

《AI Agent 开发与应用：基于大模型的智能体构建》一书

旧模型迁移到新模型的说明

据当前最新消息（截止 2025 年 8 月 1 日），从 2025 年 3 月起，即 OpenAI 发布 GPT-4.1 后，OpenAI 等厂商宣布将逐步弃用 text-davinci-003 等旧版本模型，现在已经统一迁移至可支持函数调用（Function Calling）、多轮上下文保持、Tool 调用的新一代模型上了，比如：

- (1) OpenAI GPT-4 (gpt-4-0613, gpt-4-0125-preview)
- (2) OpenAI GPT-3.5 (gpt-3.5-turbo-0613, gpt-3.5-turbo-0125)

为了帮助读者顺利将书中基于 text-davinci-003 的实现迁移至新模型上，现整理如下迁移指南。

(1) 旧版示例（OpenAI text-davinci-003 调用）：

```
import openai

response = openai.Completion.create(
    engine="text-davinci-003", (其实就是这里以及下方的“prompt”有问题，其他都不需要修改！)
    prompt="请用中文解释什么是 AI Agent",
    max_tokens=1000,
    temperature=0.7
)
print(response["choices"][0]["text"])
```

新版推荐（OpenAI gpt-4 或 gpt-3.5-turbo 调用）：

```
import openai

response = openai.ChatCompletion.create(
    model="gpt-4-0613",
    messages=[
        {"role": "system", "content": "你是一位 AI Agent 专家"},
        {"role": "user", "content": "请用中文解释什么是 AI Agent"}
    ],
    temperature=0.7
)
print(response["choices"][0]["message"]["content"])
```

主要区别在于，新模型使用 ChatCompletion 替代旧的 Completion 接口，使用多轮对话结

构（messages）组织对话，并且支持内置函数调用、工具集成等高级功能。

（2）函数式工具集成与 Action 调用：

新版模型支持函数调用(Function Calling)与工具集成（Tool Calling），可用于构建结构化 Agent，如任务调度、插件调用、参数控制等：

新版函数调用示例（OpenAI gpt-4-0613）：

```
response = openai.ChatCompletion.create(
    model="gpt-4-0613",
    messages=[{"role": "user", "content": "今天北京的天气怎么样?"}],
    functions=[
        {
            "name": "get_weather",
            "description": "查询城市天气",
            "parameters": {
                "type": "object",
                "properties": {
                    "location": {"type": "string", "description": "城市名"}
                },
                "required": ["location"]
            }
        }
    ],
    function_call="auto"
)
```

以上是为了说明改动原理的例子，具体怎么改看下面。

总结一下代码更新建议：

旧代码逻辑：

```
prompt = f"请帮我分解任务：{任务描述}"
response = openai.Completion.create(
    engine="text-davinci-003",
    prompt=prompt,
    max_tokens=800
)
```

推荐新写法（注意，对于 messages 来说，`{ }`和`f" "`格式均可以）：

```
messages = [
    {"role": "system", "content": "你是一位项目任务分析专家"},
    {"role": "user", "content": f"请帮我分解任务：{任务描述}"}
]
```

```
response = openai.ChatCompletion.create(  
    model="gpt-4-0125-preview",  
    messages=messages,  
    max_tokens=800  
)
```

总计一下就是，以前是 `prompt` 和 `text-davinci-003` 配合，现在是 `messages` 和新模型配合，改动地方不多，主要就是改模型名和交互结构（`prompt` 改成 `messages`，`completion` 改成 `Chatcompletion`），响应写法、其他核心参数都不用改，代码大部分逻辑不用动。

如需配合 `LangChain` 或 `LangGraph` 等构建结构化 Agent，可替换 OpenAI 模型配置如下：

```
from langchain.chat_models import ChatOpenAI  
  
llm = ChatOpenAI(  
    model_name="gpt-4",  
    temperature=0.7,  
    openai_api_key="your-key"  
)
```

如使用 `Claude` 或 `Gemini` 模型，请根据对应 SDK 替换调用方式与模型标识，与上面完全一致。