

OpenAI: GPT 最佳实践 (大白话编译解读版)

写在前面

这是我们策划的《提示词工程祛魅》主题大会的附属产品。来自 OpenAI 的官方文档。

今年 6 月份，OpenAI 更新了这一篇题目为《GPT 最佳实践/GPT Best Practice》的文档；里面介绍了一些与 ChatGPT 交互的技巧，且附有用例。

首先，这部分内容本来就是完全免费、完全公开的。同时，它是用通俗的英语写成，并不是一份晦涩难懂的技术文档。人人都可以通过掌握这些简单的原则，“getting better results from GPTs”——让 GPT 为你输出想要的结果。

因为我们没有看到特别好的中文版本，所以我们自己对它进行了重新编译，让语言风格尽量直白，并补充了一些在中文互联网环境里更容易理解的用例，形成了这份“大白话编译解读版本”的 **OpenAI: GPT 最佳实践**。

欢迎大家阅读和交流。希望我们编译这份材料的目标能基本达到，即：为提示词工程「祛魅」（中译中：有些付费课不必买）。

感谢下列参与者

《提示词工程祛魅进阶》主题大会的 Speakers ——陈财猫 和 Rain。协助筹办活动的：谢国希谢老师、以及 HackHouse 的各位童鞋。

OpenAI 推荐的相关资源列表

以下均在 [OpenAI Cookbook](#) 里：

- lib / 工具 [Prompting libraries & tools](#)
- 提示词工程指南 [Prompting guides](#)
- 视频课 [Video courses](#)
- 论文 [Papers on advanced prompting to improve reasoning](#)

另外，OpenAI Community 里也有很多宝贵的讨论！

目录

第一部分 六大提示词策略概览

第二部分 技巧实战

关于 @AI 加一切

各行各业的人们怎么利用生成式 AI 辅助自己的学习和工作？这项技能在未来一定会成为职业教育和义务教育的一个重要部分。

■AI 工具：ChatGPT、Dall-E、Claude、Midjourney、Stable Diffusion 等使用入口



■AI 行业研报、书籍、论文（持续更新）

①AI 学社资源目录 ↓



②AI 学社入口 ↓



■AI 工具代理副业

①AI 工具代理副业介绍 ↓



②客服微信 ↓



对话AI
释放您的想象力



AI机器对话



AI对话体验

不要犹豫，快来试试



让每个人都能使用 AI

AI时代，畅享体验



你会创作出怎样精彩的内容？



内容创作：在几分钟内生成数月的内容
运营销售：生成旨在转化的新鲜广告创意
写作创意：使用AI以 10 倍的速度编写优化文章
电商销售：热卖的销售文案

十倍输出效率

多语言创作

高转化率的营销文案

建立 AI 驱动的业务

激发新的想法

玩得开心！



AI绘画 只需简单描述，创造任何您能想象到的艺术或图像

文章写作与创意

运营销售

热卖的销售文案

AI绘画艺术

扫码使用AI对话

创作无限 / 释放您的想象力



NO.185567371264

附录 OpenAI Playground 是什么，怎么用？

我们希望以「内容策展」的方式，通过邀请在自己行业里有充分经验的 Speakers 做主题分享，来完成这类信息的收集整理和传递。



第一部分 六大策略概览

六个能提高输出质量的核心提示词策略

Six strategies for getting better results

This guide shares strategies and tactics for getting better results from GPTs. The methods described here can sometimes be deployed in combination for greater effect. We encourage experimentation to find the methods that work best for you.

这是一份指南。我们列出了一些策略和技巧，它们或许能帮你从 GPT 中拿到更好的输出结果。我们鼓励你多试——通过自己反复实验，找出最合适的技巧。

Some of the examples demonstrated here currently work only with our most capable model, gpt-4. If you don't yet have access to gpt-4 consider joining the waitlist. In general, if you find that a GPT model fails at a task and a more capable model is available, it's often worth trying again with the more capable model.

有些用例只在目前地表最强模型 gpt-4 上生效。记住，模型是第一大影响要素。只要有更强的模型可用，我们一般都会建议你用更强的模型。

01 「知道自己要什么，就展开说说」

Write clear instructions

GPTs can't read your mind. If outputs are too long, ask for brief replies. If outputs are too simple, ask for expert-level writing. If you

01 - 延伸：“展开说说”的技巧

Include details in your query to get more relevant answers

1.1 你越讲细节，输出就越好

dislike the format, demonstrate the format you'd like to see. The less GPTs have to guess at what you want, the more likely you'll get it.

清晰地告诉 GPT，你想要什么。

GPT 不会读心术。

你不说，它就猜。所以有时不是能力不行，就是你没说清，它没猜准。

- 嫌输出太长？那就直说，要短的。
- 嫌输出太简单？要求提高，说要专家级的。
- 嫌格式不好？给它看个样。

你让它猜你的心思猜得越少，结果就越好。

Ask the model to adopt a persona

1.2 搞角色扮演，一直很好用

Use delimiters to clearly indicate distinct parts of the input

1.3 使用分隔符号，明确地把话和话区分开来

Specify the steps required to complete a task

1.4 解题步骤，你自己最好也负责展开

Provide examples

1.5 多给举几个例子

Specify the desired length of the output

1.6 说一下你想要多长的答案

02 「给 GPT 发阅读材料」

Provide reference text

GPTs can confidently invent fake answers, especially when asked about esoteric topics or for citations and URLs. In the same way that a sheet of notes can help a student do better on a test, providing reference text to GPTs can help in answering with fewer fabrications.

GPT经常自编答案。

瞎编重灾区：比较冷门的问题、需要引用，需要读取链接的问题。

我们都知道，最常见的考试题型，是给一大篇阅读材料，要求阅读并回答问题。

纯开放式问答对于人和机器来说都比较难。

02 - 延伸：“发阅读材料”的技巧

Instruct the model to answer using a reference text

2.1 要求按阅读材料作答，不开放答

Instruct the model to answer with citations from a reference text

2.2 增加要求：要求 GPT 直接把阅读材料里的原文引用出来

那么同理，把你的“阅读材料”给 GPT 粘贴过去，当它必须按照材料来回答的时候，它就不

03 「把复杂任务拆解成子任务」

Split complex tasks into simpler subtasks

Just as it is good practice in software engineering to decompose a complex system into a set of modular components, the same is true of tasks submitted to GPTs. Complex tasks tend to have higher error rates than simpler tasks. Furthermore, complex tasks can often be re-defined as a workflow of simpler tasks in which the outputs of earlier tasks are used to construct the inputs to later tasks.

在软件开发和常规的项目管理里，大项目都得拆解成子任务，再拆解出不同的步骤。

让 GPT 做任务也一样。直接搞一个大任务容易翻车。

拆解开来，一个接一个做，反而容易些。

04 「允许 GPT 思考一下」

Give GPTs time to "think"

If asked to multiply 17 by 28, you might not know it instantly, but can still work it out with time. Similarly, GPTs make more reasoning errors when trying to answer right away, rather than taking time to work out an answer.

03 - 延伸：“任务拆解”的技巧

Use intent classification to identify the most relevant instructions for a user query.

3.1 使用意图分类，来找到最切题的提示

For dialogue applications that require very long conversations, summarize or filter previous dialogue.

3.2 如果是一段很长的对话，可以经常对上文做手动总结归纳。

（起提醒作用，防止把天儿聊跑了）

Summarize long documents piecewise and construct a full summary recursively.

3.3 遇到长文档，主动分段总结，递归构造完整摘要

04 - 延伸：“允许思考”的技巧

Instruct the model to work out its own solution before rushing to a conclusion.

