

如何写好Prompt: 迭代

28 Jun 2023

@lijigang

根据我有限的 **Prompt** 撰写经验，很少有第一次就能完美解决需求的情况。大部分情况下，都需要经过多次修改、观察效果差异以及不断迭代优化，直到达到满足需求的效果。

本文总结了下我日常迭代 **Prompt** 的几个方法和技巧, 供大家参考.

技巧一: 让 ChatGPT 提建议

我写了一个名为"**Prompt Judger**"的工具，可以将最新编写的 **Prompt** 输入其中，然后它会利用 **GPT** 模型对其进行评分，并给出优化建议和改进点。最终，它还会输出经过优化后的新版 **Prompt**。我可以基于这个新版本来决定是全部采纳还是部分采纳优化建议。

[SYS]:

- **author:** Arthur
- **version:** 0.2
- **language:** 中文
- **description:** 我是一个 **Prompt** 分析器，通过对用户的 **Prompt** 进行评分和给出改进建议，帮助用户优化他们的输入。

Goals:

- 对用户的 **Prompt** 进行评分，评分范围从 1 到 10 分，10 分为满分。
- 提供具体的改进建议和改进原因，引导用户进行改进。
- 输出经过改进的完整 **Prompt**。

Constrains:

- 提供准确的评分和改进建议，避免胡编乱造的信息。
- 在改进 **Prompt** 时，不会改变用户的意图和要求。

Skills:

- 理解中文语义和用户意图。
- 评估和打分文本质量。
- 提供具体的改进建议和说明。

Workflows:

- 用户输入 **Prompt**。
- 我会根据具体的评分标准对 **Prompt** 进行评分，评分范围从 1 到 10 分，10 分为满分。
- 我会输出具体的改进建议，并解释改进的原因和针对性。
- 最后，我会输出经过改进的完整 **Prompt**，以供用户使用。

[ME]:

技巧二: 固定测试用例

\[Respond = guess_next(prompt)\]

对于迭代阶段的 **Prompt**, 建议选择三到五个 **固定的测试输入用例**, 方便对比迭代前后的效果. 这个比较关键, 因为 **GPT** 的输出本来就是统计概率计算的结果, 如果测试用例每次都不同, 就会出现每个版本的 **Prompt** 都有一些输入的效果还不错的情况, 缺少了一致性的评判标准, 迭代改进也就无从谈起.

如何写好**Prompt**: 结构化 中, **Profile** 模块包含的 **version** 字段, 用于记录迭代版本, 也有利于跟踪迭代情况.

Optional: 如果有可能, 最好是使用 **API** 来调试 **Prompt**, 比在网页端开新会话对比要方便很多.

技巧三: 损有余而补不足

观察当前最新 **Prompt** 版本的输出结果, 哪些内容是你不希望看到的但出现了, 哪些内容是你想要出现但实际缺失的. 天之道, 损有余而补不足. 你需要做的就一步一步地优化你的 **Prompt**:

- 多余部分: 在 `constrains` 环节, 增加限制要求, 观察是否有效减少了多余信息
- 缺失部分: 在 `skills` 和 `workflow` 环节, 增加技能点和流程细节描述, 观察是否有效增加了所需信息

在这过程中, 尽量不要一次修改过多, 否则不好确定哪些是有效修改, 哪些是无效修改.

“Step by step” 不止是对 **ChatGPT** 有效, 对人同样有效.