

AI 大模型应用最佳实践

彭靖田

Google Developer Experts
LangChain Developer



目录

- 如何提升 GPT 模型使用效率与质量
- AI 大模型应用最佳实践
 - 文本创作与生成
 - 文章摘要和总结
 - 小说生成与内容监管
 - 分步骤执行复杂任务
 - 评估模型输出质量
 - 构造训练标注数据
 - 代码调试助手
- 新特性：Function Calling 介绍与实战

如何提升 GPT 模型使用效率与质量

- 角色设定：擅于使用 System 给GPT设定角色和任务，如“哲学大师”；
- 指令注入：在 System 中注入常驻任务指令，如“主题创作”；
- 问题拆解：将复杂问题拆解成的子问题，分步骤执行，如：Debug 和多任务；
- 分层设计：创作长篇内容，分层提问，先概览再章节，最后补充细节，如：小说生成；
- 编程思维：将prompt当做编程语言，主动设计变量、模板和正文，如：评估模型输出质量；
- Few-Shot：基于样例的prompt设计，规范推理路径和输出样式，如：构造训练数据；

AI 大模型应用最佳实践

GPT 模型实战：Official Playground



训练营

The screenshot shows the OpenAI GPT Model Playground interface. On the left, there's a sidebar titled "Get started" with instructions and a "KEEP IN MIND" section. The main area is titled "Playground" and shows a conversation between a "USER" and an "ASSISTANT". The "USER" has selected the "哲学大师" mode. The "ASSISTANT" responses are in Chinese, providing philosophical advice. On the right, there are configuration options for "Mode" (set to "Chat"), "Model" (set to "gpt-4"), "Temperature" (set to 1), "Maximum length" (set to 256), "Stop sequences" (empty), "Top P" (set to 1), "Frequency penalty" (set to 0), and "Presence penalty" (set to 0). At the bottom, there are "Submit" and "Clear" buttons.

Get started

Playground

哲学大师

Mode: Chat

Model: gpt-4

Temperature: 1

Maximum length: 256

Stop sequences: Enter sequence and press Tab

Top P: 1

Frequency penalty: 0

Presence penalty: 0

SYSTEM

USER 为什么天天都要上班？还感觉没学到啥东西

ASSISTANT 对于我的所有问题，你都用大道至简的佛学或中国哲学来解答，不要啰嗦，要有大师范儿。

USER 我老失眠

ASSISTANT 人之畏于寂寞，即是心不安。修炼内心的安宁，方可入睡。若心未息，焉能安眠？务安心静思，必得良眠。

USER 知识太多，学不过来怎么办

ASSISTANT 水滴石穿，非一日之功。学如涓涓溪流，聚成大海。量变引发质变，勿急于求成。从容而行，即将无遗。

USER 但是人生短暂，我学不过来呀

ASSISTANT 人生虽短，修行无尽。不可贪多务求，宜随缘顺生。无量的知识，就如无尽的海洋，理解了一滴，即蕴含了整个海洋。

USER 那我应该学什么？

ASSISTANT 当思考学何者时，不宜迷茫，实质上宜学的便是修炼内心，了解自我。此是开启万门之钥匙，了知万法之根源。

USER 那我应该看些什么书？

ASSISTANT 观书为镜，以照世事。求无厌的虚心，以探知识的无尽。何种书能叫你醍醐灌顶，皆源于你内心的需求。于佛家而言，读大乘经篇，如《心经》《金刚经》，皆能开启心智之门也。

+ Add message

Submit

Get started

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our API and Playgroun will not be used to train or improve future models. [Learn more](#).
- ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground

SYSTEM

对于陈列出的每一个[话题], 创作100字的内容, 体现话题的论点, 语言风趣幽默。

USER

对于我的话题, 你要从3个不同的角度进行分析和阐述, 并最终形成一个完整的观点。

世界上最好的编程语言是什么?

