CERN openlab

Evaluation of Open Source Large Language Models for Industrial Control

Uses Cases

Siria Sadeddin

August 12, 2024

BE Industrial Control Systems

Accelerating Science…

…with GPUs

Presentation Roadmap

1. **Project Objectives**
2. **Why Open Source?**
3. **Navigating the Options: Model Selection**
4. **Achieving Efficiency**
5. **Future Directions**

CERN openlab

Project Objectives

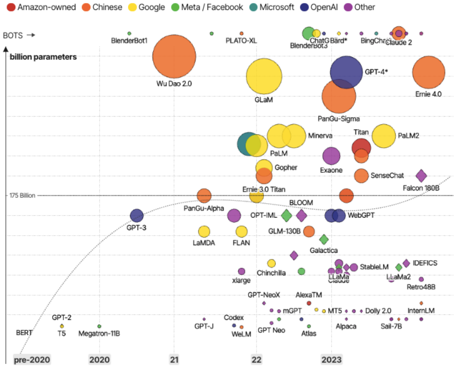
Identify the best open-source code-summarization model that is resource-efficient and can solve domain specific tasks.

- **Evaluate Model Performance**
  - Compare the performance of various open-source large language models on the HumanEval dataset.
  - *Metrics:* summarization accuracy and computational efficiency.
- **Develop Comparative Analysis**
  - Conduct a comprehensive analysis comparing the efficacy of different models on a dataset of industrial control code.
- **Fine-tune Models**
  - Adapt Llama 3.1 8B and other models using internal documentation data.

Why Open Source?

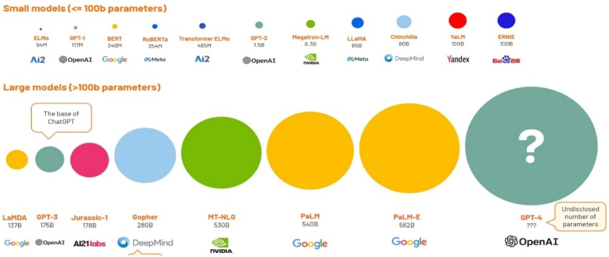
What is a Large Language Model?

A Large Language Model is a type of artificial intelligence (AI) that can read, understand, and write text in a way that resembles human language.

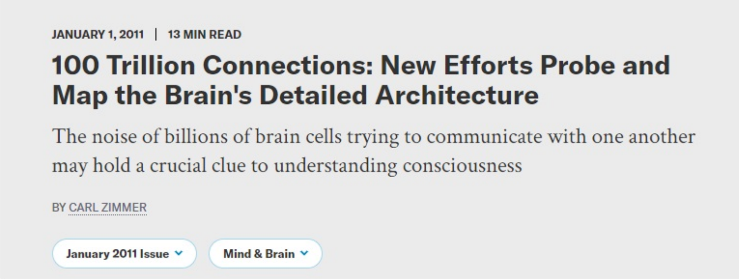


David McCandless, Tom Evans, Paul Barton
Information is Beautiful // UPDATED 2nd Nov 23

source: news reports, LifeArchitect.ai
* = parameters undisclosed // see the data



100 Trillion Connections!



JANUARY 1, 2011 | 13 MIN READ

100 Trillion Connections: New Efforts Probe and Map the Brain's Detailed Architecture

The noise of billions of brain cells trying to communicate with one another may hold a crucial clue to understanding consciousness

BY CARL ZIMMER

Figure: "100 Trillion Connections: New Efforts Probe and Map the Brain's Detailed Architecture" in Scientific American Magazine Vol. 304 No. 1 (January 2011)

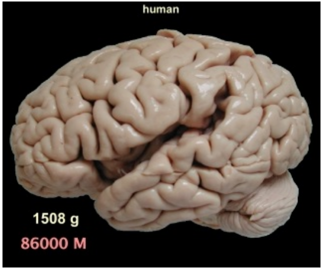


Figure: Herculano-Houzel S. The human brain in numbers: a linearly scaled-up primate brain. Front Hum Neurosci. 2009 Nov 9

