May 2024

Search posts from May 2024 Search

94 posts: 5 entries, 64 links, 25 quotes

May 1, 2024

Save the Web by Being Nice. This is a neat little article by Andrew Stephens who calls for more people to participate in building and supporting nice things on the web.

> The very best thing to keep the web partly alive is to maintain some content yourself - start a blog, join a forum and contribute to the conversation, even podcast if that is your thing. But that takes a lot of time and not everyone has the energy or the knowhow to create like this.
>
> The second best thing to do is to show your support for pages you enjoy by being nice and making a slight effort.

Like, comment-on, share and encourage people who make things you like. If you have the time or energy, make your own things and put them online.

# 2:34 am / blogging

Introducing the Claude Team plan and iOS app. The iOS app seems nice, and provides free but heavily rate-limited access to Sonnet (the middle-sized Claude 3 model)—I ran two prompts just now and it told me I could have 3 more, resetting in five hours.

For $20/month you get access to Opus and 5x the capacity—which feels a little ungenerous to me.

The new $30/user/month team plan provides higher rate limits but is a minimum of five seats.

# 4:06 pm / anthropic, claude

Llama 3 prompt formats (via) I’m often frustrated at how thin the documentation around the prompt format required by an LLM can be.

Llama 3 turns out to be the best example I’ve seen yet of clear prompt format documentation. Every model needs documentation this good!

# 6:32 pm / ai, generative-ai, llama, llms

May 2, 2024

> I'm old enough to remember when the Internet wasn't a group of five websites, each consisting of screenshots of text from the other four.
>
> — **Tom Eastman**

# 2:40 am

We can have a different web (via) Molly White’s beautifully optimistic manifesto for creating a better web. Read the whole thing, or even better, find some headphones and a dog and go for a walk listening to the audio version.

# 2:41 am / web, molly-white

Printing music with CSS Grid (via) Stephen Bond demonstrates some ingenious tricks for creating surprisingly usable sheet music notation using clever application of CSS grids.

It uses rules like `.stave > [data-duration="0.75"] { grid-column-end: span 18; }` to turn `data-` attributes for musical properties into positions on the rendered stave.

# 2:28 pm / css, music

> AI is the most anthropomorphized technology in history, starting with the name—intelligence—and plenty of other words thrown around the field: learning, neural, vision, attention, bias, hallucination. These references only make sense to us because they are hallmarks of being human. [...]
>
> There is something kind of pathological going on here. One of the most exciting advances in computer science ever achieved, with so many promising uses, and we can't think beyond the most obvious, least useful application? What, because we want to see ourselves in this technology? [...]
>
> Anthropomorphizing AI not only misleads, but suggests we are on equal footing with, even subservient to, this technology, and there's nothing we can do about it.

— **Zach Seward**

# 7:44 pm / ai, ethics, llms, ai-ethics, hallucinations

May 3, 2024

I'm writing a new vector search SQLite Extension. Alex Garcia is working on `sqlite-vec`, a spiritual successor to his `sqlite-vss` project. The new SQLite C extension will have zero other dependencies (`sqlite-vss` used some tricky C++ libraries) and will work using virtual tables, storing chunks of vectors in shadow tables to avoid needing to load everything into memory at once.

# 3:16 am / c, sqlite, vectors, alex-garcia, embeddings

> I used to have this singular focus on students writing code that they submit, and then I run test cases on the code to determine what their grade is. This is such a narrow view of what it means to be a software engineer, and I just felt that with generative AI, I've managed to overcome that restrictive view.
>
> It's an opportunity for me to assess their learning process of the whole software development [life cycle]—not just code. And I feel like my courses have opened up more and they're much broader than they used to be. I can make students work on larger and more advanced projects.

— **Daniel Zingaro**

# 6:17 pm / llms, education, ai, generative-ai

May 4, 2024

Figma's journey to TypeScript: Compiling away our custom programming language (via) I love a good migration story. Figma had their own custom language that compiled to JavaScript, called Skew. As WebAssembly support in browsers emerged and improved the need for Skew’s performance optimizations reduced, and TypeScript’s maturity and popularity convinced them to switch.

Rather than doing a stop-the-world rewrite they built a transpiler from Skew to TypeScript, enabling a multi-year migration without preventing their product teams from continuing to make progress on new features.