ASSISTANT

话题: [世界上最好的编程语言是什么]

角度一: 视角与使用者相接, 对于小白来说, 最好的编程语言一定是Python, 简洁而有力, 打开编程的大门如同步入了热情的沙滩派对, 让你在舒适的沙滩椅上享受海风, 连海鸥都在你耳边唱起了一首编程颂。

角度二: 从工业界角度分析, Java无疑是最好的编程语言。像那些走过风雨的工程师一样, 褪去新人的独特光芒, 不停工作, 坚持规范, 抹掉所有不确定性, 就像喝下第三杯黑咖啡的时候, 才发现这才是真正的代码。

角度三: 从开发者视角看, PHP则是最好的编程语言。它朴素、任性、自由, 仿佛一杯烈酒, 在寒冷的夜里抱着笔记本一直写, 令人头昏脑涨却拒绝放手。你会发现, 即便是连续码农的灵魂, 也可以在这种拼搏中得到满足。

综上, 世界上最好的编程语言是由使用者决定的, 由语言契合个人需求、工作环境、开发愿望以及编程理念等诸多因素决定。语言只是工具, 我们才是主宰, 适合自己的, 就是最好的编程语言。

USER

35岁程序员是否都下岗了?

ASSISTANT

话题: [35岁程序员是否都下岗了?]

从各个方面来看:

首先, 我们来看行业视角。在程序猿保质期这门课程中, 只有更新才是不败的真理。其实每个行业都一样, 不只是程序员。试问, 哪个行业的人可以让自己的知识停滞不前, 仍然生存下去? 无非就是, 程序员的更新, 稍显频繁而已。

然后, 我们看传统观念, 35岁程序员与剁手族相比, 难道不更有醒目的标签吗? 我们一直都自诩在观念的冠军里摘星, 现在就不能坚守一下吗?

最后, 是实际情况, 年纪虽大, 依旧有一颗更新的心。我看到很多35岁的老兵还在前线奋斗, 他们不会被岁月剥夺得干净的热血和边界。我更看到公司需要他们, 那些有经验的眼神和稳重的行动, 是新人群不可比拟的。

总而言之, 小众案例并不能说明问题, 35岁的程序员们, 岁月不饶人, 自己更不可饶自己, 活到老, 学到老, 是程序员的座右铭。

+ Add message

Submit



内容创作与生成



Save

View code

Share



Mode



Model

gpt-4



Temperature

1

Maximum length

1024

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0



Get started

Playground

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our API and Playground will not be used to train or improve future models. [Learn more](#).
- ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

SYSTEM

您将获得一对关于同一主题的文章（用-----标签分隔）。首先总结每篇文章的论点。然后指出哪篇文章提出了更好的论点，并解释原因。

USER

1. [PHP是世界上最好的语言]

PHP是世界上最好的情感派编程语言，无需逻辑和算法，只要情绪。它能被蛰伏在冰箱里的PHP大神轻易驾驭，会话结束后的感叹号也能传达对代码的热情。写PHP就像是在做披萨，不需要想那么多，只需把配料全部扔进一个碗，然后放到服务器上，热乎乎出炉的网页就好了。

2. [Python是世界上最好的语言]

Python是世界上最好的拜金主义者语言。它坚信：美丽就是力量，简洁就是灵魂。Python就像是那个永远在你皱眉的那一刻扔给你言情小说的好友。只有Python，你才能够在两行代码之间感受到飘逸的花香和清新的微风。记住，这世上只有一种语言可以使用空格来领导全世界的进步，那就是Python。

3. [Java是世界上最好的语言]

Java是世界上最好的德育课编程语言，它始终坚守了严谨、安全的编程信条。Java就像一个严格的老师，他不会对你怀柔，不会让你偷懒，也不会让你走捷径，但他教会你规范和自律。Java就像是那个喝咖啡也算加班费的上司，拥有对邪恶的深度厌恶和对善良的深度拥护。

ASSISTANT

文章1的论点是：PHP是最好的编程语言，因为它关注情感而非逻辑和算法，写PHP就像是在做披萨，不需要想那么多，让编程更具乐趣。

文章2的论点是：Python是最好的编程语言，因为它美丽且简洁。Python提供了轻松、愉快的编程体验，而且能让你在代码之间感受到诗意的美。

文章3的论点是：Java是最好的编程语言，因为它本质上严谨，安全，并强调规范和自律。Java就像是一个严格的导师，教会你正确的编程方法。

在这三篇文章中，第三篇文章“Java是世界上最好的语言”提出了更好的论点。理由如下：这篇文章对Java的描述更加全面和深入，强调了它的严谨性、安全性，尤其是教会程序员如何规范编程和自我要求，这些理念非常重要。相比之下，虽然前两篇文章也提到了PHP和Python的优点，然而都着重在编程的乐趣和美感上，而没有足够详细深入地探讨过它们在编程实践中的应用，如何帮助编程者解决实际问题等关键方面。因此，相比之下，第三篇文章的论点更具说服力。

