28, 2025



Executive Summary

- Chatbots are easy to build but often fail to deliver accuracy, trust, or compliance.
- Scientific and clinical users require precision, traceability, and speed—not open-ended conversations.
- Generative AI can transform how guidelines are accessed, but only when deployed with structured retrieval, not casual chat.
- This article explores why traditional chatbots fall short and outlines a better path forward for organizations managing high-stakes medical knowledge.
- For healthcare associations and research bodies, the opportunity isn't AI hype—it's building tools that preserve authority while improving access.

One thing you have to understand about chatbots is they're not new. While yes, the rise of generative AI has fundamentally changed what they're capable of, the core problem most businesses are trying to solve with a chatbot has not.

It's almost a right of passage in every executive's career to consider a chatbot within their brand ecosystem at this point. Why is that the case?

Well, I find there's really two scenarios leadership finds themselves in that naturally lead themselves towards thinking a chatbot is a fantastic idea.

Scenario 1 (Informed Conversion)

Imagine you're a business savvy executive leader in 2013. You attend all the big conferences. Read all the thought leadership on Linkedin. You even consume podcasts and YouTube videos in your spare time. All to make sure you're ahead of the curve over your competition.

Most of the time, your mind circles around how to bring in more leads and convert them to paying customers. Especially in the early 2010s, you'd have to live under a rock to not get smashed in the face with the idea that "personalization is king" in marketing. You hear that the advantage of personalization is that you can communicate the value of your offering. But you do it in a way that resonates with the individual instead of trying to over generalize your offering and losing people in vague language.

Thinking personalization is your key to conversion, you do your research. Then it dawns on you: there's nothing more personalized than a conversation with your prospect. As luck would have it, your CRM tool comes with a chatbot embed out of the box. So, you decide to put it on your website in the hopes it increases conversion.

This same scenario repeats hundreds of times. The internet is littered with websites that chime and ping with messages from an army of bots desperate to get your business.

Scenario 2 (Informed Retrieval)

Fast forward a bit, it's now 2018. Your company has been in business for a while. Time in the market comes with a torrent of information. Anyone new to your brand will drown in an unending fractal of complex knowledge.

How on earth can you make this information easier to access?

Well, one of the tools you use for documentation offers a chatbot that can use natural language processing to surface information. You can simply point it at your knowledge base and let it work its magic.

This sort of works. It does some basic semantic matching and surfaces text and articles based on loose fuzzy matching of what the user inputs into the chat window.

Fast forward to 2025.

Generative AI has pushed out every other topic. It's become the answer to everything, even problems you didn't even know you had. It's borderline magical with how well these large language models are able to answer and respond to every single question you have for it.

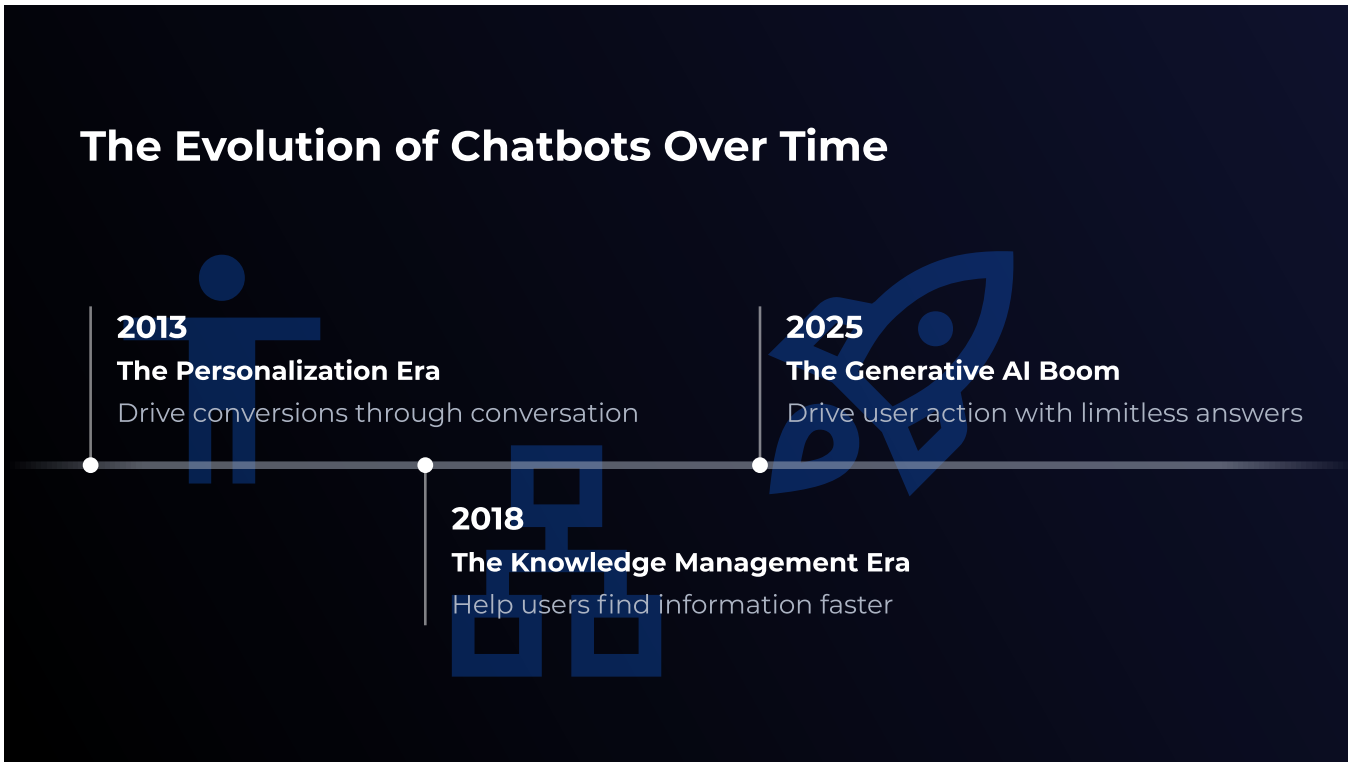
Then, once again, the thought appears in your mind.

