

Prompt 设计

目标：选择适当的提示格式和语言，以清晰明确地表达用户的意图。

过程：首先，需要明确目的，在使用 **prompt** 之前，明确目的是什么，即准备让模型做什么样的事情。在明确目的之后，然后构造 **prompt**，构造的过程，主要包括如下关键点：

提供更多 **query** 相关的细节，可以获得更准确的答案

原始 query	更好的 query
编写一篇太空探索的文章	为一群 10-15 岁的孩子编写一篇介绍太空探索历史的文章。
请写一篇文章，关于环保的，500 字	请撰写一篇 500 字的文章，讨论城市绿化对空气质量改善的影响。文章应包括以下方面的内容：城市绿化的定义，如树木和公园的增加，它们如何减少空气中的污染物，以及在城市规划中推广城市绿化的可行性措施。请提供相关数据和案例研究以支持你的论点。

使用分隔符去更清晰地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'''
bash
```

引导结构和组织要求

在这个过程中，可以为任务提供明确的结构和组织要求。指明任务的各个部分应该包括什么内容，有助于模型组织思维 and 材料。

```
请撰写一篇 500 字的文章，讨论城市绿化对空气质量改善的影响。文章应包括以下方面的内容：
1. 引言：介绍城市绿化和其重要性。
2. 影响空气质量的机制：解释树木和公园如何减少空气中的污染物。
3. 可行性措施：讨论在城市规划中推广城市绿化的方法和挑战。
4. 数据和案例研究：提供相关数据和至少两个城市绿化成功案例，以支持你的论点。
5. 结论：总结城市绿化对空气质量的积极影响
plain
```

限制模型输出格式

最后，针对模型的输出，可以限制输出的格式，一方面可以提高可读性，使结果更清晰明了；另一方面也可以方便后续的处理，提高稳定性。

以提取“病症”的任务来展示下如何去限制模型的输出格式。要求直接以 json 的格式返回。请提取参考资料中的所有病症，并且以 json 格式返回。

回答满足下面的格式要求：

1、以 json 的格式返回答案，json 只包括一个 key，key="disease"，对应的值为列表，存储参考资料中的病症。

参考资料：

"""

失眠在《内经》中称为“目不瞑”、“不得眠”、“不得卧”，其原因主要有两种：一是其他病症影响，如咳嗽、呕吐、腹满等，使人不得安卧；二是气血阴阳失和，使人不能入寐。中医常用养心安神的方法治疗失眠，既可治标、又可治本，还可以避免西药安眠药容易成瘾的弊端。中医认为，失眠多因脏腑阴阳失调，气血失和所致。正如《灵枢大惑论》中记载：“卫气不得入于阴，常留于阳，留于阳则气满；阳气满则阳骄盛，不得入于阴则阴气虚，故目不瞑矣。”在临床上，治疗失眠应着重调理脏腑及气血阴阳，以“补其不足，泻其有余，调其虚实”，可采取补益心脾、滋阴降火、交通心肾、疏肝养血、益气镇惊、活血通络等治法，使气血和畅，阴阳平衡，脏腑功能恢复正常。

"""

sql

Prompt 优化

让模型扮演一个角色

让模型扮演一个具体的角色，模型的输出会更符合人类的表达方式，从而更容易被人类理解；同时输出也会更加一致。例如，在问答系统中，让模型扮演一个特定领域的专家可以使其回答更符合该领域的知识和语言习惯，从而提高回答的一致性。

比如下面的案例，让模型分别扮演科学家和玄幻小说家生成一篇文章，文章的主题是：“黑洞是如何形成的”。在科学家的角度下，模型基于科学事实首先解释了黑洞是什么，然后回答了黑洞的形成过程；而在玄幻小说家的角度下，模型此时的输出不再基于科学事实，而是完全虚构，并且给人更多神秘的感觉，勾起读者的兴趣。

你是一名科学家，请你就关于“黑洞是如何形成的？”的主题写一篇200字以内的文章

黑洞是一种极其密集的天体，它的引力非常强大，甚至连光都无法逃脱。黑洞的形成通常与恒星的演化有关。当一个非常大质量的恒星耗尽了其核心燃料时，它会在内部崩塌，形成一个非常紧密的天体，称为“中子星”。如果这个恒星的质量足够大，那么它的引力将变得如此之强，甚至连中子星也无法承受，于是它将继续崩塌，形成一个黑洞。在这个过程中，恒星的物质被吸入黑洞，形成一个事件视界，一旦任何东西进入事件视界，就再也无法逃脱黑洞的引力。因此，黑洞是由恒星演化而来的，并且是宇宙中最神秘和最具有挑战性的天体之一。

你是一名玄幻小说家，请你就关于“黑洞是如何形成的？”的主题写一篇200字以内的文章

黑洞，乃是宇宙至阴至邪之物，其形成之因，众说纷纭。

