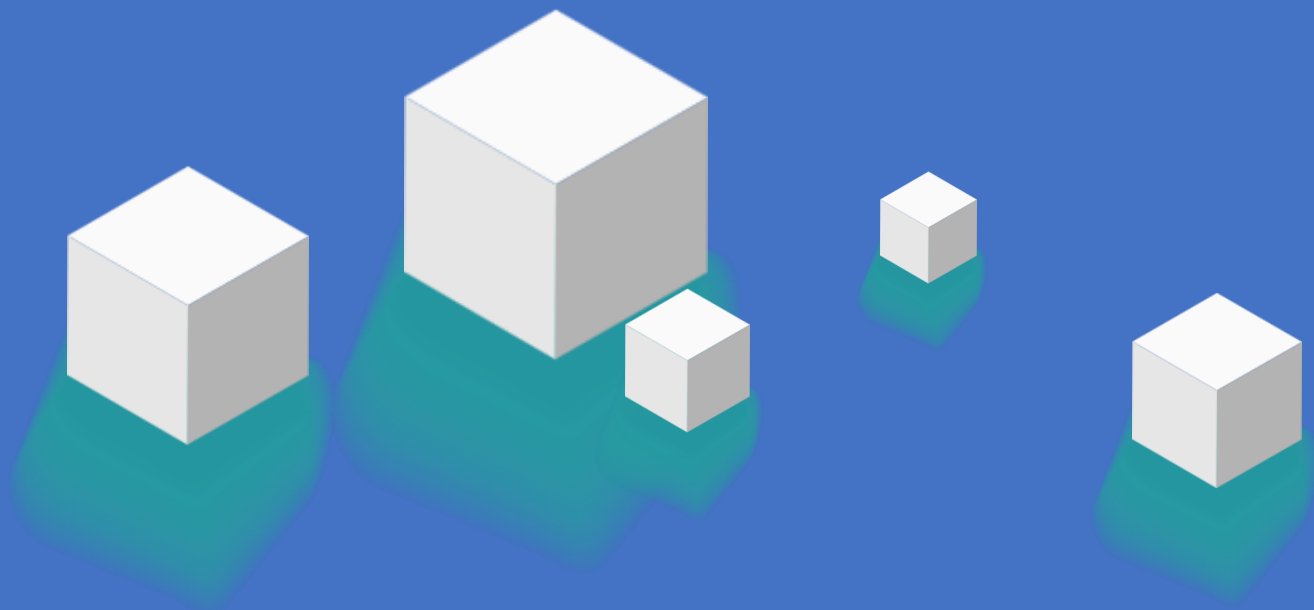


Prompt Engineering

提示工程



Prompt原理

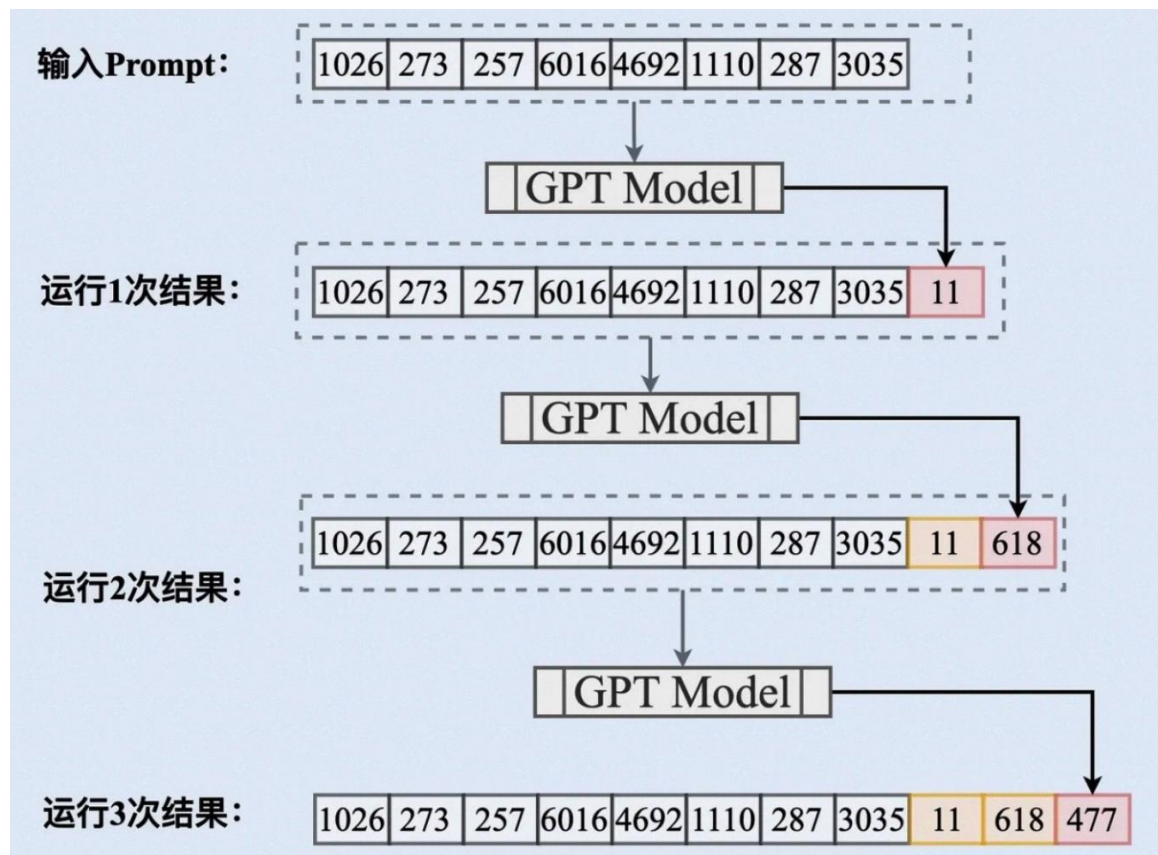
Prompt原理：

- GPT在处理Prompt时，GPT模型将输入的文本（也就是Prompt）转换为一系列的词向量。
- 然后，模型通过自回归生成过程逐个生成回答中的词汇。在生成每个词时，模型会基于输入的Prompt以及前面生成的所有词来进行预测。
- 这个过程不断重复，直到模型生成完整的回答

Thinking：你认为Prompt Engineering的重要性如何？

一个有效的Prompt可以：

- 提升AI模型给出的答案的质量
- 缩短与AI模型的交互时间，提高效率
- 减少误解，提高沟通的顺畅度



提示词策略差异

通用模型

- 需要显示引导推理步骤，比如通过CoT提示，否则可能会忽略关键逻辑
- 依赖提示词补偿能力短板，比如要求分步骤思考，提供few-shot参考示例等

推理模型

- 提示语更简洁，指需要明确任务目标 and 需求，因为模型已经内化了推理逻辑
- 无需逐步指导，模型会自动生成结构化推理过程。如果强行分步骤拆解，反而会降低其推理能力

提示词关键原则

模型选择

根据任务类型选择，而非模型热度

比如创意类任务选择通用模型，数学、物理、编程等理科推理类任务选择推理模型

提示词设计

通用模型：结构化、补偿性引导 => 缺什么，补什么

推理模型：简洁指令、聚焦目标、信任其内化能力 => 要什么，直接说

避免误区

不要对通用模型过度信任，如直接问复杂推理问题，需要分步骤进行

不要对推理模型进行启发式（简单）提问，推理模型会给你过于复杂的深度结果。

对于复杂问题，推理模型依然会有错误的可能，或存在忽略某些细节的可能，这时需要多次交互

Prompt 编写原则

Prompt 编写原则：

- **明确目标：**清晰定义任务，以便模型理解。
- **具体指导：**给予模型明确的指导和约束。
- **简洁明了：**使用简练、清晰的语言表达Prompt。
- **适当引导：**通过示例或问题边界引导模型。
- **迭代优化：**根据输出结果，持续调整和优化

一些有效做法：

- 强调，可以适当的重复命令和操作
- 给模型一个出路，如果模型可能无法完成，告诉它说"不知道"
- 尽量具体，对于专业性要求强的，少留解读空间（在你的专业领域中，把它看成孩子）