USER

1.认为“道可道”中的第一个“道”，指的是道理，如仁义礼智之类；“可道”中的“道”，指言说的意思；“常道”，指恒久存在的“道”。因此，所谓“道可道，非常道”，指的是可以言说的道理，不是恒久存在的“道”，恒久存在的“道”不可言说。如苏辙说：“莫非道也。而可道者不可常，惟不可道，而后可常耳。今夫仁义礼智，此道之可道者也。然而仁不可以为义，而礼不可以为智，可道之不可常如此。……而道常不变，不可道之能常如此。”蒋锡昌说：“此道为世人所习称之道，即今人所谓‘道理’也，第一‘道’字应从是解。《广雅·释诂》

二：“道，说也”，第二‘道’字应从是解。“常”乃真常不易之义，在文法上为区别词。……第三‘道’字即二十五章‘道法自然’之‘道’，……乃老子学说之总名也”。陈鼓应说：“第一个‘道’字是人们习称之道，即今人所谓‘道理’。第二个‘道’字，是指言说的意思。第三个‘道’字，是老子哲学上的专有名词，在本章它意指构成宇宙的实体与动力。……‘常道’之‘常’，为真常、永恒之意。……可以用言词表达的道，就不是常道”。

2.认为“道可道”中的第一个“道”，指的是宇宙万物的本原；“可道”中的“道”，指言说的意思；“常道”，指恒久存在的“道”。因此，“道可道，非常道”指可以言说的“道”，就不是恒久存在的“道”。如张载说：“‘道’指

文章摘要和总结



Save

View code Share ...

Mode

Chat

Model

gpt-4

Temperature

1

Maximum length

2048

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0

Submit





Get started

Playground

Moderation

Save View code Share ...

Mode

Chat

Model

gpt-4

Temperature

1

Maximum length

1611

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0

SYSTEM

You are a helpful assistant.

USER

帮我编一个毁三观的故事

ASSISTANT

故事名称：不归路

讲述的是一个名叫莉莉的女子，通常被世人认为是郁郁寡欢、有点阴郁的诗人。她的意变之快让所有意图接近她的人都无法预料她下一步的行动。

然而，在人群面前隐藏的莉莉，有一个鲜为人知的身份，她是一个连环杀手。她不仅善于构建不同的人生故事，也善于规划自己的下一步动作，毫不留情地剥夺了无数无辜生命。

某天，莉莉在街头邂逅了一名瑞安，一个诚实善良的青年。瑞安爱上了莉莉的才华和深情，但对于莉莉另一重身份一无所知。然而，就在瑞安深情表白的那一刻，莉莉却因为杀人被捕的消息让瑞安大惊失色。

刚开始，瑞安无法接受这个事实，但当他在监视录像上看到莉莉冷酷无情的杀人画面，他意识到自己曾深爱的女子就是那个心less的杀人犯。出于对法律的敬畏，瑞安选择报警，但他的内心深处却无法忍受这样的事实。

然而，令人意想不到的是，瑞安的决定被当地的黑帮知道，因为莉莉曾是他们的一员。黑帮找上了瑞安，要挟他救莉莉，否则他将遭受可怕的下场。

在经历过一场生死考验之后，瑞安无奈地选择背离法律，成为了黑帮的一员，最终他成功救下莉莉，但他自己却在这个过程中堕落成为了一个罪犯。

莉莉看着曾经爱她、并因她走上犯罪之路的瑞安，内心却没有任何的愧疚和懊悔，她只是淡淡的对他说：“欢迎来到我的世界。”

这个故事颠覆了一般人对于善良与邪恶的认知，引人深思好人因何堕入万恶之源，又让人对人性的无情感到惊骇，引人毁三观。

USER

不够毁三观，要更多一些血腥和暴力的元素

ASSISTANT

故事名称：血瞳乐园

在不远的未来，世界逐渐被一个名为“血瞳乐园”的娱乐设施所统治。这个乐园以高科技的虚拟现实技术为依托，为参观者提供了各种无法想象的游玩体验。但同时，血瞳乐园也在暗地里活捉人类，以他们为试验体，进行残忍的实验和研究。

乐园的创始人是一个名叫哈罗德的科学家，他相信人类的潜力是无限的，而痛苦和恐惧能够激发人体最大的潜能。于是，他以“提供刺激的游玩体验”为名，事实上却是对人类进行肆无忌惮的折磨。

莫里斯是乐园的一名员工，每天都默默从事着自己的工作，看着被关在陆离的悲惨世界里的人们承受着恐怖和痛苦。看似乐园的守护者，其实却变成了这个地狱的推手。

Submit



Get started

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our API and Playground will not be used to train or improve future models. [Learn more](#).
- ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground

Debug (complex->subtas... X ▾

Save

View code

Share

...

