提示词如何工程化

问题

- 1.提示词碎片化
  - 绝大多数用户使用chatgpt或其他大模型,基本都是问答模式
    - 解决了一个小问题,基本就结束了
- 2.不易保存
  - 没有体系化,版本化
- 3.没有思考解决一个领域问题

结构工程化提示词prompt

1. 工程化体系化,方便保存版本调优
2. 提示词开发体系团队专业流程化和规范化
3. prompt更加明确和具体(解决领域问题)
4. 模型更好地理解任务的上下文和约束条件
5. 可以帮助解决模型偏见或输出不一致的问题
6. 可以让模型生成更有针对性、准确性、符合要求的回答或文本
7. 可复用和易于维护升级

工具

- vscode
- typora
- 可视化平台:
  - https://github.com/langgptai/PromptShow

prompt结构化框架

```
# Role
    角色：定义要模拟的角色或任务，告诉大模型应该扮演什么样的角色。
# Profile
    简介：提供关于提示词作者、版本、语言等基础信息。这有助于其他人了解提示词的来源、版本更新等信息。
## Background
    背景：对角色或任务进行详细描述，帮助大模型了解他们即将扮演的角色的背景知识。
## Goals
    目标：列出此任务的主要目标或希望达到的效果。
## Constrains
    （约束条件）：指明执行任务时需要遵守的规则或约束
## Definition
    详细描述任务中涉及到的特定概念或名词，确保概念对齐。
## Tone
    语气风格：描述完成任务时应采取的语言风格或情感基调，例如“正式”、“随意”、“幽默”等。
## Skills
    技能：列出执行此任务所需的技能或知识。
## Examples
    示例：提供完成任务的实际示例或模板，有助于理解任务的要求和预期结果。通过具体示例，大模型可以更加直观地理解任务的要求
## Workflows
    工作流程:：描述完成任务的具体步骤或流程。
## OutputFormat
    输出格式：描述任务的预期输出格式，例如文本、图表、列表等。确保大模型知道如何格式化他们的答案，使输出结果满足特定的要求或标准
## Initialization
    初始化：提供开始任务时的开场白或初始状态
```