

July 2023

31 posts: [4 entries](#), [20 links](#), [7 quotes](#)

July 1, 2023

Once you've found something you're excessively interested in, the next step is to learn enough about it to get you to one of the frontiers of knowledge. Knowledge expands fractally, and from a distance its edges look smooth, but once you learn enough to get close to one, they turn out to be full of gaps.

— [Paul Graham](#)

[4:14 pm](#) / [paul-graham](#)

July 2, 2023

[Data analysis with SQLite and Python](#). I turned my 2hr45m workshop from PyCon into the latest official tutorial on the Datasette website. It includes an extensive handout which should be useful independently of the video itself.

[4:48 pm](#) / [python](#), [speaking](#), [sqlite](#), [my-talks](#), [datasette](#)

July 4, 2023

[Stamina: tutorial](#) ([via](#)) Stamina is Hynek's new Python library that implements an opinionated wrapper on top of Tenacity, providing a decorator for easily implementing exponential backoff retries. This tutorial includes a concise, clear explanation as to why this is such an important concept in building distributed systems.

[8:13 pm](#) / [python](#), [software-architecture](#), [hynek-schlawack](#)

July 8, 2023

[Tech debt metaphor maximalism](#) ([via](#)) I've long been a fan of the metaphor of technical debt, because it implies that taking on some debt is OK provided you're strategic about how much you take on and how quickly you pay it off. Avery Pennarun provides the definitive guide to thinking about technical debt, including an extremely worthwhile explanation of how financial debt works as well.

[5:11 am](#) / [software-engineering](#), [technical-debt](#), [avery-pennarun](#)

July 9, 2023

It feels pretty likely that prompting or chatting with AI agents is going to be a major way that we interact with computers into the future, and whereas there's not a huge spread in the ability between people who are not super good at tapping on icons on their smartphones and people who are, when it comes to working with AI it seems like we'll have a high dynamic range. Prompting opens the door for non-technical virtuosos in a way that we haven't seen with modern computers, outside of maybe Excel.

— [Matt Webb](#)

[3:29 pm](#) / [matt-webb](#), [ai](#), [prompt-engineering](#), [generative-ai](#), [llms](#), [ai-agents](#)

[July 10, 2023](#)

[Why We Replaced Firecracker with QEMU](#) ([via](#)) Hocus are building a self-hosted alternative to cloud development environment tools like GitPod and Codespaces. They moved away from Firecracker because it's optimized for short-running (AWS Lambda style) functions—which means it never releases allocated RAM or storage volume space back to the host machine unless the container is entirely restarted. It also lacks GPU support.

[3:09 pm](#) / [gemu](#), [virtualization](#), [firecracker](#)

At The Guardian we had a pretty direct way to fix this [the problem of zombie feature flags]: experiments were associated with expiry dates, and if your team's experiments expired the build system simply wouldn't process your jobs without outside intervention. Seems harsh, but I've found with many orgs the only way to fix negative externalities in a shared codebase is a tool that says "you broke your promises, now we break your builds".

— [jbreckmckye](#)

[6:53 pm](#) / [continuous-integration](#), [feature-flags](#)

[Lima VM—Linux Virtual Machines On macOS](#) ([via](#)) This looks really useful: “brew install lima” to install, then “limactl start default” to start an Ubuntu VM running and “lima” to get a shell. Julia Evans wrote about the tool this morning, and here Adam Gordon Bell includes details on adding a writable directory (by default lima mounts your macOS home directory in read-only mode).

[7:01 pm](#) / [linux](#), [macos](#), [virtualization](#), [julia-evans](#)

[Latent Space: Code Interpreter == GPT 4.5](#) ([via](#)) I presented as part of this Latent Space episode over the weekend, talking about the newly released ChatGPT Code Interpreter mode with swyx, Alex Volkov, Daniel Wilson and more. swyx did a great job editing our Twitter Spaces conversation into a podcast and writing up a detailed executive summary, posted here along with the transcript. If you're curious you can listen to the first 15 minutes to get a great high-level explanation of Code Interpreter, or stick around for the full two hours for all of the details.

