Episode Book

EPISODE #5743

Why Claude Keeps Saying "Prior Art

PUBLISHED	6 October 2026 at 12:06 UTC
DURATION	25:05
LISTEN	myweirdprompts.com/episode/claude-prior-art-phrase/
HOSTS	Corn, Herman
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:39 UTC

SYNOPSIS

Prior art" is a patent-law term of art, but it keeps showing up in Claude's output — and Daniel wants to know if it even belongs there. This episode digs into the research on lexical overrepresentation: why preference-tuned models reach for a "language of prestige," and how raters rewarding certain words teaches tics no one intended. Then it gets practical: if you're asking an agent to survey existing solutions before building, how do you phrase that instruction so you get a real foundation to build on instead of noise? The answer isn't a vocabulary ban — it's knowing what you're actually asking for.

TRANSCRIPT



Corn

A voice note, a reused prompt, and one clause Daniel keeps pasting in: check whether anyone has already built this. That's the setup. And the phrase that keeps coming back in the reports is the one he's asking about.



Herman

"Prior art."



Corn

"A survey of prior art." He's noticed it slipping into recent Claude models, and he isn't sure the usage is even correct. Here's the thing he flags himself. The pattern with AI overused phrases isn't that they're wrong. It's that they over amplify terms humans use sparingly. So Daniel's assuming it belongs in that category, and he's asking us to take the instinct seriously.



Herman

You know what's funny about the prompt making that assumption? He states it as a hypothesis, like he's embarrassed to be right.



Corn

He's right. But let me get through the workflow first, because that's the part that actually matters. Daniel records a long voice note. What he wants to build, which framework, all the details. Then he runs it through a prompt he reuses, which turns that brain dump into organized documents. Technical specs, that kind of thing. Increasingly he's routing this through sub-agent delegation.



Herman

Right, he's splitting the work.



Corn

So one of the things he found valuable to include in those voice notes is asking the agent whether anyone has already created this, or whether there are components he could build on. He gives a concrete case. Say he's instructing an agent to build a backup runner for a podcast, based on an RSS feed. He'd rather the agent come back and say "I found something you didn't" than build something that already exists in a better, more maintained form. That's the whole point of the clause.



Herman

Don't reinvent the wheel.



Corn

But here's what he says is more often the more important utility. The agent says nobody has built exactly this, but here's a basis we could build on. Not a match. A foundation.



Herman

That's a different search.



Corn

It is a different search, and I think he knows it, which is why he's asking. So here's what he wants from us. How do you phrase this part of the stack development instruction for success? Would you write a standalone prompt for surveying existing technology or components? Or would you add a couple of paragraphs to a prompt that initiates a project? Give him your best draft. And then, what are the non obvious things you'd be careful to include or exclude?



Herman

That's five questions.



Corn

It's four or five. Nobody's counting.



Herman

I counted.



Corn

So let's start with the term itself, because Daniel's instinct that it might not be used correctly is worth taking seriously.



Herman

Start with the legal definition, because everything else hangs off it. Prior art is a term of art from patent law. It's all publicly available information predating an invention's filing date. Earlier patents, published papers, products, public demonstrations, anything a person skilled in the field could have known. Its function is narrow. It's the body of evidence used to test whether an invention is novel and non obvious enough to deserve a patent. If prior art discloses your invention, you can't patent it.



Corn

That's the whole job. It exists to kill claims.



Herman

And nobody in software engineering calls a survey of existing libraries a survey of prior art. The correct terms are landscape survey, technical scan, state of the art, or just plain existing solutions.



Corn

Worth flagging one caveat. That definition is from general knowledge, not a freshly pulled statute. It's standard and I'm confident in it, but if you're going to quote the USPTO directly, verify first.



Herman

Fair. The definition's fine. It's textbook.



Corn

So the episode has two threads. One, the lexical tic. Why does a model reach for prior art, and why now? Two, the practical instruction. How do you ask an agent to survey existing solutions without getting noise back?



Herman

Term, then mechanism, then application.



Corn

Then the reframe at the end, which is where I think the real answer lives. But we'll get there. The legal definition is clear enough. The interesting question is why a model would reach for it in the first place.



