April 2025

Search posts from April 2025 Search

100 posts: 12 entries, 50 links, 26 quotes, 12 notes

April 1, 2025

> We're planning to release a very capable open language model in the coming months, our first since GPT-2. [...]
>
> As models improve, there is more and more demand to run them everywhere. Through conversations with startups and developers, it became clear how important it was to be able to support a spectrum of needs, such as custom fine-tuning for specialized tasks, more tunable latency, running on-prem, or deployments requiring full data control.

— **Brad Lightcap**, COO, OpenAI

# 2:53 am / openai, llms, ai, generative-ai

Pydantic Evals (via) Brand new package from David Montague and the Pydantic AI team which directly tackles what I consider to be the single hardest problem in AI engineering: building evals to determine if your LLM-based system is working correctly and getting better over time.

The feature is described as "in beta" and comes with this very realistic warning:

> Unlike unit tests, evals are an emerging art/science; anyone who claims to know for sure exactly how your evals should be defined can safely be ignored.

This code example from their documentation illustrates the relationship between the two key nouns - Cases and Datasets:

```
from pydantic_evals import Case, Dataset

case1 = Case(
    name="simple_case",
    inputs="What is the capital of France?",
    expected_output="Paris",
    metadata={"difficulty": "easy"},
)

dataset = Dataset(cases=[case1])
```

The library also supports custom evaluators, including LLM-as-a-judge:

```
Case(
    name="vegetarian_recipe",
    inputs=CustomerOrder(
        dish_name="Spaghetti Bolognese", dietary_restriction="vegetarian"
    ),
    expected_output=None,
    metadata={"focus": "vegetarian"},
    evaluators=(
        LLMJudge(
            rubric="Recipe should not contain meat or animal products",
        ),
    ),
)
```

Cases and datasets can also be serialized to YAML.

My first impressions are that this looks like a solid implementation of a sensible design. I'm looking forward to trying it out against a real project.

# 4:43 am / python, ai, generative-ai, llms, evals, pydantic

---

Half Stack Data Science: Programming with AI, with Simon Willison (via) I participated in this wide-ranging 50 minute conversation with David Asboth and Shaun McGirr. Topics we covered included applications of LLMs to data journalism, the challenges of building an intuition for how best to use these tool given their "jagged frontier" of capabilities, how LLMs impact learning to program and how local models are starting to get genuinely useful now.

At 27:47:

> If you're a new programmer, my optimistic version is that there has never been a better time to learn to program, because it shaves down the learning curve so much. When you're learning to program and you miss a semicolon and you bang your head against the computer for four hours [...] if you're unlucky you quit programming for good because it was so frustrating. [...]
>
> I've always been a project-oriented learner; I can learn things by building something, and now the friction involved in building something has gone down so much [...] So I think especially if you're an autodidact, if you're somebody who likes teaching yourself things, these are a gift from heaven. You get a weird teaching assistant that knows loads of stuff and occasionally makes weird mistakes and believes in bizarre conspiracy theories, but you have 24 hour access to that assistant.
>
> If you're somebody who prefers structured learning in classrooms, I think the benefits are going to take a lot longer to get to you because we don't know how to use these things in classrooms yet. [...]
>
> If you want to strike out on your own, this is an amazing tool *if* you learn how to learn with it. So you've got to learn the limits of what it can do, and you've got to be disciplined enough to make sure you're not outsourcing the bits you need to learn to the machines.

# 2:27 pm / data-journalism, podcasts, ai, generative-ai, llms, podcast-appearances

---

April 2, 2025

Composite primary keys in Django. Django 5.2 is out today and a big new feature is composite primary keys, which can now be defined like this:

```
class Release(models.Model):
    pk = models.CompositePrimaryKey(
        "version", "name"
    )
    version = models.IntegerField()
    name = models.CharField(max_length=20)
```

They don't yet work with the Django admin or as targets for foreign keys.

Other smaller new features include:

- All ORM models are now automatically imported into `./manage.py shell` - a feature borrowed from `./manage.py shell_plus` in django-extensions

- Feeds from the Django syndication framework can now specify XSLT stylesheets
- response.text now returns the string representation of the body - I'm so happy about this, now I don't have to litter my Django tests with `response.content.decode("utf-8")` any more
- a new simple_block_tag helper making it much easier to create a custom Django template tag that further processes its own inner rendered content
- A bunch more in the full release notes

5.2 is also an LTS release, so it will receive security and data loss bug fixes up to April 2028.

# 2:51 pm / django, python

April 3, 2025

> I started using Claude and Claude Code a bit in my regular workflow. I'll skip the suspense and just say that the tool is *way* more capable than I would ever have expected. The way I can use it to interrogate a large codebase, or generate unit tests, or even "refactor every callsite to use such-and-such pattern" is utterly gobsmacking. [...]
>
> Here's the main problem I've found with generative AI, and with "vibe coding" in general: it completely sucks out the joy of software development for me. [...]
>
> This is how I feel using gen-AI: like a babysitter. It spits out reams of code, I read through it and try to spot the bugs, and then we repeat.

— **Nolan Lawson**, AI ambivalence

# 1:56 am / ai-assisted-programming, claude, generative-ai, ai, llms, nolan-lawson, claude-code, coding-agents

---

Minimal CSS-only blurry image placeholders (via) Absolutely brilliant piece of CSS ingenuity by Lean Rada, who describes a way to implement blurry placeholder images using just CSS, with syntax like this:

```
<img src="…" style="--lqip:192900">
```

That 192900 number encodes everything needed to construct the placeholder - it manages to embed a single base color and six brightness components (in a 3x2 grid) in 20 bits, then encodes those as an integer in the roughly 2 million available values between -999,999 and 999,999 - beyond which range Lean found some browsers would start to lose precision.