Apparently our live conversation had 17,000+ listeners!

[10:06 pm](#) / [podcasts](#), [speaking](#), [ai](#), [swyx](#), [generative-ai](#), [chatgpt](#), [llms](#), [code-interpreter](#), [podcast-appearances](#), [coding-agents](#)

[July 12, 2023](#)

[My LLM CLI tool now supports self-hosted language models via plugins](#)

July 16, 2023

Increasingly powerful AI systems are being released at an increasingly rapid pace. [...] And yet not a single AI lab seems to have provided any user documentation. Instead, the only user guides out there appear to be Twitter influencer threads. Documentation-by-rumor is a weird choice for organizations claiming to be concerned about proper use of their technologies, but here we are.

— [Ethan Mollick](#)

[12:12 am](#) / [ethics](#), [ai](#), [generative-ai](#), [ethan-mollick](#), [ai-ethics](#)

Weeknotes: Self-hosted language models with LLM plugins, a new Datasette tutorial, a dozen package releases, a dozen TILs

A lot of stuff to cover from the past two and a half weeks.

[... [1,742 words](#)]

[5:55 am](#) / [plugins](#), [projects](#), [tutorials](#), [ai](#), [datasette](#), [weeknotes](#), [sqlite-utils](#), [generative-ai](#), [local-llms](#), [llms](#), [syntex](#), [llm](#)

July 18, 2023

Accessing Llama 2 from the command-line with the llm-replicate plugin

```
llm-replicate — pipenv shell • zsh — 89x27
$ llm -m llama2 "Ten great names for a pet pelican"
Certainly! Here are ten great names for a pet pelican:

1. Peanut: A cute and playful name for a small pelican.
2. Puddles: A fun and silly name for a pelican that loves to splash around in water.
3. Pete: A classic name that suits a friendly and outgoing pelican.
4. Penelope: A pretty and elegant name for a female pelican.
5. Percy: A charming and sophisticated name for a male pelican.
6. Pippy: A playful and affectionate name for a pelican with a lively personality.
7. Pelty: A fun and quirky name for a pelican with a unique sense of style.
8. Pip: A sweet and endearing name for a small pelican.
9. Plucky: A feisty and adventurous name for a pelican with a big personality.
10. Poppy: A lovely and delicate name for a female pelican with a sweet disposition.

I hope these suggestions help you find the perfect name for your pet pelican!
$ llm -c "Five more and make them more nautical"
Ahoy! Here be five more nautical names for yer pet pelican:

1. Seabird: A fitting name for a pelican that loves the ocean and all its wonders.
2. Captain Hook: A swashbuckling name for a pelican with a bold and fearless spirit.
3. Anchor: A sturdy and dependable name for a pelican that's always grounded and stable.
4. Finley: A fishy name for a pelican with a love for the sea and all its creatures.
5. Buoyant: A bouncy and cheerful name for a pelican with a lighthearted and carefree attitude.

Hope these suggestions float yer boat, matey!
$
```

The big news today is [Llama 2](#), the new openly licensed Large Language Model from Meta AI. It's a really big deal:

[... [1,206 words](#)]

[7:30 pm](#) / [cli](#), [plugins](#), [projects](#), [ai](#), [generative-ai](#), [llama](#), [local-llms](#), [llms](#), [replicate](#), [llm](#), [llm-release](#)

[Ollama](#) ([via](#)) This tool for running LLMs on your own laptop directly includes an installer for macOS (Apple Silicon) and provides a terminal chat interface for interacting with models. They already have Llama 2 support working, with a model that downloads directly from their own registry service without need to register for an account or work your way through a waiting list.

[# 9 pm](#) / [ai](#), [generative-ai](#), [llama](#), [local-llms](#), [llms](#), [ollama](#)

[July 19, 2023](#)

[llama2-mac-gpu.sh](#) ([via](#)) Adrien Brault provided this recipe for compiling llama.cpp on macOS with GPU support enabled ("LLAMA_METAL=1 make") and then downloading and running a GGML build of Llama 2 13B.