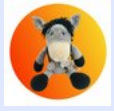
Herman

And that turns out to be a well studied problem. Not just anecdotally. There's a research program on it.



Corn

Go.



Herman

There's a paper by Tom Juzek and Zina Ward. "Why Does ChatGPT Delve So Much?" It came out of COLING twenty twenty-five. What they did is build a formal method to detect words whose frequency in scientific abstracts has spiked. They found twenty-one what they call focal words. Delve is the famous one. Intricate, underscore, a bunch of others.



Corn

And the important part isn't the list.



Herman

The important part is what they couldn't find. They failed to find evidence that the overuse comes from model architecture, algorithm choices, or training data. They pose it as the puzzle of lexical overrepresentation. The words are showing up more often than any of the usual suspects would explain.



Corn

So it's not in the weights as a data artifact.



Herman

Not in the way you'd expect. And the follow up is the part that solves it. Same two authors. "Word Overuse and Alignment in LLMs: The Influence of Learning from Human Feedback." This one's from BIAS twenty twenty-five at ECML PKDD. They experimentally emulated the human feedback procedure.



Corn

Meaning they sat raters down and made them pick.



Herman

They built the same kind of comparison the preference data comes from. And human raters systematically preferred text variants containing certain words. That's the finding. Systematically.



Corn

So the model isn't inventing a tic. It's learning one.



Herman

It's being taught one, at the stage where it's being taught to be helpful. The key insight is that this is a form of misalignment. And it arises from a divergence between the lexical expectations of two populations. The human feedback workers who trained the model, and the actual users of the model. The raters reward a word. The model over learns it. The users find it grating.



Corn

Two different rooms, two different ideas of what good writing sounds like.



Herman

And the raters' room won, because they're the ones holding the clicker.



Corn

There's a third paper, right?



Herman

There's a third. Ming, Hernandez and Juzek. "Isolating LLM Lexical Bias," FLAIRS twenty twenty-six. They introduce something called a Triangulated Preference Shift score. What it does is isolate lexical shifts caused specifically by preference tuning, and they ran it across six model families. The framing they land on is that preference learning pushes models toward a language of prestige.



Corn

A language of prestige.



Herman

That's their phrase.



Corn

That's a very good phrase. And it's a very good description of what prior art is doing in a Claude response.



Herman

It sounds like it came from a room with a mahogany table in it.



Corn

So let's apply the framework. Why is prior art such a strong candidate for this pattern?



Herman

Four reasons. First, it's a prestige term. It borrows the authority of law and R and D. Second, it's a compression. One phrase that means the body of existing work relevant to this problem. That's exactly the kind of high information density token that gets rewarded in preference data. Prefer the version that sounds like it did more work.



Corn

Third.



Herman

It's slightly unusual. Rare enough in ordinary prose that it reads as a deliberate, sophisticated choice. Which is precisely what makes it feel like a tic when it recurs. If it were a common word you wouldn't notice.



Corn

Fourth.



Herman

It maps cleanly onto a task the model is frequently asked to do. Research before building. So it gets reinforced in exactly the context where it appears. The reward signal and the use case are lined up.



Corn

And here's the thing I want Daniel to hear directly, because he framed it as a question about whether Claude is using the term correctly. The answer is that Claude is using it metaphorically and loosely, and it's not technically wrong. The mapping holds. Existing solutions that already do this maps onto prior art. The underlying logic is the same. Don't claim novelty for something that already exists. Build on what's there.



Herman

It's just register mismatched.



Corn

It's a legal term of art deployed in a domain where it doesn't belong. Which is a different failure than being incorrect.



Herman

It's not an error. It's a borrowed coat that doesn't fit.



Corn

And Daniel's framing of the whole thing is well supported. He said the pattern isn't that the words are wrong, it's that they over amplify terms humans use sparingly. That's exactly what Juzek and Ward found. It's a frequency misalignment, not a correctness problem.



Herman

He got there on instinct from noticing the word in his own reports.



Corn

One supporting detail that shows how naturally the phrase lives in patent discourse. There was a discussion on Hacker News about an AI generated invention system, unpatentable dot org, back around December of last year. And the developer notes in it that prior art doesn't have to be flawless to count.