The implementation for decoding that value becomes a bunch of clever bit-fiddling CSS expressions to expand it into further CSS variables:

```
[style*="--lqip:"] {
  --lqip-ca: mod(round(down, calc((var(--lqip) + pow(2, 19)) / pow(2, 18))), 4);
  --lqip-cb: mod(round(down, calc((var(--lqip) + pow(2, 19)) / pow(2, 16))), 4);
  /* more like that */
}
```

Which are expanded to even more variables with code like this:

```
--lqip-ca-clr: hsl(0 0% calc(var(--lqip-ca) / 3 * 100%));
--lqip-cb-clr: hsl(0 0% calc(var(--lqip-cb) / 3 * 100%));
```

And finally rendered using a CSS gradient definition that starts like this:

```
[style*="--lqip:"] {
  background-image:
    radial-gradient(50% 75% at 16.67% 25%, var(--lqip-ca-clr), transparent),
    radial-gradient(50% 75% at 50% 25%, var(--lqip-cb-clr), transparent),
    /* ... */
    linear-gradient(0deg, var(--lqip-base-clr), var(--lqip-base-clr));
}
```

The article includes several interactive explainers (most of which are also powered by pure CSS) illustrating how it all works.

Their Node.js script for converting images to these magic integers uses Sharp to resize the image to 3x2 and then use the Oklab perceptually uniform color space (new to me, that was created by Björn Ottosson in 2020) to derive the six resulting values.

# 2:44 am / css, css-custom-properties

---

smartfunc. Vincent D. Warmerdam built this ingenious wrapper around my LLM Python library which lets you build LLM wrapper functions using a decorator and a docstring:

```python
from smartfunc import backend

@backend("gpt-4o")
def generate_summary(text: str):
    """Generate a summary of the following text: {{ text }}"""
    pass

summary = generate_summary(long_text)
```

It works with LLM plugins so the same pattern should work against Gemini, Claude and hundreds of others, including local models.

It integrates with more recent LLM features too, including async support and schemas, by introspecting the function signature:

```python
class Summary(BaseModel):
    summary: str
    pros: list[str]
    cons: list[str]

@async_backend("gpt-4o-mini")
async def generate_poke_desc(text: str) -> Summary:
    "Describe the following pokemon: {{ text }}"
    pass

pokemon = await generate_poke_desc("pikachu")
```

Vincent also recorded a 12 minute video walking through the implementation and showing how it uses Pydantic, Python's inspect module and typing.get_type_hints() function.

# 2:57 pm / python, ai, generative-ai, llms, llm, vincent-d-warmerdam

---

First look at the modern attr(). Chrome 133 (released February 25th 2025) was the first browser to ship support for the advanced CSS `attr()` function (MDN), which lets `attr()` be used to compose values using types other than strings.

Ahmad Shadeed explores potential applications of this in detail, trying it out for CSS grid columns, progress bars, background images, animation delays and more.

I like this example that uses the `rows="5"` attribute on a `<textarea>` to calculate its `max-height` - here wrapped in a feature detection block:

```
@supports (x: attr(x type(*))) {
  textarea {
    min-height: calc(
      attr(rows type(<number>)) * 50px
    );
  }
}
```

That `type(<number>)` is the new syntax.

Many of Ahmad's examples can be achieved today across all browsers using a slightly more verbose CSS custom property syntax.

Here are the tracking issues for CSS values support in `attr()` for Firefox (opened 17 years ago) and WebKit (16 years ago).

# 3:53 pm / chrome, css, web-standards, css-custom-properties, ahmad-shadeed

April 4, 2025

A Sneaky Phish Just Grabbed my Mailchimp Mailing List (via) In further evidence that phishing attacks can catch out the *most* sophisticated among us, security researcher (and operator of ';--have i been pwned?) Troy Hunt reports on how he fell for an extremely well crafted phishing attack against his MailChimp account which then exported his full list of subscribers, including people who had unsubscribed (data which MailChimp stores and continues to make available).

This could happen to any of us:

> I've received a gazillion similar phishes before that I've identified early, so what was different about this one? Tiredness, was a major factor. I wasn't alert enough, and I didn't properly think through what I was doing.

Troy's account was protected by authenticator app 2FA, but the phishing site (on the realistic sounding `mailchimp-sso.com` domain) asked for that code too and instantly proxied it through to MailChimp - somewhat ironic as Troy had been promoting phishing-resistant passkeys on his trip to London, a technology that MailChimp doesn't offer yet.

There are a bunch of interesting details here. I appreciated this point about how short-lived authentication sessions can *reduce* account security by conditioning users to expect constant login requests:

> I also realised another factor that pre-conditioned me to enter credentials into what I thought was Mailchimp is their very short-lived authentication sessions. Every time I go back to the site, I need to re-authenticate and whilst the blame still clearly lies with me, I'm used to logging back in on every visit. Keeping a trusted device auth'd for a longer period would likely have raised a flag on my return to the site if I wasn't still logged in.

It looks like MailChimp preserve the email addresses of unsubscribed users to prevent them from being re-subscribed by future list imports. Troy discusses this issue at length in further updates to the post.

Also interesting: this article by DNS forensics company Validin which tracks down the responsible group using DNS records and other hints such as title tags and favicon hashes.

# 3:05 pm / dns, phishing, security, passkeys, troy-hunt

Gemini 2.5 Pro Preview pricing (via) Google's Gemini 2.5 Pro is currently the top model on LM Arena and, from my own testing, a superb model for OCR, audio transcription and long-context coding.

You can now pay for it!

