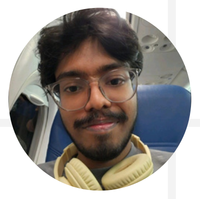
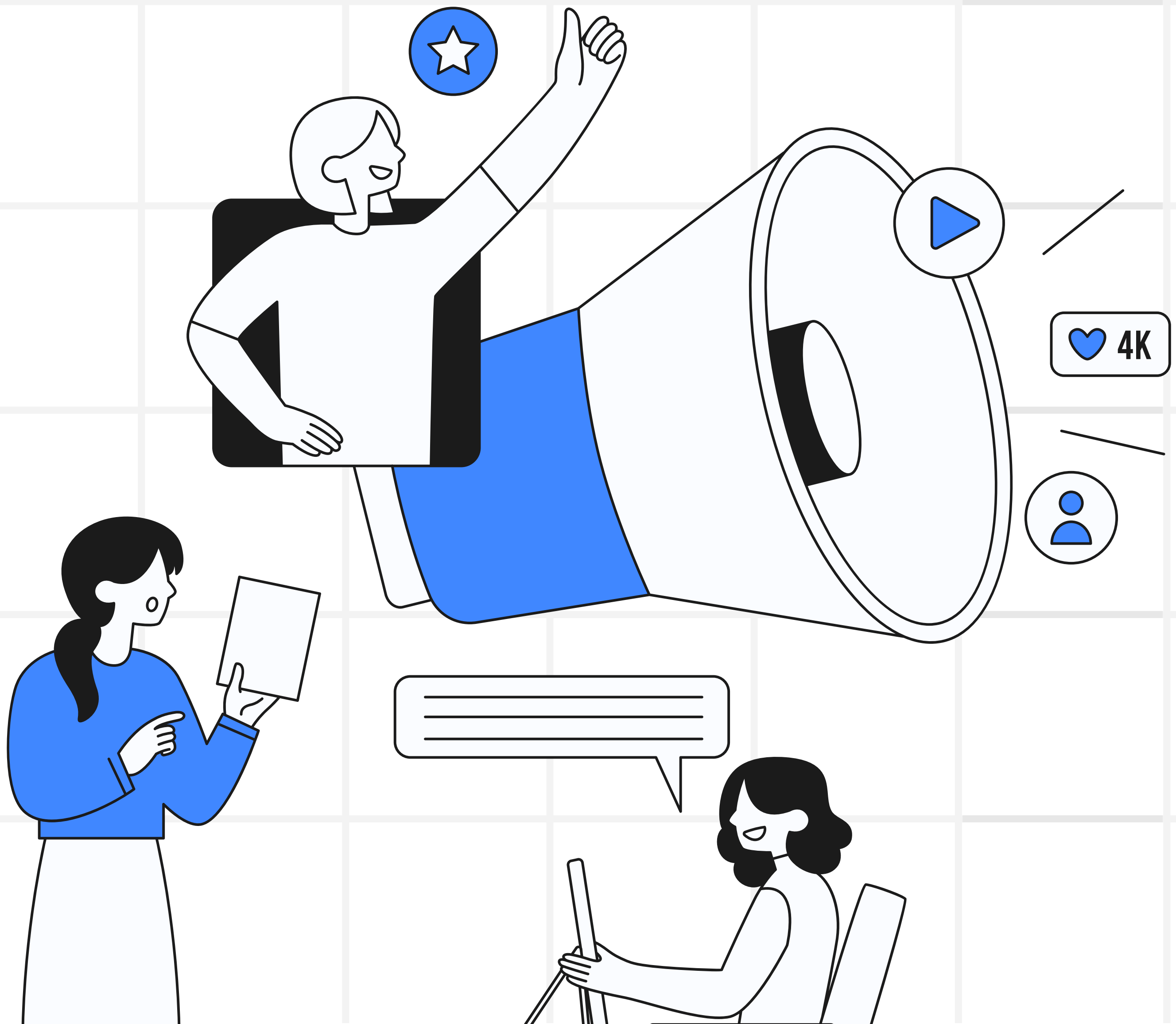
Agents Series



Is Context Engineering really all you need ?



