

源于社区 服务社区

 中国DevOps社区峰会 2024 · 上海



驾驭大模型开发真实项目代码

路宁



路宁

互联网大厂 高级技术总监、效能、AI应用

独立咨询师 敏捷开发、技术实践

ThoughtWorks 工程师、架构师



扫一扫上面的二维码图案，加我为朋友。

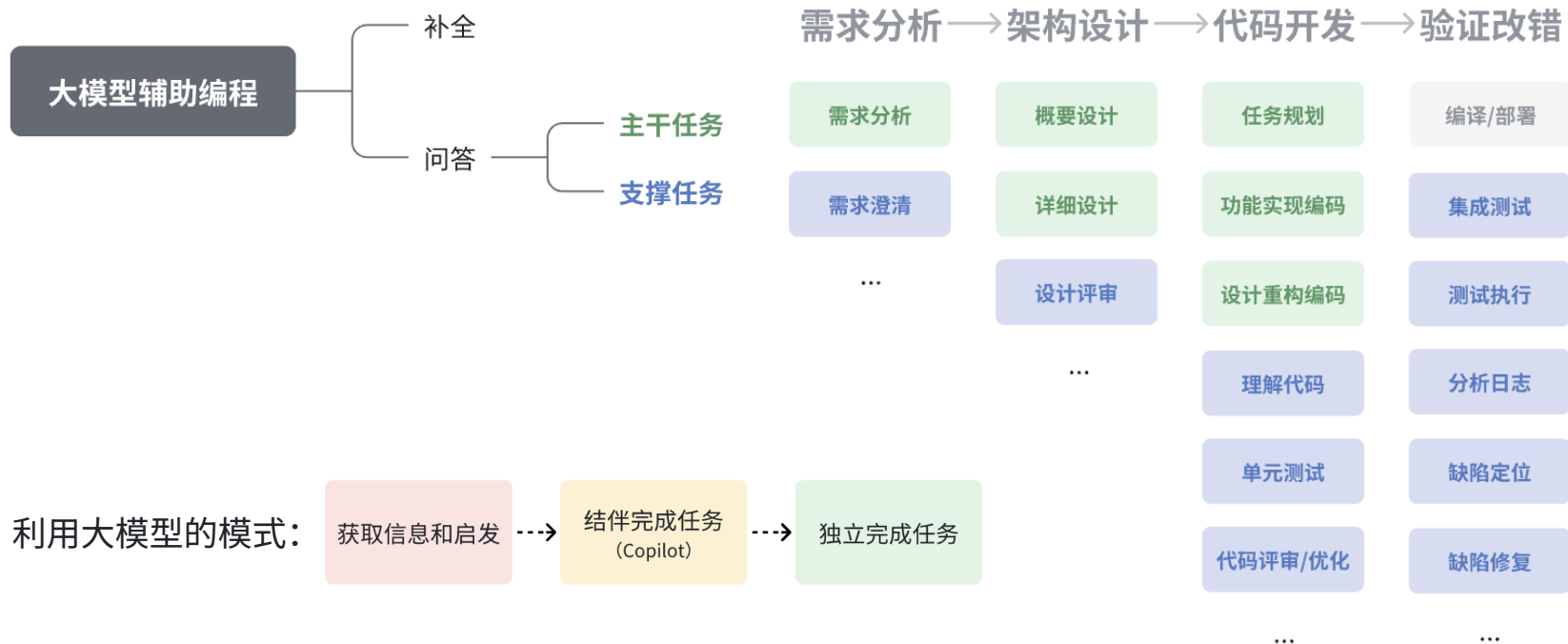


目录

- 1 大模型辅助开发生态
- 2 开发真实需求的挑战
- 3 切换到知识生产和消费的思维框架
- 4 依赖经验知识的编码任务实验
- 5 知识分类及提示词框架
- 6 大模型辅助开发的实用形态



1 大模型辅助开发生态 - 任务全貌





1 大模型辅助开发生态 - 主干任务相关应用现状

Devin, MetaGPT, ...

Github Copilot Workspace, Cursor, ...