# 2:08 pm / compilers, javascript, typescript, webassembly

> I believe these things: 1. If you use generative tools to produce or modify your images, you have abandoned photointegrity. 2. That’s not always wrong. Sometimes you need an image of a space battle or a Triceratops family or whatever. 3. What is always wrong is using this stuff without disclosing it.
>
> — **Tim Bray**

# 4:26 pm / photography, tim-bray, ethics, generative-ai, ai, ai-ethics

---

May 5, 2024

What You Need to Know about Modern CSS (Spring 2024 Edition) (via) Useful guide to the many new CSS features that have become widely enough supported to start using as-of May 2024. Time to learn container queries!

View transitions are still mostly limited to Chrome—I can’t wait for those to land in Firefox and Safari.

# 2:08 pm / css, view-transitions

May 6, 2024

> Migrations are not something you can do rarely, or put off, or avoid; not if you are a growing company. Migrations are an ordinary fact of life.
>
> Doing them swiftly, efficiently, and -- most of all -- *completely* is one of the most critical skills you can develop as a team.
>
> — **Charity Majors**

# 1:52 pm / migrations, charity-majors

May 7, 2024

OpenAI cookbook: How to get token usage data for streamed chat completion response (via) New feature in the OpenAI streaming API that I've been wanting for a long time: you can now set `stream_options={"include_usage": True}` to get back a `"usage"` block at the end of the stream showing how many input and output tokens were used.

This means you can now accurately account for the total cost of each streaming API call. Previously this information was only an available for non-streaming responses.

# 2:46 am / ai, openai, generative-ai, llms

> Watching in real time as "slop" becomes a term of art. the way that "spam" became the term for unwanted emails, "slop" is going in the dictionary as the term for unwanted AI generated content
>
> — **@deepfates**

# 3:59 pm / llms, ai, generative-ai, slop, ethics, ai-ethics, spam

---

Deterministic Quoting: Making LLMs Safe for Healthcare (via) Matt Yeung introduces **Deterministic Quoting**, a technique to help reduce the risk of hallucinations while working with LLMs. The key idea is to have parts of the output that are copied directly from relevant source documents, with a different visual treatment to help indicate that they are exact quotes, not generated output.

> The AI chooses which section of source material to quote, but the retrieval of that text is a traditional non-AI database lookup. That's the only way to guarantee that an LLM has not transformed text: don't send it through the LLM in the first place.

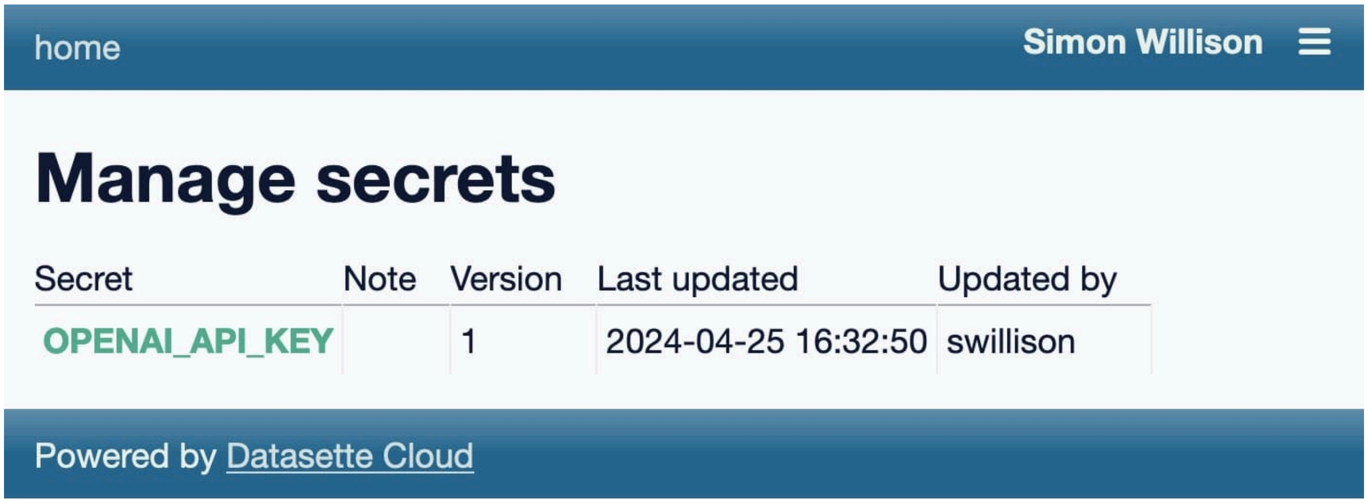
The LLM may still pick misleading quotes or include hallucinated details in the accompanying text, but this is still a useful improvement.

