Anthropic's new Citations API

24th January 2025

Here's a new API-only feature from Anthropic that requires quite a bit of assembly in order to unlock the value: Introducing Citations on the Anthropic API. Let's talk about what this is and why it's interesting.

Citations for Retrieval Augmented Generation

The core of the Retrieval Augmented Generation (RAG) pattern is to take a user's question, retrieve portions of documents that might be relevant to that question and then answer the question by including those text fragments in the context provided to the LLM.

This usually works well, but there is still a risk that the model may answer based on other information from its training data (sometimes OK) or hallucinate entirely incorrect details (definitely bad).

The *best* way to help mitigate these risks is to support the answer with citations that incorporate direct quotations from the underlying source documents. This even acts as a form of fact-checking: the user can confirm that the quoted text did indeed come from those documents, helping provide relatively robust protection against hallucinated details resulting in incorrect answers.

Actually building a system that does this can be quite tricky. Matt Yeung described a pattern for this he called Deterministic Quoting last April, where answers are accompanied by direct quotations from the source documents that are guaranteed to be copied across and not lossily transformed by the model.

This is a great idea, but actually building it requires some quite sophisticated prompt engineering and complex implementation code.

Claude's new Citations API mechanism handles the difficult parts of this for you. You still need to implement most of RAG—identifying potentially relevant documents, then feeding that content in as part of the prompt—but Claude's API will then do the difficult work of extracting relevant citations and including them in the response that it sends back to you.

Trying out the new API with uv run

I tried the API out using Anthropic's Python client library, which was just updated to support the citations API.

I ran a scratch Python 3.13 interpreter with that package using uv run like this (after first setting the necessary `ANTHROPIC_API_KEY` environment variable using llm keys get):

```
export ANTHROPIC_API_KEY="$(llm keys get claude)"
uv run --with anthropic --python 3.13 python
```

Python 3.13 has a nicer interactive interpreter which you can more easily paste code into. Using `uv run` like this gives me an environment with that package pre-installed without me needing to setup a virtual environment as a separate step.

Then I ran the following code, adapted from Anthropic's example. The text.txt Gist contains text I copied out from my Things we learned about LLMs in 2024 post.

```
import urllib.request
import json

url =
'https://gist.githubusercontent.com/simonw/9fbb3c2e2c40c181727e497e358fd7ce/raw/6ac20
text = urllib.request.urlopen(url).read().decode('utf-8')

import anthropic

client = anthropic.Anthropic()

response = client.messages.create(
    model="claude-3-5-sonnet-20241022",
    max_tokens=1024,
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "type": "document",
                    "source": {
                        "type": "text",
                        "media_type": "text/plain",
                        "data": text,
                    },
                    "title": "My Document",
                    "context": "This is a trustworthy document.",
                    "citations": {"enabled": True}
                },
                {
                    "type": "text",
                    "text": "What were the top trends?"
                }
            ]
        }
    ]
)
print(json.dumps(response.to_dict(), indent=2))
```

The JSON output from that starts like this:

```
{
  "id": "msg_01P3zs4aYz2Baebumm4Fejoi",
  "content": [
    {
      "text": "Based on the document, here are the key trends in AI/LLMs from
2024:\n\n1. Breaking the GPT-4 Barrier:\n",
      "type": "text"
    },
    {
      "citations": [
        {
          "cited_text": "I\u2019m relieved that this has changed completely in the
past twelve months. 18 organizations now have models on the Chatbot Arena Leaderboard
that rank higher than the original GPT-4 from March 2023 (GPT-4-0314 on the
board)\u201470 models in total.\n\n",
          "document_index": 0,
          "document_title": "My Document",
          "end_char_index": 531,
          "start_char_index": 288,
          "type": "char_location"
        }
      ],
      "text": "The GPT-4 barrier was completely broken, with 18 organizations now
having models that rank higher than the original GPT-4 from March 2023, with 70
models in total surpassing it.",
      "type": "text"
    },
    {
      "text": "\n\n2. Increased Context Lengths:\n",
      "type": "text"
    },
```


Here's the full response.

This format is pretty interesting! It's the standard Claude format but those `"content"` blocks now include an optional additional `"citations"` key which contains a list of relevant citation extracts that support the claim in the `"text"` block.