"You know what we need? A chatbot."

Every time, you're trying to nudge people towards an action.

Every time, you're trying to communicate information that leads to an action.

Every time, you take the path of least resistance to solve that problem.



What We Want: Fast, Trustworthy Access to Knowledge

Don't get me wrong. I love LLMs. I genuinely think they've forever altered the landscape of the digital world. That said, I (clearly) have a bit of a bone to pick with chatbots in particular.

My biggest issue with chatbots is that they're exceptionally easy to build. In fact, with tools like LangChain, your engineers can bang out a proof of concept in a weekend and have something to demonstrate to executive leadership by that Friday. They're an easy answer to nuanced problems and often are rarely the best answer.

Clinicians, researchers, and members need clarity, speed, and trust when accessing scientific knowledge. They aren't looking to be entertained or engaged in conversation for its own sake—they want accurate answers they can act on.

At first glance, conversational UX seems intuitive. After all, humans communicate through language, so why not interact with machines the same way? But scientific audiences have different expectations than consumers. They value speed, traceability, and precision over friendliness or personality.

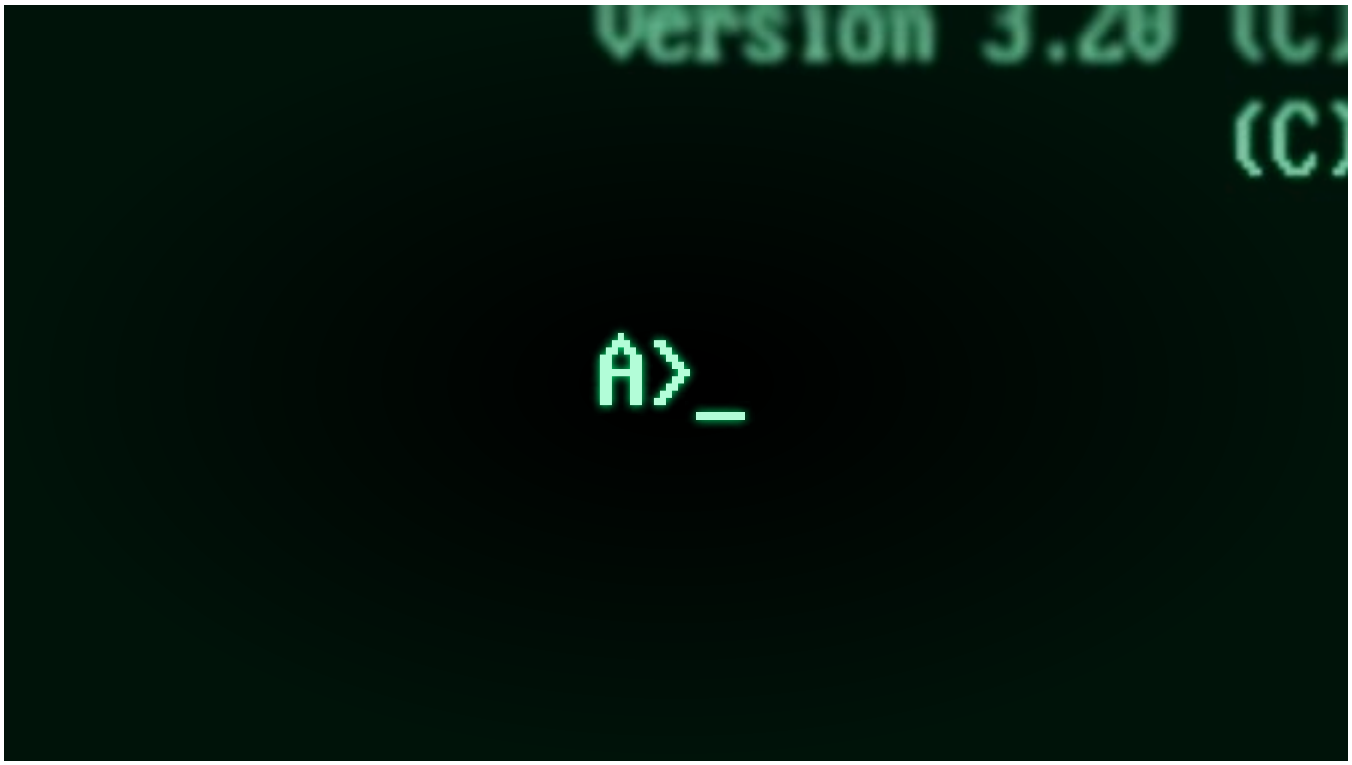
> ***They aren't looking to be entertained or engaged in conversation for its own sake—they want accurate answers they can act on.***

As your use case and problem space become more defined, so too should your tooling. General-purpose conversational agents like ChatGPT shine in open-ended, wide-ranging scenarios. But when the goal is to support clinical reasoning or guideline adherence, the bar is much higher. You need targeted systems that surface authoritative information with minimal ambiguity.

That's why the end goal shouldn't be to simply create a chatbot. The goal should be frictionless access to validated knowledge with the right answer, from the right source, at the right time.

Why Chatbots Fall Short

The Blank-Screen Problem