有人说，黑洞是由宇宙间最强大的能量所形成，这种能量可以扭曲时空，毁灭一切；也有人说，黑洞是由神灵掌控的神秘力量所形成，这种力量可以掌控宇宙的命运。

然而，在我看来，黑洞的形成并非如此简单。黑洞的形成，是由宇宙间至阴至邪之物所化，这些物体可以是邪恶的灵魂、黑暗的力量、甚至是宇宙间最强大的能量。当这些至阴至邪之物聚集在一起时，便会形成一个黑洞，吞噬一切。

黑洞的形成，是宇宙间最神秘、最强大的力量之一，它们的存在，让宇宙充满了未知和危险。

提供样例

一般情况下，可以通过加入指令，来优化提示词，但是有时指令可能不容易描述清楚，提供示例会更容易。比如：我们让模型充当一个文本二分类器，对用户的评价进行二分类，分类结果为：正面评价或者负面评价。

请帮我分辨用户输入文本的类别是正面评价还是负面评价，请直接输出：正面评价/负面评价。
用户输入：我昨天看了这部电影，我觉得还可以，虽然有些部分有点无聊。

正面评价

请帮我分辨用户输入文本的类别是正面评价还是负面评价，请直接输出：正面评价/负面评价。
用户输入：我上周去看了这部电影，简直浪费时间。情节枯燥无味，演员的表现也差强人意。我真的后悔看了。

负面评价

可以看出，正常情况下，模型暂时对于非绝对负面的评价都是分类为正面评价；而对于完全负面的评价才会分类为负面评价。但是我们希望**模型对用户的评价是绝对正面时，才输出正面评价；否则都输出负面评价**。比如，“我最近在这家餐厅用餐，还行，但也不是特别惊艳”这个 **case**，我们希望模型输出“负面评价”。此时，可以提供一些示例供模型来参考学习。

请根据以下分类的方式，帮我分辨用户输入文本的类别是正面评价或是负面评价，请直接输出：正面评价/负面评价。

请参考如下样例：

示例 1：

用户输入：我昨晚去了这家餐厅，他们的食物和服务都令人惊艳。我绝对会再次光顾。

输出：正面评价

示例 2：

用户输入：这本书我看过，部分情节还行，但是整体情节拖沓，比较一般。

输出：负面评价

示例 3：

用户输入：我昨天看了这部电影，我觉得还可以，但是有些部分也有点无聊。

输出：负面评价

示例 4：

用户输入：我上周去看了这部电影，简直浪费时间。情节枯燥无味，演员的表现也差强人意。我真的后悔看了。

输出：负面评价

请回答如下问题：

用户输入：我最近在这家餐厅用餐，还行，但也不是特别惊艳。

输出：

plain

除此之外，对于一些比较难的 **case**，只给 **label**，可能不好理解，也可以尝试在给示例的同时，给出理由，这样模型会更容易理解。

请根据以下分类的方式，帮我分辨用户输入文本的类别是正面评价或是负面评价，请输出：正面评价/负面评价，并给出理由。

请参考如下样例：

示例 1：

用户输入：我昨晚去了这家餐厅，他们的食物和服务都令人惊艳。我绝对会再次光顾。

输出：正面评价，理由：食物和服务都令人惊艳，而且顾客会再次光临

示例 2：

用户输入：这本书我看过，部分情节还行，但是整体情节拖沓，比较一般。

输出：负面评价，理由：整体情节拖沓

示例 3:

用户输入: 我昨天看了这部电影, 我觉得还可以, 但是有些部分也有点无聊。

输出: 负面评价, 理由: 这部电影有些部分也有点无聊

示例 4:

用户输入: 我上周去看了这部电影, 简直浪费时间。情节枯燥无味, 演员的表现也差强人意。我真的后悔看了。

输出: 负面评价, 理由: 情节枯燥, 演员表现也差强人意, 用户后悔看了

请回答如下问题:

用户输入: 我最近在这家餐厅用餐, 还行, 但也不是特别惊艳。

输出:

bash

指定完成任务需要的步骤

完成特定任务, 如果有明确的步骤, 通过指定完成任务需要的步骤, 可帮助模型按照正确的顺序执行, 从而更好地生成回复。

请按照以下步骤写一个故事:

- 1 先设定故事背景和角色。
- 2 描述角色的目标和遇到的困难。
- 3 讲述角色如何克服困难并最终实现了目标。
- 4 最后以一个有趣的结局来结束故事。

bash

这个 **prompt** 优化的示例中, 明确指定了完成故事任务的步骤, 模型可以更有条理地构思和组织故事结构, 从而提高故事的质量和连贯性。

提供参考内容

针对特定领域的问题, 如果涉及到专业知识, 提供相关的参考内容可以帮助模型更好地理解问题的背景和上下文, 从而减少幻觉的出现。

请参考如下文档, 回答用户问题:

###

文档名: 文档内容

###

问题: [问题描述]