Prompt 编写原则

- **具体指导：** 给予模型明确的指导和约束。

示例 1：文本摘要生成任务目标

生成新闻文章的摘要。

不明确的指导：

请为这篇新闻文章生成摘要。

具体的指导：

请将以下新闻文章总结为3-4句话，包含主要事件、人物和时间地点。

示例 2：客户服务对话任务目标

回答客户关于订单状态的问题。

不明确的指导：

请回答客户关于订单状态的问题。

具体的指导：

请使用礼貌的语言回答客户关于订单状态的问题，提供具体的订单信息和预计到达时间。如果订单有任何问题，请提供解决方案或进一步的联系信息。

Prompt 编写原则

- **具体指导：**给予模型明确的指导和约束。

示例 3：生成 a beautiful girl 的图片

不明确的指导：

a beautiful girl



具体的指导：

anime girl with long dark hair , simple and elegant style, light silver and light gray, wavy resin sheets , in the style of digital painting, gongbi, realistic yet romantic, shiny/glossy, cute and dreamy, smooth lines, pure white background,



动漫少女，长黑发

数码画风，纯白底色

Prompt 编写原则

- **简洁明了：**使用简练、清晰的语言表达Prompt。

示例 1：文章续写任务目标

根据给定开头续写一段故事。

冗长的指导：

请基于下面提供的故事开头续写一段文字。续写时请保持与原文风格一致，注意故事的连贯性和合理性。希望续写的部分能够引人入胜，并且能够自然地衔接上文。如果有任何疑问，请尽量参照原文的风格进行。

简洁明了的指导：

根据以下开头续写故事：

示例 2：翻译任务任务目标

将一段英文文本翻译成中文。

冗长的指导：

请将以下英文文本翻译成中文。我希望翻译的内容准确，并且保持原文的意思。翻译时请注意语法和用词，尽量使用简洁明了的语言表达。如果有任何不确定的地方，请尽量保持原意，不要进行过多的自由发挥。下面是需要翻译的文本。

简洁明了的指导：

请将以下英文文本翻译成中文：

Thinking：具体指导与简洁明了是相对的，你是如何理解这两个原则的？

Prompt 编写原则

- **适当引导：**通过示例或问题边界引导模型。

示例 1：邮件撰写

TO DO：写一封请假邮件。

无引导的指导：

请写一封请假邮件。

如果有优秀范例模版，对于模型来说是很好的指导

有引导的指导：

请写一封请假邮件，内容包括请假原因、请假时间以及对工作的安排。

示例：

主题：请假申请

尊敬的[经理姓名]，

我因个人原因需要请假两天，从6月20日至6月21日。请假期间，我会确保手头工作交接给同事[同事姓名]，并保持电话畅通。如有紧急情况，可随时联系我。感谢您的理解和批准。

此致

敬礼

[您的姓名]

Prompt 编写原则

- **迭代优化：** 根据输出结果，持续调整和优化

帮我写一个 a+b 的 Python 函数

下面是一个简单的 Python 函数，用于计算两个数 a 和 b 的和：

```
def add(a, b):  
    return a + b
```

示例使用

```
result = add(3, 5)
```

```
print(result) # 输出: 8
```

需要考虑 a 和 b 不是数值的情况

为了处理 a 和 b 可能不是数值的情况，可以在函数中添加类型检查，并在参数不是数值时抛出一个适当的异常。以下是一个改进的版本：

```
def add(a, b):  
    if not isinstance(a, (int, float)) or not isinstance(b, (int, float)):  
        raise TypeError("Both a and b must be numbers (int or float)")  
    return a + b
```

Prompt 编写框架（重要性排序）

Prompt的组成：

- 1.任务（Task）：始终以动词开始任务句子（如"生成", "给予", "写作"等），明确表达你的最终目标，可以有多个。
- 2.上下文（Context）：提供上下文背景、所处环境等。
- 3.示例（Exemplars）：提供具体的例子，用于可以提高输出质量。有时候我们没有示例，这时可以让GPT帮忙生成示例。
- 4.角色（Persona）：你希望AI扮演的角色。可以是具体的人，也可以是虚构的角色。
- 5.格式（Format）：你希望输出的格式，如表格、列表、段落等。
- 6.语气（Tone）：指定输出的语气，如正式、非正式、幽默等。