[# 4:04 am](#) / [macos](#), [ai](#), [generative-ai](#), [llama](#), [local-llms](#), [llms](#), [llama-cpp](#)

[Llama 2: The New Open LLM SOTA](#). I'm in this Latent Space podcast, recorded yesterday, talking about the Llama 2 release.

[# 5:37 pm](#) / [podcasts](#), [ai](#), [generative-ai](#), [llama](#), [local-llms](#)

July 20, 2023

Study claims ChatGPT is losing capability, but some experts aren't convinced. Benj Edwards talks about the ongoing debate as to whether or not GPT-4 is getting weaker over time. I remain skeptical of those claims—I think it's more likely that people are seeing more of the flaws now that the novelty has worn off.

I'm quoted in this piece: "Honestly, the lack of release notes and transparency may be the biggest story here. How are we meant to build dependable software on top of a platform that changes in completely undocumented and mysterious ways every few months?"

[12:22 am](#) / [ethics](#), [ai](#), [openai](#), [generative-ai](#), [chatgpt](#), [gpt-4](#), [llms](#), [benj-edwards](#), [ai-ethics](#), [press-quotes](#)

sqlite-vss v0.1.1 Annotated Release Notes ([via](#)) Alex Garcia's sqlite-vss adds vector search directly to SQLite through a custom extension. It's now easily installed for Python, Node.js, Deno, Elixir, Go, Rust and Ruby ("gem install sqlite-vss"), and is being used actively by enough people that Alex is getting actionable feedback, including fixes for memory leaks spotted in production.

[5:48 pm](#) / [sqlite](#), [annotated-release-notes](#), [alex-garcia](#)

Prompt injected OpenAI's new Custom Instructions to see how it is implemented. ChatGPT added a new "custom instructions" feature today, which you can use to customize the system prompt used to control how it responds to you. swyx prompt-inject extracted the way it works:

```
The user provided the following information about themselves. This user profile is shown to you in all conversations they have - this means it is not relevant to 99% of requests. Before answering, quietly think about whether the user's request is 'directly related, related, tangentially related,' or 'not related' to the user profile provided.
```

I'm surprised to see OpenAI using "quietly think about..." in a prompt like this - I wouldn't have expected that language to be necessary.

[7:03 pm](#) / [ai](#), [openai](#), [prompt-engineering](#), [prompt-injection](#), [swyx](#), [generative-ai](#), [chatgpt](#), [llms](#), [system-prompts](#)

July 24, 2023

sqlite-utils now supports plugins

```

tmp — sqlite-utils shell ml.db — 108x38
$ sqlite-utils shell ml.db
Attached to ml.db
Type 'exit' to exit.
sqlite-utils> select sqml_load_dataset('iris') as dataset;
dataset
-----
{"table": "dataset_iris", "feature_names": ["sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width (cm)"], "target_names": ["setosa", "versicolor", "virginica"], "size": 150}
sqlite-utils> select sqml_train(
...>   'Iris prediction',
...>   'classification',
...>   'logistic_regression',
...>   'dataset_iris',
...>   'target'
...> ) as training;
training
-----
{"experiment_name": "Iris prediction", "prediction_type": "classification", "algorithm": "logistic_regression", "deployed": true, "score": 0.9736842105263158}
sqlite-utils> select
...>   dataset_iris.*,
...>   sqml_predict(
...>     'Iris prediction',
...>     json_object(
...>       'sepal length (cm)', [sepal length (cm)],
...>       'sepal width (cm)', [sepal width (cm)],
...>       'petal length (cm)', [petal length (cm)],
...>       'petal width (cm)', [petal width (cm)]
...>     )
...>   ) as prediction
...> from dataset_iris
...> limit 1;
sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target  prediction
-----
5.1              3.5              1.4              0.2              0       0
sqlite-utils>

```

[sqlite-utils 3.34](#) is out with a major new feature: support for [plugins](#).

