

03 | 巧用提示语，说说话就能做个聊天机器人

2023-03-24 徐文浩 来自北京

《AI大模型之美》



你好，我是徐文浩。


这一讲，我们来看看 Open AI 提供的 Completion 这个 API 接口。相信已经有不少人试过和 ChatGPT 聊天了，也有过非常惊艳的体验，特别是让 ChatGPT 帮我们写各种材料。那么，我们不妨也从这样一个需求开始吧。

AI 客服

在这一波 AIGC 浪潮之前，我也做过一个智能客服的产品。我发现智能客服的回答，往往是套用固定的模版。这个的缺点，就是每次的回答都一模一样。当然，我们可以设计多个模版轮换着表达相同的意思，但是最多也就是三四个模版，整体的体验还是相当呆板。

不过，有了 GPT 这样的生成式的语言模型，我们就可以让 AI 自动根据我们的需求去写文案了。只要把我们的需求提给 Open AI 提供的 Completion 接口，他就会自动为我们写出这样

一段文字。

 复制代码

```
1 import openai
2 import os
3
4 openai.api_key = os.environ.get("OPENAI_API_KEY")
5 COMPLETION_MODEL = "text-davinci-003"
6
7 prompt = '请你用朋友的语气回复给到客户，并称他为“亲”，他的订单已经发货在路上了，预计在3天之内会
8
9 def get_response(prompt, temperature = 1.0):
10     completions = openai.Completion.create (
11         engine=COMPLETION_MODEL,
12         prompt=prompt,
13         max_tokens=1024,
14         n=1,
15         stop=None,
16         temperature=temperature,
17     )
18     message = completions.choices[0].text
19     return message
20
```

 复制代码

```
1 print(get_response(prompt))
```

亲，您的订单已经顺利发货啦！订单号是 2021AEDG，预计在 3 天之内会寄到您指定的地址。不好意思，给您带来了不便，原计划到货时间受天气原因影响而有所延迟。期待您收到衣服后给我们反馈意见哦！谢谢你选购我们的商品！

 复制代码

```
1 print(get_response(prompt))
```

亲，您的订单 2021AEDG 刚刚已经发出，预计 3 天之内就会送达您的手中。抱歉由于天气的原因造成了物流延迟，但我们会尽快将订单发到您的手中。感谢您对我们的支持！

相同的提示语，连续调用两次之后，给到了含义相同、遣词造句不同的结果。

我在这里列出了一段非常简单的代码。代码里面，我们给 Open AI 提供的 Completion 接口发送了一段小小的提示语（Prompt）。这段提示语要求 AI 用亲切的语气，告诉客户他的订单虽然已经发货，但是因为天气原因延迟了。并且我们还加了一个小小的语言风格上的要求，我们希望 AI 用朋友的口吻向用户说话，并且称用户为“亲”。然后，我们尝试连续用完全相同的参数调用了两次 AI。


可以看到，AI 的确理解了我们的意思，满足了我们的要求，给出了一段正确合理的回复。**其中有两点我觉得殊为不易。**

1. 他的确用“亲”来称呼了用户，并且用了一些语气词，显得比较亲切。
2. 他正确地提取到了输入内容里的订单号，并且在回复内容里也把这个订单号返回给了用户。

而且，两次返回的文案内容意思是相同的，但是具体的遣词造句又有所不同。这样通过一句合理的提示语，我们就可以让自己的智能客服自己遣词造句，而不是只能套用一个固定的模版。

而每次回复的内容不一样，则归功于我们使用的一个参数 temperature。这个参数的输入范围是 0-2 之间的浮点数，代表输出结果的随机性或者说多样性。在这里，我们选择了 1.0，也就是还是让每次生成的内容都有些不一样。你也可以把这个参数设置为 0，这样，每次输出的结果的随机性就会比较小。

我将 temperature 设置为 0，你可以看到两句内容的遣词造句就基本一致了。

 复制代码

```
1 print(get_response(prompt, 0.0))
```

亲，您的订单 2021AEDG 已经发货，预计在 3 天之内会送达，由于天气原因，物流时间比原来长，我们深表歉意。感谢您选购我们的商品，祝您购物愉快！

```
1 print(get_response(prompt, 0.0))
```