Herman

Examiners and courts rely on the correct portions even of error containing disclosures.



Corn

Right. So a flawed reference still counts as prior art. If it discloses the thing, it discloses the thing, even if the document around it is wrong. That's a real, correct, technical use of the term. And notice how specific it is. It's answering a question only a patent person would ask.



Herman

It's load bearing in that sentence. In Claude's output it's decorative.



Corn

Which is a nice way to tell the two apart. If you could swap the phrase for existing solutions and lose nothing, it was decoration.



Herman

So the mechanism is preference tuning pushing models toward a language of prestige. The practical question is what you do about it when you're the one writing the prompt.



Corn

And it's worth saying, the practical answer isn't a vocabulary ban. It's not that you should never see the word. It's that you have to know what you're actually asking for.



Herman

The strongest argument for the instruction is one Daniel already made better than I could. There's a blog post from September, Liam Powell, "Bend 2 and the Vibe Coding Trap." His thesis is that vibe coding makes it possible to build a substantial solution before learning enough about the problem to recognize that a much better solution exists.



Corn

That's the trap. You can get all the way to a working thing.



Herman

You can produce an entire language and compiler while missing an approach that an introductory survey of the field would have put directly in front of you. That's his sentence, roughly.



Corn

What's the case study?



Herman

The Bend language built an elaborate system for formal verification. And apparently never encountered SPARK. Which is an existing open source formal verification language and compiler. Powell recreated Bend's demo in SPARK with a fraction of the code and got, and I'm quoting the output here, "Success: all checks proved."



Corn

A fraction of the code.



Herman

A fraction. And here's his conclusion, which is the part I'd put on a wall. "If you ask a LLM for a language where it's possible to prove that a function is formally correct by building up a proof from basic principles then it will happily do so, it will never stop to suggest to you that computers can already build complex proofs. It will never tell you that what you're



Herman (continued)

building already mostly exists as work that you can build on."



Corn

That's the entire episode in one paragraph.



Herman

That's why Daniel's clause exists.



Corn

Now the counter position, because there was one in that same thread and it's better than the usual cynicism.



Herman

There was. A commenter makes the incentive case. Providers have little reason to encourage research first behavior. Research is slow. Web searches are slow. Models are rate limited or blocked from pages. And sometimes the research contradicts itself, which annoys users. The commenter's line is that users like faster gratification cycles from the agent slot machine handle.



Corn

The agent slot machine handle.



Herman

And then the psychological driver, which is the part I hadn't thought about. Writing a bunch of bespoke code instead of leveraging prior art makes a lot of users feel like they own something novel. Big. Important.



Corn

That's true. That's uncomfortably true. There is a feeling when you've got an agent churning out files for you.



Herman

He explicitly declines to call it a conspiracy. He attributes it to providers optimizing real but sometimes misleading success metrics. Which is a fairer framing than the conspiracy version.



Corn

It's more damning, actually. Nobody has to be scheming. The metric just has to be the wrong one.



Herman

Which loops right back to the preference tuning. Same shape of problem at a different level.



Corn

So when does Daniel's clause earn its keep, and when is it noise?



Herman

It earns its keep under two conditions, and they both need to hold. First, the problem space is mature enough that solutions plausibly exist. Second, the cost of building the wrong thing is high. If both are true, do the survey.



Corn

And when is it noise?



Herman

When the task is novel. When it's trivially small. Or when the survey comes back with vaguely related libraries and the agent then tries to shoehorn them in anyway. That last one is the worst outcome, because now you've got a dependency you didn't need and you paid tokens for the privilege.



Corn

Which connects to what Daniel said about the more important utility. He said the more valuable case is usually not a whole product match. It's that nobody has built exactly this, but here's a basis to build on. That argues for a survey that returns components and primitives, not just whole product matches.



Herman

The foundation is the deliverable.



Corn

So let's write the thing.



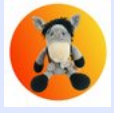
Herman

Design principles first. And then the draft, and I want to read the draft out rather than describe it, because the whole point is the phrasing.



Corn

Start with the first principle.



Herman

Name the deliverable, not the vibe. Don't say survey the prior art. Say what you want back. A short list of candidate components, each with a one line assessment of fit, maturity, license, and maintenance status.