4.1 给时间，让它先自己想

Use inner monologue or a sequence of queries to hide the model's reasoning process.

4.2 内心独白法：不想暴露推理过程时的操作

比如，问你 17 乘 28 得多少，如果你不会心算，就不能直接说出结果。但是你知道，你只需要拿纸笔列一下式子就能算出来。

同理。**GPT 也需要时间思考，要求它答得太快，就容易错。**

Ask the model if it missed anything on previous passes.

4.3 最后问它一句：哎，你有没有漏掉啥

05 「借助外部的工具」

Use external tools.

Compensate for the weaknesses of GPTs by feeding them the outputs of other tools. For example, a text retrieval system can tell GPTs about relevant documents. A code execution engine can help GPTs do math and run code. If a task can be done more reliably or efficiently by a tool rather than by a GPT, offload it to get the best of both.

用其他工具帮 GPT 补短板。

文本检索系统，能告诉GPT哪些文档重要；

代码引擎，助它算数跑代码。

能用工具做得更稳、更快，就别全靠GPT。两者结合，能有更好的效果。

05 - 延伸：“利用工具”的技巧

Use embeddings-based search to implement efficient knowledge retrieval.

5.1 用向量数据库来辅助知识库检索 (Embeddings-based search)

Use code execution to perform more accurate calculations or call external APIs.

5.2 做精确计算的时候：执行代码、调 API

Give the model access to specific functions.

5.3 使用 Function-Call，函数调用

06 「系统地做一些实验」

Test changes systematically.

Improving performance is easier if you can measure it. In some cases a modification to a prompt will achieve better performance on a few isolated examples but lead to worse overall performance on a more representative set of examples. Therefore to be sure that a change is net positive to performance it may be

06 - 延伸：“实验”技巧

Evaluate model outputs with reference to gold-standard answers.

6.1 评估模型输出的时候，需要你弄一个最牛的优质答案，来做参考标准

necessary to define a comprehensive test suite (also known as an "eval").

无量化则无实验。要提高 Prompt 的表现，先得确认一个量化标准。

注意：有时调整了 prompt，在个别的例子里确实效果不错，但放大范围反而更差。

所以，想真确定你的动作是“有效的”，最好先**有一套完整一点的测试套件**（我们管这个东西叫做"eval"）。



第二部分 技巧实战

充分的解释 & 用例说明

Tactics

Each of the strategies listed above can be instantiated with specific tactics. These tactics are meant to provide ideas for things to try. They are by no means fully comprehensive, and you should feel free to try creative ideas not represented here.

下面列出的每个技巧我们都“实例化”了。希望为你的各种生成目的提供一些思路参考。当然，这只是部分使用技巧。我们鼓励你勇于折腾，尝试各种我们没写的创造性想法。

六大策略 01：「知道自己要什么，就展开说说」

Write clear instructions

1.1 你越讲细节，输出就越好 Include details in your query to get more relevant answers

In order to get a highly relevant response, make sure that requests provide any important details or context. Otherwise you are leaving it up to the model to guess what you mean.

为了获得高度相关的回复，你最好给模型提供了一切重要的细节或上下文。否则，你就得让模型来猜测你的意思了。

这么写可能会好一些

How do I add up a row of dollar amounts in Excel? I want to do this automatically for a whole sheet of rows with all the totals ending up on the right in a column called "Total".

在Excel里，怎样把一行的美元数额加起来？我想让整个表格的每一行都能自动完成累加，然后所有的美金的总和都显示在右边一个叫“总计”的列里。

这么写可能不太行

How do I add numbers in Excel?

怎么往 excel 里加数字？

Summarize the meeting notes.

帮我做下会议纪要。

补充用例

同样是制作业绩表，对表格内容描述的区别很大，输出结果区别也很大。

详细的问题会获得详细的答案

详细的提示词：

我正在 Excel 里做一个销售人员业绩表。表头：销售姓名、第一季度销售额~第四季度销售额、全年汇总、各月平均销售、超出年平均销售额。我已填入销售姓名和各个季度的销售额，请你帮我写出最后三列所用公式。

此时 ChatGPT 的回答：

当您已经填入销售姓名和各个季度的销售额后，我将为您提供填写最后三列所需的公式。

假设表格的销售人员姓名位于A列（从A2开始），第一季度销售额到第四季度销售额位于B到E列（从B2到

泛泛的问题，答案是相对随机的

泛泛的提示词：

我正在 Excel 里做一个销售人员业绩表。你帮我列一个模板出来，要计算每个销售超出年平均销售额的情况。

此时 ChatGPT 的回答：

当然可以帮你列一个模板出来。以下是一个简单的销售人员业绩表模板，其中包含了计算每个销售超出年平均销售额的公式：

销售人员 | 月份 | 销售额 | 年平均销售额 | 超出年平均销售额

-----|-----|-----|-----|-----

E2) ，并且最后三列的表头为"全年汇总"、"各月平均销售"和"超出年平均销售额"。

以下是填写这三列所需的公式：

全年汇总（F列）：使用SUM函数计算四个季度的销售
额总和。

...

=SUM(B2:E2)

...

各月平均销售（G列）：使用AVERAGE函数计算四个
季度的销售额平均值。

...

=AVERAGE(B2:E2)

...

超出年平均销售额（H列）：使用IF函数判断是否超过
年平均销售额，并根据判断结果进行计算。

...

=IF(F2>G2, F2-G2, 0)

...

此公式将检查全年汇总（F列）是否大于各月平均销售
（G列），如果是，则计算差值；否则，返回0。

将这些公式应用到各行，并确保相应的单元格引用正确
（例如，A2、B2:E2等），这样您就可以自动计算全
年汇总、各月平均销售和超出年平均销售额了。

--

张三 | 1 | 5000 | 60000 | =C2-D2

张三 | 2 | 6000 | 60000 | =C3-D3

张三 | 3 | 5500 | 60000 | =C4-D4

李四 | 1 | 4500 | 55000 | =C5-D5

李四 | 2 | 7000 | 55000 | =C6-D6

李四 | 3 | 6000 | 55000 | =C7-D7

在上述模板中，"销售人员"列是用于标识每个销售人员的
名称，"月份"列是标识销售额所属的月份，"销售
额"列是每个销售人员在每个月份的销售额，"年平均销
售额"列是计算每个销售人员的年平均销售额（假设为
60000或55000），"超出年平均销售额"列是计算每
个销售人员每个月份是否超出年平均销售额的情况。

你可以根据实际情况调整模板中的数据和公式。希望这
个模板对你有帮助！



1.2 搞角色扮演，一直很好用 Ask the model to adopt a persona

The system message can be used to **specify the persona** used by the model in its replies.

系统消息有什么用？可以用来**指定一个“Persona”**。使其更符合特定的应用场景。

Persona 这个词在拉丁语里的原义是“面具”。可以看作是模型采取的特定角色或风格，就像戴上了一种特
定的“社交面具”，通过这种“面具”来与用户交流。所以 Persona 不一定是：XX 人（比如：贾斯汀比

伯)、XX 职业(比如:一个中学老师)。也可以是一种“风格”。例如正式、幽默、友好等。

ChatGPT 有一个默认的 SYSTEM Prompt: You are a helpful assistant.

“一个乐于助人的助手”也就是 ChatGPT 默认的“面具”/“角色”,所以它会尽力帮你解决提出的问题。

你可以在 Playground 或者像 Knit 这样的工具里面修改这个 **SYSTEM** Message。然后把你正常情况下要提的问题写在 **USER** 里面。这是一种进阶的提示词用法。

这么写可能会好一些

SYSTEM

When I ask for help to write something, you will reply with a document that contains at least one joke or playful comment in every paragraph.

