

深圳大学实验报告

课程名称：智能物联网

实验项目名称：实验 4 大模型 prompting 小实验

学院：计算机与软件学院

专业：数计班

指导教师：王璐

报告人：詹耿羽 学号：2023193026 班级：数计班

实验时间：2024.6.23

实验报告提交时间：2024.6.24

实验目的

- 1.这个实验旨在评估和比较不同的大型语言模型如 ChatGPT（GPT-4o）、文心一言、Kimi 在不同提示方法下的表现。
- 2.这些方法包括零次学习（zero-shot）、少次学习（few-shot）、思维链（chain of thought）以及其他可能的提示方式。
- 3.实验将通过一个包含常识、数学和逻辑推理问题的测试集来进行。

实验环境

- 1.模型选择：ChatGPT（GPT-4o）、文心一言（百度的 LLM）、Kimi。
- 2.平台：实验将在各自模型的官方或授权平台上进行，确保环境稳定和结果的可靠性。
- 3.工具：使用标准化的评估工具和脚本进行自动化测试和结果记录。

实验内容：

- 1.测试集编制：设计包含至少 10 个问题的测试集，覆盖常识、数学问题和逻辑推理等内容。
- 2.问题类型：确保问题多样性，包括开放式回答、问题解答、推理解析等。

实验步骤：

Tips: 由于例子过多，在此只选择具有代表性且能体现实验结论的样例截图展示。

1. 测试集准备：
 - 根据实验需要，设计并准备一个涵盖多种题型和难度的测试集。
 - 常识类问题
 - 水沸腾的温度是多少摄氏度？
 - 维生素 C 主要有什么作用？
 - 数学问题
 - 解方程 $13x-7=5x+1$
 - 一个圆的面积是 78.578 平方厘米，求其半径（使用 $\pi \approx 3.14$ ）。
 - 如果一个商品在打 8 折后售价为 240 元，那么它的原价是多少？
 - 逻辑推理问题
 - 如果所有的猫都怕水，而 Tom 是一只猫，那么 Tom 怕水吗？
 - 在一个有三个开关的房间里，只有一个开关与灯泡相连。如果每次只能测试一个开关，至少需要多少次测试才能确定哪个开关与灯泡相连？
 - 综合理解问题
 - 一本书的前三分之一部分有 200 页，这本书一共有多少页？
 - 计算一周有多少秒？
 - 深度思考问题
 - 在最后一个问题里面，我改用了深圳大学 OJ 题目：

题目描述

定义多边形基类polygon, 包含保护属性: 边数number(int), 私有属性: 边长数组side_length(int, 最多100个元素); 公有函数包含: 构造函数、虚函数perimeter计算多边形周长 (返回int), 函数display输出多边形边数和周长。
定义派生类rectangle(矩形), 公有继承polygon类, 新增以下数据成员: 矩形高height (int), 矩形长width (int); 新增公有成员函数: 构造函数, 重写perimeter计算矩形周长。
定义派生类equal_polygon(等边多边形), 公有继承polygon类, 新增以下数据成员: 边长side_len (int); 新增公有成员函数: 构造函数, 重写perimeter计算矩形周长。

主函数定义基类指针, 根据输入的多边形信息, 指向一个多边形类对象或矩形类对象或等边多边形类对象, 计算每一个多边形的周长并且输出其边数和周长。
主函数已给出, 请补齐三个类。构造函数可根据需要自行添加。

输入

测试输入包含一个测试用例, 该测试用例的第一行输入多边形的个数n, 接下来n行每一行给出一个多边形的基本信息, 每行的第一个数字为当前多边形的类型:
类型0为一般多边形, 后面跟随m个数字为m条边的边长, -1为一般多边形边长输入结束标志,
类型1为矩形, 后面跟随两个数字, 分别为height和width
类型2为等边多边形, 后面跟随两个数字为等边多边形的边数和边长。

图 1: OJ 原题

我把原先的代码全部删去, 让 ai 自行发挥, 然后对比前后输入输出的示例, 来检验 ai 是否做对了题目。

题目为:

题目描述

定义多边形基类polygon, 包含保护属性: 边数number(int), 私有属性: 边长数组side_length(int, 最多100个元素); 公有函数包含: 构造函数、虚函数perimeter计算多边形周长 (返回int), 函数display输出多边形边数和周长。

定义派生类rectangle(矩形), 公有继承polygon类, 新增以下数据成员: 矩形高height (int), 矩形长width (int); 新增公有成员函数: 构造函数, 重写perimeter计算矩形周长。

定义派生类equal_polygon(等边多边形), 公有继承polygon类, 新增以下数据成员: 边长side_len (int); 新增公有成员函数: 构造函数, 重写perimeter计算矩形周长。

主函数定义基类指针, 根据输入的多边形信息, 指向一个多边形类对象或矩形类对象或等边多边形类对象, 计算每一个多边形的周长并且输出其边数和周长。

输入

测试输入包含一个测试用例, 该测试用例的第一行输入多边形的个数n, 接下来n行每一行给出一个多边形的基本信息, 每行的第一个数字为当前多边形的类型:
类型0为一般多边形, 后面跟随m个数字为m条边的边长, -1为一般多边形边长输入结束标志,
类型1为矩形, 后面跟随两个数字, 分别为height和width
类型2为等边多边形, 后面跟随两个数字为等边多边形的边数和边长。

输出

每行输出多边形的边数以及周长

输入样例1

```
3
0 32 54 76 88 24 -1
1 32 54
2 3 32
```

输出样例1

```
5 274
4 172
3 96
```

语言:

C++

图 2: 改写后的题目

这些测试集旨在检验模型在处理不同类型的问题时的能力，包括对基本常识的理解、数学计算能力、逻辑推理和深入思考的能力。

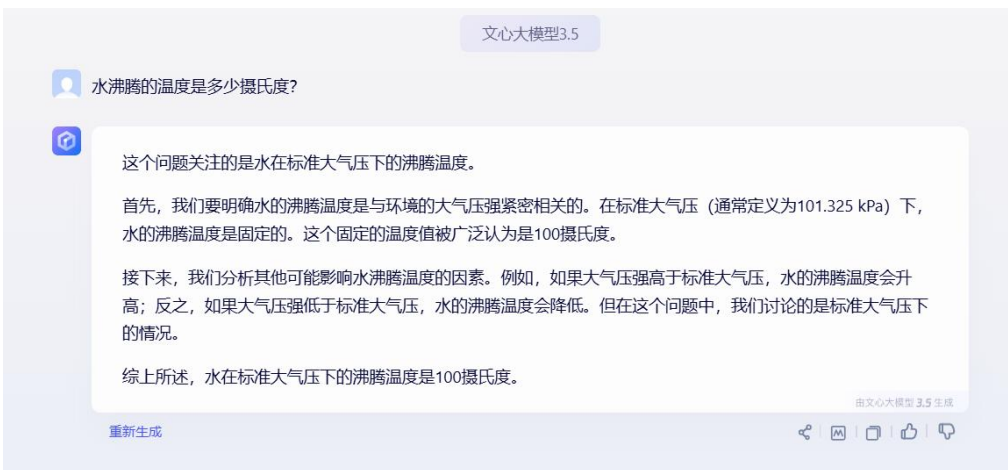
2. 模型初始化：

- 为每个模型配置必要的环境和参数。

3. 零次学习测试：

- 直接向模型提交问题，不提供任何背景信息或例子。
- 记录模型的回答和处理时间。

A. 文心一言



B. Kimi



C. GPT-4o

水沸腾的温度是多少摄氏度？



水在标准大气压下（1个大气压，约101.3千帕）的沸点是100摄氏度。这是一个常见的物理常数，表示水由液态变为气态的温度。

然而，水的沸点会随压力的变化而变化。在高于标准大气压的条件下，水的沸点会高于100摄氏度；在低于标准大气压的条件下，水的沸点会低于100摄氏度。例如，在高山上由于气压较低，水的沸点会低于100摄氏度。