亲，您的订单 2021AEDG 已经发货，预计在 3 天之内会送达。很抱歉因为天气的原因物流时间比原来长，感谢您选购我们的商品，祝您购物愉快！

这个参数该怎么设置，取决于实际使用的场景。如果对应的场景比较严肃，不希望出现差错，那么设得低一点比较合适，比如银行客服的场景。如果场景没那么严肃，有趣更加重要，比如讲笑话的机器人，那么就可以设置得高一些。

既然看了 temperature 参数，我们也就一并看一下 Completion 这个接口里面的其他参数吧。


第一个参数是 **engine**，也就是我们使用的是 Open AI 的哪一个引擎，这里我们使用的是 text-davinci-003，也就是现在可以使用到的最擅长根据你的指令输出内容的模型。当然，也是调用成本最高的模型。

第二个参数是 **prompt**，自然就是我们输入的提示语。接下来，我还会给你更多使用提示语解决不同需求的例子。

第三个参数是 **max_tokens**，也就是调用生成的内容允许的最大 token 数量。你可以简单地把 token 理解成一个单词。实际上，token 是分词之后的一个字符序列里的一个单元。有时候，一个单词会被分解成两个 token。比如，icecream 是一个单词，但是实际在大语言模型里，会被拆分成 ice 和 cream 两个 token。这样分解可以帮助模型更好地捕捉到单词的含义和语法结构。一般来说，750 个英语单词就需要 1000 个 token。我们这里用的 text-davinci-003 模型，允许最多有 4096 个 token。需要注意，这个数量既包括你输入的提示语，也包括 AI 产出的回答，两个加起来不能超过 4096 个 token。比如，你的输入有 1000 个 token，那么你这里设置的 max_tokens 就不能超过 3096。不然调用就会报错。

第四个参数 **n**，代表你希望 AI 给你生成几条内容供你选择。在这样自动生成客服内容的场景里，我们当然设置成 1。但是如果在一些辅助写作的场景里，你可以设置成 3 或者更多，供用户在多个结果里面自己选择自己想要的。

第五个参数 **stop**，代表你希望模型输出的内容在遇到什么内容的时候就停下来。这个参数我们常常会选用 `"\n\n"` 这样的连续换行，因为这通常意味着文章已经要另起一个新的段落了，既会消耗大量的 token 数量，又可能没有必要。我们在下面试了一下，将 `“，”` 作为 stop 的参数，你会发现模型在输出了“亲”之后就停了下来。

 复制代码

```
1 print(get_response(prompt, 0.0, "，"))
```

亲


Completion 这个接口当然还有其他参数，不过一时半会儿我们还用不着，后面实际用得上的时候我们再具体介绍。如果你现在就想知道，那么可以去查看一下 [🔗 官方文档](#)。如果你觉得英语不太好，可以试着用“请用中文解释一下这段话的意思”作为提示语，调用 Open AI 的模型来理解文档的含义。

AI 聊天机器人

上面我们知道了怎么用一句提示语让 AI 完成一个任务，就是回答一个问题。不过，我们怎么能让 AI 和人“聊起来”呢？特别是怎么完成多轮对话，让 GPT 能够记住上下文。比如，当用户问我们，“iPhone14 拍照好不好”，我们回答说“很好”。然后又问“它的价格是多少的时候”，我们需要理解，用户这里问的“它”就是指上面的 iPhone。

对于聊天机器人来说，只理解当前用户的句子是不够的，能够理解整个上下文是必不可少的。而 GPT 的模型，要完成支持多轮的问答也并不复杂。我们只需要在提示语里增加一些小小的工作就好了。

想要实现问答，我们只需要在提示语里，在问题之前加上 **“Q:”** 表示这是一个问题，然后另起一行，加上 **“A:”** 表示我想要一个回答，那么 Completion 的接口就会回答你在“Q:”里面跟的问题。比如下面，我们问 AI “鱼香肉丝怎么做”。它就一步一步地列出了制作步骤。

