Local LLMs - Why, what, how?

Gergő Bocsárdi

2025-02-18

Outline

1. Why?

1. Privacy
2. Efficiency
3. Ethics

2. What?

1. Models
2. Compute

3. How?

1. Ollama
2. Open-WebUI

WHY?

Privacy

- Sometimes you don’t want Sam Altman to get your data
- Company private info, personal data (subjects or yourself)

Efficiency

- Local hosting is free! (minus the compute costs, if applicable)
- Pick the size of model for the task
  - Smaller but faster model might work for your usecase
- Endless customization
  - From prompts to parameters, it’s all open

Ethics

- Scam Altman
- Amazonian forest
- Nuclear reactor data centers
- Theft

WHAT?

How it started

TECH / AI

Meta’s powerful AI language model has leaked online — what happens now?

Illustration: Alex Castro / The Verge

/ Meta’s LLaMA model was created to help researchers but leaked on 4chan a week after it was announced. Some worry the technology will be used for harm; others say greater access will improve AI safety.

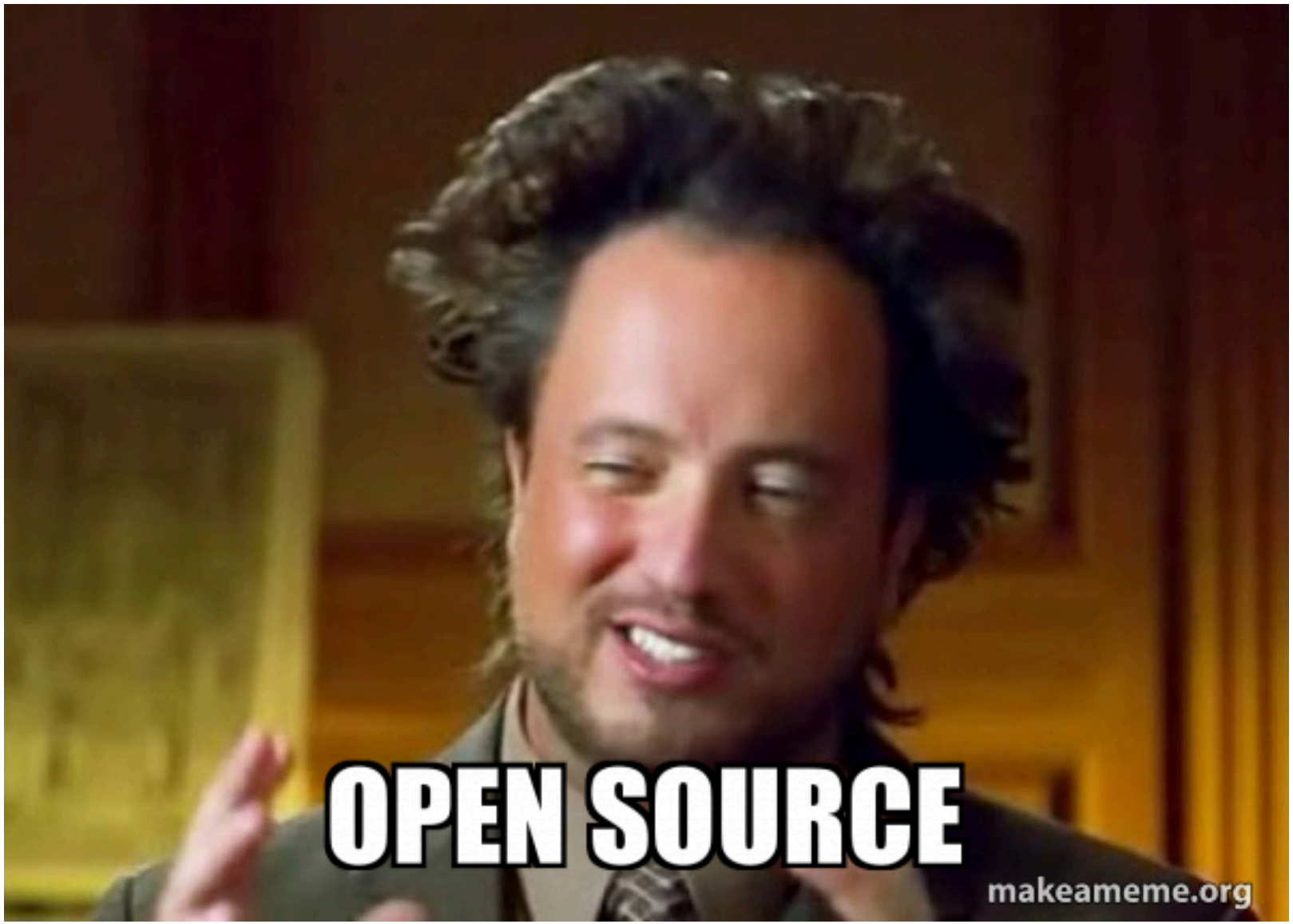
by **James Vincent**
Mar 8, 2023, 2:15 PM GMT+1

4 Comments (4 New)

If you buy something from a Verge link, Vox Media may earn a commission. See our ethics statement.

source

Solution?



Open Source LLMs

- Huge community
- Lots of options
- Constant new things rolling out

Factors to consider

- Model size:
  - how smart it is (inversely proportional to speed, just like humans)
- Context length:
  - how much info can it “remember”?
- Quantization:
  - how much it is “compressed”

About compute

- Two bottlenecks
  - Memory (RAM/VRAM): model size restrictions
    - You can run as many B-s as GB-s of RAM you have
  - CPU/GPU: inference speed restrictions

So a lot of RAM without a GPU will give you great answers from a large model, very slowly.

Model selection

- Pick the right tool for the job
  - Conversation/reasoning has different needs to Fill-in-the-Middle code completion
- Consider what you’re priorities are
  - For RAG I use a small model with a large context window
  - For editing text, a large model with a small window

Quantization?

- Specifies at which precision the model parameters are stored.
  - `f16` = “half precision”, 16bit floats
  - `q_8` = 8bit floats, very little performance hit compared to `f16`
  - …
  - `q_4` = 4bit floats, noticeable performance hit, but half the size of `f16`
    - this is the Ollama default, keep it in mind

Opinionated Primer on Models

- My daily driver: Phi-4, at q_8
- Best Small Model: Phi-3.5
  - Runner up: Llama 3.2
- For code-completion:
  - qwen2.5-coder
  - starcoder-2

HOW?

Tooling

- Ollama
  - Manage and run local language models
- Open-WebUI
  - Amazing interface to interact with models

Instructions

1. install ollama
2. download appropriate model (start small)
    1. run `ollama pull phi3.5`
3. make sure ollama is running
4. set up open-webui
    1. easiest is using `uv` https://docs.astral.sh/uv/
    2. run: `uvx --python 3.11 open-webui@latest serve`
5. go to `http://localhost:8080/` and voila!

What can you do?

- Standard chat interface
  - Even has the code interpreter
- But also RAG

RAG demo

- Using the Knowledge Feature of OpenWebUI
- Here's a tutorial
- Pro tip: use MarkItDown
  - If you have `uv` installed, you can just do `uvx markitdown example.docs` against any file

Nerd stuff

- Ollama environment variables
  - set `OLLAMA_FLASH_ATTENTION` to 1
  - set `OLLAMA_KV_CACHE_TYPE` to q8_0

Both of these will make the context use less memory, without meaningful hits to performance. Read this blogpost to learn more about this!

Resources

- RTFM – all of these tools come with amazing docs, use it!
- Simon Willison
  - his blog
  - absolute legend, one of the best people covering ai stuff
  - he does all sort of social media too (X, Bluesky, etc)