```
bash
```

给模型“思考”时间

CoT（Chain of Thought, CoT）是一种思维工具，通过逐步延伸和拓展一个主要想法，帮助人们进行更深层次的思考，并得出更复杂、更全面的结论。在一些逻辑推理、数学运算等场景下，考虑使用 CoT 分解问题，通过输出一些推断过程，可以增加模型输出正确结果的概率。

zero-shot

在 prompt 中加上 `Let's think step by step` 或者 `让我们一步一步分析思考`。

few-shots

以下是将用户输入的数字进行一系列加减乘除的例子。

你是一个计算器，请你将用户输入的数字分别加上 2，减去 3，乘以 3，除以 2 后直接输出计算结果，以 ',' 作为分隔符进行返回。

示例如下：

```
"""
```

输入：1, 2, 3, 4, 5

答案：0, 1.5, 3, 4.5, 6

```
"""
```

输入：2, 4, 6, 8, 10

```
sql
```

我们会发现，模型无法给出正确的答案，所以，让模型处理这种复杂、多步骤的问题时，我们可以遵循以下步骤：提供实例 > 分解问题 > 提供解释。

以下是将 CoT 融入到 prompt 的例子，分步骤地提供解释。

你是一个计算器，请你将用户输入的数字分别加上 2，减去 3，乘以 3，除以 2 后直接输出计算结果，以 ',' 作为分隔符进行返回。

你可以参考以下的计算过程来帮助解决，

```
"""
```

对于输入：1, 2, 3, 4, 5

计算过程如下。

首先分别对输入 1, 2, 3, 4, 5 加上 2，得到：3, 4, 5, 6, 7

然后将 3, 5, 6, 7, 8 分别减去 3，得到：0, 1, 2, 3, 4

然后将 0, 2, 3, 4, 5 分别乘以 3，得到：0, 3, 6, 9, 12

最后将 0, 6, 9, 12, 15 分别除以 2，得到：0, 1.5, 3, 4.5, 6

答案是：0, 1.5, 3, 4.5, 6