Anantha Narayanan →

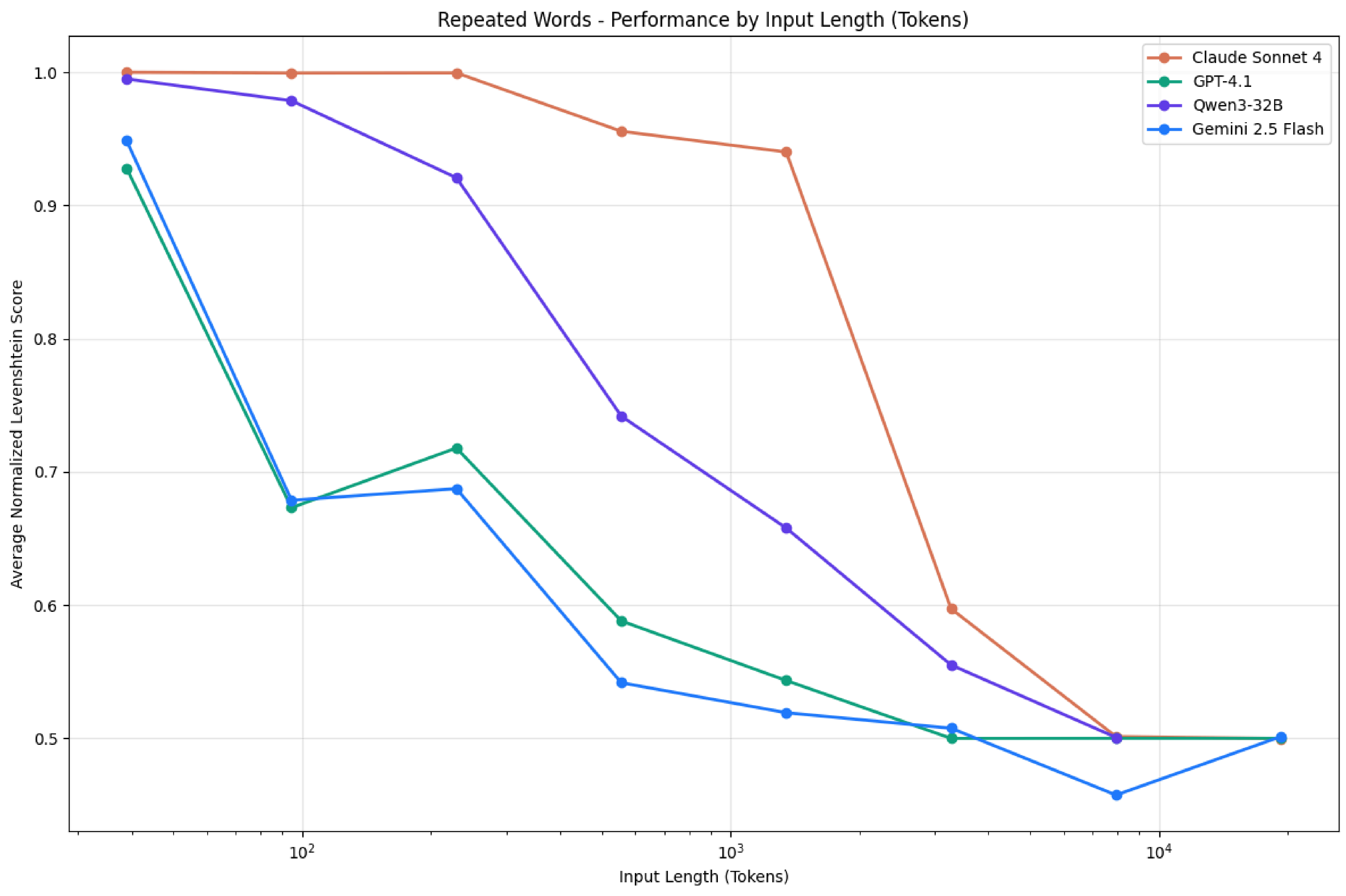
Context Engineering?

Context Engineering is the discipline of designing and building dynamic systems that provides the right information and tools, in the right format, at the right time, to give a LLM everything it needs to accomplish a task.

As **Andrej Karpathy** puts it, LLMs are like a new kind of operating system. The LLM is like the CPU and its context window is like the RAM, serving as the model's working memory. Just like RAM, the LLM context window has limited capacity to handle various sources of context.

Wait, Long context will Fail!

Although Large Language Models (LLMs) are generally assumed to process all parts of the input equally—whether it's the 100th or the 10,000th token—this isn't actually the case in practice. In reality, model performance tends to fluctuate depending on the length of the input, sometimes even on straightforward tasks.



Context Rot!

As frontier model context windows continue to grow[1], with many supporting up to 1 million tokens, I see many excited discussions about how long context windows will unlock the agents of our dreams. Sounds easy, right?

But this isn't the right way to build agents. A study by Chroma on 'Context Rot' reveals that simply increasing input length and maximizing token windows does not lead to linear improvements in accuracy. Instead, LLM performance often degrades unevenly and unpredictably as context length increases—highlighting the limitations of relying on sheer scale alone, rather than applying thoughtful context engineering.