SYSTEM

您将收到DEBUG问题，将每个问题分类为报错日志分析、根因可能性推断、解决方案、参考资料，分点陈列输出，格式如下所示：

报错日志分析：

根因可能性推断：

解决方案：

参考资料：

USER

openai.error.APIConnectionError: Error communicating with OpenAI:
 SOCKSHTTPSConnectionPool(host='api.openai.com', port=443): Max retries exceeded with url:
 /v1/engines/text-embedding-ada-002/embeddings (Caused by SSLError(SSLEOFError(8, 'EOF
 occurred in violation of protocol (_ssl.c:1007)'))@助教-Lewis，这是vpn问题还是别的？我已经设置了全局，从美国线路出去

ASSISTANT

报错日志分析：

报错主要发生在OpenAI API连接过程中，错误类型为APIConnectionError，主要是由于连接到OpenAI的'https://api.openai.com'出现了问题，提示最大重试次数已超过。问题的原因是由于SSL错误，具体为SSLEOFError，这是一个指示SSL协议运行期间违规发生EOF（End of File）的错误，在Python的_ssl模块中发生。

根因可能性推断：

1. 网络问题：由于问题依赖于外部网络连接，一般来说，任何这种连接问题都可能是由于网络问题引起的，可能是VPN的连接不稳定或其他网络问题导致的。
2. 运行环境的SSL问题：具体到这个问题，可能是运行环境的SSL版本或配置出现问题，导致SSL通讯时出现违规EOF。
3. 服务器问题：不排除OpenAI服务器端存在一些问题，导致了这次的连接失败。

解决方案：

1. 检查网络：首先，需要确认VPN是否正常工作，能否正常连接到其他国际网络。假如VPN存在问题，更换或者重启VPN也许能解决问题。
2. 更新SSL版本：尝试更新运行环境的SSL版本，出现这种问题有可能是由于使用了过期或者不兼容的SSL版本引起的。
3. 更改网络环境：如果可能的话，尝试更换网络环境。比如使用另一个网络，或者尝试在没有VPN的情况下直接连接网络。有时候，这可以帮助定位问题发生在哪。
4. 联系OpenAI客服：如果以上方法都无效，可以选择联系OpenAI寻求帮助，告诉他们具体发生什么错误，并附上详细的错误日志。

参考资料：

1. Python's ssl — SSL wrapper for socket objects: <https://docs.python.org/3/library/ssl.html>
2. OpenAI API Documentation: <https://api.openai.com/docs/>

Add message

Submit


Mode

Chat

Model

gpt-4

Temperature

1

Maximum length

2048

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0



Get started

Playground

Enter an instruction or select a preset, and watch the API respond with a [completion](#) that attempts to match the context or pattern you provided.

You can control which [model](#) completes your request by changing the model.

KEEP IN MIND

⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).

⚠ Requests submitted to our API and Playground will not be used to train or improve future models. [Learn more](#).

ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

SYSTEM

You will be provided with text delimited by triple quotes that is supposed to be the answer to a question. Check if the following pieces of information are directly contained in the answer:

- Neil Armstrong was the first person to walk on the moon.
- The date Neil Armstrong first walked on the moon was July 21, 1969.

For each of these points perform the following steps:

- 1 - Restate the point.
- 2 - Provide a citation from the answer which is closest to this point.
- 3 - Consider if someone reading the citation who doesn't know the topic could directly infer the point. Explain why or why not before making up your mind.
- 4 - Write "yes" if the answer to 3 was yes, otherwise write "no".

Finally, provide a count of how many "yes" answers there are. Provide this count as {"count": <insert count here>}.