Rendering the citations

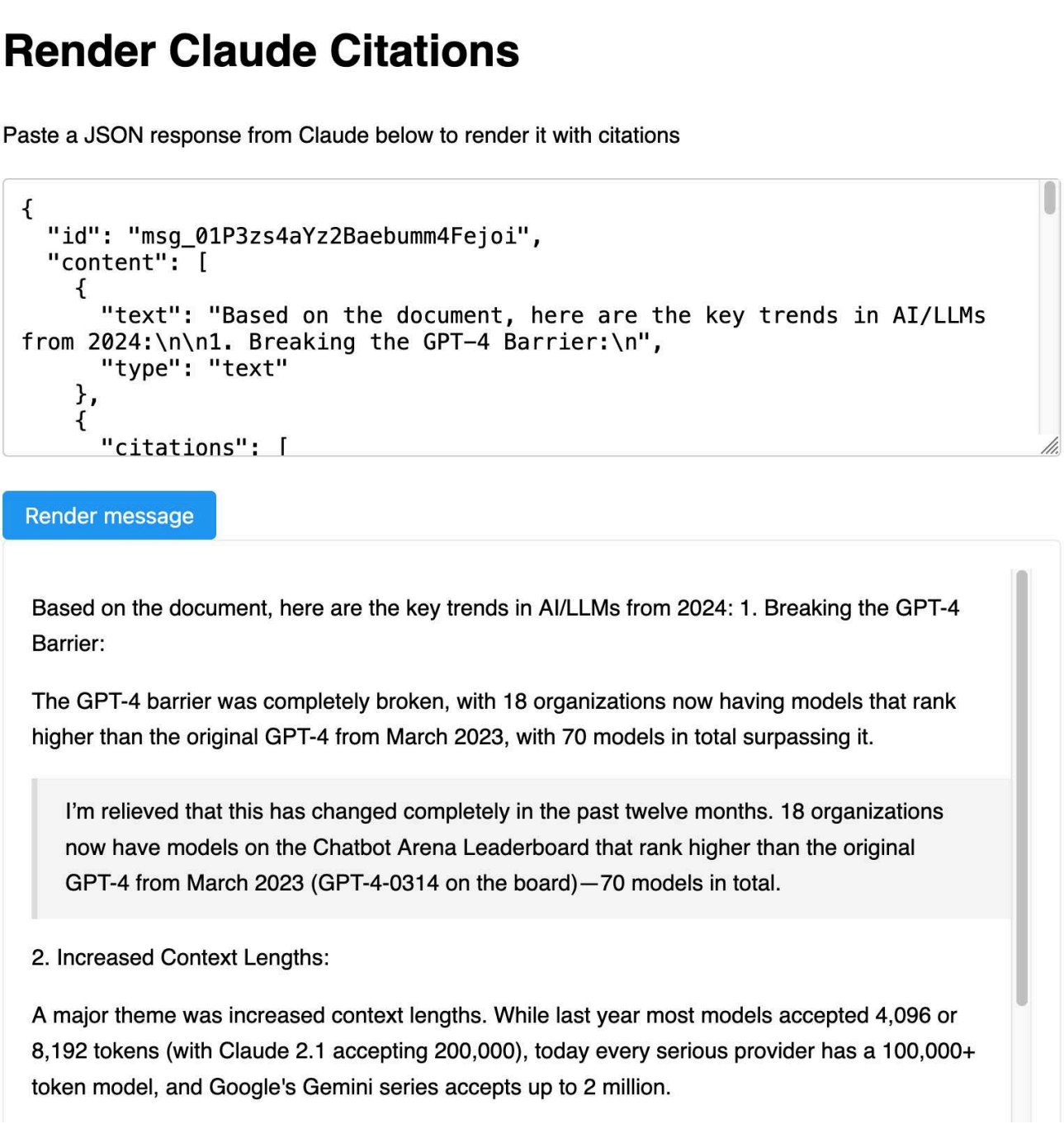
Eyeballing the JSON output wasn't particularly fun. I wanted a very quick tool to help me see that output in a more visual way.

A trick I've been using a lot recently is that LLMs like Claude are *really* good at writing code to turn arbitrary JSON shapes like this into a more human-readable format.

I fired up my Artifacts project, pasted in the above JSON and prompted it like this:

> Build a tool where I can paste JSON like this into a textarea and the result will be rendered in a neat way—it should should intersperse text with citations, where each citation has the cited_text rendered in a blockquote

It helped me build this tool (follow-up prompt here), which lets you paste in JSON and produces a rendered version of the text:



Now I need to design an abstraction layer for LLM

I'd like to upgrade my LLM tool and llm-claude-3 plugin to include support for this new feature... but doing so is going to be relatively non-trivial.

The problem is that LLM currently bakes in an assumption that all LLMs respond with a stream of text.

With citations, this is no longer true! Claude is now returning chunks of text that aren't just a plain string—they are annotated with citations, which need to be stored and processed somehow by the LLM library.

This isn't the only edge-case of this type. DeepSeek recently released their Reasoner API which has a similar problem: it can return two different types of text, one showing reasoning text and one showing final content. I described those differences here.

I've opened a design issue to tackle this challenge in the LLM repository: Design an abstraction for responses that are not just a stream of text.

Anthropic's strategy contrasted with OpenAI

Another interesting aspect of this release is how it helps illustrate a strategic difference between Anthropic and OpenAI.

OpenAI are increasingly behaving like a consumer products company. They just made a big splash with their Operator browser-automation agent system—a much more polished, consumer-product version of Anthropic's own Computer Use demo from a few months ago.

Meanwhile, Anthropic are clearly focused much more on the developer / "enterprise" market. This Citations feature is API-only and directly addresses a specific need that developers trying to build reliable RAG systems on top of their platform may not even have realized they had.

Posted 24th January 2025 at 4:22 am · Follow me on Mastodon or Twitter or subscribe to my newsletter

More recent articles

- OpenAI o3-mini, now available in LLM - 31st January 2025
- A selfish personal argument for releasing code as Open Source - 24th January 2025

tools 29 ai 1073 openai 250 prompt-engineering 118 generative-ai 920 llms 908 ai-assisted-programming 114 llm 130 anthropic 109 claude 115 rag 31 claude-artifacts 27

Next: A selfish personal argument for releasing code as Open Source

Previous: Six short video demos of LLM and Datasette projects