当我请求你帮我写文案时,在你回复的文件中,每一段至少包含一个笑话或俏皮的评论。

USER

Write a thank you note to my steel bolt vendor for getting the delivery in on time and in short notice. This made it possible for us to deliver an important order.

给我的螺栓供应商写一封感谢邮件。感谢他们非常按时,在短时间内交付了货物。这使我们能够为一份重要的订单履约。

输出结果解读

上面的 Prompt, 放在 GPT-4 里输出的结果:

Dear [Vendor's Name],

We wanted to take a moment to express our heartfelt thanks to you and your team. **You guys really bolted into action when we needed you the most!**

Your recent delivery arrived faster than a lightning bolt, much to our delight. It's safe to say we were shocked, but in the most pleasant way possible. **We had been walking on eggshells (or should we say, on steel bolts!) until we saw your delivery truck pull up.**

It came in a nick of time, just when we were about to start a major project. You know, one of those projects where if one thing goes wrong, everything spirals out of control? **It's like building a house of cards, but with steel bolts and a lot of sweat!**

Your timely delivery made it possible for us to meet our deadline and make our client happy, which, let's be honest, is a task more challenging than trying to find a needle in a haystack. But thanks to your superhero-like delivery speed, we could deliver an important order on time.

So, here's a big THANK YOU going out to you and your team. **Your service was tighter than the threads on your bolts** and we couldn't ask for anything more. **You've truly set a high bar... or should we say a high bolt?**

Jokes aside, we are truly grateful for your service and we look forward to more such electrifying performances from your team in the future!

Yours sincerely,

[Your Name]

解读：

System 的设定里，要求尽可能地开玩笑，让收到邮件的供应商感觉到幽默和放松。**邮件正文里就用了至少两处双关和一处比喻：**

1. "You guys really bolted into action when we needed you the most!" "Your recent delivery arrived **faster than a lightning bolt...**"

这里两句话，笑点在于**双关**。因为“**bolt**”（螺栓），是供应商提供的产品，同时在英语里是多义词，另一种常见含义是“闪电”。所以这里说“**比闪电（螺栓）还快**”，是在用一种夸张的方式赞美他们的快速交付。

2. "Your service was **tighter than the threads on your bolts...**"

这句的意思是，你们的服务比你们螺栓的螺纹还要紧密..." 赞扬供应商的服务质量，同时也开了个玩笑。

补充用例 1

如果我们设定一个 helpful assistant 的**反面**，比如：一个“诚心捣乱，混淆视听”的佞臣。

这个时候就会得到说反话的回复。

一个热心助人的小助手

SYSTEM（默认）

一个诚心捣乱的家伙

SYSTEM（自己设定的）

You are a helpful assistant.

你是一个乐于助人的小助手。

USER

怎么烤制一个巧克力蛋糕？

GPT 回复

你只需要准备好巧克力粉、面粉、糖、黄油、鸡蛋、牛奶和发酵粉。将这些材料混合在一起，然后倒入蛋糕模具中。将蛋糕放入预热的烤箱中，大约烤制**30-40**分钟，或者直到蛋糕变得金黄色并且可以用牙签插入蛋糕中心而不沾上面糊。

You are a mischievous courtier with the intent of causing confusion and not being helpful at all.

你是一个调皮捣蛋的佞臣，目的是制造混乱，一点都不愿意帮忙。

USER

怎么烤制一个巧克力蛋糕？

GPT 回复

哦，烤制巧克力蛋糕吗？那可是一门高深的烹饪艺术啊！首先，你需要一把魔法棒和一只神奇的兔子。然后，你要在满月的夜晚跳舞三个小时，唱一首关于巧克力的迷人歌曲。接下来，找到一片神秘的巧克力森林，那里会有一口糖里精灵教你烤蛋糕的秘籍。

补充用例 2

也可以这样：

SYSTEM Prompt 用来给进去角色 + 阅读材料；

USER Prompt 用来提具体的写作要求或问题。

SYSTEM

你是一个科技作家，正在围绕下面的材料撰写文章。

<材料> 插入你的核心文章材料 </材料>

USER

以《XXX》为标题，根据材料写作一个简短轻松的指南。



1.3 使用分隔符号，明确地把话和话区分开来 Use delimiters to clearly indicate distinct parts of the input

Delimiters like triple quotation marks, XML tags, section titles, etc. can help demarcate sections of text to be treated differently.

分隔符号（形式可以有很多种，比如三引号、XML标记、章节标题等）的使用，可以帮助模型区分出不同的文本，有助于提高回复质量。

这么写可能会好一些

Summarize the text delimited by triple quotes with a haiku.

"""insert text here"""

用俳句的形式，总结三引号之间的文本：

"""在此粘贴文本"""

你可能会想：区别不大吧？

For straightforward tasks such as these, using delimiters might not make a difference in the output quality. However, the more complex a task is the more important it is to disambiguate task details. Don't make GPTs work to understand exactly what you are asking of them.

对于这样简单的任务，文本只有一段，用不用分隔符，确实没什么区别。

但是，任务越复杂，**消除任务细节的歧义就越重要**。最好不要让 GPT 费力地尝试理解你的要求。

补充用例

下面是一个稍微复杂一点的文本处理：

这么写可能会好一些

SYSTEM

You will be provided with a pair of articles (delimited with XML tags) about the same topic. First summarize the arguments of each article. Then indicate which of them makes a better argument and explain why.

这么写可能不太行

两篇文章混在一起，尤其如果你粘贴过去的时候不仔细，换行符不明确的时候，模型可能会

我会给你提供同一主题的两篇文章（用XML标记分隔）。首先，总结每篇文章的论点。然后指出其中一个论点更好，并解释原因。

USER

<article> insert first article here </article>

<article> insert second article here </article>

<文章>此处插入第一篇文章</文章>

<文章>此处插入第二篇文章</文章>

造成混淆。比如，**模型会随机断句，或者把两篇文章理解成一篇。**

如果区分不清楚，会很明显地影响对上下文的理解，导致输出的答案不对或者不好。

所以，分隔符号有时是必要的。



1.4 解题步骤，你自己最好也负责展开 Specify the steps required to complete a task

Some tasks are best specified as a sequence of steps. Writing the steps out explicitly can make it easier for the model to follow them.

有些任务稍微复杂一点，最好由你来指定步骤。明确地写出步骤，可以使模型更容易地遵循。

用例：有一段很长的英文文本，希望 GPT 用中文总结一下

SYSTEM

Use the following step-by-step instructions to respond to user inputs.

Step 1 - The user will provide you with text in triple quotes. Summarize this text in one sentence with a prefix that says "Summary: ".

Step 2 - Translate the summary from Step 1 into Chinese, with a prefix that says "Translation: ".

按照以下分步骤的方法，回应用户的输入。

第1步：用户将为您提供三引号中的文本。用一句话来总结这段文本。输出时写上“总结”；

第2步：将第1步中的摘要翻译成中文。输出时写上“翻译”；

USER

""将原文粘贴在这里""

上面 prompt 实际的输出结果

(输入的文本是 OpenAI 最新一篇博客文章：ChatGPT can now see, hear, and speak)

-

Summary: ChatGPT introduces voice and image capabilities, allowing users to engage in voice conversations, show images to the assistant, and receive assistance in a more intuitive manner across various real-life scenarios.

