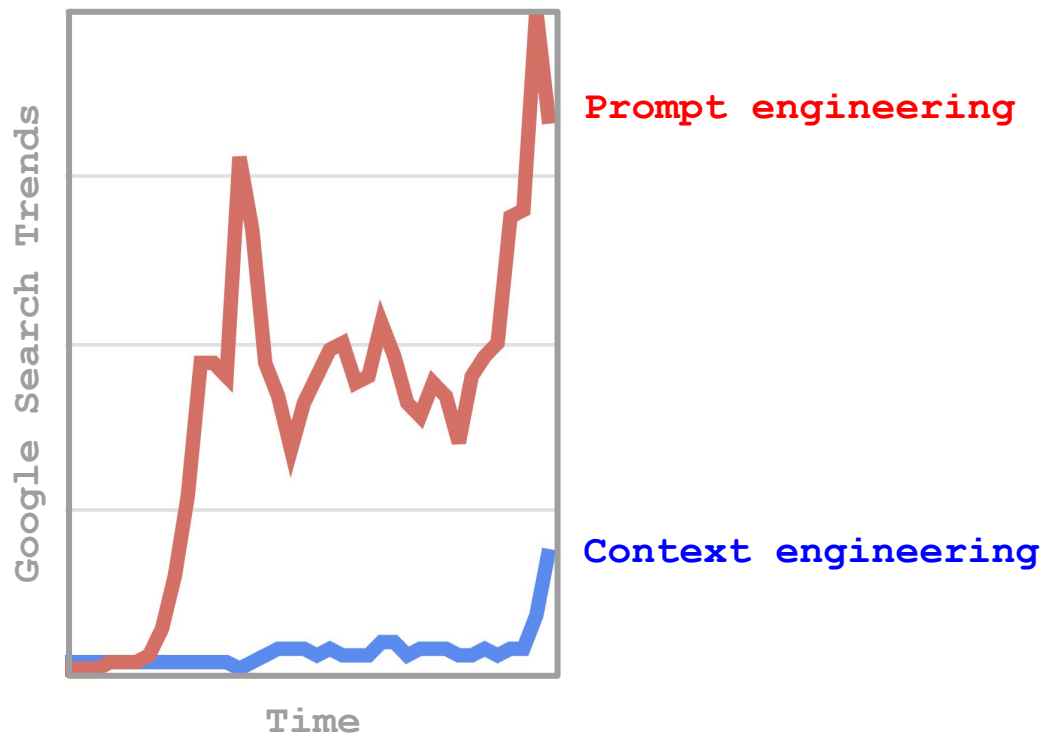


Context Engineering
@rlancemartin

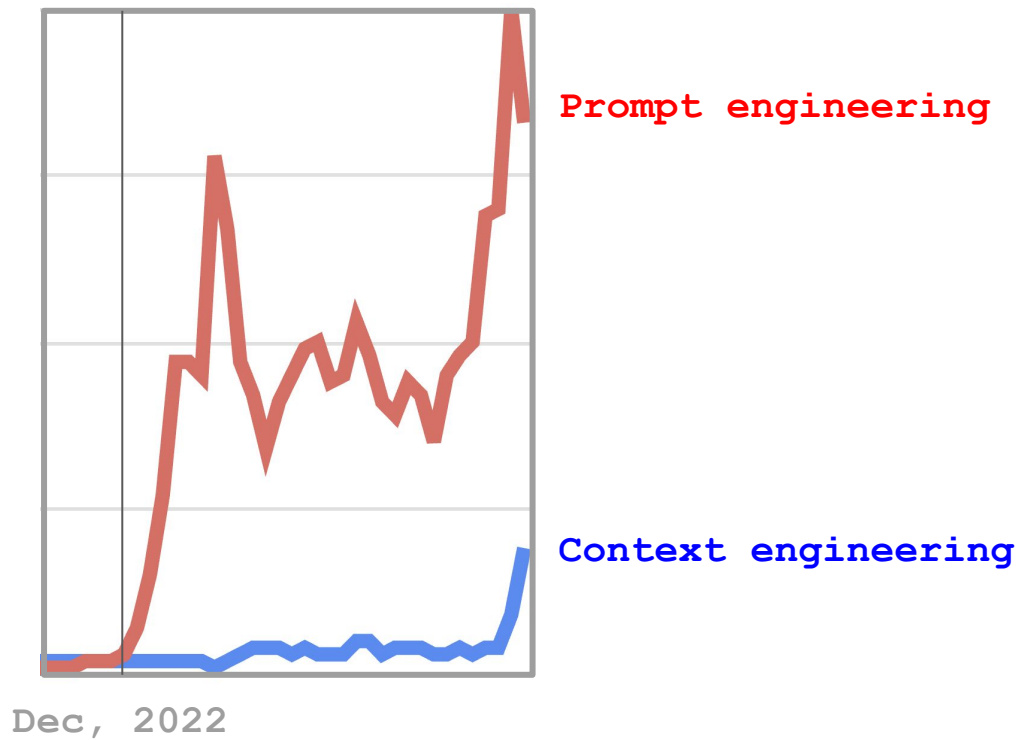