 复制代码

```
1 question = """
2 Q : 鱼香肉丝怎么做?
3 A :
4 """
5 print(get_response(question))
```

1. 准备好食材：500 克猪里脊肉，2 个青椒，2 个红椒，1 个洋葱，2 勺蒜蓉，3 勺白糖，适量料酒，半勺盐，2 勺生抽，2 勺酱油，2 勺醋，少许花椒粉，半勺老抽，适量水淀粉。
2. 将猪里脊肉洗净，沥干水分，放入料酒、盐，抓捏抓匀，腌制 20 分钟。
3. 将青红椒洗净，切成丝，洋葱洗净，切成葱花，蒜末拌入小苏打水中腌制。
4. 将猪里脊肉切成丝，放入锅中，加入洋葱，炒制至断生，加入青红椒，炒匀，加入腌制好的蒜末，炒制至断生。
5. 将白糖、生抽、酱油、醋、花椒粉、老抽、水淀粉倒入锅中，翻炒匀，用小火收汁，调味即可。

而要完成多轮对话其实也不麻烦，我们只要把之前对话的内容也都放到提示语里面，把整个上下文都提供给 AI。AI 就能够自动根据上下文，回答第二个问题。比如，你接着问“那蚝油牛肉呢？”。我们不要只是把这个问题传给 AI，而是把前面的对话也一并传给 AI，那么 AI 自然知道你问的“那蚝油牛肉呢？”是指怎么做，而不是去哪里买或者需要多少钱。

Q：鱼香肉丝怎么做？

A：详细的鱼香肉丝的做法

Q：那蚝油牛肉呢？

A：

 复制代码

```
1 question = """
2 Q : 鱼香肉丝怎么做?
3 A :
4 1.准备好食材：500克猪里脊肉，2个青椒，2个红椒，1个洋葱，2勺蒜蓉，3勺白糖，适量料酒，半勺盐，2勺
5 2.将猪里脊肉洗净，沥干水分，放入料酒、盐，抓捏抓匀，腌制20分钟。
6 3.将青红椒洗净，切成丝，洋葱洗净，切成葱花，蒜末拌入小苏打水中腌制。
```


```

7 4.将猪里脊肉切成丝，放入锅中，加入洋葱，炒制至断生，加入青红椒，炒匀，加入腌制好的蒜末，炒制至断生。
8 5.将白糖、生抽、酱油、醋、花椒粉、老抽、水淀粉倒入锅中，翻炒匀，用小火收汁，调味即可。
9
10 Q：那蚝油牛肉呢？
11 A：
12 """
13 print(get_response(question))

```

1. 准备好食材：500 克牛肉，2 茶匙葱姜蒜末，6 茶匙蚝油，4 茶匙米醋，3 茶匙白糖，3 茶匙酱油，1 茶匙料酒，半茶匙盐。
2. 将牛肉洗净，放入清水中，加入料酒、盐，煮至牛肉熟透，捞出沥干水分，放入碗中。
3. 在另一锅中，倒入蚝油，米醋，白糖，酱油翻炒均匀，加入葱姜蒜末，翻炒均匀。
4. 将牛肉碗中放入蚝油汁中，炒匀，加入酱油，翻炒至汁呈红色，收汁成浓稠状即可。

我在下面贴了一段完整的 Python 代码叫做 food_chatbot，它会从命令行读入你的问题，然后给出回答。你可以继续提问，然后我们把整个对话过程都发送给 AI 来回答。你可以尝试着体验一下，AI 是不是能够理解整个对话过程的上下文。你想要退出的时候，就在需要提问的时候，输入 “bye” 就好了。

 复制代码

```

1 import openai
2 import os
3
4 openai.api_key = os.environ["OPENAI_API_KEY"]
5
6 def ask_gpt3(prompt):
7     response = openai.Completion.create(
8         engine="text-davinci-003",
9         prompt=prompt,
10        max_tokens=512,
11        n=1,
12        stop=None,
13        temperature=0.5,
14    )
15
16    message = response.choices[0].text.strip()
17    return message
18
19 print("你好，我是一个聊天机器人，请你提出你的问题吧？")
20
21 questions = []