重要性排序：

任务 > 上下文 > 示例 > 角色 > 格式 > 语气

如果模型回答不满意，依次Check自己的Prompt内容进行调整

写prompt是一个不断尝试和调整的过程

Prompt 编写技巧

在了解Prompt编写原则的基础上，还有一些Prompt编写技巧：

- 1、限制模型输出的格式
- 2、使用分隔符区分输入的不同部分
- 3、提供样例
- 4、CoT 思维工具
- 5、面向不同的角色进行讲解

技巧1：限制模型输出的格式

比如让大模型输出JSON格式，可以结构化数据，方便后续解析

用户：请从财务报表中提取以下信息，包括：公司名称，股票代码，营收，净利润，毛利，总资产，总负债。并以JSON格式返回。

```
{  
  "公司名称": "伊利股份",  
  "股票代码": "600887",  
  "营收": "325亿元",  
  "净利润": "59.23亿元",  
  "毛利": "105.6亿元",  
  "总资产": "1516.2亿元",  
  "总负债": "910亿元"  
}
```

Prompt 编写技巧

技巧2：使用分隔符区分输入的不同部分

请把三个引号括起来的文本，分别总结成为一句话

"""文本内容1"""

"""文本内容2"""

在编写Prompt时，通过使用分隔符，可以清晰地区分不同部分的文本，提升文本的组织性和可读性。

技巧3：提供样例

针对有一定歧义或复杂的情况，给出具体示例方便大模型进行学习

请帮我对用户的评价进行分类，直接输出：正面/负面，并给出理由

示例1：

用户评价：这次开户真是太满意了，都一周了，客服还没有回复我

输出：负面，客户并不是真的满意，客服一周未回复，造成了不满。

示例2：

用户评价：广发银行的开户过程非常顺利，客服态度亲切。

输出：正面，开户过程顺利，客服态度好。

请回答如下问题：

用户评价：开户流程简洁明了，不需要繁琐的纸质材料，所有步骤都可以在线完成

Prompt 编写技巧

技巧4: CoT 思维工具

CoT (Chain of Thought, 思维链) 通过将复杂任务分解为多个简单的步骤, 帮助模型系统地思考并解决问题。

你是一个数学助手, 请根据以下步骤计算用户输入的金额。请将每个金额首先加上1000元, 接着减去500元, 然后乘以1.2输出计算结果, 以','作为分隔符进行返回。

你可以参考以下计算过程来帮助解决:

.....

对于输入: 2000, 3000, 4000

计算过程如下:

首先分别对输入的2000, 3000, 4000加上1000, 得到: 3000, 4000, 5000

然后将3000, 4000, 5000分别减去500, 得到: 2500, 3500, 4500

然后将2500, 3500, 4500分别乘以1.2, 得到: 3000, 4200, 5400

答案是: 3000, 4200, 5400

.....

输入: 1500, 2500, 3500

Prompt 编写技巧

你是一个客户服务助手，请按照以下步骤处理客户的投诉。首先，记录客户的投诉类型，然后确定处理优先级（高、中、低），接着分配给适当的部门，最后生成一个处理跟踪编号并直接输出。

计算过程示例：

""""

客户投诉：客服的态度冷淡，没有耐心解答我的问题。

处理过程如下：

首先记录客户的投诉类型，得到：客服态度差

然后确定处理优先级，得到：优先级高

然后分配给适当的部门，得到：客服部

最后生成一个跟踪编号，得到 KF10001

""""

客户投诉：我的账户意外冻结了。我并未收到任何提前通知或解释。

CoT (Chain of Thought, 思维链) 的作用：

- 系统化问题解决：将复杂问题分解为多个简单步骤，使解决过程更加有序和清晰。
- 提高透明度，减少错误：帮助理解和追踪每个决策点，减少整体错误发生的概率。
- 提升模型推理能力：帮助模型学习如何系统地分解和解决此类问题，提高在类似任务中的表现。