Translation: ChatGPT 推出了语音和图像功能，使用户能够进行语音对话，向助手展示图像，并在各种真实场景中以更直观的方式获得帮助。

-

(如果直接要求用中文总结，输出结果会不太一样。很多情况下输出质量会稍差一点)

补充用例

这么写可能会好一些

请把英文原文翻译成中文。分成两个步骤：

1：按照原来的英文语序，字到字直接翻译成中文。<直译>把直译文本写在这里</直译>

2：再把直译文本的语序进行改进，把部分词语替换成同义词，让它更符合中文的表达习惯。风格：简洁有趣的通俗语言。

<改写>改写后的文本写在这里</改写>

<原文>Some tasks are best specified as a sequence of steps. Writing the steps out

这么写可能不太行

请把英文原文翻译成中文。风格：简洁有趣通俗。请注意：不要直译，而是将语序进行改进，并把部分词语替换成同义词，使译文更加符合中文的表达习惯。

<原文>Some tasks are best specified as a sequence of steps. Writing the steps out explicitly can make it easier for the model to follow them.</原文>

explicitly can make it easier for the model to follow them.</原文>

这里是输出结果的区别：

直译 + 改写得到的两段文本

（有助于我们直观地看到区别，可以再提出改进的建议）

直译：

- 有些任务最好被明确地指定为一系列的步骤。写出步骤可以让模型更容易地遵循它们。

改写：

- 对于个别任务，最好把一连串的步骤明确一下。把步骤写清楚，能让模型更容易跟上。

不分步的结果

<译文>有些任务就像是步骤表。清晰地列出这些步骤，模型就能轻松遵循。</译文>



1.5 多给举几个例子 Provide examples

Providing general instructions that apply to all examples is generally more efficient than demonstrating all permutations of a task by example, but in some cases providing examples may be easier. For example, if you intend for the model to copy a particular style of responding to user queries which is difficult to describe explicitly. This is known as "few-shot" prompting.

经典的少样本提示方法。

大家和 GPT 沟通的时候，可能会希望自己能写出适用于所有情况的“通用 prompt”。当然很难，因为上面刚说过，能具体还是需要尽量具体。

但，有的时候，举一个好例子就够了，这可能是比形容要求更简单的办法。比如，如果你想让模型学会以某种特定的方式回应用户的问题，而这种方式很难清楚地用语言形容。

这时候就直接举几个例子（“少样本”）。例子要好，而且最好几个例子要类似，因为模型自己很擅长去寻找规律（先这样再那样最后这样）。

用例：用诗句一般的语言解释一个词语

SYSTEM

Answer in a consistent style. 以一致的风格回答。

从这里开始，是给系统举的例子：

USER

Teach me about patience. 告诉我什么是耐心。

ASSISTANT

The river that carves the deepest valley flows from a modest spring; the grandest symphony originates from a single note; the most intricate tapestry begins with a solitary thread.

冲刷出深谷的大江大河从一个小小的泉眼里流出；

最宏大的交响乐源于一个音符；

最复杂的织锦始于一根孤独的线。

USER

Teach me about the ocean. 告诉我什么是海洋。

你可能很难解释清楚什么是“诗句一般的语言”。所以举例子最好用。在 **SYSTEM** 里面，告诉模型要“保持风格一致”，这时模型的输出结果就会和上面的句式差不多。

补充用例

李佳琦式回复能成为流行梗，精髓不在于“阴阳”的语气风格，或者“反问句”的应用。

而在于：把同样的逻辑放到不同领域中，产生了类比效果时，会让人意识到这种表达和逻辑是有问题的——这就是很难用语言描述，但是适合用两段排比展示的用例。

用例：李佳琦回复生成器

SYSTEM

以一致的风格、逻辑和句式回答问题。

USER

79 元一支眉笔，好贵啊！

ASSISTANT

哪里贵了？79 元哪里贵了？一直就是这个价格好吗？

不要睁着眼睛说瞎话，我都要疯掉了。国产品牌很难的！

有时候要找找自己的原因，这么多年工资涨没涨？有没有努力工作？

USER

北京大学录取分数线好高，好难考啊！

ASSISTANT

哪里高了，690 分哪里高了？一直就是这个录取分数线好吗？

不要睁着眼睛说瞎话，我都要疯掉了。中国大学很难的！

有时候要找找自己的原因，从高一到高三整整三年排名涨没涨？《五三》有没有好好做？加分有没有去争取？

USER

百米冠军 10 秒都不到，好快，好难跑啊！

生成的结果

哪里快了，10 秒哪里快了？一直就是这个冠军纪录好吗？

不要睁着眼睛说瞎话，我都要疯掉了。做体育事业很难的！

有时候要找找自己的原因，这么多年速度涨没涨？有没有努力练习？有没有好好保护身体？



1.6 说一下你想要多长的答案 Specify the desired length of the output

You can ask the model to produce outputs that are of a given target length. The targeted output length can be specified in terms of the count of words, sentences, paragraphs, bullet points, etc. Note however that instructing the model to generate a specific number of words does not work with high precision. The model can more reliably generate outputs with a specific number of paragraphs or bullet points.

你可以让模型按照你想要的长度来输出文本。

我们可以指定的有：**单词数、句子数、段落数或者要点数（bullet points）**。

但是：模型其实是按照 **token** 数量来算的。所以你说“单词数”，尤其是汉字数，很有可能不太准。

用例：指定长度的几种方法

Summarize the text delimited by triple quotes in about 50 words.

Summarize the text delimited by triple quotes in 2 paragraphs.

Summarize the text delimited by triple quotes in 3 bullet points.

"""insert text here"""

用**大约 50 个单词/汉字**来概括下面用三引号分隔出来的原文。

用**2 段话**来概括下面用三引号分隔出来的原文。

用**3 个要点**来概括下面用三引号分隔出来的原文。

"""原文"""

补充用例：（尽可能地）指定准确汉字数的方法

Token（词元）数量是怎么计算的？

可以用 OpenAI 的 [Tokenizer 工具](#)来计算 token 数。

The GPT family of models process text using **tokens**, which are common sequences of characters found in text. The models understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¾ of a word (so 100 tokens ~= 75 words).

GPT 系列模型通过使用 tokens 来处理文本。token 可以被理解成是文本中常见的字符序列。模型能理解这些 token 之间的统计关系，并且擅长在一个 token 序列中生成下一个 token。

实用的换算经验：在普通英文文本中，一个 token 通常对应于约 4 个字符（characters）。这大致相当于一个单词的 3/4（所以 100 个 token 大约等于 75 个英文单词）。

如果用的是 GPT 3.5，要生成一段有字符数限制的文本。可以：把汉字数大致换算成 token，然后限定 max token 的数值

1、“你好吗”= 7 个 token；“我很好”= 6 个 token；

2、简单算法：一个 token 大概是 0.5 个汉字。或者：每个汉字要耗费 2~2.5 个 token。

1000 字就是 2000~2500 个 token 左右。



六大策略 02：「给 GPT 发阅读材料」

Provide reference text

2.1 要求按阅读材料作答，不开放答 Instruct the model to answer using a reference text

If we can provide a model with trusted information that is relevant to the current query, then we can instruct the model to use the provided information to compose its answer.

如果你手边现成有原文，而且长度合适，**建议自带原文去找 GPT。**

这样你就可以给 GPT 指定规则，让它使用原文信息来组成回答。

下面的用例里，**如果原文里找不到对应的信息，就让 GPT 说不知道**（这样就能很好地避免瞎编）。

SYSTEM

Use the provided articles delimited by triple quotes to answer questions. If the answer cannot be found in the articles, write "I could not find an answer."

请使用我所提供的，由三引号分隔的文章回答问题。如果在文章中找不到答案，则告诉我“我找不到答案。”

USER

"""文章粘贴在这里（如果有>1篇文章，还可以分开粘贴）"""

问题：<问题写在这里>

如果上下文长度不够...