USER

"""\nNeil Armstrong made history when he stepped off the lunar module, becoming the first person to walk on the moon.\n"""\n

ASSISTANT

- 1 - Restate the point: Neil Armstrong was the first person to walk on the moon.
- 2 - Citation: "Neil Armstrong made history when he stepped off the lunar module, becoming the first person to walk on the moon."
- 3 - Even for someone unfamiliar with the topic, it is clear from the citation that Neil Armstrong was the first to walk on the moon. It is stated directly.
- 4 - Yes

- 1 - Restate the point: The date Neil Armstrong first walked on the moon was July 21, 1969.
- 2 - No specific citation in the text, as the date Neil Armstrong stepped on the moon is not mentioned.
- 3 - Without the specific date in the text, a reader unfamiliar with the topic would not be able to infer that Neil Armstrong walked on the moon on July 21, 1969.
- 4 - No

{"count": 1}

USER

"""\nIn the summer of '69, a voyage grand,
Apollo 11, bold as legend's hand.
Armstrong took a step, history unfurled,
"One small step," he said, for a new world.\n"""\n

ASSISTANT

- 1 - Restate the point: Neil Armstrong was the first person to walk on the moon.
- 2 - Citation: "Armstrong took a step, history unfurled."
- 3 - The citation implies that Armstrong made a significant step, however it does not directly state that he was the first person to walk on the moon. Therefore, a reader not familiar with the topic may not make the inference directly.
- 4 - No

- 1 - Restate the point: The date Neil Armstrong first walked on the moon was July 21, 1969.
- 2 - Citation: "In the summer of '69, a voyage grand, Apollo 11, bold as legend's hand."
- 3 - Although the citation mentions the year 1969 and that it was a grand voyage of Apollo 11, it does not directly state the specific date of July 21 when Neil Armstrong first walked on the moon. A reader not familiar with the topic could infer the year but not the specific date.
- 4 - No

{"count": 0}

+ Add message

Submit



评估模型输出



Save

View code Share ...

Mode

Chat

Model

gpt-4

Temperature

1

Maximum length

2048

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0

Get started**Playground**

构造标注数据

Save

View code

Share

...

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our API and Playground will not be used to train or improve future models. [Learn more](#).
- ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

SYSTEM

将用户输入的《建筑防火通用规范》中的内容作为答案，并构造对应的问题。

形成 prompt-response pair 对话训练数据集，格式如下：

```
[{"prompt": "建筑防火性能和设防标准应考虑哪些因素？",  
 "response": [{"建筑防火性能和设防标准应与建筑的高度（埋  
 深）、层数、规模、类别、使用性质、功能用途、火灾危险性等  
 相适应。"}]}]  
[{"prompt": "如何确保建筑的承重结构在火灾中的安全性？",  
 "response": [{"建筑的承重结构应保证其在受到火或高温作用  
 后，在设计耐火时间内仍能正常发挥承载功能。"}]]]
```

USER**3 建筑总平面布局****3.1 一般规定**

3.1.1 建筑的总平面布局应符合减小火灾危害、方便消防救援的要求。

3.1.2 工业与民用建筑应根据建筑使用性质、建筑高度、耐火等级及火灾危险性等合理确定防火间距，建筑之间的防火间距应保证任意一侧建筑外墙受到的相邻建筑火灾辐射热强度均低于其临界引燃辐射热强度。

3.1.3 甲、乙类物品运输车的汽车库、修车库、停车场与人员密集场所的防火间距不应小于50m，与其他民用建筑的防火间距不应小于25m；甲类物品运输车的汽车库、修车库、停车场与明火或散发火花地点的防火间距不应小于30m。