LangChain

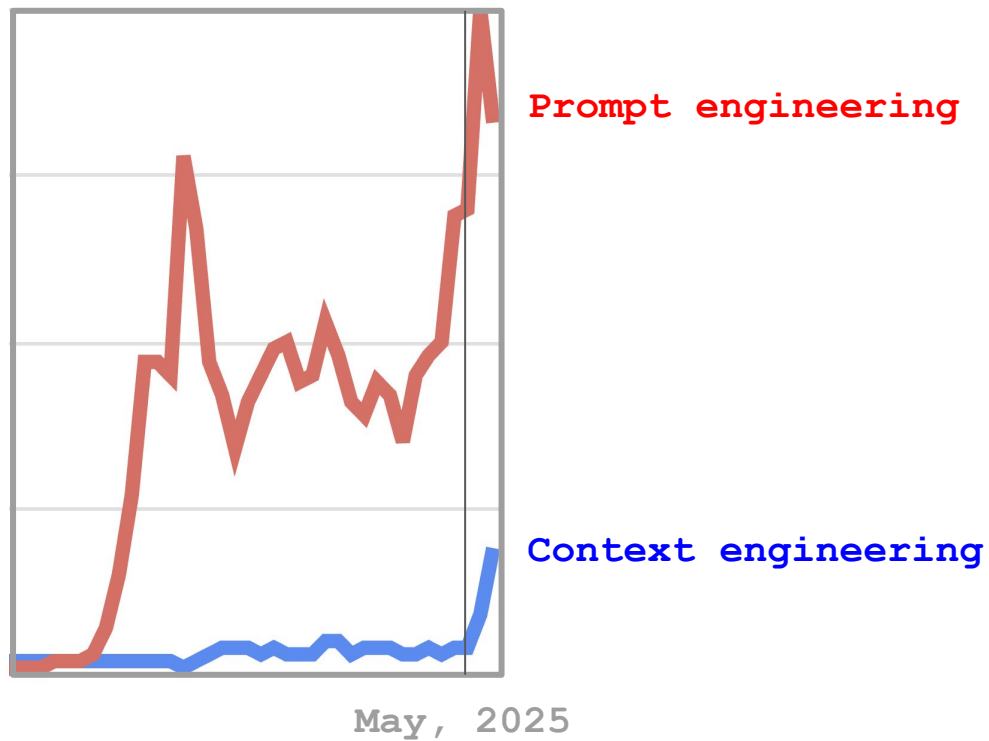
Like Drew said, buzzwords identify common experiences