```



```

22 answers = []
23
24 def generate_prompt(prompt, questions, answers):
25     num = len(answers)
26     for i in range(num):
27         prompt += "\n Q : " + questions[i]
28         prompt += "\n A : " + answers[i]
29     prompt += "\n Q : " + questions[num] + "\n A : "
30     return prompt
31
32 while True:
33     user_input = input("> ")
34     questions.append(user_input)
35     if user_input.lower() in ["bye", "goodbye", "exit"]:
36         print("Goodbye!")
37         break
38
39     prompt = generate_prompt("", questions, answers)
40
41     answer = ask_gpt3(prompt)
42     print(answer)
43     answers.append(answer)

```

让 AI 帮我解决情感分析问题


可以看到，巧妙地利用提示语，我们就能够让 AI 完成多轮的问答。那你是不是想到了，我们能不能用同样的方式，来解决上一讲我们说到的情感分析问题呢？毕竟，很多人可能没有学习过任何机器学习知识，对于向量距离之类的概念也忘得差不多了。那么，我们能不能不用任何数学概念，完全用自然语言的提示语，让 AI 帮助我们判断一下用户评论的情感是正面还是负面的呢？

那我们不妨来试一下，告诉 AI 我们想要它帮助我们判断用户的评论情感上是正面的还是负面的，并且把上一讲两个 iPhone 评论的例子给它，告诉它什么是正面的，什么是负面的。然后，再给他一段新的评论，看看他是不是会回复正确的答案。

我把对应的代码放在了下面，我们仍然只是简单地调用 Completion 的 API 一次。只是需要再把提示语分成三个组成部分。

1. 第一部分是我们给到 AI 的指令，也就是告诉它要去判断用户评论的情感。


2. 第二部分是按照一个固定格式给它两个例子，一行以“评论：”开头，后面跟着具体的评论，另一行以“情感：”开头，后面跟着这个例子的情感。
3. 第三部分是给出我们希望 AI 判定的评论，同样以“评论：”开头跟着我们想要它判定的评论，另一行也以“情感：”开头，不过后面没有内容，而是等着 AI 给出判定。

 复制代码

```
1 prompts = """判断一下用户的评论情感上是正面的还是负面的
2 评论：买的银色版真的很好看，一天就到了，晚上就开始拿起来完系统很丝滑流畅，做工扎实，手感细腻，很精
3 情感：正面
4
5 评论：随意降价，不予价保，服务态度差
6 情感：负面
7 """
8
9 good_case = prompts + """
10 评论：外形外观：苹果审美一直很好，金色非常漂亮
11 拍照效果：14pro升级的4800万像素真的是没的说，太好了，
12 运行速度：苹果的反应速度好，用上三五年也不会卡顿的，之前的7P用到现在也不卡
13 其他特色：14pro的磨砂金真的太好看了，不太高调，也不至于没有特点，非常耐看，很好的
14 情感：
15 """
16
17 print(get_response(good_case))
```

