

# 日报

## 20250302

想了一上午到底自己是在做什么  
下午按照计划打了一场算竞比赛  
晚上跑cf网站的爬虫被反爬打趴

## 20250303

爬虫依然被打趴。  
向刘健师兄请教到底自己该怎么做，通过交流明确了很多基本概念，消除了我的很多疑惑并明确了需要做的事情——找到感兴趣的论文就去认真读并复现。

## 20250304

想起老师所说的ChatDev，成功运行了它的代码，我尝试让他“生成一个五子棋程序”(ChatGPT3.5)，我认为这不算一个很难的要求，但发现好像效果不是很好。  
注：原论文评估时就是用的GPT3.5。  
同时阅读论文，大概读了小半。  
另外群u给了个镜像站地址，爬虫站起来了，虽然已经没大用了。

## 20250305

读完了论文，开始阅读代码，学习到了很多之前不会的python技巧，如使用包装器函数将对Message的访问重定向str类中。  
回顾所学，写这篇周报。

# ChatDev 论文阅读

[论文链接](#)

# Summary

仿照软件工程中分工模式，让模型分别扮演CEO(design), CTO(communicate with programmer), Programmer(coding), Reviewer(code review), Tester(testing)来开发一个程序，提高程序设计成功率。其中，CEO负责理解用户需求并规划任务，CTO负责理解CEO给出的任务并与Programmer交互直到确认代码编写完成，接下来是Reviewer和Tester依次指出问题（两者区别是执行代码与否），由Programmer修复漏洞。

## 学习到的知识以及感觉需要记录的东西

### Memory

由于显然交流多次后聊天记录会很长，而目前模型无法处理巨长无比的聊天记录，因此作者整了一个长短期记忆，具体而言，短期记忆就是存双方近期聊天记录，长期记忆存储'solution'，我理解是只存了代码，关于这点需要继续阅读代码。

By sharing only the solutions of each subtask rather than the entire communication history, ChatDev minimizes the risk of being overwhelmed by too much information, enhancing concentration on each task and encouraging more targeted cooperation, while simultaneously facilitating cross-phase context continuity.(3.1节最后一段)

如果真的是这样的话是不是武断了点？

### 执行者提问环节

图例：

A: assistant 执行者

I: instructor 构思者

-> : 提要求/提问

~> : 回复/给方案

正常来说一般是：

I -> A, A ~> I

作者加入了一个环节，要求 A 可以向 I 提问更多详细信息。

I -> A, (A -> I, I ~> A)(这里可循环), A ~> I

作者称这能够有效减少沟通性幻觉。

# 评估

作者制作了一个专门的评估数据集。

评估标准分为完成性(能否独立完成)，可执行性，一致性(符不符合要求)，质量(这个是前面三个的一个加权结果)。作者给出了它们的计算方式。

# 其他感受

这篇文章的作者英文水平比之前看的好的多，用词仿佛莎士比亚，导致我到处不认识单词。直观感觉这篇文章中所提到的东西都称不上难，这是一种错觉吗（？，或许是文章写得好容易理解。

之前在组会上红温(字面义，我的语言功能因各种原因紧张而丧失时会整个脑袋升温，表现为面部耳部一秒变红肿大)时想表达的东西，大致就是给Programmer增加一个Planner以及一个监督者，类似于这篇文章中CEO和CTO的定位。