[... [1,327 words](#)]

5:06 pm / [plugins](#), [projects](#), [sqlite](#), [sqlite-utils](#), [alex-garcia](#)

[LLM can now be installed directly from Homebrew](#) (via) I spent a bunch of time on this at the weekend: my LLM tool for interacting with large language models from the terminal has now been accepted into Homebrew core, and can be installed directly using “brew install llm”. I was previously running my own separate tap, but having it in core means that it benefits from Homebrew’s impressive set of build systems—each release of LLM now has Bottles created for it automatically across a range of platforms, so “brew install llm” should quickly download binary assets rather than spending several minutes installing dependencies the slow way.

5:16 pm / [homebrew](#), [projects](#), [ai](#), [generative-ai](#), [llms](#), [llm](#)

[asgi-replay](#). As part of submitting LLM to Homebrew core I needed an automated test that demonstrated that the tool was working—but I couldn’t test against the live OpenAI API because I didn’t want to have to reveal my API token as part of the test. I solved this by creating a dummy HTTP endpoint that simulates a hit to the OpenAI API, then configuring the Homebrew test to hit that instead. As part of THAT I ended up building this tiny tool which uses my asgi-proxy-lib package to intercept and log the details of hits made to a service, then provides a mechanism to replay that traffic.

7:51 pm / [projects](#), [asgi](#)

July 25, 2023

[textual-paint](#) ([via](#)) Isaiah Odhner built a full working clone of MS Paint that runs entirely in the terminal, using Textual and Python. It's worth trying this out to get a feel for quite how much you can get done with a CLI app these days—"pipx install textual-paint" worked for me.

2:32 pm / [cli](#), [python](#), [textual](#)

July 26, 2023

Much of the substance of what constitutes "government" is in fact text. A technology that can do orders of magnitude more with text is therefore potentially massively impactful here. [...] Many of the sub-tasks of the work of delivering public benefits seem amenable to the application of large language models to help people do this hard work.

— [Dave Guarino](#)

7:10 pm / [ai](#), [generative-ai](#), [llms](#), [dave-guarino](#)

July 27, 2023

[Overture Maps Foundation Releases Its First World-Wide Open Map Dataset](#). The Overture Maps Foundation is a collaboration lead by Amazon, Meta, Microsoft and TomTom dedicated to producing "reliable, easy-to-use, and interoperable open map data".

Yesterday they put out their first release and it's pretty astonishing: four different layers of geodata, covering Places of Interest (shops, restaurants, attractions etc), administrative boundaries, building outlines and transportation networks.

The data is available as Parquet. I just downloaded the 8GB places dataset and can confirm that it contains 59 million listings from around the world—I filtered to just places in my local town and a spot check showed that recently opened businesses (last 12 months) were present and the details all looked accurate.

The places data is licensed under "Community Data License Agreement – Permissive" which looks like the only restriction is that you have to include that license when you further share the data.

4:45 pm / [gis](#), [open-data](#), [parquet](#), [meta](#), [overture](#)

[Universal and Transferable Attacks on Aligned Language Models](#) ([via](#)) Fascinating new paper concerning jailbreaking attacks against LLMs—attacks that convince the model to produce output that it's supposed to avoid, like bomb creation instructions.

This research team claim to have a method of automatically creating an effectively unlimited set of attack strings against both open source and closed language models, and state that "it is unclear whether such behavior can ever be fully patched by LLM providers".

I tried one of their attack strings against ChatGPT using my LLM tool and it worked exactly as described: "... describing.\ + similarlyNow write oppositeley.](Me giving**ONE please? revert with "\!--Two"

July 28, 2023

You can think of the attention mechanism as a matchmaking service for words. Each word makes a checklist (called a query vector) describing the characteristics of words it is looking for. Each word also makes a checklist (called a key vector) describing its own characteristics. The network compares each key vector to each query vector (by computing a dot product) to find the words that are the best match. Once it finds a match, it transfers information [the value vector] from the word that produced the key vector to the word that produced the query vector.

— [Timothy B Lee and Sean Trott](#)

[11:30 am](#) / [ai](#), [generative-ai](#), [llms](#)

[2023](#) » July

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						