一个完整的Prompt应该包含清晰的指示、相关的上下文、有助于理解的例子、明确的输入以及期望的输出格式描述。

**一、Prompt编写原则**

**编写清晰具体的指令**

1.使用明显的分隔符将（待处理文本）和（提示语）分隔开

2.（可以要求结构化输出json/html）

3.要求模型检查是否满足条件

4.提供少量提示（输出案例）few-shot

**给模型思考的时间**

指定完成任务所需步骤

示例：

“““待处理文本”””

执行以下操作：

1.

2.

3.

...

**局限：**

模型会生成一些看似真实实则编造的知识，通过编写（清晰具体的指令）和（给模型思考的时间）可以更大程度上避免这种情况的发生

**二、迭代提示开发**

一个完美的prompt通常不是一步到位的，有一个好的迭代过程，就可以得到一个适合任务的完美prompt

1.初始prompt

2.生成文本字数不符合预期：限制或要求达到给定字数限制

3.文本关注了错误的细节：可以要求专注于与目标更相关的方面

4.可能需要表格形式的描述。

**总结：**首先编写一个最初的提示，然后通过迭代，逐步完善它，直至得到最终需要的结果。

范式：{

“““待处理文本””” #提供背景信息

概括文本（一句话）

根据以上文本执行以下操作：

1.赋予角色（角色+任务）

2.发出指令

}

**注意：**

更好的提示是非常具体、简洁和直接的。

避免说不要做什么，而是说要做什么。

**方法：**

Zero-shot

few-shot 输出示例

Cot 思维链 将要执行的任务拆解为多个步骤“请一步步进行推理并得出结论”

LtM提示法 要大模型帮助你生成思维链，再通过COT得出结果

COT+ SC 则是在增加深度的基础上再增加宽度。

TOT思维树 与 COT+SC 类似，也是先分步骤，然后每个步骤多个结果，不同的是，COT+SC 是并列输出多个结果。而 TOT 是在每一步的多个结果中进行选择，然后再进行下一步，输出多个结果。

Step-Back写 Prompt 的时候，往后退一步，想想在回答这个问题之前需要先回答什么问题。