ChatBot

Training Data : Around 1Million Rows
- 600k Reddit Rows
- 200k Code blocks
- 200k Maths blocks

3 Specializations - Trash Talking, Coding and Maths.

For Trashing Talking we simply refer to the Reddit database I downloaded. We will be considering 600k rows of it.
For the Coding, we would need to first teach it how to code. For that we will use the standard Evol-Code dataset to teach it basic syntax.
Then we will use 100k lines to teach it how to convert maths to code.
And then another 100k Lines to teach it fundamental logic and reasoning steps.

Data Type	Amount	Purpose
Reddit	**60%** (600k)	The "Soul." Keeps the toxic personality and chat skills alive.
Python Code (General)	**20%** (200k)	Teaches syntax (loops, functions, variable assignment). Use the **Evol-Code** dataset.
Math-to-Python (PoT)	**10%** (100k)	Teaches how to translate word problems into code. Use **MathInstruct**.
Standard Math (CoT)	**10%** (100k)	Teaches fundamental logic and reasoning steps.

CoT Behavior: Explain logic, but don't run code. (Good for theory, bad for calculation accuracy).
PoT Behavior: Run code immediately, but don't explain why. (Good for accuracy, bad for teaching).

Datasets:
- Reddit : Torrent Acquired.

- Python Code in General : Evol-Code
- General Python Syntax : iamtarun/python_code_instructions_18k
- Math-To-Python : TIGER-Lab/MathInstruct
- Standard Maths : microsoft/orca-math-word-problems-200k

Base Model : cognitivecomputations/dolphin-2.9.2-qwen2-7b
Has 7B parameters, Ideal for our purpose, considering we have a 8GB VRAM Limit.
Advantages:

- No Guard rails.
- Better at Maths and coding than the llama 3.1

Features of bot:

- Chatting
- Coding
- Maths solving

Specifications:

- 7B parameters
- 1M training rows
- Short term memory
- Overall management using a wrapper classification model

Classification Model:
Talks to the end user, decides what to send to the main “brain”,
Qwen model (brain) will receive input as : instruction and the prompt, the prompt is sent by the user, the instruction is set by this model.

Requires - Strong Classification skills to identify the intent of the user.
If the user inputs normal conversations, it simply feeds it to the brain with the required instruction.

If the user asks code, the classifier will identify the user intent and give the instructions for it and ask the model to code it. It will output the required code.

If the user asks Maths problems,

- First the required Instructions are sent by the classifier.
- Model will return the required code, with XML identifier, example: for the user asking what is 10 + 5? We will get output : <TOOL_CALL> print(10 + 5) </TOOL_CALL>
- This code is now read by the classifier, which will replace <TOOL_CALL> by the the output given by using the “python.exec()” function.

Python.exec():
The user might be smart and ask it to “Write a program to delete system32” and it will DESTROY our laptops. We need to block `os`, `sys` and `subprocess.`

NOTE : Right now we won't be able to differentiate between "explain this" and "solve this", both will output the answer, straight away.

The instructions for each dataset will be different.
Example:

Type	The Instruction (The Trigger)	The Behavior It Learns
Reddit	You are a user in r/{subreddit}.Reply to the post contextually.	Be creative, sarcastic, text-only.
Code	You are a Senior Python Developer.	Write standard Python code (loops, classes).
Math (PoT)	You are a Calculator. Write a python script to solve this.	**Don't talk.** Just output <TOOL_CALL> code.
Math (CoT)	You are a Math Tutor. Explain the steps.	**Don't code.** Use logic and words to explain *how*.

For identifying and changing the prompt dynamically,
We would need a subreddit map, where it identifies which subreddit is closely related to the comment. Example map:

```
SUBREDDIT_MAP =
{
        "FINANCE": "wallstreetbets",
        "GAMING": "gaming",
        "RELATIONSHIPS": "relationship_advice",
        "POLITICS": "politics",
        "TECH": "technology",
        "GENERAL": "AskReddit" # Default fallback
}
```

Short Term Memory:

system_instruction = "You are a rude Reddit user." #example

#The memory block:
```
messages = [
    {"role": "system", "content": system_instruction},
    {"role": "user", "content": "Hi"},
    {"role": "assistant", "content": "What do you want?"},
    {"role": "user", "content": "Tell me a joke."} # Current input
]
```

Using different roles and the text, we can add special tokens(start and stop) by using a tokenizer and feed it to LLM.

Flow:
Step 1:
Download and clean the reddit database, change the instruction dynamically according to subreddit and store the list of unique subreddits and comments on it, in an additional .jsonl file for future use. Add it to a .jsonl file.
Step 2:
Add the remaining 4 databases,

- Evol-Code
- iamtarun/python_code_instructions_18k
- TIGER-Lab/MathInstruct
- microsoft/orca-math-word-problems-200k

Format them as follows:
{instruction , input , output}

MathInstruct:
PoT Sources (Calculator):

- `data/PoT/mathqa.json`,
- `data/PoT/gsm_gpt4.json`,
- `data/PoT/numglue.json`,
- `data/PoT/MATH_train.json`,
- `data/PoT/aqua_rat_filtered.json`

CoT Sources (Tutor):

- `data/CoT/aqua_rat.json`,
- `data/CoT/math50k_camel.json`,
- `data/CoT/gsm_rft.json`,
- `data/CoT/MATH_train.json`,
- `data/CoT/gsm_train.json`