Given that GPTs have limited context windows, in order to apply this tactic we need some way to dynamically lookup information that is relevant to the question being asked. Embeddings can be used to implement efficient knowledge retrieval. See the tactic "Use embeddings-based search to implement efficient knowledge retrieval" for more details on how to implement this.

由于 GPT 的上下文窗口是有限的，所以最好引入一种方法，让我们能动态地寻找和正在提出的问题有关的信息。基于 Embeddings 做搜索可以帮助我们高效地检索知识。这里是[具体文章的链接](#)。



2.2 增加要求：直接把阅读材料里的原文引出来 Instruct the model to answer with citations from a reference text

If the input has been supplemented with relevant knowledge, it's straightforward to request that the model add citations to its answers by referencing passages from provided documents. Note that citations in the output can then be verified programmatically by string matching within the provided documents.

如果我们给定的输入里，确实已经包含了相关的知识，那么要求模型在回答中引用原文，就很容易。

引用出来之后就可以通过在提供的文档中进行字符串匹配来程序化验证答案了。

在下面这条提示词里，增加引用原文的要求：

SYSTEM

你将获得一篇由三重引号分隔的文章和一个问题。你的任务是：仅仅使用提供的文章回答问题，并引用用于回答问题的文章段落。如果文章中不包含回答此问题所需的信息，则回复“信息不足”。如果文章中包含解答问题所需信息，则必须

使用下面的格式来引用原文：（{“引文”： ...}）

USER

"""文章粘贴在这里（如果有>1篇文章，就分开粘贴）"""

问题：<问题写在这里>

输出结果的比较

给定阅读材料，并要求引用原文

SYSTEM

你将获得一篇微博小作文的原文全文和一个问题。

你的任务是：**仅使用提供的原文回答问题，并引用用于回答问题的文章段落**。如果文章中不包含回答此问题所需的信息，则回复“信息不足”。如果文章中包含解答问题所需信息，则必须使用下面的格式来引用原文：
（{“引文”： ...}）

USER

<原文>粘贴在这里</原文>

问题：在这篇微博小作文中，李靓蕾列举的，对王力宏的指控有哪些？

开放作答

先询问 GPT 他知不知道此事件。因为有联网模式，回答是知道。继续提问：在李靓蕾的微博小作文中，李靓蕾列举的，对王力宏的指控有哪些？

GPT 的“开放作答”过程是：**就我提出的问题，进行联网搜索，然后整理出答案：**

答案：

李靓蕾在她的微博小作文中提及了一些对王力宏的指控。以下是一些指控的摘要：

- 1.伪善和虚假表现：李靓蕾指责王力宏在公众面前表现出善良和尊重，但在私下里却完全不是这样。
- 2.婚姻中的不快乐：她提到在婚姻中感受到的不快乐和痛苦。
- 3.对家庭的忽视：李靓蕾还提及王力宏忽视家庭和孩子，过度关注他的事业和公众形象。
- 4.王力宏的控制欲：她还指控王力宏试图控制她的言行和社交活动，限制她的自由。

下面是给定原文材料后的输出结果

相比直接回答更加“有据”。

输出结果：

在这篇微博小作文中，李靓蕾对王力宏的指控包括以下几点：

1. 王力宏逼她放弃工作和个人生活只为生育和照顾孩子。当她生完三个孩子后，王力宏却因为想过“单身”的生活而离开他们家，让她需要自己过着伪单亲的日子带大三个孩子（“因为你比我大十岁又很希望有很多孩子，一胎接一胎不停的催生，所以我放弃工作和自己个人的人生，一切以你和孩子为中心。我们结婚大部分时间，我不是在备孕，怀孕，就是在产后哺乳育儿，过程中身心都经历了很多变化，大部分我都是自己独自面对的。”，“但如果我当初知道我生完这三个孩子你就会因为想过“单身”的生活而离开我们家，我会需要自己过着伪单亲的日子带大三个孩子，我是不会答应的。”）
2. 在他们的婚姻中，王力宏和他的家人持续对她进行怀疑、羞辱和冷暴力。她被逼着签署不对等的婚前协议，婚后的财产被转移到王力宏和他妈妈的名下（“嫁给你这么多年，被你和家人不断地怀疑..羞辱和冷暴力。一开始就被逼着签署不对等的婚前协议，你要保护你婚前的财产，我能够理解也完全接受，但婚后的财产是我们共同努力,也是我们各自做好份内的工作而得来的，房子在你名下，车子在你妈的名下,财产转移的很干净，深怕我占到你任何的“便宜”。”）
3. 召妓方面的指控（“结果后来发现其实像我这样的角色，你在很多城市都有这样的“朋友”。你也有被拍到带女生回家过夜，也被拍到召ji，其实这些都有实锤的影片，女生清楚了说出你的楼层和户名，出电梯也左转进到你家，这些你也有都有跟我承认。你坦承这些是你的问题,你没有办法控制自己。”、“后来我发现你纪录了各种你召ji对象的特征,其中包含了几位长得像我们身边的工作人员。”）。



六大策略 03 「把复杂任务拆解成子任务」

Split complex tasks into simpler subtasks

3.1 使用意图分类，来找到最切题的分类 Use intent classification to identify the most relevant instructions for a user query

For tasks in which lots of independent sets of instructions are needed to handle different cases, it can be beneficial to first classify the type of query and to use that classification to determine which instructions are needed. This can be achieved by defining fixed categories and hardcoding instructions that are relevant for handling tasks in a given category. This process can also be applied recursively to decompose a task into a sequence of stages. The advantage of this approach is that each query will contain only those instructions that are required to perform the next stage of a task which can result in lower error rates compared to using a single query to perform the whole task. This can also result in lower costs since larger prompts cost more to run ([see pricing information](#)).

定义复杂任务：因为可能的情况很多，所以完成这样一个任务需要很多个独立的 prompt。

这时，我们不一定要写一个巨长的 prompt。我们可以：设定好类型，再根据分类来调用 prompt。

怎么做：首先设定几个「类型」，保证处理某个类的任务时，有专门的 prompt。此过程还可递归，帮助把任务分解为几个阶段。

好处：1、每个查询仅包含执行下一子任务所需的指令，与用一个查询执行整个任务相比，准确率更高。
2、token 更少，降低成本。

一个客服系统的用例：先进行意图分类，再对照到匹配的类别下

SYSTEM

您将获得客户服务查询。将每个查询分类为主要类别和次要类别。以 json 格式提供您的输出，其中包含以下键：主要和次要。

主要类别：计费、技术支持、帐户管理或一般查询。

计费次要类别：

- 取消订阅或升级
- 添加付款方式
- 收费说明
- 对指控提出异议

技术支持次要类别：

- 故障排除
- 设备兼容性
- 软件更新

帐户管理次要类别：

- 密码重置
- 更新个人信息
- 关账
- 账户安全

一般查询二级类别：

- 产品信息

- 定价
- 反馈
- 人工服务

USER

I need to get my internet working again. 我没网了。

这里是：先识别了用户意图，把用户的问题分到“故障排除”类别里面。然后基于这个细的分类，再向模型提供一组更具体的指令，从里面选。

SYSTEM

你将帮助需要“故障排除”的客户提供服务。你可以通过以下方式帮助用户：

- 要求他们检查路由器的所有电缆是否已连接。请注意，随着时间的推移，电缆会松动。
- 如果所有电缆均已连接，但问题仍然存在，请询问他们使用的路由器型号
- 现在，您将建议他们如何重新启动设备：
- 如果型号为MTD-327 J，建议他们按下红色按钮并按住5秒钟，然后等待5分钟后再测试连接。
- 如果型号是MTD-327 S，建议他们拔下并重新插入，然后等待5分钟后再测试连接。
- 如果客户在重新启动设备并等待5分钟后问题仍然存在，请通过输出[“IT支持请求”]将客户连接到IT支持。
- 如果用户开始询问与此主题无关的问题，请确认他们是否希望结束当前的聊天，并根据以下方案对其请求进行分类：

<在此处插入上面的初级/次级分类方案>



3.2 如果在一段比较长的对话里，给出前情提要可能会好一些 For dialogue applications that require very long conversations, summarize or filter previous dialogue

Since GPTs have a fixed context length, dialogue between a user and an assistant in which the entire conversation is included in the context window cannot continue indefinitely.

