

## 人工智能基础大作业报告

咸浩哲 信科学院，北京大学，预科

叶拉曼 信科学院，北京大学，大一

杨其融 信科学院，北京大学，大一

兰艺慧 信科学院，北京大学，大一

娜迪娅 信科学院，北京大学，大一

## 摘要

我们小组所做的项目是一个算法竞赛的助手。它支持录入题面和题解后让 AI 提示用户一些 hint，或者在用户解出题目后和用户进行一些讨论。也支持录入题面，然后让 AI 给一些情况绘图。

## 1 需求分析

需求来自于我中学时参加算法竞赛时的想法。初始的想法是希望它能直接做题，或者给出一些思路上的启发，但是这对 AI 来说太难了。所以退而求其次，选择直接给 AI 录入题面和题解让它来引导用户，以加强一道题的训练效果。并集成画图软件，让 AI 可以根据用户的要求绘制一些图片，以方便用户思考。

## 2 技术选型

选用 camel ai 搭建应用。

- 对于让 AI 录入题面和题解，使用知识库增强可以让 AI 找到

一些题目之间的共性，以对一些可能不那么完整的题解进行查漏补缺。使用模型 sentence-transformer 并嵌入 moka-ai/m3e-base，以完成高质量编码，检索策略为 faiss。

- 对于第一部分，即录入题面和题解让 AI 提供 hint。使用模型 Pro/deepseek-ai/DeepSeek-R1，这是因为看中了该模型的推理能力。
- 对于第二部分，即录入题面让 AI 根据用户的要求绘制一些图片。也使用模型 Pro/deepseek-ai/DeepSeek-R1，这是因为看中了该模型的抽象能力。
- 绘致图片需要集成一些能够绘制抽象软件。在 deepseek 推荐下最终选用了 matplotlib，它比较轻量，并支持对数据结构中的序列，图论中的树和图进行绘制。它的缺点是无法绘制太复杂的情况，而需求也并不需要十分复杂的图。（毕竟最终是

给人看的)

入知识库的 json 即可。代码见 teach.py。

### 3 实验细节

使用 camel AI 开发。

- 对知识库的信息来源,选择 <https://yhx12243.github.io/OI-transit/> 爬取。由于不需要爬取太复杂的网页,且并没有找到能够直接对 html 爬取 markdown 的代码,所以手动爬下来然后用 BeautifulSoup 解析,并最后将字符用 GBK 编码。代码见 knowledge\_crawler.py。

- 对于构建知识库,将知识库用 json 保存在本地以避免问一道题就要重新编码出一遍知识库。知识库允许导出为 json 或者用 json 导入。并对知识库的调用,允许传入 model 和提示词,从而增加灵活性。代码见 knowledge\_generator.py。

- 对于第一部分,直接使用上一份代码即可。temperature 取 0.1,这可以让大模型生成的随机性变低,比较适合比较数学的问题。而且需求也不需要太多的创造性。token 取 16384,以便模型可以充分思考有概率挖掘出一些比较深层的性质和信息。实现上,把身份,题目和题解传入 Agent 作为提示词,然后导

- 对于第二部分,集成 matplotlib,并将内容细分为树,图和序列。分别作为工具集成。这里也可以加入更多工具,以适应更多的需求。temperature 取 0.3,比起第一部分随机性稍高更有助于抽象,但依然会较为按部就班的思考如何绘图。token 取 4096,因为这部分对寻找深层信息的要求较低,而更多的是按部就班的理解并执行命令。把身份,题目传入 Agent 作为提示词,然后引导它使用正确的工具即可。代码见 draw.py。

- 对于两部分,都允许 AI 在不理解时询问用户。这样一方面不会产生过于严重的错误,也可以引导用户进行思考。也要求 AI 检测用户的要求是否恰当,相当于敏感词防护与防止指令注入。

### 4 评估对比

- 在 teach.py 中,由于生成结果的效果不好量化,加之 agent 接受的提示词足够多,导致其对 temperature 比较鲁棒。但依然选择取 0.1 以加强它的数学推理能力。
- 在 draw.py 中,其对 tempera-

ture 也比较鲁棒。但可以观察到,在 `temperature=0.3` 中它仅用括号描述点权,在 `temperature=0.9` 时则自作主张将点权命名为 `a`。并会很外向的询问用户要不要继续一些分析。

- 关于与前人的对比,尚不清楚是否有类似的东西。而对于它的效果,我认为它生成的东西质量尚可,所以对信息学竞赛选手确能提升一定训练质量。但由于其有着生成速度过慢,正确性未知等问题,提升是很有有限的,不妨仅仅将其看作一次尝试。

## 5 反思

- 在需求上, AI 无法解决一些较为困难的事情。这当然与算法竞赛没有充足的资料,推理难度大,且需要极强的抽象思维有关,这是没有办法的事。但一些整理,以及并不十分困难的事, AI 的确能做得很好。
- 在选型上,使用优秀的模型以及参数是重要的,可以通过对比来进行调参。提示词也是重要的。需要良好的提示词才能让 AI 按想法做事。
- 在细节上,代码主要由 AI 生成。AI 在代码上有时会犯一些

小错,有时也会因为各种不兼容的问题出现错误,调试有些麻烦,但确实能省下写冗长代码的时间。

一些可以拓展的内容:

- 提供更多的格式转化以及更多的工具,如输入题目和题解的网址而非文件,以及一些拓展的画图。没做的原因一方面是工作量,一方面是工具是永远越找越多的。
- 开发记忆系统,描述用户的水平和薄弱点擅长点等。没做的原因一方面是工作量,一方面是 AI 并不能完全理解题目,于是也不太能对题目进行评估。如果记忆错误,原本的生成质量反而会下降。