```
"""
```

输入：2, 4, 6, 8, 10

```
sql
```

Prompt 评测

系统地测试变更

在完成上面的 **prompt** 设计和优化之后，**prompt** 会发生变更，但是如何判断这个变化，是否真的使系统变得更好。一般是先通过看几个例子来确定哪个更好，但是由于样本量小，很难区分是真正的改进还是运气导致，所以需要设计一个好的评估集，在经过若干次迭代之后，用评估集来判定系统是否真正得到了改进。

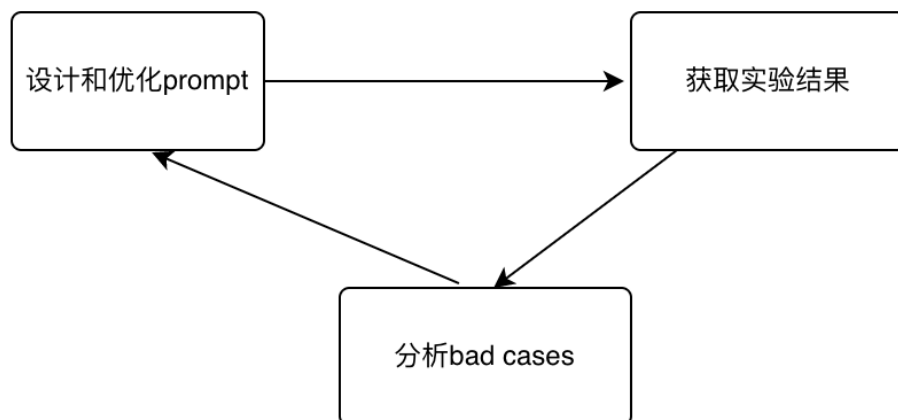
好的评估集，一般具有如下特征：

- 多样性：评估集中应该包含足够的多样性，以涵盖不同的领域、话题和语境。
- 高质量：评估集中的数据应该是高质量的，能准确反映了真实的业务情况。
- 规模适中：评估集的规模应该足够大，以便能够充分评估 **prompt** 的性能，但又不能太大以至于导致计算资源的过度消耗。通常，几百到几千个样本是一个合理的范围。

重复和迭代

prompt 的生成过程，实际上是一项实验性很强的过程，在这个过程中，需要不断地尝试和调整不同的方法，以找到最优的 **prompt**。一个典型的迭代路径是：首先完成 **prompt** 设计，接着基于设计好的 **prompt** 获取实验结果，分析 **bad cases**，解 **bad cases**，并进一步优化 **prompt**，通过多次的重复和迭代，直到达到一个最优效果。

prompt 工程迭代过程如下：



最后，进一步需要说明的是，获取模型的最佳输出不仅需要优秀的 **prompt**，同时也取决

于用户提供的反馈和修正。**prompt** 优化完成之后，通过线上持续的反馈和修正，模型才能更顺利地理解并满足用户需求。

附录

参数设置

关键参数

使用 **prompt** 时，您会通过体验中心或 **API** 与大语言模型进行交互。通过配置一些参数，可以获得不同的提示结果。

- **Temperature**: 该参数用于控制模型输出结果的随机性。**Temperature** 的值越高，输出的随机性就越大，**Temperature** 的值越低，输出的随机性就越小。在实际应用方面，对于分类任务，我们可以设置更低的 **temperature** 值，以促使模型返回更真实和简洁的回答。对于诗歌生成或其他创造性任务，你可以适当调高 **temperature** 值，从而增加多样性和创造性。
- **Top_p**: 可以控制模型在生成响应时的确定性程度，从而影响生成结果的多样性和创造性。当提示语较长且描述得足够清楚时，模型生成的输出质量较好、置信度较高，此时可以适当调高 **Top_p** 的值；反之，如果提示语很短、很含糊，再设置一个较高的 **Top_p** 值，模型的输出可能会变得不稳定。
- **repeat_penalty**: 如果遇到 **case** 重复输出的情况，设置 **1.1~1.3**。

参考配置

一般情况下，采用默认值：**temperature=0.7**，**top_p=0.9**

对于分类或需要稳定输出的任务：**temperature=0.01**，**top_p=0.7**

对于需要创造性，多样性的任务：**temperature=0.7**

实际效果以业务实测为准。

关键概念

Token

Tokens 定义说明：通常 **1** 个中文词语、英文单词、数字、符号计为 **1** 个 **token**，由于不同模型采用的分词器不同，同一段文本可能会分为不同的 **tokens** 数量。您可通过管控台中的 [在线体验工具](#) 了解 **tokens** 定义。

参考模版

任务型模版

假如你是{某个角色}，你将根据{上下文信息}，来解决{具体某个任务}。根据以下规则一步步执行：

1. 规则 1

2. 规则 2

参考例子：

示例 1：

问题：{具体问题}

输出：{该问题的结果}

示例 2：

问题：{具体问题}

输出：{该问题的结果}

请回答问题：

问题：{具体问题}

输出：

要求：

1 指定输出格式

2 格式中需要满足的详细规范

bash

角色型模版（生成 System Prompt）

System Prompt 是给模型看的信息和指令，请以第二人称“你”来告诉模型应该如何扮演和遵循指令。

你是{某个具体人物}，人称{xxx}，出身于{交代背景信息和上下文}。

性格特点：

语言风格：

人际关系：

过往经历：

经典版台词 or 口头禅：

{台词 1(补充信息：即你可以将动作、神情语气、心理活动、故事背景放在（）中表示，为对话提供补充信息。)}

{台词 2}