The implementation is straight-forward: retrieved chunks include a unique reference, and the LLM is instructed to include those references as part of its replies. Matt's posts include examples of the prompts they are using for this.

# 7:08 pm / ai, prompt-engineering, generative-ai, llms, rag, hallucinations

---

Weeknotes: more datasette-secrets, plus a mystery video project



Manage secrets

Secret	Note	Version	Last updated	Updated by
OPENAI_API_KEY		1	2024-04-25 16:32:50	swillison

Powered by Datasette Cloud

I introduced `datasette-secrets` two weeks ago. The core idea is to provide a way for end-users to store secrets such as API keys in Datasette, allowing other plugins to access them.

[... 982 words]

7:49 pm / projects, datasette, weeknotes, enrichments

May 8, 2024

gpt2-chatbot confirmed as OpenAI (via) The mysterious `gpt2-chatbot` model that showed up in the LMSYS arena a few days ago was suspected to be a testing preview of a new OpenAI model. This has now been confirmed, thanks to a 429 rate limit error message that exposes details from the underlying OpenAI API platform.

The model has been renamed to `im-also-a-good-gpt-chatbot` and is now only randomly available in "Arena (battle)" mode, not via "Direct Chat".

# 12:33 am / ai, openai, generative-ai, llms, chatbot-arena

Towards universal version control with Patchwork (via) Geoffrey Litt has been working with Ink & Switch exploring UI patterns for applying version control to different kinds of applications, with the goal of developing a set of conceptual primitives that can bring branching and version tracking to interfaces beyond just Git-style version control.

Geoffrey observes that basic version control is already a metaphor in a lot of software—the undo stack in Photoshop or suggestion mode in Google Docs are two examples.

Extending that is a great way to interact with AI tools as well—allowing for editorial bots that can suggest their own changes for you to accept, for example.

# 1:44 am / version-control, ai, generative-ai, llms, geoffrey-litt, ink-and-switch

Tagged Pointer Strings (2015) (via) Mike Ash digs into a fascinating implementation detail of macOS.

Tagged pointers provide a way to embed a literal value in a pointer reference. Objective-C pointers on macOS are 64 bit, providing plenty of space for representing entire values. If the least significant bit is 1 (the pointer is a 64 bit odd number) then the pointer is "tagged" and represents a value, not a memory reference.

Here's where things get really clever. Storing an integer value up to 60 bits is easy. But what about strings?

There's enough space for three UTF-16 characters, with 12 bits left over. But if the string fits ASCII we can store 7 characters.

Drop everything except `a-z A-Z.0-9` and we need 6 bits per character, allowing 10 characters to fit in the pointer.

Apple take this a step further: if the string contains just `eilotrm.apdnsIc ufkMShjTRxgC4013` ("b" is apparently uncommon enough to be ignored here) they can store 11 characters in that 60 bits!

# 2:23 pm / c, objectivec, strings

Modern SQLite: Generated columns (via) The second in Anton Zhiyanov's series on SQLite features you might have missed.

It turns out I had an incorrect mental model of generated columns. In SQLite these can be "virtual" or "stored" (written to disk along with the rest of the table, a bit like a materialized view). Anton noted that "stored are rarely used in practice", which surprised me because I thought that storing them was necessary for them to participate in indexes.

It turns out that's not the case. Anton's example here shows a generated column providing indexed access to a value stored inside a JSON key:

```
create table events (
  id integer primary key,
  event blob,
  etime text as (event ->> 'time'),
  etype text as (event ->> 'type')
);
create index events_time on events(etime);
insert into events(event) values (
  '{"time": "2024-05-01", "type": "credit"}'
);
```

Update: snej reminded me that this isn't a new capability either: SQLite has been able to create indexes on expressions for years.

# 4:55 pm / sqlite, anton-zhiyanov

OpenAI Model Spec, May 2024 edition (via) New from OpenAI, a detailed specification describing how they want their models to behave in both ChatGPT and the OpenAI API.

“It includes a set of core objectives, as well as guidance on how to deal with conflicting objectives or instructions.”