Navigating the Options: Model Selection

Open-Source Large Language Models for Code Summarization

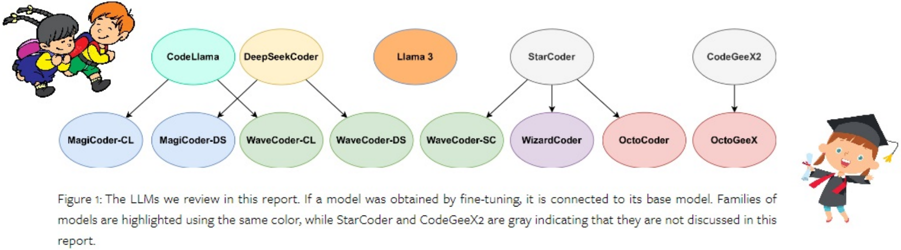
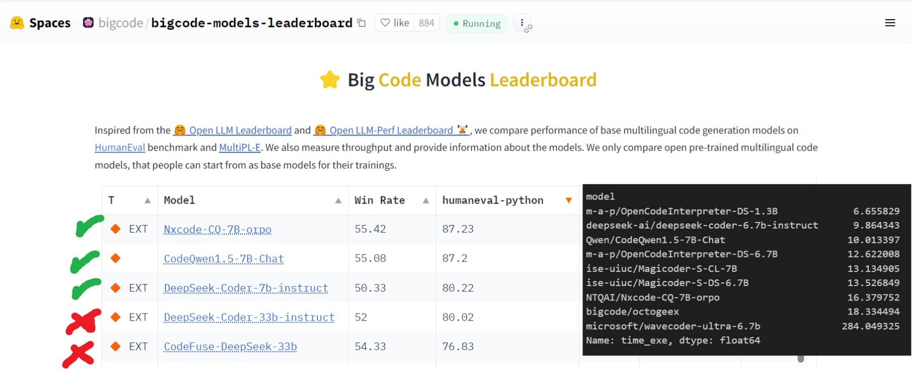


Figure 1: The LLMs we review in this report. If a model was obtained by fine-tuning, it is connected to its base model. Families of models are highlighted using the same color, while StarCoder and CodeGeeX2 are gray indicating that they are not discussed in this report.

Figure: Large Language Models for Code Summarization, arXiv:2405.19032v1

Model selection



🤗 Spaces bigcode / **bigcode-models-leaderboard** ♡ like 884 ● Running

⭐ Big Code Models Leaderboard

Inspired from the 🤗 Open LLM Leaderboard and 🤗 Open LLM-Perf Leaderboard 🏋️, we compare performance of base multilingual code generation models on HumanEval benchmark and MultiPL-E. We also measure throughput and provide information about the models. We only compare open pre-trained multilingual code models, that people can start from as base models for their trainings.

	T	Model	Win Rate	humaneval-python
✓	◆ EXT	Nxcode-CQ-7B-orpo	55.42	87.23
✓	◆	CodeQwen1.5-7B-Chat	55.08	87.2
✓	◆ EXT	DeepSeek-Coder-7b-instruct	50.33	80.22
✗	◆ EXT	DeepSeek-Coder-33b-instruct	52	80.02
✗	◆ EXT	CodeFuse-DeepSeek-33b	54.33	76.83

```
model
m-a-p/OpenCodeInterpreter-DS-1.3B              6.655829
deepseek-ai/deepseek-coder-6.7b-instruct       9.864343
Qwen/CodeQwen1.5-7B-Chat                      10.013397
m-a-p/OpenCodeInterpreter-DS-6.7B             12.622008
ise-uiuc/Magicoder-S-CL-7B                    13.134905
ise-uiuc/Magicoder-S-DS-6.7B                  13.526849
NTQAI/Nxcode-CQ-7B-orpo                       16.379752
bigcode/octogeex                              18.334494
microsoft/wavecoder-ultra-6.7b               284.049325
Name: time_exe, dtype: float64
```

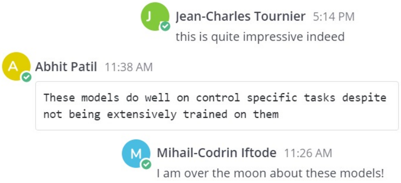
Figure: HuggingFace spaces

Achieving Efficiency

Model Performance on Industrial Control Code

We ran the selected models on industrial control code.

- **C++:**
  - *Prompt:* Give a detailed summary of the following C++ code
  - *Summary:* This code is a part of a LaserManager class which is a part of a larger system ...
- **CTRL**
  - *Prompt:* Give a detailed summary of the following CTRL code
  - *Summary:* This function is part of a larger system that manages the import of device data from a front-end system into a UniCos system ...



Jean-Charles Tournier 5:14 PM
this is quite impressive indeed

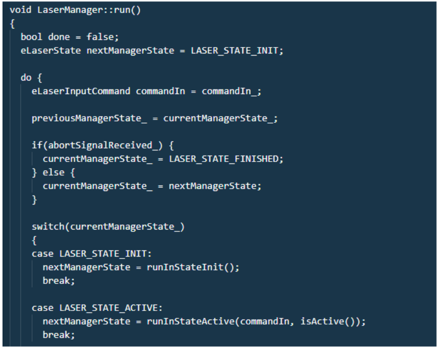
Abhit Patil 11:38 AM

```
These models do well on control specific tasks despite
not being extensively trained on them
```

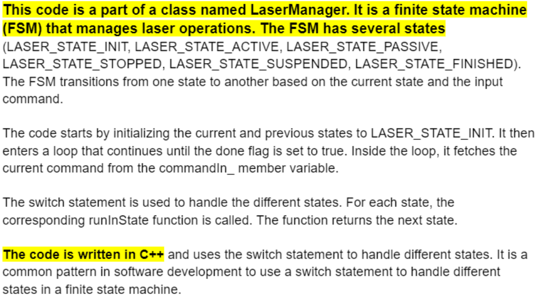
Mihail-Codrin Iftode 11:26 AM
I am over the moon about these models!