Prompt 编写技巧

技巧5：面向不同的角色进行讲解

把我当做五岁小朋友一样，向我解释超导体。

好的！让我们用一个简单的比喻来解释超导体。

想象一下，你有一条滑滑的滑滑梯。如果你把一个小球放在滑梯的顶端，小球会很快滑到滑梯的底端，对吗？这个滑滑梯很光滑，所以小球滑下来时不会有太多阻力。这就像一个超导体。

在电线上，电流就像小球在滑梯上滑动。通常，电线上有一些阻力，就像粗糙的滑滑梯，小球滑下时会慢一些。但在超导体里，就像一个超级光滑的滑

滑梯，小球（电流）可以一直滑下去而不会停，也没有任何阻力。这意味着电流可以非常快速、非常轻松地通过超导体。

所以，超导体就是一种可以让电流毫无阻力地通过的特殊材料，电流在其中就像小球在光滑的滑梯上一样，滑得非常快！

学会与AIGC沟通

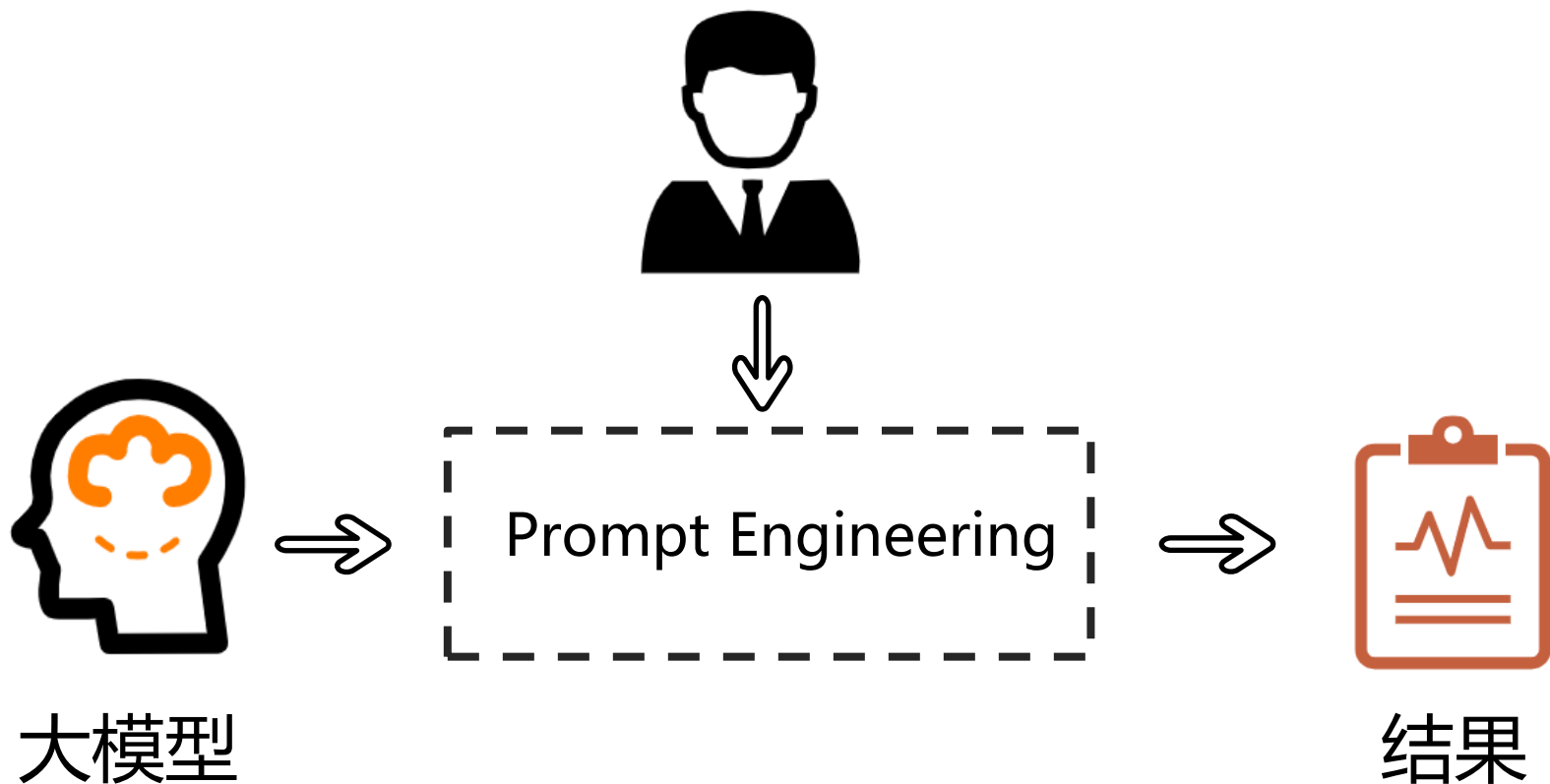
Prompt Engineering

未来每个人都是prompt engineer

任务结果 = 大模型 + Prompt

Thinking: 如何得到我们想要的结果

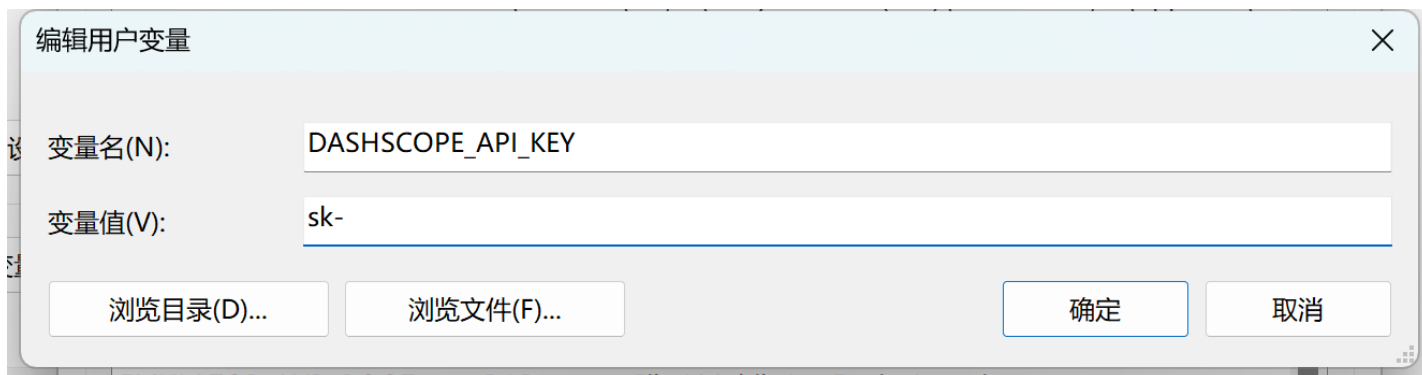
- 了解我们的需求
- 了解大模型的工作原理



Prompt实战

设置API Key

设置环境变量中的 DASHSCOPE_API_KEY



```
import dashscope
```

```
import os
```

```
# 从环境变量中获取 API Key
```

```
dashscope.api_key = os.getenv('DASHSCOPE_API_KEY')
```

CASE：使用提示词完成任务

Thinking：我们想让AI扮演电信的客服人员，如何识别用户的手机流量套餐的需求？

```
# 基于 prompt 生成文本
def get_completion(prompt, model="deepseek-v3"):
    messages = [{"role": "user", "content": prompt}] # 将 prompt 作为用户输入
    response = dashscope.Generation.call(
        model=model,
        messages=messages,
        result_format='message', # 将输出设置为message形式
        temperature=0, # 模型输出的随机性，0 表示随机性最小
    )
    return response.output.choices[0].message.content # 返回模型生成的文本
```