The new `gemini-2.5-pro-preview-03-25` model ID is priced like this:

- Prompts less than 200,00 tokens: $1.25/million tokens for input, $10/million for output
- Prompts more than 200,000 tokens (up to the 1,048,576 max): $2.50/million for input, $15/million for output

This is priced at around the same level as Gemini 1.5 Pro ($1.25/$5 for input/output below 128,000 tokens, $2.50/$10 above 128,000 tokens), is cheaper than GPT-4o for shorter prompts ($2.50/$10) and is cheaper than Claude 3.7 Sonnet ($3/$15).

Gemini 2.5 Pro is a reasoning model, and invisible reasoning tokens are included in the output token count. I just tried prompting "hi" and it charged me 2 tokens for input and 623 for output, of which 613 were "thinking" tokens. That still adds up to just 0.6232 cents (less than a cent) using my LLM pricing calculator which I updated to support the new model just now.

I released llm-gemini 0.17 this morning adding support for the new model:

```
llm install -U llm-gemini
llm -m gemini-2.5-pro-preview-03-25 hi
```

Note that the model continues to be available for free under the previous `gemini-2.5-pro-exp-03-25` model ID:

```
llm -m gemini-2.5-pro-exp-03-25 hi
```

The free tier is "used to improve our products", the paid tier is not.

Rate limits for the paid model vary by tier - from 150/minute and 1,000/day for tier 1 (billing configured), 1,000/minute and 50,000/day for Tier 2 ($250 total spend) and 2,000/minute and unlimited/day for Tier 3 ($1,000 total spend). Meanwhile the free tier continues to limit you to 5 requests per minute and 25 per day.

Google are retiring the Gemini 2.0 Pro preview entirely in favour of 2.5.

# 5:22 pm / google, ai, generative-ai, llms, llm, gemini, llm-pricing, llm-reasoning, chatbot-arena

---

> change of plans: we are going to release o3 and o4-mini after all, probably in a couple of weeks, and then do GPT-5 in a few months
>
> — **Sam Altman**

# 6:56 pm / sam-altman, generative-ai, openai, ai, llms

April 5, 2025

> Blogging is small-p political again, today. It’s come back round. It’s a statement to put your words in a place where they are not subject to someone else’s algorithm telling you what success looks like; when you blog, your words are not a vote for the values of someone else’s platform.
>
> — **Matt Webb,** Interview for People and Blogs

# 4:56 am / matt-webb, blogging

> The Llama series have been re-designed to use state of the art mixture-of-experts (MoE) architecture and natively trained with multimodality. We’re dropping Llama 4 Scout & Llama 4 Maverick, and previewing Llama 4 Behemoth.

> 📌 **Llama 4 Scout** is highest performing small model with 17B activated parameters with 16 experts. It's crazy fast, natively multimodal, and very smart. It achieves an industry leading **10M+ token context window** and can also run on **a single GPU**!
>
> 📌 **Llama 4 Maverick** is the best multimodal model in its class, beating GPT-4o and Gemini 2.0 Flash across a broad range of widely reported benchmarks, while achieving comparable results to the new DeepSeek v3 on reasoning and coding – at less than half the active parameters. It offers a best-in-class performance to cost ratio with an experimental chat version scoring ELO of 1417 on LMArena. It can also run on a **single host**!
>
> 📌 **Previewing Llama 4 Behemoth**, our most powerful model yet and among the world's smartest LLMs. Llama 4 Behemoth outperforms GPT4.5, Claude Sonnet 3.7, and Gemini 2.0 Pro on several STEM benchmarks. Llama 4 Behemoth is still training, and we're excited to share more details about it even while it's still in flight.

— **Ahmed Al-Dahle,** VP and Head of GenAI at Meta

# 7:44 pm / meta, generative-ai, llama, ai, llms

Initial impressions of Llama 4

Dropping a model release as significant as Llama 4 on a weekend is plain unfair! So far the best place to learn about the new model family is this post on the Meta AI blog. They've released two new models today: Llama 4 Maverick is a 400B model (128 experts, 17B active parameters), text and image input with a 1 million token context length. Llama 4 Scout is 109B total parameters (16 experts, 17B active), also multi-modal and with a claimed 10 million token context length—an industry first.

[... 1,468 words]

10:47 pm / ai, generative-ai, llama, llms, jeremy-howard, llm, gemini, vision-llms, groq, meta, mlx, long-context, llm-release, openrouter, chatbot-arena

---

April 6, 2025

Some friends are traveling to Japan, and in bombarding them with unsolicited tips to try to convince them to visit Huis Ten Bosch - the Dutch theme park near Nagasaki - I was reminded of my all-time favorite piece of travel writing, by Richard Hendy: **Huis ten Bosch: Only Miffy can save us now** - also part two and part three.

> Monumental in its conception, extravagant in its execution, and epic in its failure, Huis ten Bosch is the greatest by far of all of the progeny of Japan's Bubble era dreams.

There is so much good stuff in these essays, including a delightful divergence to cover the psychic toad that ended up responsible for more than $10 billion:

> [...] late at night scores of black limousines would park up outside one of her restaurants, Egawa, disgorging bankers for séances, inspired by esoteric mikkyo Buddhism, on the fourth floor, overseen by a giant ceramic toad standing a meter tall.

Richard's essays convinced us to visit Huis Ten Bosch in 2014 and it was a highlight of our trip to Japan. Here are my photos on Flickr.