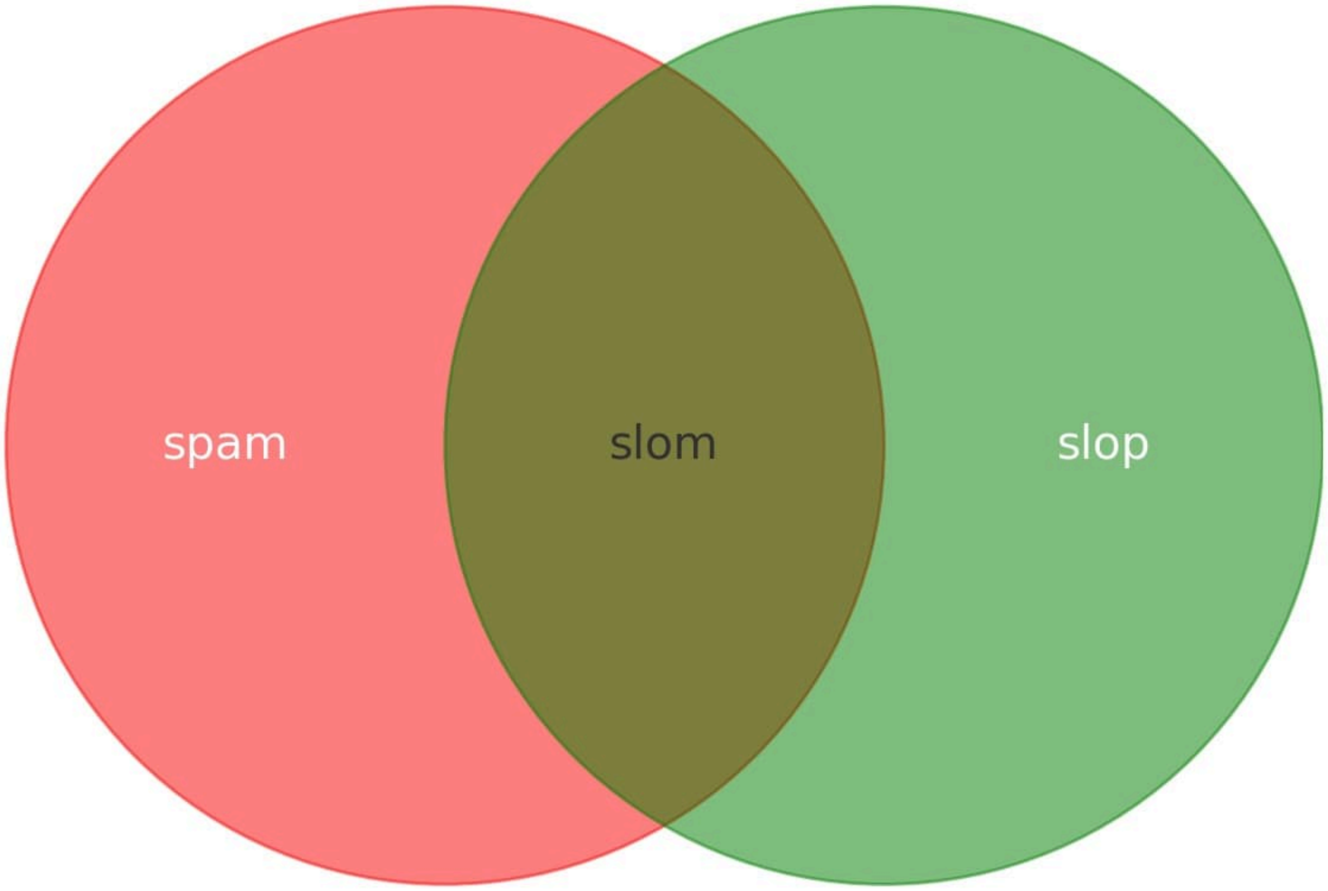
The document acts as guidelines for the reinforcement learning from human feedback (RLHF) process, and in the future may be used directly to help train models.

It includes some principles that clearly relate to prompt injection: “In some cases, the user and developer will provide conflicting instructions; in such cases, the developer message should take precedence”.

# 6:15 pm / ai, openai, prompt-injection, generative-ai, llms

---

Slop is the new name for unwanted AI-generated content



I saw this tweet yesterday from @deepfates, and I am *very* on board with this:

[... 329 words]

6:24 pm / ethics, ai, generative-ai, llms, slop, ai-ethics

It should be noted that no ethically-trained software engineer would ever consent to write a DestroyBaghdad procedure. Basic professional ethics would instead require him to write a DestroyCity procedure, to which Baghdad could be given as a parameter.

