Emacs and Private AI

December 6, 2025

By Aaron Grothe
AlphaWall LLC

Introduction

Slides will be available on my website at https://grothe.us in the presentations section tonight or tomorrow

This is a quick intro to one way to do Private AI and Emacs, there are a lot of other ways to do this.

Overview of Talk

- Why Private AI?
- What do I need for Private AI?
- Emacs and Private AI
- Pieces for an AI Emacs Solution
- Demo - Minimum Viable Product
- Summary

Why Private AI?

Quite simply read the Terms & Conditions for any AI system

If you're using the free tiers. Your queries/code/uploaded information is being used to train the models. In some case you are giving the company's a perpetual license to your data

If you're using the paid tiers. You may be able to opt out of the data. Keep in mind this can change, or they may charge for that

Your costs vary. Right now the services are being heavily subsidized. When they start charging their real costs plus profit it is going to change

What do I need for Private AI?

If you're going to run your own AI you're going to need a system with either some cores, a GPU, or an NPU.

I currently have 4 systems I'm experimenting with

- System 76 Pangolin AMD Ryzen 7 7840u with Radeon 780M integrated graphics
- HP Z620 Intel Xeons with 4 Nvidia K2200 graphics cards
- Macbook Air with M1 processor
- Acer Aspire One with an AMD Ryzen 5700H in it

Using the Pangolin for this demonstration

What do I need for Private AI?

- Apple's M4 chip has 38 teraflops of NPU performance
- Microsoft Co-Pilot PCs require a minimum of 45 teraflops of NPU
- Raspberry PI's new AI top - is about 18 teraflops, and is $70 on top of the cost of a Raspberry Pi 5

There is going to be a lot of local power available in the future.

Emacs and Private AI

There are a couple popular solutions

- Gptel - simple interface, minimal interface integrate easily to workflow
- Ollama Buddy - more full featured, menu interface, has quick actions for things like code refactoring, text reformatting, etc.
- Ellama - provides more full featured interface, has more capabilities
- Aidermac - pair programming with your AI in Emacs. Most like Cursor in Emacs
- There are some others as well

Pieces for an AI Emacs Solution

A minimum viable product can be done with just two pieces of software

- Llamafile - this is a whole LLM contained in one file. The same files runs on Mac OS X, Linux, Windows and the BSDs
- GPtel - Plugin for Emacs

Pieces for an AI Emacs Solution

Setting up the LLM

$ wget https://huggingface.co/Mozilla/Llama-3.2-1B-Instruct-llamafile/blob/main/Llama-3.2-1B-Instruct.Q6_K.llamafile?download=true

$ chmod +x Llama-3.2-1B-Instruct.Q6_K.llamafile

$ ./Llama-3.2-1B-Instruct.Q6_K.llamafile –server

Pieces for an AI Emacs Solution

Install GPT tel into emacs

$ emacs

M-x install-package<RET>
gptel<RET>

Pieces for an AI Emacs Solution

Setup config file

$ emacs ~/.emac/config.el

```
(require 'gptel) ; Ensure gptel is loaded

;; Define the llamafile backend
(setq gptel-llamafile-backend
      (gptel-make-openai
       "Llamafile-Local"    ; A name for your backend
       :stream t            ; Enable streaming responses
       :protocol "http"     ; The protocol used by the local
server
```

Pieces for an AI Emacs Solution

Config file continued

```
      :host "127.0.0.1:8080" ; Host and port where llamafile is running
      :key nil               ; No API key is needed for a local server
      :models '("llama-model-name") ; A list of model names. It can be a dummy name like "llama-model" if the server doesn't require a specific one.
      ))

;; Optionally, set it as the default backend
(setq gptel-backend gptel-llamafile-backend)
```

Pieces for an AI Emacs Solution

Time for a Quick test

$ emacs

M-x gptel who was David Bowie?

More In-Depth Solution

Pieces for a Better Solution

- Ollama or LM Studio - More capable than LLamafile, can accept a lot of different models
- Open WebUI - provides - More capabilities including Retrieval-Augmented Generation (RAG)
- Vector Database alternative to RAG - ChromaDB, PostgreSQL with pgvector, etc.
- GPtel - Emacs interface or one of the alternatives, such as Ollama Buddy

What about the license?

I am not a lawyer

Couple of highlights from Meta Llama 3 Community license

- Restrictions:
  - Scale: If your service exceeds 700 million monthly users, you need additional licensing.
  - Competition: You can't use the model to enhance competing models.
  - Trademark: Limited use of Meta's trademarks.

What about the license?

I am not a lawyer

Couple of highlights from Meta Llama 3 Community license

- Permissive License: It's designed to encourage innovation and open development.
- Commercial Use: Allowed, but with some restrictions.
- Modifications: You can modify the model, but you still need to abide by the license terms.
- Distribution: You can distribute the model or its derivatives.

What about the license?

Licenses vary depending on the model you're using.

Mistral is dual licensed with a commercial and the Mistral AI Non-Production License (MNPL)

Mistral AI Non-Production License (MNPL)

- Purpose: Primarily for research and development purposes.
- Restrictions: Commercial use is prohibited.
- Focus: Encourages innovation and research without the pressure of immediate commercialization.

Commercial license is based upon number of users, etc.

Are There Open Source Data Model options?

Yes, but most of them are small or specialized currently.

Molmo is a family of Open Source AI models

Molmo doesn't have as many data points as most other models. The difference is that they they use human validated data to try and improve the system.

Not currently available via ollama. Hopefully, it will be soon.

Available under the MIT license :-)

Things To Know

Tips to Help you have a better Experience

- Get ollama and Open-Webui working by themselves then setup your gptel config file
- Llamafile/Gptel is a great way to start experimenting
- Use RAG to load documents, information into your LLM environment, easy with Open-Webui
- Read Hacker Noon's - How to Build a $300 AI computer https://hackernoon.com/how-to-build-a-$300-ai-computer-for-the-gpu-poor
- You don't need a great GPU or CPU to get started. Smaller models like tinyllama can run on very small systems

Things To Know

Tips to Help you have a better Experience

- The AMD Ryzen™ AI Max+ 395 in the mini-pc format makes a nice dedicated AI host
- Macs work very well with AI - thermal throttling can be an issue
- Alex Ziskind on Youtube has a channel that does a lot of AI performance benchmarking
- Small Domain-specific LMs are happening. Imagine an LM that has all of your code, and information in it
- Huggingface.co has a lot of AI models available, find the one that works for you
- Hallucinations are real

Summary & Thanks

Summary

AI can extend and improve your Emacs experience. Give it a spin.

Thanks for Listening

Links

Local LLM Engines

Ollama

- https://ollama.co

LM Studio

- https://lmstudio.ai

Links

Llamafile

- https://github.com/mozilla-ai/llamafile

Hugging Face

- https://huggingface.co/

Links

LLM Web interface

Open Webui

- https://openwebui.com/

Links

Emacs AI Interfaces

GPtel

- https://github.com/karthink/gptel

Ellama

- https://github.com/s-kostyaev/ellama

Links

Emacs AI Interfaces

Aidermacs

- https://github.com/MatthewZMD/aidermacs

Ollama Buddy

- https://github.com/captainflasmr/ollama-buddy
- https://www.dyerdwelling.family/emacs/20250207092636-emacs--ollama-buddy-local-llm-integration-for-emacs/

Links

Alex Ziskind's Youtube Channel

- https://www.youtube.com/@AZisk