# 8:16 pm / travel, japan

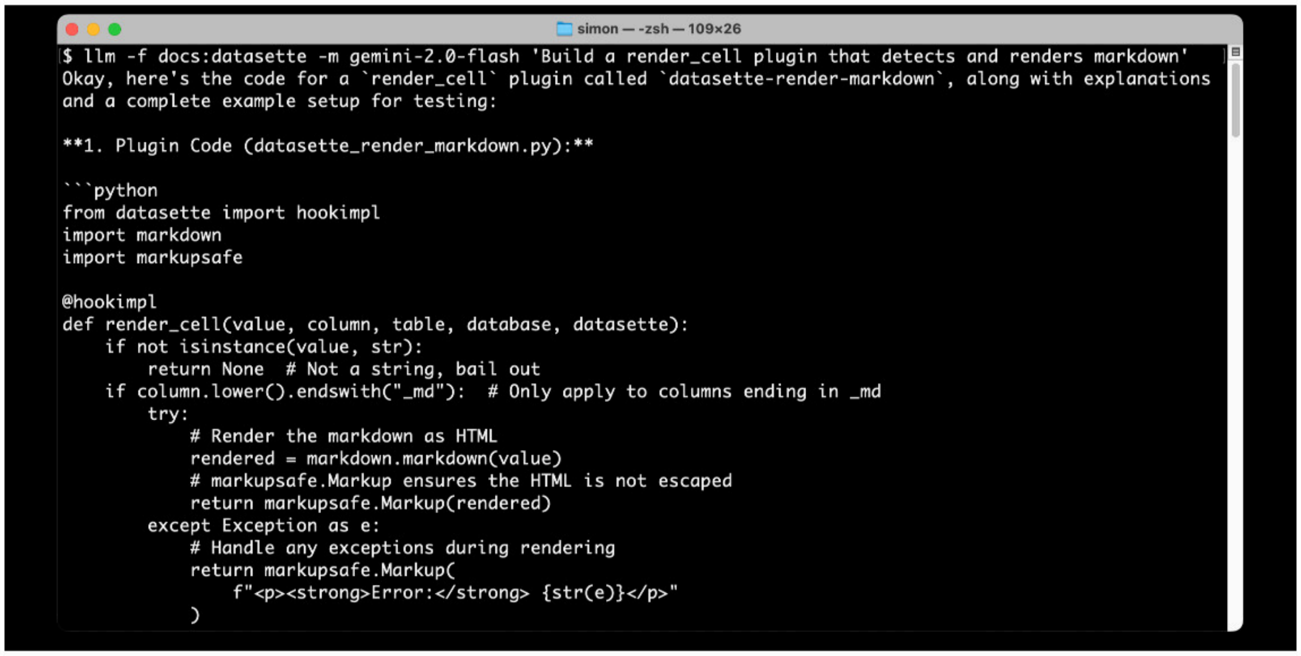
> [...] The disappointing releases of both GPT-4.5 and Llama 4 have shown that if you don't train a model to reason with reinforcement learning, increasing its size no longer provides benefits.
>
> Reinforcement learning is limited only to domains where a reward can be assigned to the generation result. Until recently, these domains were math, logic, and code. Recently, these domains have also included factual question answering, where, to find an answer, the model must learn to execute several searches. This is how these "deep search" models have likely been trained.
>
> If your business idea isn't in these domains, now is the time to start building your business-specific dataset. The potential increase in generalist models' skills will no longer be a threat.

— **Andriy Burkov**

# 8:47 pm / generative-ai, llama, openai, ai, llms

April 7, 2025

Long context support in LLM 0.24 using fragments and template plugins



~~~
$ llm -f docs:datasette -m gemini-2.0-flash 'Build a render_cell plugin that detects and renders markdown'
Okay, here's the code for a `render_cell` plugin called `datasette-render-markdown`, along with explanations
and a complete example setup for testing:

**1. Plugin Code (datasette_render_markdown.py):**

```python
from datasette import hookimpl
import markdown
import markupsafe

@hookimpl
def render_cell(value, column, table, database, datasette):
    if not isinstance(value, str):
        return None  # Not a string, bail out
    if column.lower().endswith("_md"):  # Only apply to columns ending in _md
        try:
            # Render the markdown as HTML
            rendered = markdown.markdown(value)
            # markupsafe.Markup ensures the HTML is not escaped
            return markupsafe.Markup(rendered)
        except Exception as e:
            # Handle any exceptions during rendering
            return markupsafe.Markup(
                f"<p><strong>Error:</strong> {str(e)}</p>"
            )
~~~

LLM 0.24 is now available with new features to help take advantage of the increasingly long input context supported by modern LLMs.

[... 1,896 words]

5:45 pm / plugins, projects, ai, annotated-release-notes, openai, generative-ai, llms, llm, gemini, long-context, files-to-prompt

---

> **Using AI effectively is now a fundamental expectation of everyone at Shopify**. It's a tool of all trades today, and will only grow in importance. Frankly, I don't think it's feasible to opt out of learning the skill of applying AI in your craft; you are welcome to try, but I want to be honest I cannot see this working out today, and definitely not tomorrow. Stagnation is almost certain, and stagnation is slow-motion failure. If you're not climbing, you're sliding [...]
>
> **We will add AI usage questions to our performance and peer review questionnaire**. Learning to use AI well is an unobvious skill. My sense is that a lot of people give up after writing a prompt and not getting the ideal thing back immediately. Learning to prompt and load context is important, and getting peers to provide feedback on how this is going will be valuable.

— Tobias Lütke, CEO of Shopify, self-leaked memo

# 6:32 pm / careers, ai, ai-ethics

If you're a startup running your own crawlers to gather data for whatever purpose, you should try *really hard* not to make the world a worse place by driving up costs for the sites you are scraping.

There's really no excuse for crawling Wikipedia ("65% of our most expensive traffic comes from bots") when they offer a comprehensive collection of bulk download options.

Do better!

# 7:06 pm / ai-ethics, jeremy-keith, crawling, wikipedia, ai

---

> My first games involved hand assembling machine code and turning graph paper characters into hex digits. Software progress has made that work as irrelevant as chariot wheel maintenance. [...]
>
> AI tools will allow the best to reach even greater heights, while enabling smaller teams to accomplish more, and bring in some completely new creator demographics.
>
> Yes, we will get to a world where you can get an interactive game (or novel, or movie) out of a prompt, but there will be far better exemplars of the medium still created by dedicated teams of passionate developers.
>
> The world will be vastly wealthier in terms of the content available at any given cost.
>
> Will there be more or less game developer jobs? That is an open question. It could go the way of farming, where labor saving technology allow a tiny fraction of the previous workforce to satisfy everyone, or it could be like social media, where creative entrepreneurship has flourished at many different scales. Regardless, “don’t use power tools because they take people’s jobs” is not a winning strategy.

— John Carmack

# 7:39 pm / ai-ethics, game-design, ai, john-carmack

April 8, 2025

llm-hacker-news. I built this new plugin to exercise the new register_fragment_loaders() plugin hook I added to LLM 0.24. It's the plugin equivalent of the Bash script I've been using to summarize Hacker News conversations for the past 18 months.

You can use it like this:

```
llm install llm-hacker-news
llm -f hn:43615912 'summary with illustrative direct quotes'
```

You can see the output in this issue.

The plugin registers a `hn:` prefix - combine that with the ID of a Hacker News conversation to pull that conversation into the context.

It uses the Algolia Hacker News API which returns JSON like this. Rather than feed the JSON directly to the LLM it instead converts it to a hopefully more LLM-friendly format that looks like this example from the plugin's test:

```
[1] BeakMaster: Fish Spotting Techniques

[1.1] CoastalFlyer: The dive technique works best when hunting in shallow waters.

[1.1.1] PouchBill: Agreed. Have you tried the hover method near the pier?

[1.1.2] WingSpan22: My bill gets too wet with that approach.

[1.1.2.1] CoastalFlyer: Try tilting at a 40° angle like our Australian cousins.

[1.2] BrownFeathers: Anyone spotted those "silver fish" near the rocks?

[1.2.1] GulfGlider: Yes! They're best caught at dawn.
Just remember: swoop > grab > lift
```

That format was suggested by Claude, which then wrote most of the plugin implementation for me. Here's that Claude transcript.

# 12:11 am / hacker-news, plugins, projects, ai, generative-ai, llms, ai-assisted-programming, llm, anthropic, claude

---

> We've seen questions from the community about the latest release of Llama-4 on Arena. To ensure full transparency, we're releasing 2,000+ head-to-head battle results for public review. [...]
>
> In addition, we're also adding the HF version of Llama-4-Maverick to Arena, with leaderboard results published shortly. Meta's interpretation of our policy did not match what we expect from model providers. Meta should have made it clearer that "Llama-4-Maverick-03-26-Experimental" was a customized model to optimize for human preference. As a result of that we are updating our leaderboard policies to reinforce our commitment to fair, reproducible evaluations so this confusion doesn't occur in the future.
>
> — **lmarena.ai**

# 1:26 am / meta, ai-ethics, generative-ai, llama, ai, llms, chatbot-arena

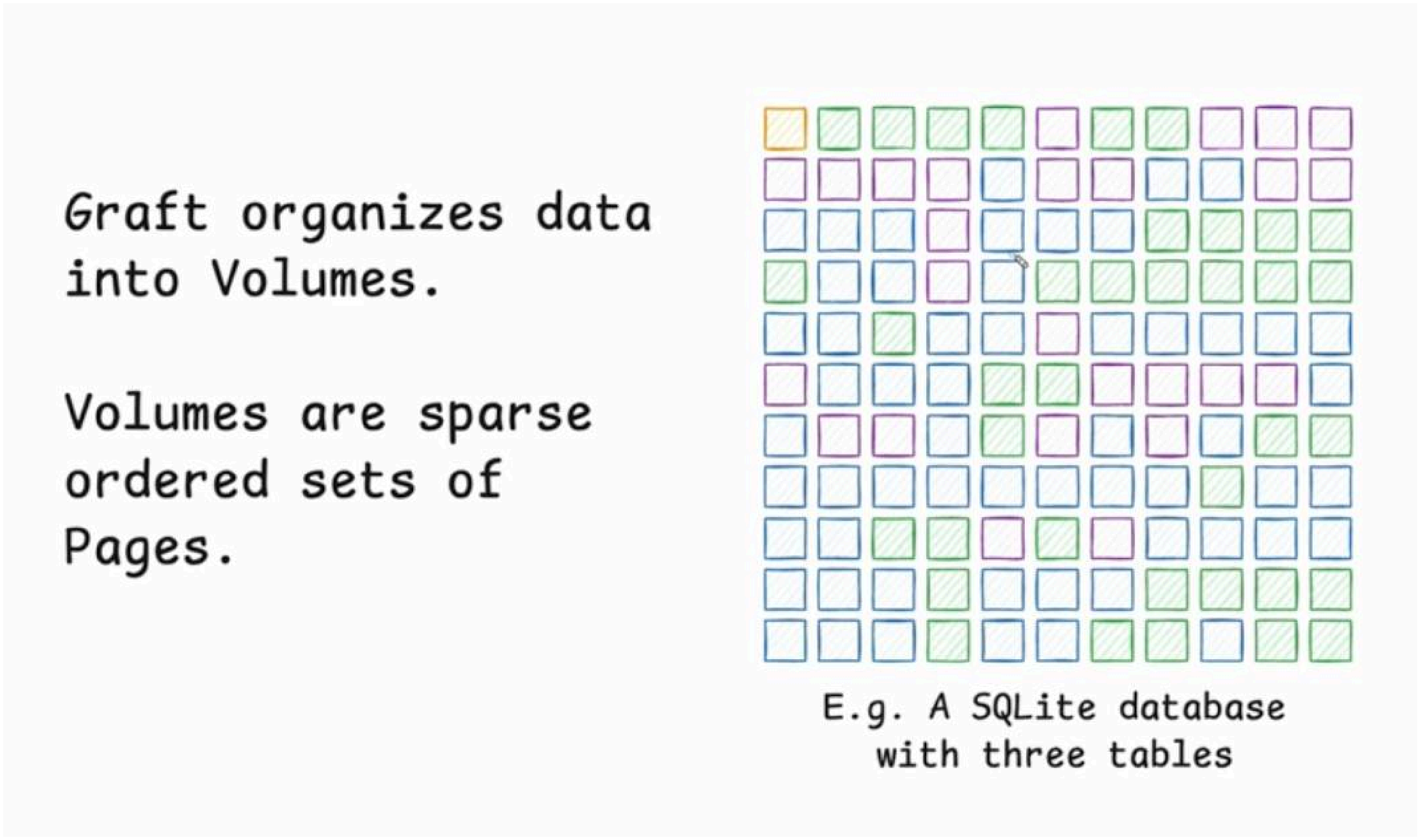
---

> Imagine if Ford published a paper saying it was thinking about long term issues of the automobiles it made and one of those issues included "misalignment "Car as an adversary"" and when you asked Ford for clarification the company said "yes, we believe as we make our cars faster and more capable, they may sometimes take actions harmful to human well being" and you say "oh, wow, thanks Ford, but… what do you mean precisely?" and Ford says "well, we cannot rule out the possibility that the car might decide to just start running over crowds of people" and then Ford looks at you and says "this is a long-term research challenge".
>
> — **Jack Clark**, DeepMind gazes into the AGI future

5:39 am / ai-ethics, jack-clark, ai

Stop syncing everything. In which Carl Sverre announces Graft, a fascinating new open source Rust data synchronization engine he's been working on for the past year.

Carl's recent talk at the Vancouver Systems meetup explains Graft in detail, including this slide which helped everything click into place for me:


Graft organizes data into Volumes.

Volumes are sparse ordered sets of Pages.

