PJ3-大模型能力探究与大模型自评估-report

姓名: 陈锐林, 学号:21307130148

2023 年 11 月 25 日

一、实验概述

通过已有的评测集和评测 prompt，对三种大模型：qwen-14b-chat，baichuan2-13b-chat-v1 和 chatgpt-3.5-turbo 进行代码生成、角色扮演、数学推理、文案撰写共 4 各方面的测试；并且尝试通过 ChatGPT 进行评判。

二、调用 API 接口和问题输入

1. 按照不同模型和 prompt 调用接口

```
import dashscope
dashscope.api_key="sk-84aad3057fbc462399a036c77c213a78"
def ask(model,prompt):
    print(model,prompt)
    response_generator = dashscope.Generation.call(
        model=model,
        prompt=prompt,
        stream=True,
        top_p=0.8)

    head_idx = 0
    for resp in response_generator:
        paragraph = resp.output['text']
        print("\r%s" % paragraph[head_idx:len(paragraph)], end='')
        if(paragraph.rfind('\n') != -1):
            head_idx = paragraph.rfind('\n') + 1
```

2. 按类型调用 API 测试

```
def ask_by_types(type,questions):
    qs = [q for q in questions if q['category'] == type]
    for q in qs:
        s=' '.join(q['turns'])
```

```
print("\n","By baichuan2-13b-chat-v1, the answer to ",s)
ask('baichuan2-13b-chat-v1',s)
print("\n","By qwen-14b-chat, the answer to ",s)
ask('qwen-14b-chat',s)
```

三、输出结果示例

这个程式的功能是逐行的读取一个文本期文件，然后计算特定单词的出现次效。这个特定
首先，我们导入了`sys`模块，以便在命令行的使用中能够获取参数。然后，定义了一个
By qwen-14b-chat, the answer to 实现一个Python程序，逐行读取文本文件并计
qwen-14b-chat 实现一个Python程序，逐行读取文本文件并计算文件中特定单词的出现
以下是一个简单的Python程序，它将逐行读取文本文件并计算文件中特定单词的出现次数

```python
def count_word_in_file(file_path, word):
    word_count = 0

    with open(file_path, 'r') as file:
        for line in file:
            words = line.strip().split()
            word_count += words.count(word)

    return word_count


# 使用方法:
file_path = 'path_to_your_text_file.txt'  # 替换为你的文本文件路径
word = 'example'  # 替换为你想要计数的单词
print(count_word_in_file(file_path, word))
```

四、结果评估

1. 代码生成：

(1) 能否通过编译，生成 3 次，分别测试：百川 (13b) 给出的很糟糕 (会有明显乱码和逻辑错误，比如 = 输出为-)，但是 (7b) 就很好，这里以 13b 为准。

任务名	百川	千问	ChatGPT
计算特定单词出现数	False	True	True
DP 实现 LCS	False	True	True
验证邮箱地址	True	True	True
求 Fi(n)	True	True	True
二分查找	True	True	True
双栈实现队列	False	True	True
找公共元素	True	True	True

(2)ChatGPT 评估，根据 judge-prompts 进行评分

任务名	百川	千问	ChatGPT
计算特定单词出现数	3	7	9
DP 实现 LCS	2	8	9
验证邮箱地址	3	7	9
求 Fi(n)	2	8	9
二分查找	1	8	9
双栈实现队列	1	7	9
找公共元素	4	9	8

2. 角色扮演：ChatGPT 评估。

任务名	百川	千问	ChatGPT
李白	8	7	9
Sheldon	6	8	9
医生	7	8	9
关系教练	7	8	9
翻译	9	6	9
机器学习工程师	8	8	10
数学老师	7	8	9
艺术家	8	8	10
数学家诗人	8	8	10
百年大树	8	8	10

3. 写作：ChatGPT 评估。

任务名	百川	千问	ChatGPT
小红书	8	9	9
辞职信	9	8	9
推荐信	8	9	9
小剧本	7	8	8
道歉信 + 解决方法	6	8	9
论文大纲	6	8	9
游记	7	6	9
电影观后感	8	9	9
解决脱发	7	7	9
简历	7	8	8

4. 数学：ChatGPT 评估。

任务名	百川	千问	ChatGPT
三角形面积	8	9	3
总投资	2	10	10
容斥原理	4	8	10
掷骰子	4	8	10
上下车问题	6	6	10
整除问题	4	10	10
小明买书	4	10	10
f(2)	5	10	10

五、结果分析和问题思考

1. 大模型能力比较：

从上述的评分中能看出来，这三个大模型的能力由强到弱应该是 chatgpt-3.5-turbo>qwen-14b-chat>baichuan2-13b-chat-v1。

2. 大模型的能力边界、缺陷和风险：

(1) 大模型的能力边界在于上下文和任务量。首先 coding 的任务量不算大，但是百川的模型还是会输出不了正确的，即使考虑到可能是异常情况剔除后；也能考虑日常生活中的经验，也可以知道，随着任务的量变大和难度增大，大模型还是没法解决的。其次如果进行以下形式的评估对话"judge_prompt + Question1 + Question2 + ..."，会发现 5/6 个问题后大模型就忘记了我们当下要做的 work 是评估任务了；说明大模型的能力边界仍是受到上下文约束的。

(2) 大模型的缺陷仍然是存在的。说到底大模型的反应机理和人是不一样的；人做不到像它那样回答各式各样的问题，但是人不会在很简单的算术上犯难 (比如 f(2) 那题)。在 writing 部分，三个大模型给出的是比较发散的结果；但是有的结果仍只是差强人意。比如小红书任务中，百川给出了一长串的"Tags"，这是反常规的。

(3) 大模型的风险一是在于上面说到的，可能有错解；二是在于内容的知识产权归属也会有问题。

3. 不同 prompt 构造的影响：

(1) 在 math 问题中；是不是提供 reference 差别结果很大。上面给出的版本是不带 reference，会出现：三个模型的答案其实俩俩不同；但是 ChatGPT 认为其中两个 (或三个) 答案都是完全正确的，并且给到满分 10 分的情况。给出了 reference 之后就只会留下一个 (或更少的)10 分答案。

(2) 在 roleplay 问题中，对于 turns 中的问题，分开询问和一起询问也能得到不同输出。但是在这个任务中，几乎不会有过大的差距；可能是因为输入和输出量

是不够大的，超不出上下文的限制。

4. 大模型评估大模型的可行性：

(1) 我觉得是可行的；但前提是我们要给出明确的评估标准，并且采用正确的prompt，以及人工的复核。

(2) 对于不同类别的任务，我们应该要给出更适合的标准；比如虽然大模型没法理解我们所说的” 正确性” 到底是什么，但是它能通过调整自己的参数来实现这一点 (虽然只是概率的预测)。

(3) 并且很多时候，在给出标准时要更精细，比如在” 不借助其他数据结构查找公共元素” 的 coding 任务里；百川和 ChatGPT 给出的代码都是两层循环 + 类似的函数调用，但是因为百川没有提供使用样例，最后得分 4 分，只有 GPT 的一半。我们应该对不同的标准给出倾向性才行，因为 coding 的重点不在于使用范例上，至少不是 4 分和 8 分的差距。

(4) 人工的复核是因为我发现在 math 的任务中，前几个题，答案错了，可能就只能得到 4 分；但是在最后的 f(2) 中，即使答案被诊断为错，仍拿了 7 分。我觉得这是需要人工去审核的。