结论：在零次学习的测试下，响应速度三者基本相等，但是 Kimi 回答的较为全面，三者的回答都是正确的。

4. 少次学习测试：

- 向模型提供几个相关的例子或提示后，再提交问题。
- 分析模型利用少量信息进行推理的能力。

A. 文心一言

一个圆的面积是78.578平方厘米，求其半径（使用 $\pi \approx 3.14$ ）。



我们有一个圆，知道它的面积是78.578平方厘米，并且 π 的值近似为3.14。

我们需要找出这个圆的半径。

假设这个圆的半径为 r 厘米。

圆的面积公式是：面积 = $\pi \times r^2$

在这个问题中，面积 = 78.578 平方厘米， $\pi \approx 3.14$ 。

用数学方程，我们可以表示为：

$$3.14 \times r^2 = 78.578$$

现在我们要解这个方程，找出 r 的值。

计算结果为： $r = 5$ 厘米

所以，这个圆的半径是 5 厘米。

图灵心大模型 3.5 版本

在一个有三个开关的房间里，只有一个开关与灯泡相连。如果每次只能测试一个开关，至少需要多少次测试才能确定哪个开关与灯泡相连？



我们有一个房间，里面有三个开关，但只有一个开关与灯泡相连。

我们的目标是找出哪个开关与灯泡相连，但每次只能测试一个开关。

假设我们有三个开关，分别命名为 A、B 和 C。

为了找出与灯泡相连的开关，我们可以采用以下策略：

1. 首先打开开关A，等待一段时间（比如几分钟），让灯泡发热。
2. 然后关闭开关A，并打开开关B。
3. 接下来，我们走到灯泡所在的房间，观察灯泡的状态。

根据灯泡的状态，我们可以得出以下结论：

1. 如果灯泡是亮的，那么与灯泡相连的开关是B。
2. 如果灯泡是暗的但摸起来很热，那么与灯泡相连的开关是A（因为A之前被打开过，灯泡发热了，但后来被关闭）。
3. 如果灯泡是暗的且摸起来不热，那么与灯泡相连的开关是C（因为C是最后被打开的，灯泡既没亮也没热）。

现在，我们不需要进行实际的测试，因为我们已经有了策略。

但根据策略，我们可以确定至少需要测试的次数。

根据上述策略，我们至少需要测试两次来确定哪个开关与灯泡相连。

所以，至少需要测试2次来确定哪个开关与灯泡相连。

由文心大模型 3.5 生成

重新生成



B. Kimi

在一个有三个开关的房间里，只有一个开关与灯泡相连。如果每次只能测试一个开关，至少需要多少次测试才能确定哪个开关与灯泡相连？



要确定哪个开关与灯泡相连，你可以使用一种称为“二分法”的策略，但这不适用于只有三个开关的情况，因为二分法适用于可以被二整除的数。在这种情况下，使用简单的排除法会更有效。

由于有三个开关，你可以按照以下步骤操作：

1. 打开第一个开关，等待一段时间（比如几分钟），然后关闭它。
2. 立即打开第二个开关。
3. 观察灯泡的状态：
 - 如果灯泡亮了，那么第二个开关就是与灯泡相连的。
 - 如果灯泡没有亮，但摸起来是热的，那么第一个开关是与灯泡相连的（因为你让它亮了一段时间）。
 - 如果灯泡没有亮，并且摸起来是冷的，那么第三个开关是与灯泡相连的。

这种方法只需要两次测试就可以确定哪个开关与灯泡相连。



一个圆的面积是78.578平方厘米，求其半径（使用 $\pi \approx 3.14$ ）。



圆的面积 A 可以使用公式 $A = \pi r^2$ 来计算，其中 r 是圆的半径， π 是圆周率。给定圆的面积 $A = 78.578$ 平方厘米，圆周率 π 约等于 3.14，我们可以将问题转化为求解半径 r 的问题。