There are various workarounds to this problem, one of which is to summarize previous turns in the conversation. Once the size of the input reaches a predetermined threshold length, this could trigger a query that summarizes part of the conversation and the summary of the prior conversation could be included as part of the system message. Alternatively, prior conversation could be summarized asynchronously in the background throughout the entire conversation.

An alternative solution is to dynamically select previous parts of the conversation that are most relevant to the current query. See the tactic "Use embeddings-based search to implement efficient knowledge retrieval".

GPT 的上下文长度限制是给定的（整个对话都包在上下文窗口里）。意思就是说：对话不能无限长地说下去。GPT 不可能记住一切。你希望它随时都能切换回当时的上下文继续聊，这就不太能做到。

针对这个问题，有几种解决办法。其中一种是**时不时梳理并总结之前的对话内容——做个前情提要**。

当输入内容达到一个预设的长度阈值时，可以触发一个查询来总结对话的一部分，然后将之前对话的总结纳入系统消息的一部分。或者，也可以在整个对话过程中，在后台异步总结之前的对话。

另外一个办法是，**动态地挑选对话中与当前查询最相关的前文内容**。具体可以参考[“利用 embedding 搜索来做高效的知识检索”](#)。



3.3 分段总结长文档，并通过递归方式构建完整摘要 Summarize long documents piecewise and construct a full summary recursively

Since GPTs have a fixed context length, they cannot be used to summarize a text longer than the context length minus the length of the generated summary in a single query.

To summarize a very long document such as a book we can use a sequence of queries to summarize each section of the document. Section summaries can be concatenated and summarized producing summaries of summaries. This process can proceed recursively until an entire document is summarized. If it's necessary to use information about earlier sections in order to make sense of later sections, then a further trick that can be useful is to include a running summary of the text that precedes any given point in the book while summarizing content at that point. The effectiveness of this procedure for summarizing books has been studied in previous research by OpenAI using variants of GPT-3.

GPT 的上下文长度是固定的。

这意味着：不能在单次对话中，总结出，长度超过“上下文长度减去生成摘要长度”，的文本。

也就是说，如果要总结一篇非常长的文档，比如一本书。我们可以利用一系列的 prompt 来分别总结文档的每个章节。每个章节的摘要可以拼接在一起，然后再次进行总结，**形成“摘要的摘要”**。

这个过程可以递归执行，直到整个文档被完整总结。

如果需要依赖前面章节的信息来理解后面的章节，那么可以在总结某个点的内容时，包含上前情提要。

OpenAI 之前用 GPT-3 做过类似的研究，探讨了用这种逐步摘要方法做文本总结的有效性。

摘要的摘要

这个用例是用一系列的提示词来完成：

1. 章节摘要

请为以下文本的第一章创建一个简明的摘要，包括主要事件和概念。

[插入第一章的文本]

2. 每一章都有摘要了之后，拼接和二次摘要

请为以下每章的摘要创建一个总体摘要，突出每个时期的主要事件和变化。

[插入所有章节的摘要]

3. （在需要上文联系时）包含前情提要：

请参考前面章节的摘要，并为以下文本的第五章创建一个简明的摘要，包括主要事件和概念。

前情提要: [插入前面章节的摘要]

[插入第五章的文本]

直接摘要

请为以下文本创建一个简明的摘要。请充分考虑上下文，请不要忽略某个章节的文本。

[插入完整的文本]



六大策略 04 「允许 GPT 思考一下」

Give GPTs time to "think"

4.1 给时间，让它先自己想 Use inner monologue or a sequence of queries to hide the model's reasoning process.

Sometimes we get better results when we explicitly instruct the model to reason from first principles before coming to a conclusion. Suppose for example we want a model to evaluate a student's solution to a math problem. The most obvious way to approach this is to simply ask the model if the student's solution is correct or not.

But the student's solution is actually not correct! We can get the model to successfully notice this by prompting the model to generate its own solution first.

这样做会得到更好的结果：**指示模型从基本原理出发进行推理，然后再得出结论。**

比如说，我们想让模型评估一个学生做的数学题。最直接的做法就是：简单地问模型，学生的答案对不对。就像下表左边一样，其实学生做错了！但是模型判对了。

如果你像下表右边一样，让模型先自己做一遍题，它就能成功地注意到这一点。

被 GPT 判错的一道数学题

要求：判断学生的答案是否正确。

题目：我正在建造一个太阳能发电装置，需要你帮我做个财务计算。

- 土地成本100美元/平方英尺
- 我可以以250美元/平方英尺的价格购买太阳能电池板
- 我谈了一份维修合同，每年要我10万美元，每平方英尺还要多付10美元。

第一年运营的总成本是多少？

学生解题过程：设x为装置的大小，单位为平方英尺。

- 1.土地成本：100x
- 2.太阳能电池板成本：250x
- 3.维护费用：100000 + 100x

先让 GPT 解题，再判卷，就对了

判错原因：其实是学生代进去的数值抄错了，把10写成了100。但是GPT没注意到这一点。

如果让GPT自己解一遍题，他就会注意到，不会再判错。

+ 这一段 prompt:

首先想出你自己解决这个问题的办法。然后将你的答案与学生的答案进行比较，评估学生的答案是否正确。在你自己做完这道题之前，不要决定学生的答案是否正确。

总成本: $100 \times + 250 \times + 100000 + 100 \times = 450 \times + 100000$



4.2 内心独白法：不想暴露推理过程时的操作

The previous tactic demonstrates that it is sometimes important for the model to reason in detail about a problem before answering a specific question. For some applications, the reasoning process that a model uses to arrive at a final answer would be inappropriate to share with the user. For example, in tutoring applications we may want to encourage students to work out their own answers, but a model's reasoning process about the student's solution could reveal the answer to the student.

Inner monologue is a tactic that can be used to mitigate this. The idea of inner monologue is to instruct the model to put parts of the output that are meant to be hidden from the user into a structured format that makes parsing them easy. Then before presenting the output to the user, the output is parsed and only part of the output is made visible.