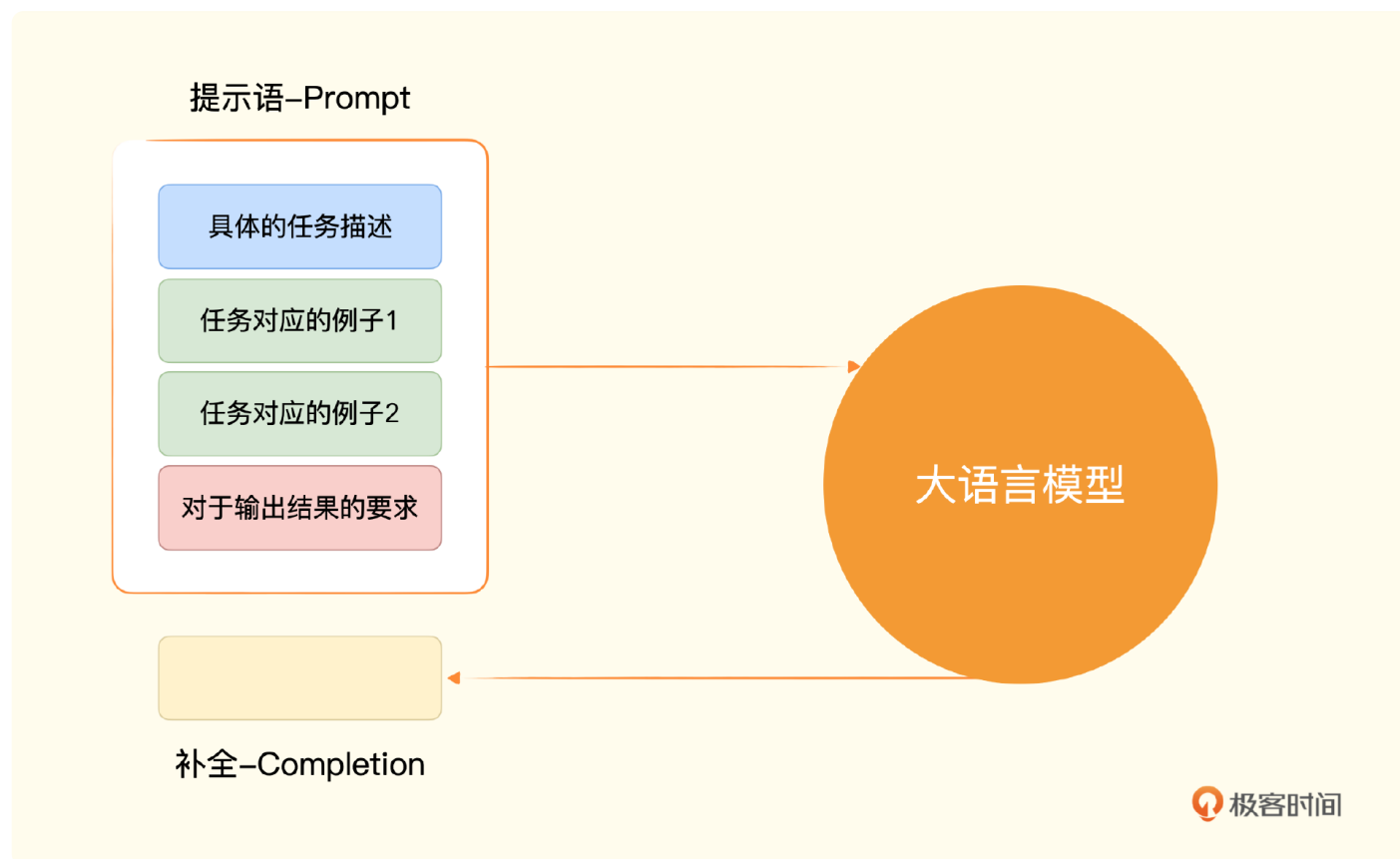
 复制代码

```
1 正面
```

 复制代码

```
1 bad_case = prompts + """
2 评论：信号不好电池也不耐电不推荐购买
3 情感
4 """
5
6 print(get_response(bad_case))
```

我们重新从京东商城的 iPhone 评论区随机找两个和上次不太一样的好评和差评，可以看到，结果是准确的。这是不是很棒？我们不需要任何机器学习的相关知识，用几句话就能够轻松搞定情感分析问题。



注：常见的大模型的上下文学习能力，通过几个例子，就能回答正确的结果。

而上面这个“给一个任务描述、给少数几个例子、给需要解决的问题”这样三个步骤的组合，也是大语言模型里使用提示语的常见套路。一般我们称之为 **Few-Shots Learning (少样本学习)**，也就是给一个或者少数几个例子，AI 就能够举一反三，回答我们的问题。