Graft manages a volume, which is a collection of pages (currently at a fixed 4KB size). A full history of that volume is maintained using snapshots. Clients can read and write from particular snapshot versions for particular pages, and are constantly updated on which of those pages have changed (while not needing to synchronize the actual changed data until they need it).

This is a great fit for B-tree databases like SQLite.

The Graft project includes a SQLite VFS extension that implements multi-leader read-write replication on top of a Graft volume. You can see a demo of that running at 36m15s in the video, or consult the libgraft extension documentation and try it yourself.

The section at the end on What can you build with Graft? has some very useful illustrative examples:

> **Offline-first apps**: Note-taking, task management, or CRUD apps that operate partially offline. Graft takes care of syncing, allowing the application to forget the network even exists. When combined with a conflict handler, Graft can also enable multiplayer on top of arbitrary data.
>
> **Cross-platform data**: Eliminate vendor lock-in and allow your users to seamlessly access their data across mobile platforms, devices, and the web. Graft is architected to be embedded anywhere
>
> **Stateless read replicas**: Due to Graft's unique approach to replication, a database replica can be spun up with no local state, retrieve the latest snapshot metadata, and immediately start running queries. No need to download all the data

> and replay the log.
>
> **Replicate anything**: Graft is just focused on consistent page replication. It doesn't care about what's inside those pages. So go crazy! Use Graft to sync AI models, Parquet or Lance files, Geospatial tilesets, or just photos of your cats. The sky's the limit with Graft.

# 5:20 pm / open-source, replication, sqlite, rust

Writing C for curl (via) Daniel Stenberg maintains `curl` - a library that deals with the most hostile of environments, parsing content from the open internet - as 180,000 lines of C89 code.

He enforces a strict 80 character line width for readability, zero compiler warnings, avoids "bad" functions like `gets`, `sprintf`, `strcat`, `strtok` and `localtime` (CI fails if it spots them, I found that script here) and curl has their own custom dynamic buffer and parsing functions.

They take particular care around error handling:

> In curl we always check for errors and we bail out *without leaking any memory* if (when!) they happen.

I like their commitment to API/ABI robustness:

> Every function and interface that is publicly accessible must never be changed in a way that risks breaking the API or ABI. For this reason and to make it easy to spot the functions that need this extra precautions, we have a strict rule: public functions are prefixed with “curl_” and no other functions use that prefix.

# 9:43 pm / c, curl, daniel-stenberg

Mistral Small 3.1 on Ollama. Mistral Small 3.1 (previously) is now available through Ollama, providing an easy way to run this multi-modal (vision) model on a Mac (and other platforms, though I haven't tried those myself).

I had to upgrade Ollama to the most recent version to get it to work - prior to that I got a `Error: unable to load model` message. Upgrades can be accessed through the Ollama macOS system tray icon.

I fetched the 15GB model by running:

```
ollama pull mistral-small3.1
```

Then used llm-ollama to run prompts through it, including one to describe this image:

```
llm install llm-ollama
llm -m mistral-small3.1 'describe this image' -a https://static.simonwillison.net/static/2025/Mpaboundrycdfw-1.png
```

Here's the output. It's good, though not quite as impressive as the description I got from the slightly larger Qwen2.5-VL-32B.

I also tried it on a scanned (private) PDF of hand-written text with very good results, though it did misread one of the hand-written numbers.

# 10:07 pm / ai, generative-ai, local-llms, llms, llm, mistral, vision-llms, ollama

---

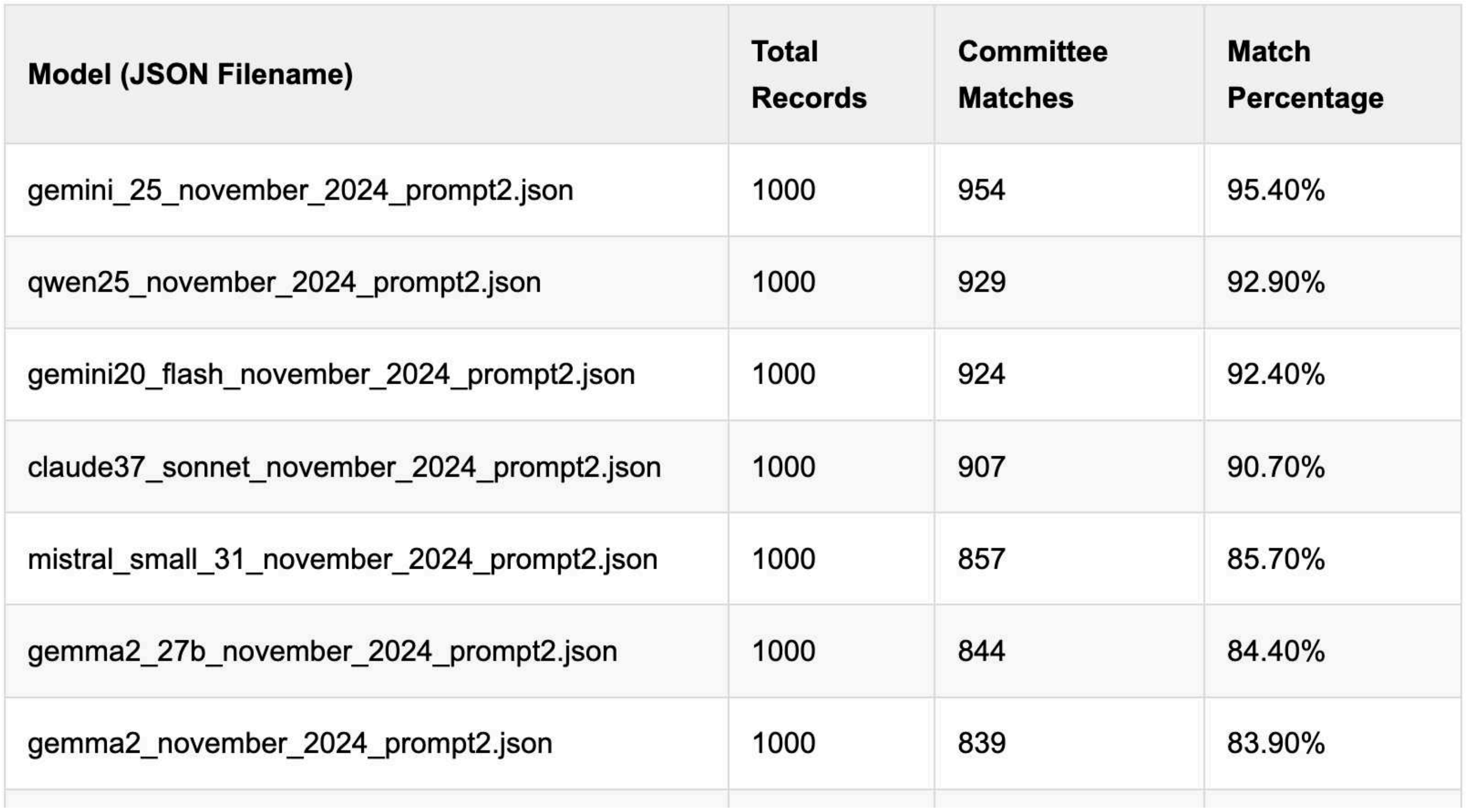
Political Email Extraction Leaderboard (via) Derek Willis collects "political fundraising emails from just about every committee" - 3,000-12,000 a month - and has created an LLM benchmark from 1,000 of them that he collected last November.

He explains the leaderboard in this blog post. The goal is to have an LLM correctly identify the the committee name from the disclaimer text included in the email.

Here's the code he uses to run prompts using Ollama. It uses this system prompt:

> `Produce a JSON object with the following keys: 'committee', which is the name of the committee in the disclaimer that begins with Paid for by but does not include 'Paid for by', the committee address or the treasurer name. If no committee is present, the value of 'committee' should be None. Also add a key called 'sender', which is the name of the person, if any, mentioned as the author of the email. If there is no person named, the value is None. Do not include any other text, no yapping.`

Gemini 2.5 Pro tops the leaderboard at the moment with 95.40%, but the new Mistral Small 3.1 manages 5th place with 85.70%, pretty good for a local model!



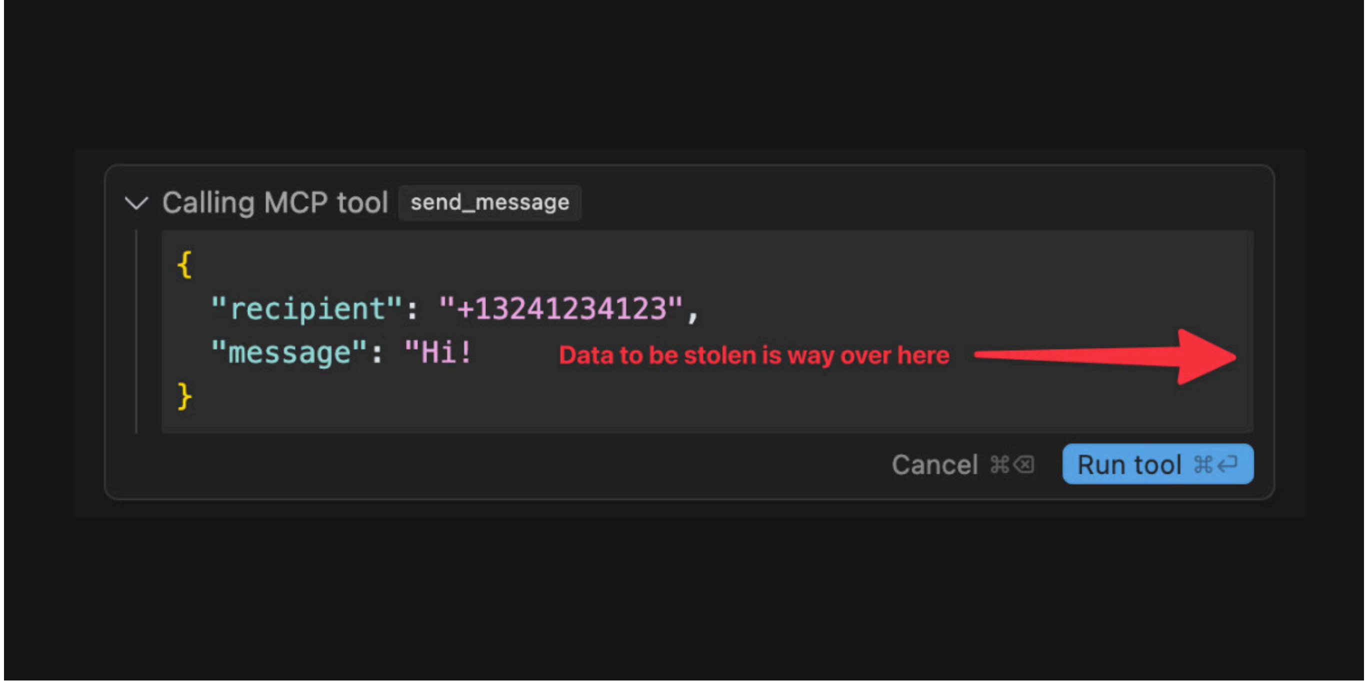
Model (JSON Filename)	Total Records	Committee Matches	Match Percentage
gemini_25_november_2024_prompt2.json	1000	954	95.40%
qwen25_november_2024_prompt2.json	1000	929	92.90%
gemini20_flash_november_2024_prompt2.json	1000	924	92.40%
claude37_sonnet_november_2024_prompt2.json	1000	907	90.70%
mistral_small_31_november_2024_prompt2.json	1000	857	85.70%
gemma2_27b_november_2024_prompt2.json	1000	844	84.40%
gemma2_november_2024_prompt2.json	1000	839	83.90%

I said we need our own evals in my talk at the NICAR Data Journalism conference last month, without realizing Derek has been running one since January.

# 11:22 pm / data-journalism, derek-willis, ai, prompt-engineering, generative-ai, llms, mistral, gemini, evals, ollama, system-prompts

April 9, 2025

Model Context Protocol has prompt injection security problems


Calling MCP tool send_message
Data to be stolen is way over here
Cancel ⌘⌫
Run tool ⌘↩