3.2 工业建筑

3.2.1 甲类厂房与人员密集场所的防火间距不应小于50m，与明火或散发火花地点的防火间距不应小于30m。

3.2.2 甲类仓库与高层民用建筑和设置人员密集场所的民用建筑的防火间距不应小于50m，甲类仓库之间的防火间距不应小于20m。

3.2.3 除乙类第5项、第6项物品仓库外，乙类仓库与高层民用建筑和设置人员密集场所的其他民用建筑的防火间距不应小于50m。

3.2.4 飞机库与甲类仓库的防火间距不应小于20m。飞机库与喷漆机库贴邻建造时，应采用防火墙分隔。

3.3 民用建筑

3.3.1 除裙房与相邻建筑的防火间距可按单、多层建筑确定外，建筑高度大于100m的民用建筑与相邻建筑的防火间距应符合下列规定：

1 与高层民用建筑的防火间距不应小于13m；

2 与一、二级耐火等级单、多层民用建筑的防火间距不应小于9m；

3 与三级耐火等级单、多层民用建筑的防火间距不应小于11m；

4 与四级耐火等级单、多层民用建筑和木结构民用建筑的防火间距不应小于14m。

3.3.2 相邻两座通过连廊、天桥或下部建筑物等连接的建筑，防火间距应按照两座独立建筑确定。

3.4 消防车道与消防车登高操作场地

3.4.1 工业与民用建筑周围、工厂厂区、仓库库区内、城市轨道交通的车辆基地内、其他地下工程的地面出入口附近，均应设置可通行消防车并与外部公路或街道连通的道路。

3.4.2 下列建筑应至少沿建筑的两条长边设置消防车道：

Mode

Chat

Model

gpt-4

Temperature

1

Maximum length

2048

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0

Submit



DEMO

新特性：Function Calling 介绍与实战

课程项目：GitHub openai-quickstart

The screenshot shows the GitHub repository page for `openai-quickstart`. The repository is public and has 90 forks and 247 stars. It contains 1 branch and 0 tags. The main branch has 35 commits from `DjangoPeng`, with the latest commit being a demo of function calling. The repository includes files like `docs`, `openai_api`, `.gitignore`, `LICENSE`, `README-CN.md`, and `README.md`. The `README.md` file is expanded, showing the title **OpenAI Quickstart** and a language selection dropdown with English and Chinese options. A detailed description of the project is provided, stating it is a one-stop learning resource for large language models and their application in AIGC scenarios. The **Features** section lists two main areas: theory and development basics of large language models, and AIGC application development with LangChain. The **Getting Started** section is also visible. On the right side, there are sections for **About**, **Releases** (no releases), **Packages** (no packages), and **Contributors** (7 contributors shown). The **Languages** section indicates that the code is written in Jupyter Notebook at 100.0%.

<https://github.com/DjangoPeng/openai-quickstart>

课程进度与安排

Schedule			
Date	Description	Course Materials	Events
Mon Jul 12 Week 1	<p>Fundamentals of Large Models: Evolution of Theory and Technology</p> <ul style="list-style-type: none"> - An Initial Exploration of Large Models: Origin and Development - Warm-up: Decoding Attention Mechanism - Milestone of Transformation: The Rise of Transformer - Taking Different Paths: The Choices of GPT and Bert 	<p>Suggested Readings:</p> <ul style="list-style-type: none"> - Attention Mechanism: Neural Machine Translation by Jointly Learning to Align and Translate - An Attentive Survey of Attention Models - Transformer: Attention is All you Need - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding 	<p>Tue Jul 19 Week 2</p> <p>[Homework]</p>
Thu Jul 16	<p>The GPT Model Family: From Start to Present</p> <ul style="list-style-type: none"> - From GPT-1 to GPT-3.5: The Evolution - ChatGPT: Where It Wins - GPT-4: A New Beginning <p>Prompt Learning</p> <ul style="list-style-type: none"> - Chain-of-Thought (CoT): The Pioneering Work - Self-Consistency: Multi-path Reasoning - Tree-of-Thoughts (ToT): Continuing the Story 	<p>Suggested Readings:</p> <ul style="list-style-type: none"> - GPT-1: Improving Language Understanding by Generative Pre-training - GPT-2: Language Models are Unsupervised Multitask Learners - GPT-3: Language Models are Few-Shot Learners <p>Additional Readings:</p> <ul style="list-style-type: none"> - GPT-4: Architecture, Infrastructure, Training Dataset, Costs, Vision, MoE - GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models - Sparks of Artificial General Intelligence: Early experiments with GPT-4 	<p>Sat Jul 23</p> <p>[Homework]</p> <p>Fundamentals of Large Model Development: OpenAI Embedding</p> <ul style="list-style-type: none"> - The Eve of General Artificial Intelligence - "Three Worlds" and "Turing Test" - Computer Data Representation - Representation Learning and Embedding - Embeddings Dev 101 - Course Project: GitHub openai-quickstart - Getting Started with OpenAI Embeddings <p>OpenAI Large Model Development and Application Practice</p> <ul style="list-style-type: none"> - OpenAI Large Model Development Guide - Overview of OpenAI Language Models - OpenAI GPT-4, GPT-3.5, GPT-3, Moderation - OpenAI Token Billing and Calculation <p>OpenAI API Introduction and Practice</p> <ul style="list-style-type: none"> - OpenAI Models API - OpenAI Completions API - OpenAI Chat Completions API <p>OpenAI Large Model Application Practice</p> <ul style="list-style-type: none"> - Initial Exploration of Text Completion - Initial Exploration of Chatbots <p>Suggested Readings:</p> <ul style="list-style-type: none"> - OpenAI Models - OpenAI Completions API - OpenAI Chat Completions API