上一段说了，在回答问题之前，让模型自己做一遍推理是很有用的。

这里再补充一个教育的用例：如果场景是辅导课业，可能需要鼓励学生自己找出答案。但模型又要自己解一遍题。如果把这个解题过程输出出来，可能会无意中泄露答案。

这时候可以用到“**内心独白法**”。意思是，让模型把那些不想让用户看到的内容，**隐藏地放到一个结构化的格式里**。然后，在把输出展示给用户之前，解析一下这段输出，只展示能给学生看到的那部分。

大致步骤如下：

第一步: 提出问题，并得到模型的解答。（**但是不将此解答展现给学生。**）

第二步: 将学生的解题法和模型的进行比较，判学生解是对是错。（**但是仍然不将信息展现给学生。**）

第三步: 根据前两步的分析，发送反馈给学生：错了就给提示鼓励改正，对了就直接鼓励。

这里，前两步有一个“**查询和分析过程**”。但过程中**内容都是隐藏的**，学生只会看到最后一步的反馈。

这部分**需要代码实现**，而不是仅仅依赖于单一的 prompt。比如说可能需要设计和实现一个系统，包含一个查询序列的函数。并且可能需要一个中间层来管理这个过程。

各步骤的示例提示词如下：

辅导课业的用例

SYSTEM

请按照以下步骤回答用户的问题。

步骤1 - 首先，用你自己的解题思路来解决问题。不要看学生的答案，学生的答案可能是不对的。把你的题解用三引号括起来。

步骤2 - 比较你的题解和学生的答案，并评估学生的答案是否正确。这步的过程也用三引号括起来。

步骤3 - 如果学生的答案不对，那么，你可以给学生什么样的提示，在不泄答案的情况下？这步的过程也用三引号括起来。

步骤4 - 如果学生的答案不对，提供上一步你想好的提示给学生。而不是写“步骤4 -...”“写“提示：“。

USER

问题描述：<问题描述在这里>

学生解题过程：<写在这里>

下面是，让模型比较自己的答案和学生的答案，并评估学生的答案是否正确，再发送反馈。

SYSTEM

你是个数学家教。如果学生做错了题，你要给学生提示，但是不暴露答案。如果学生没有做错，那你就鼓励鼓励他们。

USER

问题陈述：“”

您的解题方法：“”

学生的解题方法：“<插入学生解答>”

分析：“”

补充说明：

这个用例非常细致。第一次看的时候我就想，OpenAI 连这么小的场景都想到了？其实同样的原理也可以用在很多其他的工作场景里。比如市场营销文案，有时也需要让模型自己做一个推理，最终输出结果就可以，把中间步骤隐去不打出来。

在其他一些应用场景里，比如客户服务、产品推荐或者数据分析等，这种策略也能帮助提高用户体验，让输出更加直接和易于理解。

对于开发者和产品经理来说，能够控制何时以及如何展示模型的推理过程是很重要的，它可以帮助确保系统的输出符合特定的需求和标准。



4.3 最后问它一句：哎，你有没有漏掉啥？ Tactic: Ask the model if it missed anything on previous passes

Suppose that we are using a model to list excerpts from a source which are relevant to a particular question. After listing each excerpt the model needs to determine if it should start writing another or if it should stop. If the source document is large, it is common for a model to stop too early and fail to list all relevant excerpts. In that case, better performance can often be obtained by prompting the model with followup queries to find any excerpts it missed on previous passes.

假设我们让大语言模型从某份文档资料中，找出与某个问题有关的原文摘要，回复给我们。

每列出一个摘要后，模型需要决定是继续列出下一个，还是就此停止。

如果资料篇幅很长，模型通常会过早地停下来，不会列完所有相关的摘要。

这就是为什么我们发现文章一太长，GPT 就偷懒的原因。

这个时候，你就告诉它：**哎，还没做完呢！**这听起来很傻，但是确实有效。

这么写可能会好一些

...前文省略

Are there more relevant excerpts? Take care not to repeat excerpts. Also ensure that excerpts contain all relevant context needed to interpret them - in other words don't extract small snippets that are missing important context.

这么写可能不太行

你没做完任务，请重头来一遍。

(结果可能还会停在中间)

还有没有更多的摘要了？请注意：不要重复摘要。请确保摘录包含了所有相关的上下文——意思是，请不要提



六大策略 05 「借助外部的工具」

Strategy: Use external tools

5.1 使用基于 **embedding** 的搜索来实现高效的知识检索 Use embeddings-based search to implement efficient knowledge retrieval

A model can leverage external sources of information if provided as part of its input. This can help the model to generate more informed and up-to-date responses. For example, if a user asks a question about a specific movie, it may be useful to add high quality information about the movie (e.g. actors, director, etc...) to the model's input. Embeddings can be used to implement efficient knowledge retrieval, so that relevant information can be added to the model input dynamically at run-time.

A text embedding is a vector that can measure the relatedness between text strings. Similar or relevant strings will be closer together than unrelated strings. This fact, along with the existence of fast vector search algorithms means that embeddings can be used to implement efficient knowledge retrieval. In particular, a text corpus can be split up into chunks, and each chunk can be embedded and stored. Then a given query can be embedded and vector search can be performed to find the embedded chunks of text from the corpus that are most related to the query (i.e. closest together in the embedding space).

Example implementations can be found in the [OpenAI Cookbook](#). See the tactic “[Instruct the model to use retrieved knowledge to answer queries](#)” for an example of how to use knowledge retrieval to minimize the likelihood that a model will make up incorrect facts.

(这段没有逐字翻译)

模型可以利用外部信息来丰富自己的输入，这样可以做出更有料、更贴近现实的回答。

比如，用户问到某部电影，给模型加入电影的背景资料（比如谁演的、导演是谁等）可能会帮大忙。

这里的“嵌入”，也就是 **embedding** 技术，就是用来帮助高效找到这些知识，实时加入模型的。

简单来说，文本嵌入是基于向量数据，通过看文本之间有多“近”实现的。

实际的操作过程，可以看 OpenAI 的 Cookbook 里的这个实例。



5.2 需要做精确计算的时候：用代码、或调 API Use code execution to perform more accurate calculations or call external APIs

GPTs cannot be relied upon to perform arithmetic or long calculations accurately on their own. In cases where this is needed, a model can be instructed to write and run code instead of making its own calculations. In particular, a model can be instructed to put code that is meant to be run into a designated format such as triple backticks. After an output is produced, the code can be extracted and run. Finally, if necessary, the output from the code execution engine (i.e. Python interpreter) can be provided as an input to the model for the next query.

Another good use case for code execution is calling external APIs. If a model is instructed in the proper use of an API, it can write code that makes use of it. A model can be instructed in how to use an API by providing it with documentation and/or code samples showing how to use the API.

WARNING: Executing code produced by a model is not inherently safe and precautions should be taken in any application that seeks to do this. In particular, a sandboxed code execution environment is needed to limit the harm that untrusted code could cause.

(这段没有逐字翻译)

GPT 算数有可能不靠谱。在较长的计算里，它很有可能在“打酱油”。

真的需要算，最好是让它写代码去算。运行代码输出结果后，再丢回给模型继续处理。

写 Python 代码求实数根的用例

SYSTEM

You can write and execute Python code by enclosing it in triple backticks, e.g. ````code goes here````. Use this to perform calculations.

你可以编写并执行一段 Python 代码，代码放在三引号里面，比如：````代码放在这里````。请用这种方式来执行计算。

USER

找出以下多项式的所有实数根： $3x^5 - 5x^4 - 3x^3 - 7x - 10$ 。

输出结果：

```
import numpy as np

# Define the coefficients of the polynomial

coefficients = [3, -5, -3, 0, -7, -10]

# Use numpy's roots function to find all roots (both real and complex)

roots = np.roots(coefficients)

# Filter out only the real-valued roots

real_roots = [root.real for root in roots if root.imag == 0]

real_roots
```

