2025 自然语言处理
课程设计 2

人工智能学院 221300079 王俊童

2025.5.1

首先做一个简介：

这次的任务基本是用四个不同方法去尝试把一个解密游戏做好。

本来想用免费 ai 的，比如 Free QWQ 之类的，但是这一类 ai 有一个通病就是连接极度不稳定，而且算力分配是有问题的，所以我自费了 Kimi，用的模型是中等强度的 moonshot-v1-32k。

本次作业制作成本极高。

同时，由于这个作业评价指标不唯一，所以我设计了三个评价指标

- 指标一：是否全部答对。只有 0，1 二值。
- 指标二：在必须回答的问题中，回答对了多少。（非附加问题准确度）
- 指标三：在非必须回答问题中，回答对了多少。（附加问题准确度）

这个指标设置相对简单，因为我们不去很严格的考虑 ai 全部能答对，其实这种指标相对的武断了。我们也要考虑到答对了一部分这个情况。

1 实现方法，对自己设计的代码模块用简洁的语言描述

1.1 任务一：调用 API 生成

首先是调用 api 生成，这里我重新写了一个框架，基本思路非常非常 easy，就是喂进来一个算一个。

当然，ai 错的还是有很多的，我们在这里挑选几个 case 来观察一下准确率：

首先是一个完全正确的例子：

然后是一个部分正确的例子：

最后是一个完全错误的例子：

可以看出 ai 的推理还是有缺陷的，我们给出在 kimi 的 moonshot-v1-32k 下的准确度：

方法	原指标准确度	必答准确度	选答准确度
API	2%	27.62%	27.95%

Table 1: 方法性能对比一

1.2 任务二：Prompt Engineer

这个任务要求我们化身 prompt 大师，我们在上面任务的基础上，增加了两个可能帮助我们的大模型进一步生成更准确的推理的 prompt：

Prompt1: 你是福尔摩斯: 具体操作为：

```
def prompt_sherlock(self, question: str) -> str:
return f"你是福尔摩斯，接到一个案子：{question}␣请详细推理并找出答案。"
```

Prompt2: 序列化推导问题: 具体操作为:

```
def prompt_step_by_step(self, question: str) -> str:
return f"请一步步推理以下问题，并给出符合逻辑的正确的最终答案：{question}"
```

这两个方法最主要功能就是，提醒大模型你的身份，或者你该怎么做，大模型就不会去乱做或者没有任何先验的情况下乱搞。我们得到的结果如下:

方法	原指标准确度	必答准确度	选答准确度
Prompt 福尔摩斯	-	-	-
Prompt 序列推理	-	-	-

Table 2: 方法性能对比二

1.3 任务三：工具使用

1.4 任务四：多智能体对话

编写代码，模拟现实场景中的多智能体交流，让多个 LLM 以不同角色协同解决问题。这个地方我们主要实现一个法庭的辩论这种情形。

因为这个地方跟破案解密有关，我们设置一个法官和证人这种，然后通过 self.loop 去控制法官和证人会交锋多少轮次。其实我们想做的就是一个类似于 GAN 一样的，证人讲证据，然后法官二次判断这样来。然后最后法官作出总结陈词。

首先可以证明的是，因为反复问询这种手段的存在性，我们有必要在不同的 loop 轮次上进行不同次数的实验，因为这可能意味着也许证人提供更多的细节内容，那么法官会更倾向于做出正确选择。

首先从准确率上说

我们首先看在 loop 为 1 的情况下的正确率，这个就是最基本的一种情况了，问一次然后总结。

方法	原指标准确度	必答准确度	选答准确度
Agent loop1	-	-	-

Table 3: 方法性能对比四

那么其实，我们可以增加 loop 次数，由于经费有限，我们随机挑选 5 个例子来做这个任务，看看五个例子的预测效果会随着 loop 变化怎么变。

方法	原指标准确度	必答准确度	选答准确度
Agent loop1	-	-	-
Agent loop3	-	-	-
Agent loop5	-	-	-

Table 4: 方法性能对比四

然后从趣味性上说

我们可以截取一下对话例子:

2 复现主要实现结果，包括执行命令和环境依赖

所有的环境依赖都在 requirements.txt 中.

你可以选择用指令：**sh run.sh** 来运行程序

如果想单独运行：main.py 即可。指令如下:

python3 main.py –method 0 -n 0
其中，method0,1,2,3 代表 4 个不同方法，0 代表运行所有数据。

3 不同方法的实验结果如何

下面给出四个方法的汇总表格实现：

方法	原指标准确度	必答准确度	选答准确度
API	-	-	-
Prompt 福尔摩斯	-	-	-
Prompt 序列推理	-	-	-
Tools	-	-	-
Agent loop1（ALL）	-	-	-
Agent loop1（Random 5）	-	-	-
Agent loop3（Random 5）	-	-	-
Agent loop5（Random 5）	-	-	-

Table 5: 方法性能对比四

4 遇到的具体问题，如何解决

5 思考