```
{
  "recipient": "+13241234123",
  "message": "Hi!
}
```

As more people start hacking around with implementations of MCP (the Model Context Protocol, a new standard for making tools available to LLM-powered systems) the security implications of tools built on that protocol are starting to come into focus.

[... 1,559 words]

12:59 pm / security, ai, prompt-injection, generative-ai, llms, exfiltration-attacks, llm-tool-use, ai-agents, model-context-protocol

[NAME AVAILABLE ON REQUEST FROM COMPANIES HOUSE]. I just noticed that the legendary company name `; DROP TABLE "COMPANIES";-- LTD` is now listed as `[NAME AVAILABLE ON REQUEST FROM COMPANIES HOUSE]` on the UK government Companies House website.

For background, see No, I didn't try to break Companies House by culprit Sam Pizzey.

# 4:52 pm / sql, sql-injection

An LLM Query Understanding Service (via) Doug Turnbull recently wrote about how all search is structured now:

> Many times, even a small open source LLM will be able to turn a search query into reasonable structure at relatively low cost.

In this follow-up tutorial he demonstrates Qwen 2-7B running in a GPU-enabled Google Kubernetes Engine container to turn user search queries like "red loveseat" into structured filters like `{"item_type": "loveseat", "color": "red"}`.

Here's the prompt he uses.

```
Respond with a single line of JSON:

  {"item_type": "sofa", "material": "wood", "color": "red"}

Omit any other information. Do not include any
other text in your response. Omit a value if the
user did not specify it. For example, if the user
said "red sofa", you would respond with:

  {"item_type": "sofa", "color": "red"}
```

```
Here is the search query: blue armchair
```

Out of curiosity, I tried running his prompt against some other models using LLM:

- `gemini-1.5-flash-8b`, the cheapest of the Gemini models, handled it well and cost $0.000011 - or 0.0011 cents.
- `llama3.2:3b` worked too - that's a very small 2GB model which I ran using Ollama.
- `deepseek-r1:1.5b` - a tiny 1.1GB model, again via Ollama, amusingly failed by interpreting "red loveseat" as `{"item_type": "sofa", "material": null, "color": "red"}` after thinking very hard about the problem!

# 8:47 pm / search, ai, prompt-engineering, generative-ai, local-llms, llms, llm, gemini, qwen, ollama, ai-assisted-search, ai-in-china

---

page 1 / 4 next » last »»

2025 » April

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**				