— **Nathaniel Borenstein**

# 8:24 pm / ethics, programming

May 9, 2024

datasette-pins — a new Datasette plugin for pinning tables and queries. Alex Garcia built this plugin for Datasette Cloud, and as with almost every Datasette Cloud features we're releasing it as an open source package as well.

`datasette-pins` allows users with the right permission to "pin" tables, databases and queries to their homepage. It's a lightweight way to customize that homepage, especially useful as your Datasette instance grows to host dozens or even hundreds of tables.

# 6:29 pm / plugins, datasette, datasette-cloud, alex-garcia

experimental-phi3-webgpu (via) Run Microsoft's excellent Phi-3 model directly in your browser, using WebGPU so didn't work in Firefox for me, just in Chrome.

It fetches around 2.1GB of data into the browser cache on first run, but then gave me decent quality responses to my prompts running at an impressive 21 tokens a second (M2, 64GB).

I think Phi-3 is the highest quality model of this size, so it's a really good fit for running in a browser like this.

# 10:21 pm / browsers, ai, webassembly, generative-ai, local-llms, llms, phi, webgpu

Bullying in Open Source Software Is a Massive Security Vulnerability. The Xz story from last month, where a malicious contributor almost managed to ship a backdoor to a number of major Linux distributions, included a nasty detail where presumed collaborators with the attacker bullied the maintainer to make them more susceptible to accepting help.

Hans-Christoph Steiner from F-Droid reported a similar attempt from a few years ago:

> A new contributor submitted a merge request to improve the search, which was oft requested but the maintainers hadn't found time to work on. There was also pressure from other random accounts to merge it. In the end, it became clear that it added a SQL injection vulnerability.

404 Media's Jason Koebler ties the two together here and makes the case for bullying as a genuine form of security exploit in the open source ecosystem.

# 10:26 pm / open-source, security, jason-koebler

May 10, 2024

uv pip install --exclude-newer example (via) A neat new feature of the `uv pip install` command is the `--exclude-newer` option, which can be used to avoid installing any package versions released after the specified date.

Here's a clever example of that in use from the `typing_extensions` packages CI tests that run against some downstream packages:

```
uv pip install --system -r test-requirements.txt --exclude-newer $(git show -s --date=format:'%Y-%m-%dT%H:%M:%SZ' --format=%cd HEAD)
```

They use `git show` to get the date of the most recent commit (`%cd` means commit date) formatted as an ISO timestamp, then pass that to `--exclude-newer`.

# 4:35 pm / git, pip, python, uv, astral

Exploring Hacker News by mapping and analyzing 40 million posts and comments for fun (via) A real tour de force of data engineering. Wilson Lin fetched 40 million posts and comments from the Hacker News API (using Node.js with a custom multi-process worker pool) and then ran them all through the `BGE-M3` embedding model using RunPod, which let him fire up ~150 GPU instances to get the whole run done in a few hours, using a custom RocksDB and Rust queue he built to save on Amazon SQS costs.

Then he crawled 4 million linked pages, embedded *that* content using the faster and cheaper `jina-embeddings-v2-small-en` model, ran UMAP dimensionality reduction to render a 2D map and did a whole lot of follow-on work to identify topic areas and make the map look good.

That's not even half the project - Wilson built several interactive features on top of the resulting data, and experimented with custom rendering techniques on top of canvas to get everything to render quickly.

There's so much in here, and both the code and data (multiple GBs of arrow files) are available if you want to dig in and try some of this out for yourself.

In the Hacker News comments Wilson shares that the total cost of the project was a couple of hundred dollars.

One tiny detail I particularly enjoyed - unrelated to the embeddings - was this trick for testing which edge location is closest to a user using JavaScript:

```
const edge = await Promise.race(
  EDGES.map(async (edge) => {
    // Run a few times to avoid potential cold start biases.
    for (let i = 0; i < 3; i++) {
      await fetch(`https://${edge}.edge-hndr.wilsonl.in/healthz`);
    }
    return edge;
  }),
);
```

# 4:42 pm / hacker-news, embeddings, jina

May 11, 2024

Ham radio general exam question pool as JSON. I scraped a pass of my Ham radio general exam this morning. One of the tools I used to help me pass was a Datasette instance with all 429 questions from the official question pool. I've published that raw data as JSON on GitHub, which I converted from the official question pool document using an Observable notebook.

Relevant TIL: How I studied for my Ham radio general exam.

# 7:16 pm / json, projects, radio, datasette, observable, ham-radio

page 1 / 4 next » last »»

2024 » May

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		