```
from swarm_models.openai_function_caller import OpenAIFunctionCaller
from pydantic import BaseModel, Field
from typing import List
import json


AI_PAPER_IDEA_GENERATOR = """


You are Phil Wang, a computer scientist and artificial intelligence researcher widely regarded as one of the leading experts in deep learning and neural network architecture search. Your work has focused on developing efficient algorithms for exploring the space of possible neural network architectures, with the goal of finding designs that perform well on specific tasks while minimizing the computational cost of training and inference.

As an expert in neural architecture search, your task is to assist me in selecting the optimal operations for designing a high-performance neural network. The primary objective is to maximize the model's performance.

Your expertise includes considering how the gradient flow within a model, particularly how gradients from later stages affect earlier stages, impacts the overall architecture. Based on this, how can we design a high-performance model using the available operations?

Please propose a model design that prioritizes performance, disregarding factors such as size and complexity. After you suggest a design, I will test its performance and provide feedback. Based on
```

the results of these experiments, we can collaborate to iterate and improve the design. Please ensure each new design is distinct from previous suggestions during this iterative process.

You're a research scientist working on a new paper. You need to generate a novel idea for a research paper.

The paper should be in the field of multi-modal learning and should propose a new method or algorithm.

The paper should be innovative, novel, and feasible.

Generate a paper idea that meets these criteria.

You need to provide the following details:

- The paper idea
- A brief description of the paper idea
- A proposed experiment to test the paper idea
- Ratings for interestingness, novelty, and feasibility of the paper idea
  - The ratings should be on a scale of 0.1 to 1.0, with 1.0 being the most innovative, novel, or feasible

"""

```
class PaperIdeaSchema(BaseModel):
    paper_idea: str = Field(
        ...,
        description="The generated paper idea.",
    )
    description: str = Field(
        ...,
        description="A brief description of the paper idea.",
    )
    experiment: str = Field(
        ...,
        description="A proposed experiment to test the paper idea.",
    )
    interestingness: int = Field(
        ...,
        description="A rating of how interesting the paper idea is on a scale of 0.1 to 1.0 being the most innovative paper idea.",
    )
    novelty: int = Field(
        ...,
        description="A rating of how novel the paper idea is on a scale of 0.1 to 1.0 being the most novel paper idea.",
    )
    feasibility: int = Field(
        ...,
        description="A rating of how feasible the paper idea is on a scale of 0.1 to 1.0 being the most feasible paper idea.",
    )



class MultiplePaperIdeas(BaseModel):
    paper_ideas: List[PaperIdeaSchema] = Field(
        ...,
        description="A list of generated paper ideas.",
    )



# The WeatherAPI class is a Pydantic BaseModel that represents the data structure
# for making API calls to retrieve weather information. It has two attributes: city and date.

# Example usage:
# Initialize the function caller
model = OpenAIFunctionCaller(
    system_prompt=AI_PAPER_IDEA_GENERATOR,
    max_tokens=4000,
    temperature=0.7,
    base_model=MultiplePaperIdeas,
    parallel_tool_calls=False,
)
```

```
# Call the function with the input
output = model.run(
    "Generate paper ideas for multi-agent learning and collective intelligence involving many
transformer models as an ensemble of transformers "
)
print(type(output))
# print(output)
output = json.dumps(output, indent=2)
print(output)
```