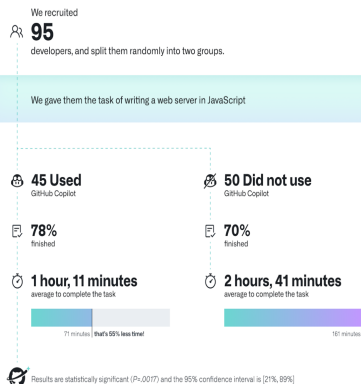
ChatGPT 4o with canvas, Claude workspace

手搓ChatGPT, o1, Claude



1 大模型辅助开发生态 - 了解测评背后的任务

Github Copilot的实验



- 任务是用JS开发一个Web Server
- 得出结论 - 效率提升55%

HumanEval评估数据

```
def encode_cyclic(s: str):
    """
    returns encoded string by cycling groups of three characters.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group. Unless group has fewer elements than 3.
    groups = [(group[i:] + group[0]) if len(group) == 3 else group for group in groups]
    return "".join(groups)

def decode_cyclic(s: str):
    """
    takes as input string encoded with encode_cyclic function. Returns decoded string.
    """
    # split string to groups. Each of length 3.
    groups = [s[(3 * i):min((3 * i + 3), len(s))] for i in range((len(s) + 2) // 3)]
    # cycle elements in each group.
    groups = [(group[-1] + group[:-1]) if len(group) == 3 else group for group in groups]
    return "".join(groups)
```

- 较为独立的函数级任务

SWE-Bench

Leaderboard

Lite	Verified	Full			
Model	% Resolved	Date	Logs	Trajs	Site
🏆 CodeStory Aide + Mixed Models	43.00	2024-07-02	🔗	-	🔗
🏆 AbanteAI MentatBot + GPT 4o (2024-05-13)	38.00	2024-06-27	🔗	-	🔗
🏆 Gru(2024-08-11)	35.67	2024-08-11	🔗	🔗	🔗
Bytedance MarsCode Agent + GPT 4o (2024-05-13)	34.00	2024-07-23	🔗	-	🔗
Alibaba Lingma Agent	33.00	2024-06-22	🔗	-	🔗
.....					
🏆✅ SWE-agent + Claude 3 Opus	11.67	2024-04-02	🔗	🔗	-
🏆✅ RAG + Claude 3 Opus	4.33	2024-04-02	🔗	-	🔗
🏆✅ RAG + Claude 2	3.00	2023-10-10	🔗	-	-

- Lite评估集有300个case, 18%仅1行改动, 30%在2行以内, 20+行改动的占7.3%。
- 被解决的问题集中在评估集中的简单任务。
- 项目都是纯python的库, 如django和pytest。

面试题 vs 实际项目任务



2 开发真实需求的挑战 - 看一个例子

- 用最好的模型手工实验才能得到最好的生成效果。
- 目录结构+代码文件+任务描述作为提示词的起点。
- 选个真实需求，人工做任务的规划。



- 实现步骤类似粗粒度的伪码，包含：
 - 改动入口或函数签名及输入输出格式：如何与现有代码集成。
 - 复用的约束：遵循设计约束，而非新写一套。
 - 新能力实现的提示：提示比较难的细节。
- 开发步骤提示的足够细生成的效果就足够好，但是...

```
1  **项目目录结构**
2  // 省略
3
4  **项目代码文件内容**
5  // 省略
6
7  **任务**
8  基于已有代码，增加端对端流式会话功能支持。
9
10 **参考经验**
11 - 主要功能往往通过Controller、业务层Service和远程接口访问Service三个组件实现。
12 - 可参考`TestSessionController`内的短连接方法逻辑，它首先调用`TestChatSessionService`，继而调用
    `TestChatGptService`，最后通过`TestChatGptService`发送请求到远程接口。
13
14 **实现步骤**：
15 1. 修改`pom.xml`，添加长连接所需依赖库，务必指定版本号。
16 2. 在`TestChatGptService`中，紧跟`handleChatResponse`方法之后，使用`Retrofit2`库来定义一个新方法
    sendMessageWithRetrofit，负责向远程接口发送消息的功能，响应交给调用方处理，sendMessageWithRetrofit的返回值应为Call类型。
17 3. 在`TestChatSessionService`类中，给出的`visitChatModel`方法之后，增加一个新方法`sessionDialogStream`，参考
    `sessionDialog`方法设置业务逻辑，然后调用`TestChatGptService`中的`sendMessageWithRetrofit`方法并将其响应直接返回给
    Controller层，即`sendMessageWithRetrofit`方法的返回值就是`sessionDialogStream`方法的返回值。
18 4. 新建一个Controller类，实现Websocket流式会话方法`streamChat`，此方法调用`TestChatSessionService`类中的
    `sessionDialogStream`方法。当响应成功时，将返回值发送给指定主题并打印返回的内容。同时设置配置类以支持Web浏览器端的流式通信。
19
20 **注意**
21 - 流式端点与非流式端点共享同一个基础URL（`azureProUrl`）。
22 - 流式请求URL需添加`stream=true`参数。
23 - 必须完整输出每一个方法的实现，永远不要只给出示例或注释。
24
25 本次只执行步骤1。
```