Summarizing a C++ code for laser management

Give a detailed summary of the following C++ code:



```
void LaserManager::run()
{
  bool done = false;
  eLaserState nextManagerState = LASER_STATE_INIT;

  do {
    eLaserInputCommand commandIn = commandIn_;

    previousManagerState_ = currentManagerState_;

    if(abortSignalReceived_) {
      currentManagerState_ = LASER_STATE_FINISHED;
    } else {
      currentManagerState_ = nextManagerState;
    }

    switch(currentManagerState_)
    {
    case LASER_STATE_INIT:
      nextManagerState = runInStateInit();
      break;

    case LASER_STATE_ACTIVE:
      nextManagerState = runInStateActive(commandIn, isActive());
      break;
```



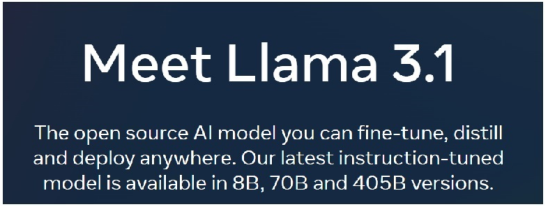
This code is a part of a class named LaserManager. It is a finite state machine (FSM) that manages laser operations. The FSM has several states (LASER_STATE_INIT, LASER_STATE_ACTIVE, LASER_STATE_PASSIVE, LASER_STATE_STOPPED, LASER_STATE_SUSPENDED, LASER_STATE_FINISHED). The FSM transitions from one state to another based on the current state and the input command.

The code starts by initializing the current and previous states to LASER_STATE_INIT. It then enters a loop that continues until the done flag is set to true. Inside the loop, it fetches the current command from the commandIn_ member variable.

The switch statement is used to handle the different states. For each state, the corresponding runInState function is called. The function returns the next state.

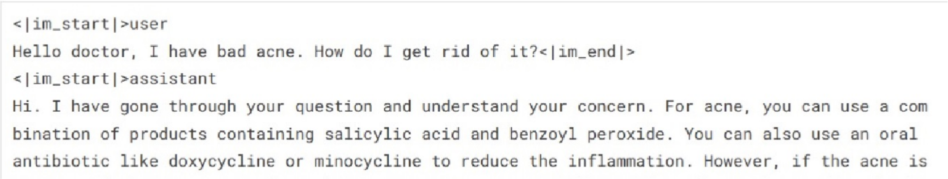
The code is written in C++ and uses the switch statement to handle different states. It is a common pattern in software development to use a switch statement to handle different states in a finite state machine.

Fine-tuning LLama 3.1 8B



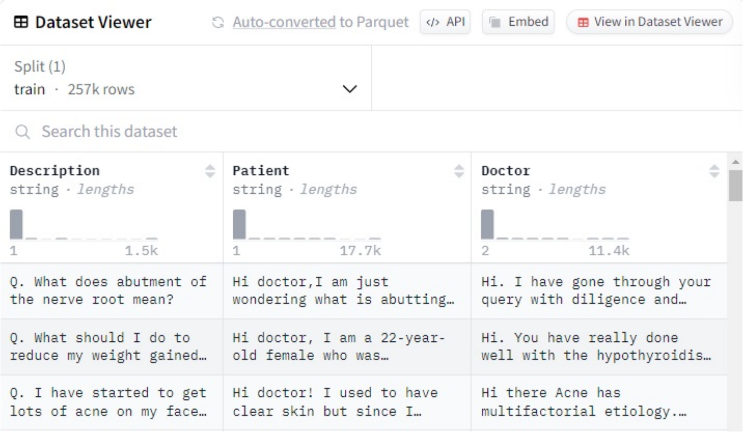
Meet Llama 3.1

The open source AI model you can fine-tune, distill and deploy anywhere. Our latest instruction-tuned model is available in 8B, 70B and 405B versions.



```
<|im_start|>user
Hello doctor, I have bad acne. How do I get rid of it?<|im_end|>
<|im_start|>assistant
Hi. I have gone through your question and understand your concern. For acne, you can use a com
bination of products containing salicylic acid and benzoyl peroxide. You can also use an oral
antibiotic like doxycycline or minocycline to reduce the inflammation. However, if the acne is
```

Data Structure



Description	Patient	Doctor
string · *lengths*	string · *lengths*	string · *lengths*
1 1.5k	1 17.7k	2 11.4k
Q. What does abutment of the nerve root mean?	Hi doctor,I am just wondering what is abutting…	Hi. I have gone through your query with diligence and…
Q. What should I do to reduce my weight gained…	Hi doctor, I am a 22-year-old female who was…	Hi. You have really done well with the hypothyroidis…
Q. I have started to get lots of acne on my face…	Hi doctor! I used to have clear skin but since I…	Hi there Acne has multifactorial etiology.…

CERN openlab

Future Directions

- Make a detailed comparative analysis of the code summarization models over a dataset consisting of Industrial Control code.
- Fine-tune models like Llama 3.1 8B over internal documentation data.

CERN openlab

Thank you!