Back in the 1980s, personal computers were just starting to enter homes and offices. At the time, the command-line terminal was the main way to interact with them. Typing commands into a blank screen gives you a sense of total control. It felt fast, direct, and even elegant in a stripped-down kind of way.

But that came at a cost.

If you didn't know the command, you were stuck. There were no menus, no hints, no visual cues to guide you. You either spoke the machine's language or you didn't get to play. And in a strange way, chatbot interfaces today fall into that same trap.

They assume the user knows what to ask and how to ask it, but offer nothing in the way of visual structure or feedback. You're just dropped into a blinking box and left to guess.

That's not intuitive. That's not discoverable. That's not even user-friendly. It's just... blank.

Imagine landing in a foreign city with no street signs, no map, and no landmarks. You're told you can ask a local for directions, but only one question at a time and only in the order you think of them. That's the chatbot experience.

It's not a guided journey, it's a guessing game with a polite assistant who can't give you a broader view of where you are or where you could go.

And even when you manage to ask the right question, good luck getting the same answer twice. The same query might yield different results depending on some internal randomness or the model's latest "mood." That kind of inconsistency chips away at trust fast. You stop believing the system is authoritative, and start treating it like a coin toss.

Engineering Trade-offs

In your mind, I want you to separate a chatbot into two parts: the conversational UX and the underlying tech that powers it.

From an engineering standpoint, chatbots are a dream. Most frameworks let you spin one up in a matter of hours. Tools like LangChain or Rasa offer out-of-the-box pipelines, and with a pretrained LLM behind the scenes, you don't even need to design a knowledge base. Just point it to some docs, embed them, wire up a UI, and you have a functional chatbot. That level of accessibility makes it incredibly appealing to teams under pressure to deliver something "AI-powered" quickly.