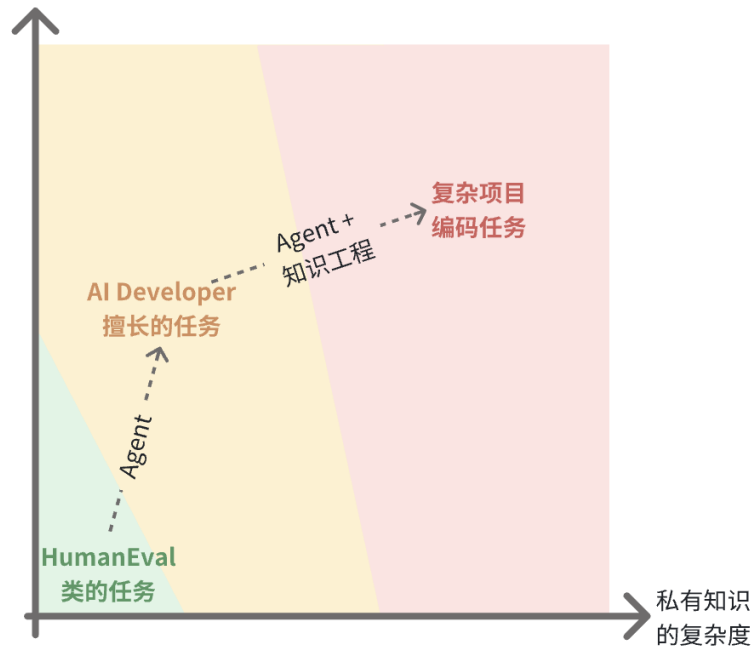


2 开发真实需求的挑战 - 分析任务难度

大模型能力：规划推理、指令遵从、窗口长度、注意力、输出倾向等

任务难度的体现	编码任务在这方面的特点
复杂任务的分解及规划难度	任务的规划分解利用模型或Agent来做效果一般，往往也不是痛点，可依赖人来做。
原子任务的理解和推理难度	任务封闭性高，推理难度极高。
所依赖可复用私有知识的复杂度	私有知识量大、隐晦，复用程度参差不齐，相关性有挑战，噪音影响明显。
为任务准备一次性信息的难度	描述和定义任务，包括实例化输入输出定义需求，实现方法和细节的提示，为提升效果展示任务特定的数据等，可能很繁琐。