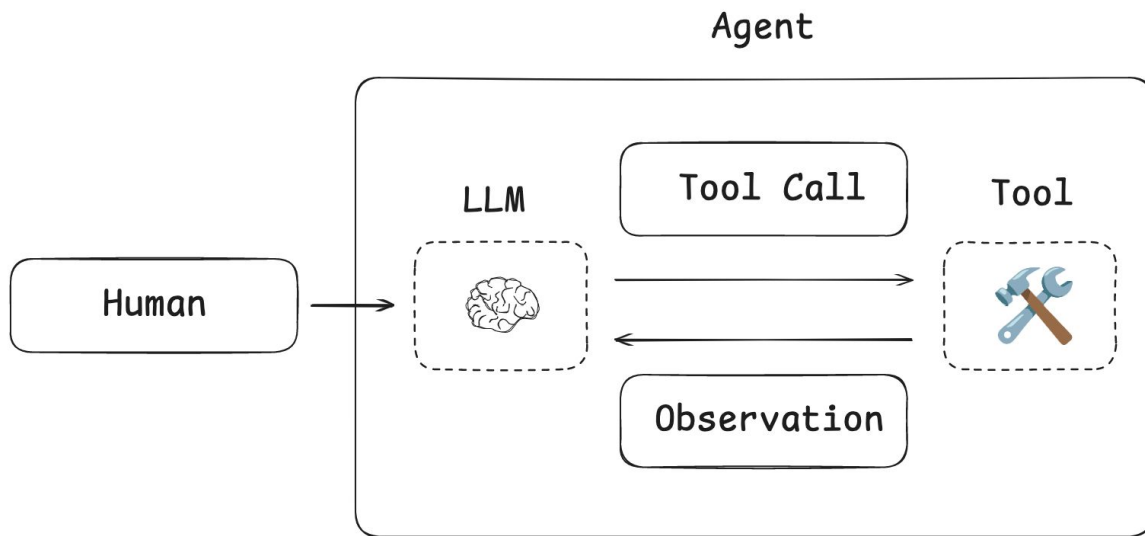
ChatGPT



"Year of agents"



Context grows w/ agents



Manus

Typical task in Manus requires around 50 tool calls

Anthropic

Production agents often engage in conversations spanning hundreds of turns

<https://www.anthropic.com/engineering/built-multi-agent-research-system>
<https://manus.im/blog/Context-Engineering-for-AI-Agents-Lessons-from-Building-Manus>

Performance drops as context grows



chroma

Context Poisoning

Ex: Gemini playing Pokemon hallucinated an item + tried to re-use it

Context Distraction

Ex: Gemini favored repeated actions over new plans as context > 100k tok

Context Confusion

Ex: Models perform worse with more tools, esp if tools are similar

Context Clash

Ex: Models perform worse if back-to-back tool calls contradict each other

"Context engineering" captures the challenges of the moment



Andrej Karpathy ✓



@karpathy



+1 for "context engineering" over "prompt engineering".

People associate prompts with short task descriptions you'd give an LLM in your day-to-day use. When in every industrial-strength LLM app, context engineering is the delicate art and science of filling the context window with just the right information for the next step. Science

00:00 / 01:08

swyx   @swyx · 1d Subscribe ...

i've been looking for "brainrot education" for the gen alphas and someone finally sent me a channel that translates @dexhorthy and @RLanceMartin's content for those who are of deficit in attention

this channel is 8 weeks old and has 22k followers
x.com/latentspacepod...
less

47 169 1.5K 131K

<https://x.com/swyx/status/1946100121038696671>

Many anecdotal experiences, no common philosophy yet . . .

. . . some common themes

Offload context

Use file system for notes (see: [Drew's post](#), [Anthropic multi-agent](#)).

Use file system (e.g., [todo.md](#)) to plan/track progress (see: [Manus](#)).

Use file system read/write tok-heavy context (see: [Manus](#)).

Use files for long-term memories (see: Ambient Agents [course](#)/[repo](#)).

Reduce context

Summarize agent message history (see: [Drew's post](#), Claude Code).

Prune irrelevant parts of message history (see: [Drew's post](#)).

Summarize / prune tool call outputs (see: [open-deep-research](#)).

Summarize / prune at agent-agent handoffs (see: [Cognition](#)).

But, care careful of information loss (see: [Cognition](#) and [Manus](#))!

Retrieve context

Mix of retrieval methods + re-ranking (see: Varun's [take](#) from Windsurf).
Systems to assemble retrievals into prompts (see: [Preempt](#) in Cursor).
Retrieve relevant tools based upon tool descriptions (see: [Drew's post](#)).

Isolate context

Split context across multi-agents (see: [Drew's post](#), [Anthropic](#)).

But, be careful (see: [Cognition/Walden Yan](#))!

Multi-agents make conflicting decisions (see: [Cognition/Walden Yan](#)).

Sub-agents lower risk if avoid decisions (see: [open-deep-research](#)).