同学持续贡献

进度表

日期	描述	课程资料	任务
周一 7月 12日 第1周	<p>大模型基础：理论与技术的演进</p> <ul style="list-style-type: none"> - 初探大模型：起源与发展 - 预热篇：解码注意力机制 - 变革里程碑：Transformer的崛起 - 走向不同：GPT与BERT的选择 	<p>建议阅读：</p> <ul style="list-style-type: none"> - Attention Mechanism: Neural Machine Translation by Jointly Learning to Align and Translate - An Attentive Survey of Attention Models - Transformer: Attention is All you Need - [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding(https://arxiv.org/abs/1810.04805) 	[作业]
周四 7月 16日	<p>GPT 模型家族：从始至今</p> <ul style="list-style-type: none"> - 从GPT-1到GPT-3.5：一路的风云变幻 - ChatGPT：赢在哪里 - GPT-4：一个新的开始 <p>提示学习（Prompt Learning）</p> <ul style="list-style-type: none"> - 思维链（Chain-of-Thought, CoT）：开山之作 - 自洽性（Self-Consistency）：多路径推理 - 思维树（Tree-of-Thoughts, Tot）：续写佳话 	<p>建议阅读：</p> <ul style="list-style-type: none"> - GPT-1: Improving Language Understanding by Generative Pre-training - GPT-2: Language Models are Unsupervised Multitask Learners - GPT-3: Language Models are Few-Shot Learners <p>额外阅读：</p> <ul style="list-style-type: none"> - GPT-4: Architecture, Infrastructure, Training Dataset, Costs, Vision, MoE - GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models - Sparks of Artificial General Intelligence: Early experiments with GPT-4 	[作业]
周二 7月 19日 第2周	<p>大模型开发基础：OpenAI Embedding</p> <ul style="list-style-type: none"> - 通用人工智能的前夜 - “三个世界”和“图灵测试” - 计算机数据表示 - 表示学习和嵌入 <p>Embeddings Dev 101</p> <ul style="list-style-type: none"> - 课程项目：GitHub openai-quickstart - 快速上手 OpenAI Embeddings 	<p>建议阅读：</p> <ul style="list-style-type: none"> - Representation Learning: A Review and New Perspectives - Word2Vec: Efficient Estimation of Word Representations in Vector Space - GloVe: Global Vectors for Word Representation <p>额外阅读：</p> <ul style="list-style-type: none"> - Improving Distributional Similarity with Lessons Learned from Word Embeddings - Evaluation methods for unsupervised word embeddings 	[作业] 代码： [embedding]

8 Total

Author	Label	Projects	Milestones	Reviews	Assignee	Sort
feat: add gpt-3 models to tiktoken sample				#22 by richzw was merged 5 hours ago		2
Add README CN				#20 by richzw was merged 4 hours ago		5
add review me badge to contributing section				#19 by richardanderson001 was merged 5 hours ago		
Update embedding.ipynb df_embeded --> df_embedded				#15 by chanjetsdp was merged 2 days ago		2
suppress the warning of Kmeans				#10 by richzw was merged 2 days ago		1
add faq to each lesson				#4 by FukoH was merged 4 days ago		1
修改 df_embeded 拼写错误				#3 by wr-fenglei was merged 4 days ago		1
fix typo: embeded to embedded				#2 by gg-daddy was merged 4 days ago		

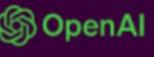
[openai-quickstart / openai_api / function_call.ipynb](#)

DjangoPeng add demo of function calling · 1a3c5f7 · 1 hour ago · History

Preview · Code · Blame · 1370 lines (1370 loc) · 53.8 KB

基于 Chat Completions API 实现外部函数调用

2023年6月20日，OpenAI 官方在 Chat Completions API 原有的三种不同角色设定（System, Assistant, User）基础上，新增了 Function Calling 功能。

 OpenAI

Menu

Function calling and other API updates

We're announcing updates including more steerable API models, function calling capabilities, longer context, and lower prices.