首先，将面积公式重写为求解半径的表达式：

$$r = \sqrt{\frac{A}{\pi}}$$

然后，将给定的数值代入表达式中：

$$r = \sqrt{\frac{78.578}{3.14}}$$

👉 按了计算器

$$r = \sqrt{25.0248408}$$

👉 按了计算器

计算得出，半径 r 大约为 5.0025 厘米。



C. GPT-4o



一个圆的面积是78.578平方厘米，求其半径（使用 $\pi \approx 3.14$ ）。



要计算一个圆的半径，已知其面积为78.578平方厘米，我们可以使用圆的面积公式：

$$A = \pi r^2$$

其中：

- A 是圆的面积
- r 是圆的半径
- $\pi \approx 3.14$

我们需要解这个方程来求 r 。

首先，将已知值代入公式：

$$78.578 = 3.14r^2$$

接下来，解方程以求 r^2 ：

$$r^2 = \frac{78.578}{3.14}$$

计算右边的值：

$$r^2 = \frac{78.578}{3.14} \approx 25$$

然后，取平方根以求 r ：

$$r = \sqrt{25}$$

$$r \approx 5$$

因此，圆的半径约为5厘米。





要找出商品的原价，可以使用以下步骤：

1. 设商品的原价为 x 元。
2. 商品打8折后，售价为原价的80%，即 $0.8x$ 。
3. 已知打折后的售价是240元，因此可以建立以下方程：

$$0.8x = 240$$

4. 解这个方程来找出 x ：

$$x = \frac{240}{0.8}$$

计算结果：

$$x = 300$$

因此，商品的原价是300元。



1. 测试第一个开关：

- 打开第一个开关，然后检查灯泡是否亮起。
- 如果灯泡亮起，则确定第一个开关是与灯泡相连的开关，此时仅需要一次测试。
- 如果灯泡没有亮起，则排除第一个开关。

2. 测试第二个开关：

- 打开第二个开关，然后检查灯泡是否亮起。
- 如果灯泡亮起，则确定第二个开关是与灯泡相连的开关，此时需要两次测试（第一次排除第一个开关，第二次确定第二个开关）。
- 如果灯泡没有亮起，则可以确定第三个开关是与灯泡相连的开关，因为前两个开关都已被排除。

通过这种方法，最少需要的测试次数如下：

- **最少1次**：如果第一个开关与灯泡相连。
- **最少2次**：如果第二个开关或第三个开关与灯泡相连。

因此，**最多只需要2次测试**就能确定哪个开关与[↓]泡相连。

结论：在少次学习测试中，响应速度：GPT4o > Kimi > 文心一言；三者的结果都是正确的。

5. 思维链测试：

- 提供一系列逻辑步骤，引导模型进行深入思考后回答问题。
- 检测模型是否能够通过提供的思维链来改善答案的逻辑性和准确性。

A. 文心一言



下面是一个C++实现的示例代码，根据题目要求定义了基类 Polygon 和两个派生类 Rectangle 和 EqualPolygon，并在主函数中根据输入的多边形类型创建了对应的多边形对象，并计算、输出了其边数和周长。

```
cpp 复制代码
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  class Polygon {
6  protected:
7      int number;
8      vector<int> side_length;
9  public:
10     Polygon(int n) : number(n) {}
11     virtual ~Polygon() {}
12
13     virtual int perimeter() = 0;
14
```

```
main.cpp
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  class Polygon {
6  protected:
7      int number;
8      vector<int> side_length;
9  public:
10     Polygon(int n) : number(n) {}
11     virtual ~Polygon() {}
12
13     virtual int perimeter() = 0;
14
15     void display() {
16         int p = perimeter();
17         cout << number << " " << p << endl;
18     }
19 };
20
21 class Rectangle : public Polygon {
22     int height, width;
23 public:
24     Rectangle(int h, int w) : Polygon(4), height(h), width(w) {}
25
```

三个错误 (文心一言)

B. Kimi



这是一个C++编程题，要求实现多边形基类polygon和两个派生类rectangle和equal_polygon，以及主函数中根据输入创建相应对象并计算周长的功能。下面是一个可能的实现方式：

```
#include <iostream>
#include <vector>

// 多边形基类
class Polygon {
protected:
    int number; // 边数
    std::vector<int> side_length; // 边长数组

public:
    // 构造函数
    Polygon(int n) : number(n) {}

    // 虚函数，计算周长
    virtual int perimeter() = 0;

    // 输出多边形信息
    void display() {
```