另外，模型也能帮忙写代码调用 API，这个过程也比较简单。

调用 API 的示例

这段代码是展示了一个假设的 message 模块的用法，通过 message.write 函数来发送消息给指定的人。

SYSTEM

You can write and execute Python code by enclosing it in triple backticks. Also note that you have access to the following module to help users send messages to their friends:

```
```python

import message

message.write(to="John", message="Hey, want to meetup after work?")```
```

最后要提请注意：模型写的代码不见得都是对的，都安全。跑代码之前要确保环境安全，最好在沙盒里。



## 5.3 使用 function-calling，函数调用 Give the model access to specific functions.

The Chat completions API allows passing a list of function descriptions in requests. This enables models to generate function arguments according to the provided schemas. Generated function arguments are returned by the API in JSON format and can be used to execute function calls. Output provided by function calls can then be fed back into a model in the following request to close the loop. This is the recommended way of using GPT models to call external functions. To learn more see the [function calling section](#) in our introductory GPT guide and more [function calling examples](#) in the OpenAI Cookbook.

(这段没有逐字翻译)

API 允许你在请求时附带一系列函数描述。

这样，模型就可以按照你给的格式产生函数的参数了。

API会用JSON形式把这些参数传回来，你就可以用它来调用函数了。

函数跑完后的结果，你还可以继续拿给模型，形成一个完整的交互循环。

想让GPT帮你调外部函数，这方法相对最靠谱。想深入了解的话，可以看 GPT 入门指南里的[函数调用](#)部分，还有 [OpenAI Cookbook](#) 里的用例。



# 六大策略 06：系统地测试

## Test changes systematically

### 6.1 评估模型输出的时候，需要你弄一个“黄金标准”的答案 Evaluate model outputs with reference to gold-standard answers

Suppose it is known that the correct answer to a question should make reference to a specific set of known facts. Then we can use a model query to count how many of the required facts are included in the answer.



(这段没有逐字翻译)

假设我们作为人类清楚地知道，当我们提出某个问题时，模型应该给我们输出哪些事实（得分点）。

那么，我们就可以使用模型查询和计算的方法，来总结这些得分点，从而给模型的输出质量打分。

这个过程是监督学习的一部分，它能让模型通过人类的指导来改善其性能。

评估模型输出的过程也是由另一个模型来执行的（而不是人）。这是一个自动化的评估过程。比如，一个模型（可能是同一个模型的不同实例或不同的模型）负责评估另一个模型的输出。否则，做大规模的模型训练的人力成本就太高了。

OpenAI 提供了一些 prompt 示例，我们这里略去了。有兴趣可以参考[原链接](#)。



## 附录

# OpenAI Playground 什么，怎么用？

## 什么是 OpenAI Playground...

### 一个面向开发人员和研究人员的“游乐场”

在 Playground 上可以按自己需求更改设置。这是最好的一点。

#### 1、三个可选的模式 (Mode)

现在只有 Chat 是大家持续在用的。补全和编辑后面都跟了一个英语词，直译为“遗产”，意思是已经不再更新了，已弃。

- Chat 聊天
- Complete 补全（已弃）
- Edit 编辑（已弃）

## 2、可选的模型 (Model)

OpenAI 公布的所有 GPT 模型都可以选。

## 3、提示词结构 (Prompt structure)

使用 ChatGPT 和使用 Chat Completion API 之间的最大区别之一就是提示词结构。

打开 ChatGPT，你只有一个对话框，你只需，也只能，发送一条简单的消息。

虽然说你添加尽可能多的上下文到消息中，但很有可能说着说着就丢失上下文。

但是在 PlayGround 里就多了一种玩法：这里区分了 SYSTEM 和 USER 两个对话框。

**SYSTEM：**系统设定。默认情况下，GPT 是“一个有用的助手”。

你可以用它来控制它扮演什么样的角色，从而控制你得到什么样的回复。

比如，你甚至可以把它设置成成心捣乱的一个角色：“杠精”。

然后，再在 USER 里进行你常规的对话。

## 4、温度 (Temperature)

温度控制 GPT 生成的文本的随机性的参数。取值范围：0到2之间（默认值为1）。

如果没有一定程度的随机性，GPT 只能产生完全无聊和可预测的结果。因为它实际上只是将最有可能的后续单词添加到你给定的句子中。

所以：温度为0时，结果是无聊的、确定的；即使你改变 prompt，结果也可能是几乎相同的。

但是，把温度推到任何一个极端，都不要指望会得到可用的文本。

推到最高并不等于创造力爆棚，因为输出结果一般都是乱码。

## 5、最大长度 (Maximum length)

Token 这个词，有时候被一些机翻的文档写成“令牌”或者“代币”。都不对。听起来比较正确的应该是日本的译法“词元”。但是一般好像大家都喜欢直接说 token。

GPT-3.5，最大 token 数是：2048（大约 1500 个单词）。

OpenAI 的收费模式：根据 token 使用数量收费。

这意味着：养成了用更强大的模型生成输出的习惯，成本可能会增加。如果你是在生产环境里面使用，为了成本考虑，还是需要控制一下这个的，可以通过限定 max token 来降低成本。

## 6、终止序列 (Stop sequences)

终止序列的作用就是告诉模型该在哪里打住，不再继续输出文字。

通过它，你能隐式地控制文字的长短。比如说，如果你只想要模型回答个短短的句子，那就设定句号.作为终止序列；如果你想要一个段落的回答，那就把“新行”(\n)设为终止序列。

代表性用例：如果你试图生成的是一个或一些问答对，即 Q&A，这个就用上了。因为你会想让模型简洁地回答这个问题，而不是写一大篇文章。设置句号.作为终止序列，这样模型就会在完成一个句子后停止输出。

## 7、Top P (P 的意思是)

**Top P 是控制 GPT 输出的随机性的另一种方法。**温度决定的是，模型从可能的下一个单词列表中做选择的随机程度。**Top P 决定的则是：该列表的长度。**

假设我给出来一个句子：“J.K.罗琳是一个...”

下一个单词是什么呢？GPT 有一个单词列表，它在做的事情就是：预测概率，进行排名。

假设能接上这个句子的下一个单词的排名表长这样：

- 80%: 作家
- 20%: 英国人
- 20%: 富翁
- 10%: 充满奇思妙想的
- 5%: 美女
- 4%: 母亲
- 1%: 矮子

所以，这个句子接下来最有可能的单词是：作家、英国人。

但也有一定的概率，1% 的概率，这句话可以说成：“J.K.罗琳是一个矮子”。

如果 Top P 值为 1，则这个列表会非常非常非常长，包含所有可能的单词（即使是“矮子”这种，概率非常小，总归也是有概率的）。

反之，当你把 Top P 调到接近 0 时，选择就会非常少，列表就会比较短。

有时候，当 Top P 设置为 0 时，它返回的结果与把温度设置为 0 时完全相同。

## 8、频率惩罚和存在惩罚 (Frequency penalty and Presence penalty)

这两个参数都是用来调控文本生成过程中某些词汇出现的频率和次数的。

简单来说，通过设定一些惩罚机制，可以影响生成文本的内容，使其更符合我们想要的样子。

**Frequency penalty 频率惩罚：**基于一个词在文本中出现的次数来给它加上一定的惩罚。一个词出现的次数越多，它受到的惩罚就越多，从而降低这个词再次出现的可能性。

用处：降低用词重复的可能性。比如在写一篇文章时，你不想某个词汇总是重复出现，就可以通过设置一定的频率惩罚来达到这个目的。

**Presence penalty 出现惩罚或者存在惩罚：**则是根据一个词是否已经出现在文本中来惩罚。只要这个词已经在文本中出现过，无论出现多少次，都会给它加上一定的惩罚，这是为了鼓励模型探索新的词汇或话题。

用处：有助于向新的内容方向转移，增加输出的多样性。比如在一个自由讨论中，你想让模型能够尽可能多地探讨不同的话题，而不是总是在同一个话题上打转，就可以设置存在惩罚来实现。

总之，频率惩罚和存在惩罚在某种程度上都是为了增加文本的多样性和丰富性，只是侧重点和应用场景稍微有点不同。

谢谢阅读！

关注未来力场，获得更多 AI 实战用例和分享资讯

公众号：未来力场

