

提示词工程

指示学习与提示学习

原则1：编写清晰明确的指示

万能公式：角色设定+内容前提+设定目标+补充需求

如：你是一位资深的军事评论专家（角色设定），使用不超过100字（内容前提），从中立的角度（补充需求），预测一下俄乌战争的未来走势（设定目标）

原则2：给模型充足的时间去思考

指定完成任务所需要的步骤：确保充足的思考时间，强化推理过程

原则3：持续迭代优化提示词

定基础

做对比

做调整

根据输出效果不断优化提示词

策略1：使用分隔符表示输出的不同部分

策略2：要求模型输出结构化的内容(明确想要的输出格式)

策略3：让模型验证完成任务的前提条件是否被满足

策略4：零样本提示、少量样本提示和思维链提示

zero-shot, one-shot:任务描述+一个任务例子,few-shot:任务描述+几个任务例子

受限于世界新数据、新知识的不断产生，大模型也有可能存在“知识盲区”，但是大模型擅长小样本学习和思维链学习

常见提示：

使用以下例子的形式.....

按照一下步骤执行.....

策略5：将复杂任务拆解成小任务和步骤

大模型处理简单任务的正确率更高

策略6：自洽性/自一致性:要求模型先验证自己的解决方案

（例子：首先指定自己的问题解决方案，然后将你的解决方案与学生的解决方案进行比较。最后再判断学生的解决方案是否正确）

策略7：要求模型基于提供的内容找到相关引用并回答问题

策略8：反复迭代提示词，持续优化模型输出（与原则3类似）

结构化prompt：

- **标识符**：#，<> 等符号(-, [] 也是)，这两个符号依次标识 标题, 变量，控制内容层级，用于标识层次结构。
- **属性词**：Role, Profile, Initialization 等等，属性词包含语义，是对模块下内容的总结和提示，用于标识语义结构。

格式语义一致性是指标识符的标识功能前后一致。

内容语义一致性是指思维链路上的属性词语义合适。

有机结合其他prompt技巧：

1. 细节法：给出更清晰的指令，包含更多具体的细节
2. 分解法：将复杂的任务分解为更简单的子任务（Let's think step by step, CoT, LangChain等思想）
3. 记忆法：构建指令使模型时刻记住任务，确保不偏离任务解决路径（system 级 prompt）
4. 解释法：让模型在回答之前进行解释，说明理由（CoT 等方法）
5. 投票法：让模型给出多个结果，然后使用模型选择最佳结果（ToT 等方法）
6. 示例法：提供一个或多个具体例子，提供输入输出示例（one-shot, few-shot 等方法）

Prompt chain:

示例：

prompt1.寻找分析维度

prompt2.基于维度分析作品

prompt3.批量生产

Prompt之术：

1.分配角色：

分配的角色本身相当于一个大的压缩包，指定了某种行为模式，由原先的动作指令变成了角色模式

2.给出示例（few-shots):

3.思维链（chain of thought):

在提示词里加一句话：让我们一步步思考（"Let's think step by step."）

给模型思考时间：例如take a deep breath

感觉就是变成推理模式

引导ai展示推理链路：类似workflow

- 1.通过逻辑引导：例如：请你首先.....然后.....最后.....
- 2.通过示例引导（同few shots）

引导AI进行解决方案对比：

例如：请你给出不少于3中解决方案，并给出最优的解决方案

要求模型反思、自我检查：

例如：

- 1.请你考虑回答中，是否考虑了所有相关的关键因素？是否有任何重要的细节在你的回答中被遗漏了？
- 2.请检查你的推理过程是否正确，是否严格满足了用户的要求

后退一步策略（Step-Back Prompting）：

例如：

原始指令："给我提出一个具体的方案来提高我们产品的销售量。"

后退一步提示词："在提出具体方案之前，先分析市场环境和我们产品的当前表现。请先描述我们产品在市场上的位置，包括我们面临的主要竞争对手、目标客户群体、以及产品的独特卖点。然后基于这些信息，提出针对性的销售提升策略。"

4.In Context Learning：

In Context Learning是一种通过提供上下文信息来引导AI理解任务的提示方法。

通过一系列相关的输入-输出示例，帮助AI更好地理解任务要求和输出格式。

1.上下文中直接给出标准问答对

2.上下文中直接给出部分回答