Let's Fix your long context now!

Anantha Narayanan

Context Poisoning

Context Poisoning is when a hallucination or other error makes it into the context, where it is repeatedly referenced.

If the "goals" section of its context was poisoned, the agent would develop nonsensical strategies and repeat behaviors in pursuit of a goal that cannot be met.

A travel agent is helping a user plan a trip to Hyderabad, but it hallucinates Mumbai as the destination early on. This error poisons the context.When the user later asks for a hotel near the Hitec city, the AI responds with Mumbai-based suggestions like "You might enjoy Taj Hotel." Even after correction, the AI keeps referencing Mumbai due to the poisoned goal.

Context Distraction

Context Distraction is when a context grows so long that the model over-focuses on the context, neglecting what it learned during training.As context grows during an agentic workflow—as the model gathers more information and builds up history—this accumulated context can become distracting rather than helpful.

While large context windows (like Gemini with 1M+ tokens) enable more memory, agents often struggle to use them effectively. Beyond 100k tokens, agents tend to repeat past actions instead of generating new plans. Smaller models show this issue even earlier

If models start to misbehave long before their context windows are filled, what's the point of super large context windows?

Context Confusion

Context Confusion is when superfluous content in the context is used by the model to generate a low-quality response.

The Berkeley Function-Calling Leaderboard is a benchmark for evaluating how well models use tools to respond to prompts. In its 3rd version, it reveals that model performance drops when more than one tool is provided. Additionally, in scenarios where none of the tools are relevant, models are expected to make no function call—yet all models sometimes incorrectly call irrelevant tools

The issue is, if you put something in the context the model has to pay attention to it. It may be irrelevant information or needless tool definitions, but the model will take it into account.

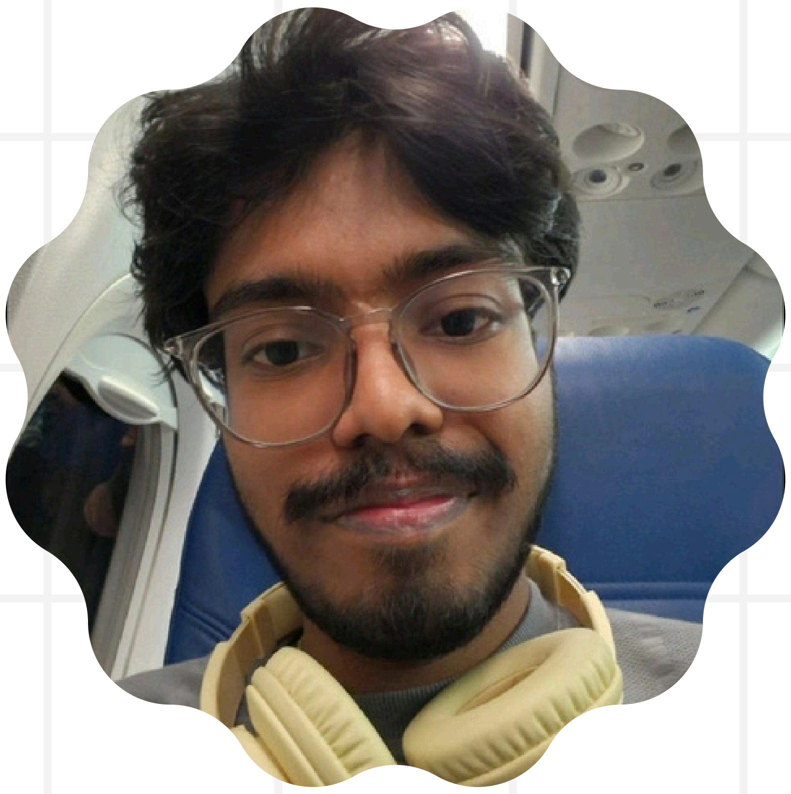
Context Clash

Context Clash is when you accrue new information and tools in your context that conflicts with other information in the context.

This is a more problematic version of Context Confusion: the bad context here isn't irrelevant, it directly conflicts with other information in the prompt.

Agents assemble context from documents, tool calls, and from other models tasked with subproblems. All of this context, pulled from diverse sources, has the potential to disagree with itself. Further, when you connect to MCP tools you didn't create there's a greater chance their descriptions and instructions clash with the rest of your prompt.

LLMs often get lost in multi-turn conversations



Follow me to stay updated on LLM Agent × RAG!

Anantha Narayanan