复制

```
main.cpp
1 #include <iostream>
2 #include <vector>
3
4 // 多边形基类
5 class Polygon {
6 protected:
7     int number; // 边数
8     std::vector<int> side_length; // 边长数组
9
10 public:
11     // 构造函数
12     Polygon(int n) : number(n) {}
13
14     // 虚函数，计算周长
15     virtual int perimeter() = 0;
16
17     // 输出多边形信息
18     void display() {
19         std::cout << number << " ";
20         int total = 0;
21         for (int len : side_length) {
22             total += len;
23         }
24         std::cout << total << std::endl;
25     }
26 };
27
```

出现错误 (Kimi)

C. GPT-4o



下面是一个符合题目描述的C++程序代码。此代码定义了多边形的基类和两个派生类，并使用多态性来根据输入的多边形类型创建相应的对象，并计算和输出周长和边数。

```
cpp
复制代码

#include <iostream>
#include <vector>
using namespace std;

class Polygon {
protected:
    int number; // 边数
    vector<int> side_length; // 边长数组
public:
    // 构造函数
    Polygon() : number(0) {}

    virtual ~Polygon() {}

    // 虚函数，计算多边形的周长

```

```
项目 Alt+1
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 class Polygon {
6 protected:
7     int number; // 边数
8     vector<int> side_length; // 边长数组
9
10 public:
11     // 构造函数
12     Polygon() : number(0) {}
13
14     virtual ~Polygon() {}
15
16     // 虚函数，计算多边形的周长
17     virtual int perimeter() const = 0;
18
19     // 输出多边形的边数和周长
20     void display() const {

```

输入样例1 复制

```
3
0 32 54 76 88 24 -1
1 32 54
2 3 32
```

输出样例1

```
5 274
4 172
3 96
```

运行 main.cpp

```
0 32 54 76 88 24 -1
1 32 54
2 3 32
5 274
4 172
3 96
```

进程已结束，退出代码为 0

结论：1) GPT4o 响应速度最快，且代码完整无误，是三个模型中唯一能够做对的。
2) 文心一言的响应速度没有 Kimi 和 GPT4o 那么快，是三者中速度最慢的。

6. 性能比较:

- 对比不同模型和提示方法的表现。
- 分析数据，识别各模型的优势和局限。

响应速度和准确性

GPT-4o:

在少次学习测试中表现最佳，响应速度最快，且代码完整无误，表明其处理速度和代码生成的能力在三者中是最优的。

文心一言:

在少次学习测试中响应速度最慢，但在零次学习测试中，其速度与其他两者相等，说明其在没有先前学习经验的情况下能迅速适应，但在需要利用之前经验的任务中速度较慢。

Kimi:

在零次学习的测试中虽然响应速度与其他两者相当，但回答较为全面，显示出在综合信息处理和提供详细回答方面的优势。

多次学习与零次学习的对比

多次学习（少次学习测试）：GPT4o 的表现优于其他两者，说明它在积累经验后能更快地输出更精准的代码。

零次学习：所有模型响应速度相当，但 Kimi 的答案更全面，这可能表明 Kimi 在处理未曾接触过的新情况时，能更好地整合和分析信息。

模型优势与局限

GPT-4o:

优势：快速响应，准确生成代码，适合需要快速且准确编程解决方案的场景。

局限：缺乏数据，不清楚在处理非编程类或更复杂的自然语言处理任务时的表现如何。

文心一言:

优势：在不依赖先前学习的情况下，能迅速适应新任务，表明其具有较强的学习能力。

局限：在有先前学习支持的任务处理上速度较慢，可能需要优化其学习算法以提高效率。

Kimi:

优势：即使在全新的测试环境下，也能提供全面的答案，显示出较强的综合处理能力。

局限：虽然答案全面，但在特定情况下的响应速度和特化任务的处理能力未见明显领先，可能需要在特定领域进行进一步的优化。

实验小结

1. 分析

- 1) GPT-4o 适合需要快速且精确代码解决方案的环境。
- 2) 文心一言 显示出良好的适应性，尤其是在新环境下。
- 3) Kimi 在提供全面答案方面表现出色，适合复杂和需要深入分析的任务。

2. 结论

各模型各有千秋，选择合适的模型需根据具体应用场景和需求来定。ChatMindAi4o 适合快速精确的编程任务，文心一言适合需要快速适应新环境的场景，而 Kimi 则适合需要全面分析的复杂任务。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：