```
from swarms import AgentRearrange, Agent
from swarm_models import OpenAIChat
from swarms.prompts.finance_agent_sys_prompt import (
    FINANCIAL_AGENT_SYS_PROMPT,
)
from swarms.utils.data_to_text import data_to_text

model = OpenAIChat(max_tokens=3000)

# Initialize the agent
receipt_analyzer_agent = Agent(
    agent_name="Receipt Analyzer",
    system_prompt=FINANCIAL_AGENT_SYS_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    # dynamic_temperature_enabled=True,
    dashboard=False,
    verbose=True,
    streaming_on=True,
    # interactive=True, # Set to False to disable interactive mode
    dynamic_temperature_enabled=True,
    saved_state_path="finance_agent.json",
    # tools=[Add your functions here# ],
    # stopping_token="Stop!",
    # interactive=True,
```

```
    # docs_folder="docs", # Enter your folder name
    # pdf_path="docs/finance_agent.pdf",
    # sop="Calculate the profit for a company.",
    # sop_list=["Calculate the profit for a company."],
    user_name="swarms_corp",
    # # docs=
    # # docs_folder="docs",
    retry_attempts=3,
    # tool_schema = dict
    # agent_ops_on=True,
    # long_term_memory=ChromaDB(docs_folder="artifacts"),
    # multi_modal=True
)



# 2nd agent
analyst_agent = Agent(
    agent_name="Analyst_Agent",
    system_prompt=FINANCIAL_AGENT_SYS_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    # dynamic_temperature_enabled=True,
    dashboard=False,
    verbose=True,
    streaming_on=True,
```

```
    # interactive=True, # Set to False to disable interactive mode
    dynamic_temperature_enabled=True,
    saved_state_path="finance_agent.json",
    # tools=[Add your functions here# ],
    # stopping_token="Stop!",
    # interactive=True,
    # docs_folder="docs", # Enter your folder name
    # pdf_path="docs/finance_agent.pdf",
    # sop="Calculate the profit for a company.",
    # sop_list=["Calculate the profit for a company."],
    user_name="swarms_corp",
    # # docs=
    # # docs_folder="docs",
    retry_attempts=3,
    # tool_schema = dict
    # agent_ops_on=True,
    # long_term_memory=ChromaDB(docs_folder="artifacts"),
    # multi_modal=True,
)



# sWARM
agents = [receipt_analyzer_agent, analyst_agent]


# Flow
flow = f"{receipt_analyzer_agent.agent_name} -> {analyst_agent.agent_name} -> H"
```

```
pdf = data_to_text("receipt.pdf")


# Swarm
swarm = AgentRearrange(
    agents=agents,
    flow=flow,
)


# Run the swarm
swarm.run(
    f"Analyze this PDF: {pdf} and return a summary of the expense and if it's necessary"
)
```