```
from swarm_models import Agent
from swarm_models.prompts.finance_agent_sys_prompt import (
    FINANCIAL_AGENT_SYS_PROMPT,
)
import torch
from swarm_models import BaseLLM
from transformers import AutoTokenizer, LlamaForCausalLM


class NvidiaLlama31B(BaseLLM):
    # Load the tokenizer and model
    def __init__(self, max_tokens: int = 2048):
        self.max_tokens = max_tokens
        model_path = "nvidia/Llama-3.1-Minitron-4B-Width-Base"
        self.tokenizer = AutoTokenizer.from_pretrained(model_path)

        device = "cuda"
        dtype = torch.bfloat16
        self.model = LlamaForCausalLM.from_pretrained(
            model_path, torch_dtype=dtype, device_map=device
        )

    def run(self, task: str):
        # Prepare the input text
        inputs = self.tokenizer.encode(task, return_tensors="pt").to(
            self.model.device
        )

        # Generate the output
        outputs = self.model.generate(
            inputs, max_length=self.max_tokens
        )

        # Decode and print the output
        output_text = self.tokenizer.decode(outputs[0])
        print(output_text)

        return output_text


# # Example usage:
# model = NvidiaLlama31B()
# out = model.run("What is the essence of quantum field theory?")
# print(out)

model = NvidiaLlama31B()

# Initialize the agent
agent = Agent(
    agent_name="Financial-Analysis-Agent_sas_chicken_eej",
    system_prompt=FINANCIAL_AGENT_SYS_PROMPT,
    llm=model,
    max_loops=2,
    autosave=True,
    dashboard=False,
    verbose=True,
    dynamic_temperature_enabled=True,
    saved_state_path="finance_agent.json",
    user_name="swarms_corp",
    retry_attempts=1,
    context_length=200000,
    return_step_meta=True,
    disable_print_every_step=True,
    output_type="json",
)


out = agent.run(
    "How can I establish a ROTH IRA to buy stocks and get a tax break? What are the criteria"
)
print(out)
```