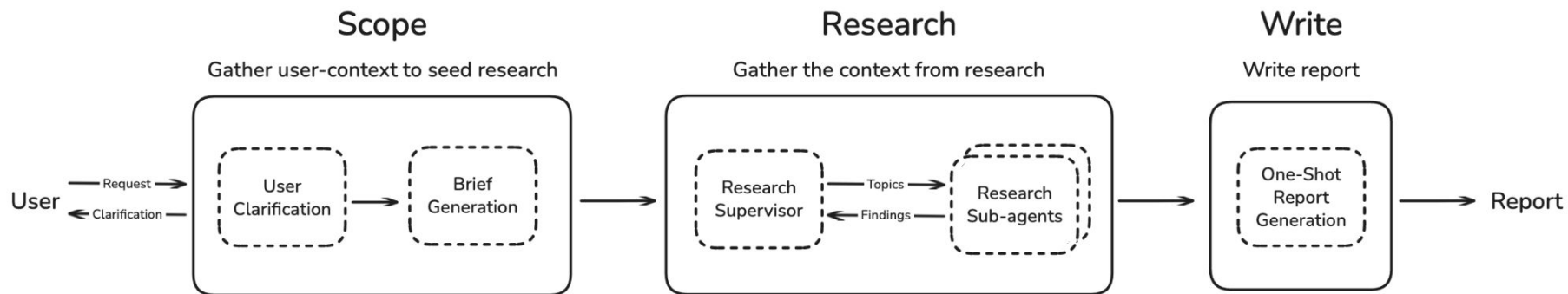
Cache Context

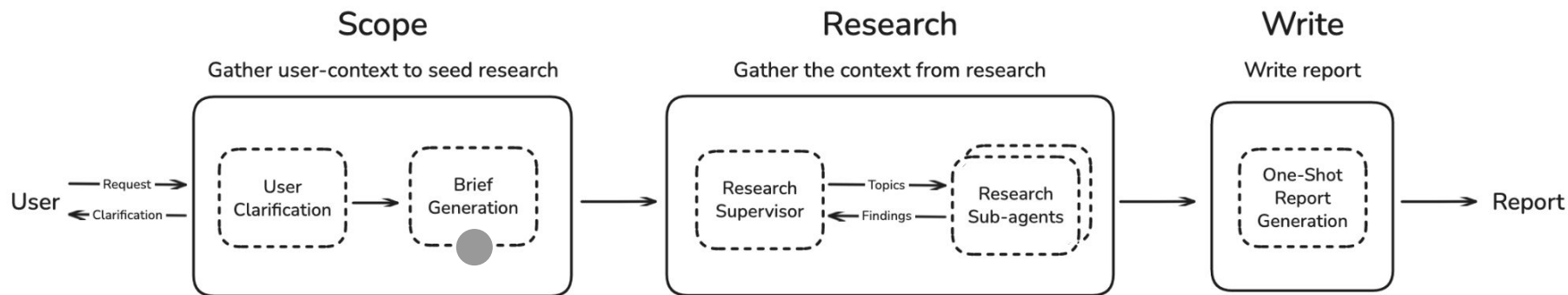
Cached input tokens for Claude-sonnet 10x cheaper!

Cache agent instructions, tool descriptions to prefix.

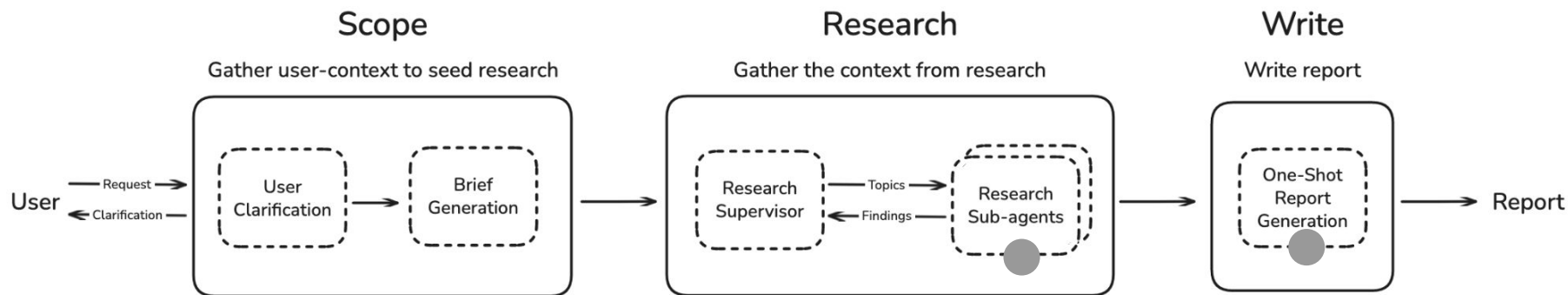
Add mutable context / recent observations to suffix.