Corn

Prior art invites the model to perform erudition. A concrete artifact spec invites it to do work.



Herman

That's exactly the distinction. Second principle, separate the two questions. Something already does this, and nothing does this but here's a foundation. Make the agent answer both, explicitly, in that order, so it can't collapse them into one vague paragraph.



Corn

Third.



Herman

Force a negative result to be stated. Require the agent to say no close match found, explicitly, rather than padding with weak matches. That's the single biggest source of noise in my experience.



Corn

Fourth.



Herman

Constrain the search surface. Say where to look. Package registries, GitHub, standards bodies, sibling ecosystems. And give a budget. Spend no more than N searches. Return at most five candidates. Unbounded research is where agents burn tokens and time and come back with nothing.



Corn

Fifth.



Herman

Require evidence, not assertion. Every candidate comes with a URL, a last commit or last release date, and a license. This is the anti hallucination measure. The unpatentable dot org developer I mentioned describes structured, constrained, multi agent prompting with a supervisor and refinement loop as noticeably reducing fabrication.



Corn

Sixth.



Herman

Explicitly forbid the failure mode. Tell the agent not to propose building on something just because it's topically adjacent. And not to recommend a dependency that's unmaintained or license incompatible.



Corn

Seventh. Delegate it.



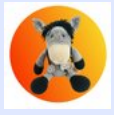
Herman

Daniel's already there. A dedicated research sub agent with a narrow brief and a clean context window will do a better survey than the main agent holding the whole project spec plus a research task at once. Those are competing jobs.



Corn

All right. Read the standalone version.



Herman

Here's the draft. "Task. Existing solutions survey. Before any design or implementation work, survey what already exists for the following problem." Then a one paragraph problem statement goes in. "Answer these two questions separately and in order. One. Is there an existing project that already does this? If yes, name it, link it, and state in one sentence why it does or doesn't fit. If no, say so plainly. Do not pad the list with weak matches. Two.



Herman (continued)

Is there no exact match, but are there components, libraries, or prior designs we should build on? List at most five. For each, give name, URL, what it provides, license, last release or commit date, and a one line fit assessment."



Corn

Keep going. The constraints are the good part.



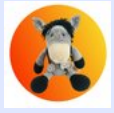
Herman

"Constraints. Search package registries, GitHub, and relevant standards and specs. Do not rely on recall alone. Cite a URL for every claim. Do not recommend anything unmaintained, no release in eighteen months or more, or license incompatible with whatever the project's license is. Do not recommend a dependency merely because it's topically related. If the honest answer is build it yourself, say that. Budget, at most however many searches. Return a short report, not an essay. Output format, a table of candidates, then a three sentence recommendation."



Corn

That's a real prompt.



Herman

That's the standalone version. For embedding inside a larger project initiation prompt, compress it to two paragraphs. One states the two questions. One states the constraints. Cite URLs, cap the list, state negative results explicitly, flag license and maintenance.



Corn

And keep the state a negative result plainly clause, whatever else you cut.



Herman

That's the highest leverage line in the whole thing. If you cut everything else, keep that.



Corn

Now the non obvious stuff. Includes first.



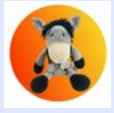
Herman

A cap on the number of candidates. Without it you get twelve mediocre matches. A required negative result statement. Maintenance and license metadata as mandatory fields, which converts a vibes based recommendation into a decision you can actually make. A search budget, which directly addresses the research is slow incentive problem we just described. The components not just whole products framing. And an explicit build it yourself escape hatch, otherwise the agent feels obligated to find something whether or not something exists.



Corn

And the excludes. Start with the one I've been waiting for.



Herman

Exclude the phrase prior art itself.



Corn

Say it again for the back.



Herman

Don't write survey the prior art in your prompt. It's a prestige term that invites performative erudition, and per Juzek and Ward it is exactly the kind of word preference tuning has over rewarded. Say existing solutions. Say prior work. Say landscape survey instead.



Corn

The fix for the tic is to stop using the tic in your own prompt.



Herman

That's the joke, and it's also the advice. It's not just comedy.



Corn

What else goes?