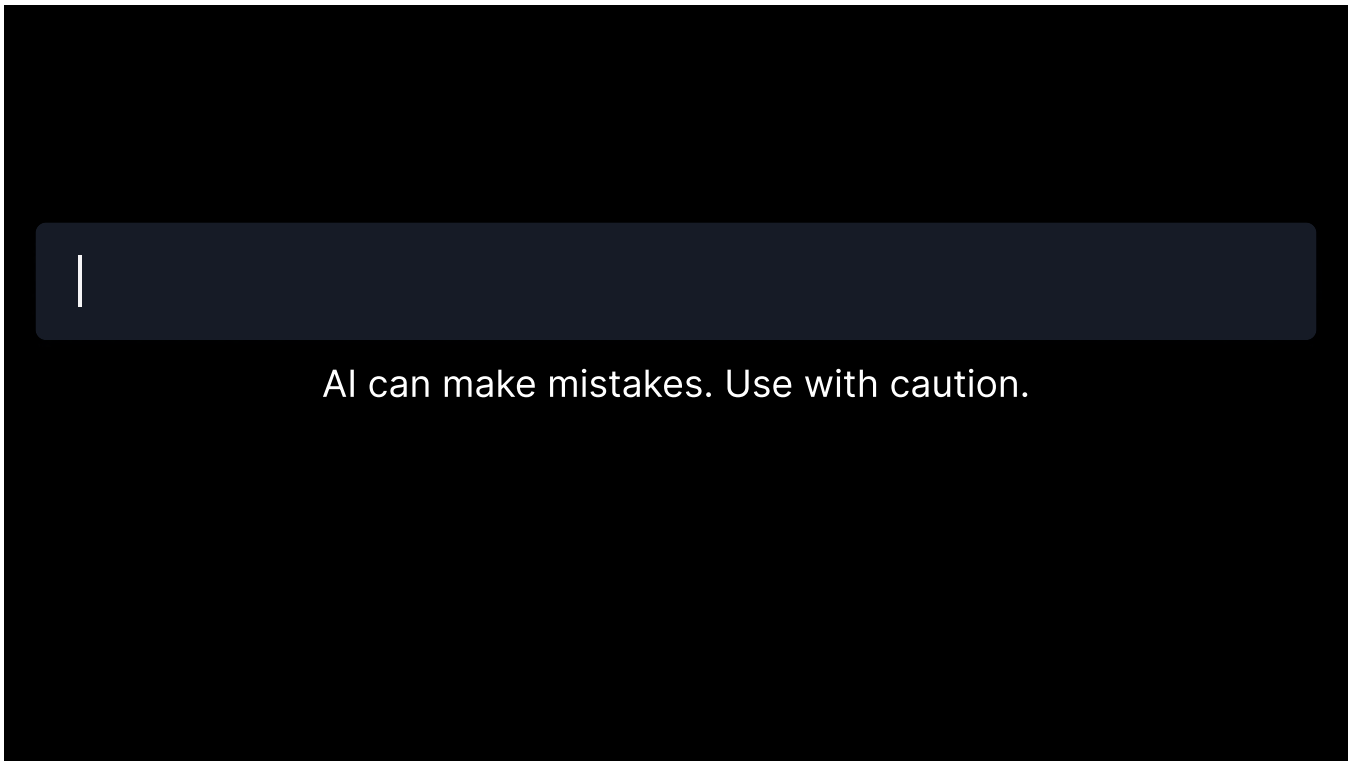
But that speed comes at the cost of depth. Most chatbot implementations rely on a stateless or pseudo-stateful conversation model. Once the thread spans beyond a few exchanges, things break down. The system forgets user intent, misinterprets prior inputs, or starts responding out of context. Engineers know this, but the simplicity of building it outweighs the complexity of fixing it. So products ship with brittle multi-turn logic and pretend it's solved.

Then there's the issue of hallucination. Because the model's job is to always return a response, the chatbot isn't designed to fail gracefully. It generates plausible-sounding output based on probability, not truth. That's fine for playful use cases, but in any environment dealing with factual precision (like healthcare), this is a critical flaw.

This leads to the bigger issue: the chatbot interaction model gets chosen not because it's the most effective interface, but because it's the easiest to implement. It gives the illusion of utility. In a demo, it looks like it understands. But in production, it

struggles with basic expectations like consistency, clarity, and trust. The entire experience is built on a convenience-first architecture that favors rapid deployment over thoughtful design.

The Safety and Security Tradeoff



Open-ended chat interfaces are inherently risky because they create a massive attack surface. There's very little friction between user input and model behavior, and that leaves the system wide open for abuse. In trying to accommodate anything a user might ask, they also end up accommodating things they shouldn't.