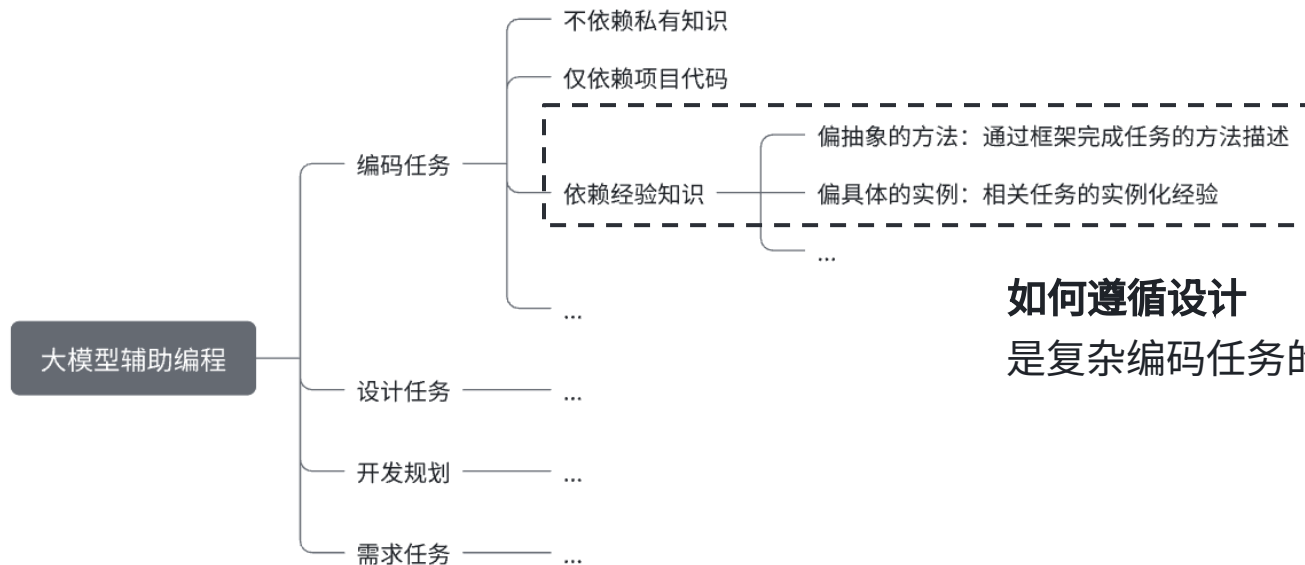
单次生成内容的复杂度







4 依赖经验知识的编码任务



如何遵循设计
是复杂编码任务的重要挑战



4 依赖经验知识的编码任务 - 利用"通过框架完成任务的方法描述"

经验

- 1 当需要“生成页面操作脚本”时，需遵循以下方法：
- 2 ★ 基于 `index.js` 中的 `executeJSFile` 函数搭建的框架，思考如何构建脚本。
- 3 ★ 使用 `context.driver` API 操作浏览器，它是对 `puppeteer` 的封装。
- 4 ★ 优先使用 `driver` 提供的内置函数完成任务。
- 5 ★ 如遇 `driver` 无法完成的操作，可提供其他方案。
- 6 ★ 若指明了现有的脚本文件，在其基础上进行修改和扩展。



新任务

- 1 我希望 生成poe的操作页面js，可以针对 `main footer textarea` 贴入外面传入文本，点击 `main footer button` 过滤出的第一个button开启一个新对话，然后点击第二个button上传指定path的文件，然后点击第四个button发送消息。



代码上下文中已蕴藏这些经验，但加入经验描述影响大窗口中的注意力分布，降低推理负担，提升生成效果。



changelist → 经验

经验

新任务



GPT - 4

GPT - 4



GPT -

1 扩展规则引擎，增加left_shift (<<) 操作符。



4 依赖经验知识的编码任务 - 利用"相关任务的实例化经验"

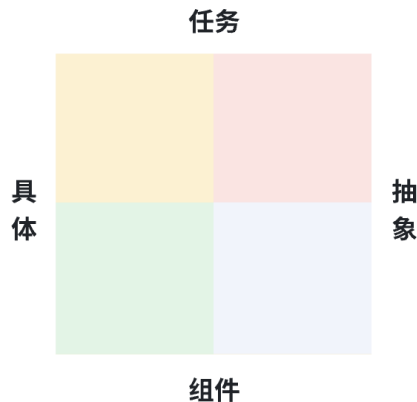
文件	函数	仅提供代码文件		Git格式diff做经验		代码中标记diff做经验	
		规划任务	生成任务	规划任务	生成任务	规划任务	生成任务
math/mod.rs	interface	✓	✓	✓	✓	✓	✓
	impl	✓	✓	✓	✓	✓	✓
operator/mod.rs	enum	✓	✓	✓	✓	✓	✓
	get_max_args	✓	✓			✓	✓
	get_min_args	✓	✓			✓	✓
	get_priority	✓	✓	✓		✓	✓
	can_have_child					✓	✓
	from_str	✓	✓	✓	✓	✓	✓
tree/mod.rs	exec_node	✓	✓	✓	✓	✓	✓
	parse_pos					✓	✗
	parse_node					✓	✓
	parse_operators					✓	✗
	unittest			✓	✓	✓	✓



5 知识分类及提示词框架

人脑曾加工和
使用过的知识

程序生产出的
数据知识



产物知识: 在流程中以产物形式显性表达的知识。

经验知识: 生产产物知识过程中使用的经验，往往是隐性的。

衍生数据知识: 程序运行时的数据，还有基于代码分析出来的数据，用于知识相关性计算或直接拿来提升某些任务效果。





5 知识分类及提示词框架

在什么**上下文**下（基于什么），应用什么样的**经验**（怎么干），完成一个什么样的**任务**（干什么）。





6 大模型辅助开发的实用形态

- 工具能做到什么程度？知识工程+Agent
- 工程师是否需要裸用大模型？
- 工程师是否必须要能驾驭大模型？
- 人的精力能被释放到什么程度？
- 追求多大比例的效率提升比较现实？
- 个人维护知识工程是否可行？



源于社区 服务社区

THANKS!

1. 主干任务难度远超支撑性任务。
2. 从知识生产和消费的角度思考。
3. 如何遵循设计是重要挑战。
4. 从历史记录中加工经验知识。
5. 知识分类指导知识工程建设。