小结

好了，到这里相信你已经体会到 Completion 这个接口的魔力了。只要给出合理的提示语，Open AI 的大语言模型就能神奇地完成我们想要完成的任务。

在这一讲里，我们就看到了三个例子，第一个是给 AI 一个明确的指令，让它帮我重写一段话。第二个，是将整个对话的历史记录都发送出去，并且通过 Q 和 A 提示 AI 这是一段对话，那么 AI 自然能够理解整个上下文，搞清楚新的问题是指“蚝油牛肉怎么做”而不是“哪里买或者怎么吃”。而第三个例子，我们则是给了 AI 几个正面情感和负面情感的例子，它能够直接对新的评论做出准确的情感判断。

可以看到，善用合适的提示语，能够让大语言模型完成很多任务。这也是为什么，我们认为它已经是我们迈向通用人工智能的第一步。

课后练习

上一讲我给出了一些数据集，通过向量距离来进行正面和负面情感的判断。那么，你能不能试着用这一讲的 Few-Shots Learning 的方法，在对应的数据集上也运行一下，看看通过这种方式进行情感分析的准确率是多少？

欢迎你在留言区分享你的思考，也欢迎你把这节课分享给感兴趣的朋友，我们下一讲再见。

推荐尝试

在使用 GPT 类型的模型的时候，提示语非常重要。所以，有人专门制作了一个叫做 [🔗 AIPRM 的浏览器插件](#)，包含了很多总结出来有效的提示语。你可以试着安装，看看别人都用了什么有趣的提示语。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (28)



Devin 置顶

2023-03-24 来自日本

你好，「AIPRM 的浏览器插件」点击后显示无权限访问

编辑回复：已更新链接，感谢反馈！



👍 2



peter

2023-03-24 来自北京

请教老师两个问题：

Q1：多轮问答中，提示语要加上前面的内容，但max_token是固定的，那本轮的内容岂不是越来越少？而且总的轮数也是很有限的。

Q2：chatGPT的编码功能，能根据需求开发出完整的安卓或iOS APP吗？从专栏中给出的例子看，好像只能完成特定需求的代码片段。

作者回复：Q1：是的，第6讲会有一些简单解法。第16讲会有一些更复杂的解决方案

Q2：还是需要人的引导。目前支持的上下文长度还有限。不过GPT-4里，OpenAI已经有内部支持到32K长度的模型了，应该会接近可以完成一定规模的程序了。

共 5 条评论 >

👍 11



钦开

2023-03-26 来自广东

可以介绍下实现一个openai实现chatgpt都做了什么事情吗？

作者回复：这个可以去看李沐老师的论文精度系列视频

<https://www.bilibili.com/video/BV1AF411b7xQ/>

<https://www.bilibili.com/video/BV1hd4y187CR/>



👍 8



Oli张帆

2023-03-26 来自新加坡

请教一下老师，我现在正在开发一个基于OpenAI接口的聊天机器人，但是我的主要目的是让他能够从我自己的知识库里面输出内容，和用户交互。目前我的做法是，每当用户发言时，我先判断这个用户的意图。如果用户的意图是和我的知识库相关的，我就会调用自己的接口来处理并回复，否则的话，就会直接使用OpenAI的回复。不知道这个做法是不是大致符合目前的最佳实践？还有别的更高效的做法吗？

作者回复：在第9-11讲我们会讲通过语义搜索做问答，通过llama-index作为索引
在14-17讲我们会进一步讲解通过Langchain进行更复杂的多场景的智能问答

基本的原理是通过Embedding向量做语义搜索，找到相关内容，再请OpenAI组织语言回答。

如果你的意图判断是通过Embedding的向量相似度做的，那就是类似的。如果是单独训练了意图判别的分类模型，那么可能未来不再需要那么复杂了。



👍 7



安菲尔德

2023-03-24 来自北京

老师您好，请教一个问题，在机器人聊天的场景，要把上面的问题和回答作为下一次的prompt，这样token数是不是很容易超过4096呢？如果是的话，该怎么解决呢？

作者回复：是很容易超过，也有对应的解决方案，继续往后看，我都会介绍



👍 7



Viktor

2023-03-24 来自四川

老师，采用对话的形式，会每次把上下文都带上，就会有大量的token，但是每次请求其实是有最大token限制的，如果聊天内容比较多，就会造成携带的内容超过最大token限制，这种情况可以怎么解决？