Let's see an example

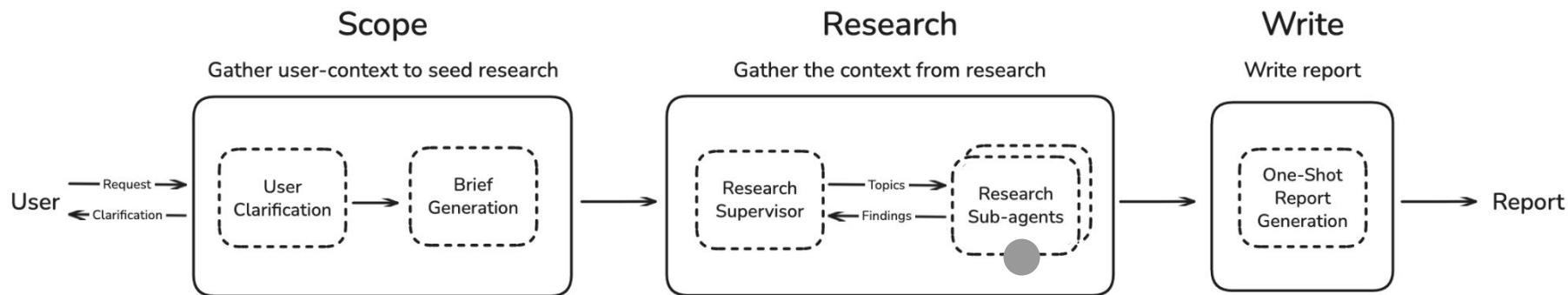




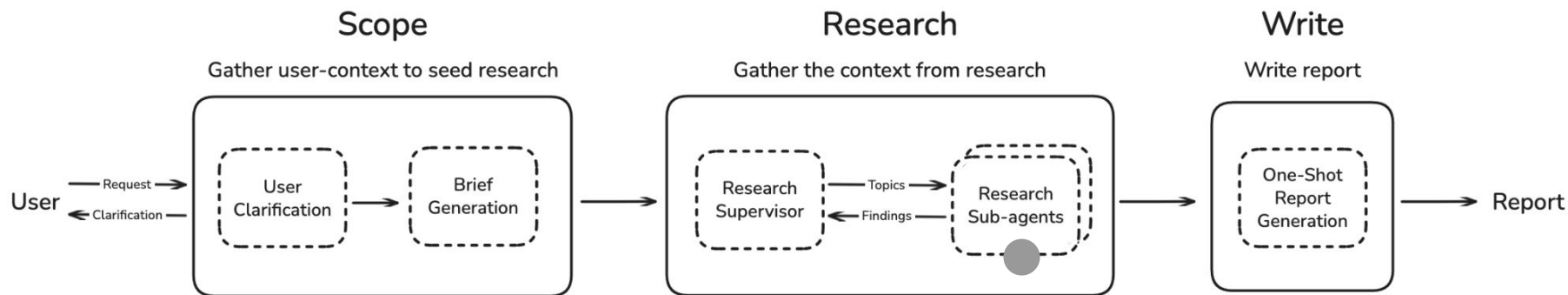
Offload: Create brief from chat and save to state.



Offload: Use brief later to steer writing/research.



Reduce: Summarize the observations from tool calls + agent research.



Isolate: Isolate context across sub-agents.

	Offload	Reduce	Retrieve	Isolate	Cache
Drew's Post	Discussed	Discussed	Discussed	Discussed	Not discussed
Manus	Used	Discouraged [1]	Not discussed	Not discussed	Used
Anthropic-researcher	Used	Used	Used	Used	Not discussed
Cognition	Not discussed	Used	Not discussed	Discouraged [2]	Not discussed
LC open-deep-research	Used	Used	Used	Used	Not discussed

[1] Information loss risk w compression is a problem. Favors offloading instead.

[2] Coordination problems w multi-agent is a risk.



Andrej Karpathy ✓

@karpathy



+1 for "context engineering" over "prompt engineering".

People associate prompts with short task descriptions you'd give an LLM in your day-to-day use. When in every industrial-strength LLM app, context engineering is the delicate art and science of filling the context window with just the right information for the next step. Science because doing this right involves task descriptions and explanations, few shot examples, RAG, related (possibly multimodal) data, tools, state and history, compacting... Too little or of the wrong form and the LLM doesn't have the right context for optimal performance. Too much or too irrelevant and the LLM costs might go up and performance might come down. Doing this well is highly non-trivial. And art because of the guiding intuition around LLM psychology of people spirits.

On top of context engineering itself, an LLM app has to:

- break up problems just right into control flows
- pack the context windows just right
- dispatch calls to LLMs of the right kind and capability
- handle generation-verification UIUX flows
- a lot more - guardrails, security, evals, parallelism, prefetching, ...

So context engineering is just one small piece of an emerging thick layer of non-trivial software that coordinates individual LLM calls (and a lot more) into full LLM apps. The term "ChatGPT wrapper" is tired and really, really wrong.

https://github.com/langchain-ai/how_to_fix_your_context