To mitigate risk, engineers often introduce content filtering, output restrictions, and instruction reinforcement layers designed to steer chatbot responses within acceptable parameters.

However, these types of guardrails create a fundamental tradeoff: the more aggressively you constrain the model to prevent inappropriate or unsafe outputs, the more you degrade its overall flexibility and user value.

Even with a multilayered safety approach, these constraints are rarely comprehensive or resilient under real-world load.

From a technical perspective, the bulk of development effort tends to shift from core feature design to reliability engineering. Developers must continuously patch edge cases, enforce token-by-token constraints, and simulate hostile or adversarial use in order to preserve some level of operational confidence. Despite this, testing and validating safety behavior in open-domain generative systems is inherently difficult. There is no standardized framework for simulating every variant of user intent or injection pattern across infinite permutations of context.

Even large-scale research teams with deep LLM expertise struggle to fully secure their models. If OpenAI, with thousands of engineers and safety researchers, hasn't yet produced a jailbreak-proof chatbot, then it's unrealistic to believe that a smaller implementation team will somehow outpace them.

The practical implication is clear:

- Chatbots introduce ongoing compliance exposure by design
- Any deployment in regulated domains should assume failure is not just possible but probable

All of this adds up to something that feels fundamentally misaligned with the expectations of clinical, scientific, or regulated environments. These fields demand accuracy, reproducibility, and auditability.

Chatbots offer none of that out of the box.

A Better Path: Semantic Search with Structured Retrieval

I've been pretty hard on chatbots, but that doesn't mean I'm down on LLMs as a whole. Quite the opposite. I think large language models are among the most consequential tools we have for information retrieval. The issue is not the model, it's the assumptions we make about how people should interact with it. Chat-first thinking limits what's possible. Interface-first thinking unlocks what these models are truly capable of.

Where LLMs shine most in a clinical or scientific context is in their ability to facilitate semantic search with structured output. We're not talking about loose keyword

matching or fuzzy heuristics. We're talking about dense vector embeddings that let the model recognize conceptual meaning across vocabulary mismatches, regional medical phrasing, or user ambiguity.

Instead of matching "heart failure therapy" verbatim, the system can resolve that to guideline-directed treatment for reduced ejection fraction, even if that exact term was never mentioned.

But the real leap comes when we constrain that output within tightly structured formats. Take medication titration protocols for example. In an open chatbot, a user might ask, "How do I increase beta-blockers in HFrEF?" and the model could easily hallucinate or oversimplify.

But in a structured retrieval interface, the same question:

- Routes through a semantic embedding space
- Maps to a validated titration chart, and
- Renders that table interactively with dosage ranges, contraindications, and references—all grounded in vetted source documents

This reduces error risk and supports clinical compliance.

Another example: risk scoring tools. In a chatbot, asking about stroke risk in atrial fibrillation could yield a narrative summary that sounds confident but lacks precise logic. But in a structured system, we can embed calculators like CHA_2DS_2-VASc directly.

The model can extract relevant inputs from user queries, retrieve the tool, and walk the user through a validated flow step-by-step, without veering off course. These workflows aren't just safer, they're auditable and reproducible.

Even something as nuanced as co-morbidity lookup across guidelines becomes more powerful. A semantic layer can normalize disparate terminology (say, distinguishing between 'renal impairment' and 'CKD stage 3') and output results using expandable cards categorized by guideline domain, each linked to its issuing authority.

The shift here is architectural. We aren't just using LLMs to generate responses. We're using them to drive retrieval, validation, and contextual assembly of structured ar-

tifacts, tables, calculators, and decision trees all mapped directly to clinician tasks. These artifacts are easier to test, easier to regulate, and less prone to dangerous drift.

So while chatbots offer immediacy, they also come with a broad surface area of risk. Structured systems narrow that surface area. They let engineers isolate failure modes, guide user behavior, and reuse existing model infrastructure within a compliant, purpose-built interface.

Let's Get This Right

Ultimately, this comes down to intentionality. If the goal is simply to check the box and deploy something "AI-powered," then sure: plug in a chatbot and call it a day. But if the goal is to build an interface that meaningfully reduces friction, preserves clinical accuracy, and scales with trust, then the solution requires more rigor.

You have to decide: are you solving for optics or outcomes?

Interested in starting a conversation?

Discuss AI chatbots or your next big idea with one of our experts.

Get In Touch