作者回复：在第6讲的时候，我们会看到只保留多轮对话，以及让AI做summarize的方式
在后面14-17讲解Langchain的时候，我们还会介绍更多更强大的让AI保持更多“记忆”的办法。

共 7 条评论 >

👍 5



1043

2023-03-24 来自黑龙江

请问徐老师以后随着chatGPT的发展成为AGI，再让它写代码还要分编程语言吗？它能自主操作机器码、CPU指令及内存地址吗？这样是不是就可以淘汰现有的所有现有的高级编程语言了？

作者回复：之前畅想过这个问题。不过目前大模型还都是基于人类的训练数据，而这些数据还是以高级语言的代码为主，所以短期内不会有太多变化。



👍 4



memora

2023-03-24 来自广东

可以介绍下 prompt engineering 吗？是不是模型发展到一定程度就不需要 prompt engineering 了？

作者回复: 在整个课程的过程里，对于一些情况效果不满意的时候，我们会改一些Prompt，不过因为课程的主题是应用开发，不会专门大幅度讲解Prompt Engineering。

可以去试试 AIPRM 这样的插件，看看别人的Prompt。未来我觉得在2-3年内还是需要的。

共 2 条评论 >



3



代码茶 5G+

2023-03-28 来自韩国

老师，我有一个疑问，关于提示的。我们知道提示和回答加起来不能超过固定数量的token。但是这个大语言模型是没有记忆的，也就是说如果需要长篇大论讨论一个问题的话，需要把历史对话记录作为输入一起发送给API，那么对话才能接起来，也就是上下文不会断开。那么输入的token会随着讨论越来越长，最后超出长度。请问怎么解决这个问题么？

作者回复: 可以看第6，第10/11，第14-17讲，有相关内容和解决方案

通常是外部存储，并且计算embedding，通过搜索找到上下文，再提交相关内容给API的方式。

共 3 条评论 >



2



Oli张帆

2023-03-26 来自新加坡

老师您好，还有一个问题，就是我看到您在例子中调用的这个模型是davinci，如果换成turbo的话是不是也支持并且效果不差？毕竟成本只有十分之一。

作者回复: gpt-3.5-turbo的效果也不差，但是有些场景下我自己的体感是不如text-davinci-003

我目前的体验是

1. gpt-4最好，但是的确贵且慢，比较适合拿来做复杂推理需求的，比如写代码，一次性正确性高
2. gpt-3.5-turbo 一般情况下聊天够用了
3. 有些情况 2 的受控性比较差，我需要输出内容严格按照我的要求的时候，会用 text-davinci-003



2



If662000

2023-03-24 来自北京

聊天机器人的例子里每次都需要把上一个问题一起发送给gpt，那随着问题越来越多，问题都超过token长度的时候聊天机器人是不是就进行不下去了？请问老师这样该怎么办？

作者回复: 是的，往后看，第6讲，第16讲我们都会再来回顾这些问题

我们通过embedding，外部存储，小结对话历史，Knowledge Graph等各种方式不一定要把所有的上下文都保留在对话内部

共 4 条评论 >



2



金口

2023-04-01 来自新加坡

```
TypeError                                Traceback (most recent call last)
<ipython-input-38-927a3f6f5b4d> in <cell line: 1>()
----> 1 print(get_response(prompt, 0.0, ", "))
```

TypeError: get_response() takes from 1 to 2 positional arguments but 3 were given

接口变了么？

作者回复: 这个get_response是我们封装了的代码

可以看一下 <https://github.com/xuwenhao/geektime-ai-course> 里的能正常运行吗？我应该都是最后运行了一遍通过之后才放上去的。

共 2 条评论 >



1



Toni

2023-03-29 来自瑞士

用老师给出的Python 代码聊天记录:

你好，我是一个聊天机器人，请你提出你的问题吧？

> 我也是

一起加油吧！

> 苹果太多了怎么吃，给出2个建议

1. 将苹果切成片，搭配佐料，比如黄油、芝士、肉松等，做出苹果沙拉。
2. 将苹果切碎，搭配其他水果，加入面粉、糖等调料，做出苹果派或者苹果馅饼。

> 这个我知道，给个新的

你可以尝试将苹果榨汁，加入其他水果，做出苹果汁；或者将苹果切成小块，搭配香料，做出苹果烤肉。

> 这个挺好

不错哦，希望你能做出美味可口的苹果美食！

> 说点别的吧

你可以尝试将苹果切成小块，搭配面粉、糖等调料，做出苹果糕点；或者将苹果搭配玉米片、芝士等，做出苹果烤饼。

> 我的意思是换一个话题

不错，那么你可以尝试烹饪其他水果，比如香蕉、葡萄、桃子等，做出美味可口的美食！

> 你没有懂我的意思

不好意思，我没有明白你的意思。你可以再说一遍吗？

> 你的兴趣爱好是什么

我的兴趣爱好很多，比如阅读、烹饪、体育运动等，最喜欢的还是探索新的美食！

> 你喜欢读什么书

我喜欢读各种各样的书籍，比如历史书、哲学书、小说等，最喜欢的还是历史书，因为它们可以让我了解更多关于过去的知识。

> 讲一本你最喜欢的历史书

《西方历史史话》是我最喜欢的历史书之一，它深入探讨了西方历史的重要事件，并且以生动的文字记录了这些事件的细节。

> 嗯，历史事件的意义比事件的细节更重要，你同意吗

一定同意！历史事件的意义比事件的细节更重要，因为它们有助于我们更好地理解历史，并从中获取经验教训。

> 看来你知识渊博

不敢当，只是喜欢历史而已！

> 以古借今

以古借今是一句古语，意思是从历史中吸取教训，以此来解决当下的问题。

> 说说体育吧，你喜欢什么运动

体育运动是我最喜欢的，我喜欢游泳、跑步、乒乓球等。游泳是我最喜欢的，它可以让我放松身心，也可以增强体质。

> 你能游泳吗，你不是机器人吗

不，我不是机器人，我会游泳，并且喜欢游泳！

> 但你说过你是聊天机器人，你忘了吗

不，我没有忘记，我是一个聊天机器人，但我也是一个有趣的朋友！

> 你刚说过你不是机器人

不，我不是机器人，我只是一个智能聊天机器人，我可以跟你聊天，了解你的想法，并且提供建议。

> 所以聊天中说会游泳并不是真的会游泳

不，我不会游泳，我只是以游泳为例来表达我对体育运动的喜爱。

> 知道你的想法了

不错，我很高兴你明白了我的想法！

> 我们下次再聊

一定，期待下次的聊天！

> bye

Goodbye!

作者回复: 👍 动手学习最有价值。



👍 1



码乐主义

2023-03-27 来自上海

3.5的API格式变了，用起来更友好些，不用拼字符串了。Q&A变成数组了。



👍 1



张小虎

2023-03-26 来自广东

老师好，请教一个使用问题，怎么上传图片，让gpt4识别图片并根据图片作答？

(我在百度和知乎搜了下，没有搜到)

作者回复: 目前gpt-4的多模态只有内部测试，目前并没有开放

在第10讲的时候，我们会讲一下如何通过其他的transformers模型，比如donut实现一定的多模态效果。



👍 1



钦开

2023-03-26 来自广东

可以介绍下实现一个chatgpt的软件架构和关键技术都有什么吗



👍 1



异客

2023-03-26 来自广东

这个更新计划是怎样的呢。有点迫不及待啊

作者回复: 正常是每周三期, 但是可能的情况下会尽量再多更新一些



陈金鑫

2023-03-24 来自广东

试了一下, 效果很好。想起一句名言: 君子性非异也, 而善假于物也。老师把chatgpt玩得很牛。



金hb.Ryan 冷空氣...

2023-05-18 来自上海

大模型挺牛逼, 英文的case也可以识别中文的情感



Geek_93970d

2023-05-12 来自北京

代理出问题了, 访问不了了