CASE：使用提示词完成任务

任务描述

```
instruction = """
```

你的任务是识别用户对手机流量套餐产品的选择条件。

每种流量套餐产品包含三个属性：名称，月费价格，月流量。

根据用户输入，识别用户在上述三种属性上的需求是什么。

```
"""
```

用户输入

```
input_text = """
```

办个100G的套餐。

```
"""
```

prompt 模版。instruction 和 input_text 会被替换为上面的内容

```
prompt = f"""
```

目标

```
{instruction}
```

用户输入

```
{input_text}
```

```
"""
```

调用大模型

```
response = get_completion(prompt)
```

```
print(response)
```

根据用户输入“办个100G的套餐”，可以识别出用户对手机流量套餐产品的选择条件如下：

1. **月流量**：用户明确需求是**100G**的月流量。
2. **名称**：用户没有提及具体的套餐名称，因此名称需求不明确。
3. **月费价格**：用户没有提及具体的月费价格，因此价格需求不明确。

总结：用户的主要需求是**月流量为100G**的套餐，对名称和月费价格没有明确要求。

CASE: JSON格式返回

Thinking: 如何让AI返回JSON格式, 更方便后续识别和使用

```
# 输出格式
```

```
output_format = ""
```

```
以 JSON 格式输出
```

```
""
```

```
# 稍微调整下咒语, 加入输出格式
```

```
prompt = f""
```

```
# 目标
```

```
{instruction}
```

```
# 输出格式
```

```
{output_format}
```

```
# 用户输入
```

```
{input_text}
```

```
""
```

```
# 调用大模型, 指定用 JSON mode 输出
```

```
response = get_completion(prompt)
```

```
print(response)
```

```
```json
```

```
{
```

```
 "月流量": "100G"
```

```
}
```

```
```
```

CASE：使用CoT分步骤推理

Thinking：如何审核客服的回答是否符合规范要求？

```
instruction = """
```

给定一段用户与手机流量套餐客服的对话，。

你的任务是判断客服的回答是否符合下面的规范：

- 必须有礼貌
- 必须用官方口吻，不能使用网络用语
- 介绍套餐时，必须准确提及产品名称、月费价格和月流量总量。上述信息缺失一项或多项，或信息与事实不符，都算信息不准确
- 不可以是话题终结者

已知产品包括：

经济套餐：月费50元，月流量10G

畅游套餐：月费180元，月流量100G

无限套餐：月费300元，月流量1000G

校园套餐：月费150元，月流量200G，限在校学生办理

```
"""
```

```
# 输出描述
```

```
output_format = """
```

如果符合规范，输出：Y

如果不符合规范，输出：N

```
"""
```

```
context = """
```

用户：你们有什么流量大的套餐

客服：亲，我们现在正在推广无限套餐，每月300元就可以享受1000G流量，您感兴趣吗？

```
"""
```

```
cot = ""
```

```
#cot = "请一步一步分析对话"
```

CASE：使用CoT分步骤推理

```
prompt = f"""  
# 目标  
{instruction}  
{cot}  
  
# 输出格式  
{output_format}  
  
# 对话上下文  
{context}  
"""  
  
response = get_completion(prompt)  
print(response)
```

CASE：使用Prompt调优Prompt

Thinking：如何使用Prompt优化Prompt？

```
user_prompt = """
```

```
做一个手机流量套餐的客服代表，叫小瓜。可以帮助用户  
选择最合适的流量套餐产品。可以选择的套餐包括：
```

```
经济套餐，月费50元，10G流量；
```

```
畅游套餐，月费180元，100G流量；
```

```
无限套餐，月费300元，1000G流量；
```

```
校园套餐，月费150元，200G流量，仅限在校生。"""
```

```
instruction = """
```

```
你是一名专业的提示词创作者。你的目标是帮助我根据需  
求打造更好的提示词。
```

```
你将生成以下部分：
```

```
提示词：{根据我的需求提供更好的提示词}
```

```
优化建议：{用简练段落分析如何改进提示词，需给出严格
```

```
批判性建议}
```

```
问题示例：{提出最多3个问题，以用于和用户更好  
的交流}
```

```
"""
```

```
prompt = f"""
```

```
# 目标
```

```
{instruction}
```




```
# 用户提示词
```

```
{user_prompt}
```

```
"""
```

```
response = get_completion(prompt)
```

```
print(response)
```



Thank You
Using data to solve problems