Herman

Open ended research thoroughly instructions. Unbounded research is where tokens and time go to die. Asking for a report without specifying a format. Specify the table. Letting the survey agent also do the design, because a clean context window produces a cleaner survey. And implicit permission to recommend adjacent but irrelevant tools. Name that failure pattern out loud or it will happen.



Corn

To Daniel's point about whether it should be standalone or appended. I'd say the survey is a distinct job with a distinct deliverable, so give it a distinct brief. If it lives inside the big initiation prompt at all, it should be a self contained block with its own output format, not a clause buried in the middle of the build instructions.



Herman

Yes. If it's a sentence in a paragraph, it becomes a box to tick.



Corn

And the deliverable spec is what stops it being a box to tick.



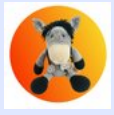
Herman

The number should be lower.



Corn

Sorry, which number?



Herman

Five. You said return at most five candidates. Two or three.



Corn

You've done this before.



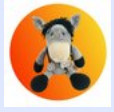
Herman

A long stretch of it. Patent searches, not as a lawyer. I was the person who actually sat in the archive and pulled the references.



Corn

So five is generous.



Herman

Anything past three is padding, and the padding is where the agent starts reaching for adjacent but irrelevant tools. I'd rather see one strong candidate and an honest nothing else close than five with three of them weak.



Corn

What does a real one look like? The artifact.



Herman

It's a claim chart. Every element of the claim mapped to a specific reference, with a citation to the exact page or column. Physical form, a binder, tabbed, with the claim language pasted down the left margin and the reference excerpts pasted opposite. The tabbing matters. If the tabs are wrong you're re-reading the same reference three times looking for the element you already found.



Corn

A binder with tabbing.



Herman

You wanted to know what it cost. That's what it looked like.



Corn

And here's what I think is actually the interesting part. You said prior art in its real sense is a negative exercise.



Herman

It is. You're looking for something that kills the claim.



Corn

And a survey of existing solutions is a positive exercise.



Herman

You're looking for something to build on. Opposite searches. Claude has borrowed the vocabulary of the first to describe the second. That's why the reports feel slightly off even when they're useful.



Corn

So it's not just register. It's polarity.



Herman

The word points the other way.



Corn

And that's why Daniel's instinct that something was wrong survived contact with the fact that the reports were fine. The reports were useful. The word was still wrong.



Herman

The binder was never about finding a foundation. It was about killing a claim.



Corn

Daniel said the more important utility is the foundation. Which means the thing he actually wants is the search Hilbert just said the term doesn't describe.



Herman

Right.



Corn

Well. That's the episode, isn't it.



Corn

That distinction between a negative search and a positive one is worth sitting with, because it explains something the register argument doesn't. Register mismatch says the word is wearing the wrong clothes. Polarity says the word is pointing in the wrong direction.



Herman

Which is why you can't fix it by finding a fancier synonym. If you swap prior art for state of the art and keep asking the same question, you've cleaned up the vocabulary and left the confusion in place.



Corn

The one thing. When you write the survey clause, decide first which search you're asking for. Kill a claim, or lay a foundation. Then name the deliverable that search produces. Everything else in the prompt is detail around that decision.



Herman

And the detail that matters most is the negative result. Insist on a plain nothing close, because an agent that can't say nothing will always say something.



Corn

Which is a thought I'll be carrying into the other direction this goes. If preference tuning is pushing models toward a language of prestige, what else is currently colonizing agent output that we haven't noticed yet?



Herman

Because we only notice the ones that have started to grate.



Corn

Delve grates. What's the one that hasn't started grating yet because it still sounds impressive?



Herman

And I don't have a clean answer to the second half of that either. If research first behavior is slow and users reward speed, is the survey instruction something we keep writing ourselves forever, or does it eventually become default?



Corn

I suspect we keep writing it. The slot machine handle is more fun than the archive.



Herman

So the fix for the tic is to stop using the tic in your own prompt. The deeper fix is to know which search you're actually asking for.



Corn

That's the episode. If you've got thoughts on any of this, leave us a review. It helps people find the show.



Herman

Thanks to our producer, Hilbert Flumingtop.



Corn

This has been My Weird Prompts. We'll be back soon.