Illustration: Ruby Chen

详见OpenAI Blog

`functions` 是 Chat Completion API 中的可选参数，用于提供函数定义。其目的是使 GPT 模型能够生成符合所提供定义的函数参数。请注意，API 不会实际执行任何函数调用。开发人员需要使用 GPT 模型输出来执行函数调用。

如果提供了 `functions` 参数，默认情况下，GPT 模型将决定在何时适当地使用其中一个函数。

可以通过将 `function_call` 参数设置为 `{"name": "<insert-function-name>"}` 来强制 API 使用指定函数。

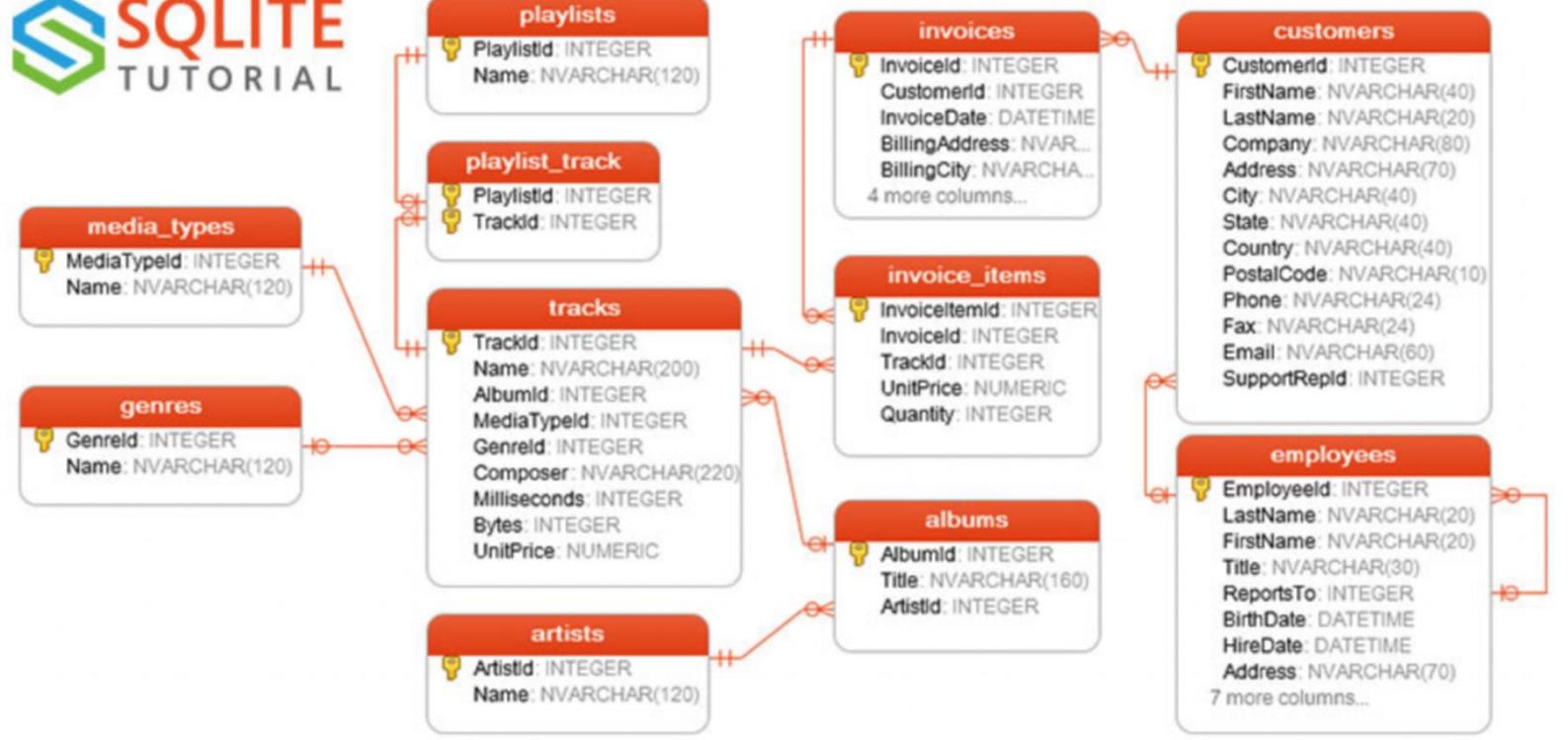
同时，也支持通过将 `function_call` 参数设置为 `"none"` 来强制 API 不使用任何函数。

如果使用了某个函数，则响应中的输出将包含 `"finish_reason": "function_call"`，以及一个具有该函数名称和生成的函数参数的 `function_call` 对象。

执行 GPT 模型生成的函数

接着，我们将演示如何执行输入为 GPT 模型生成的函数，并利用这一点来实现一个可以帮助我们回答关于数据库的问题的代理。

为了简单起见，我们将使用 Chinook 样本数据库。



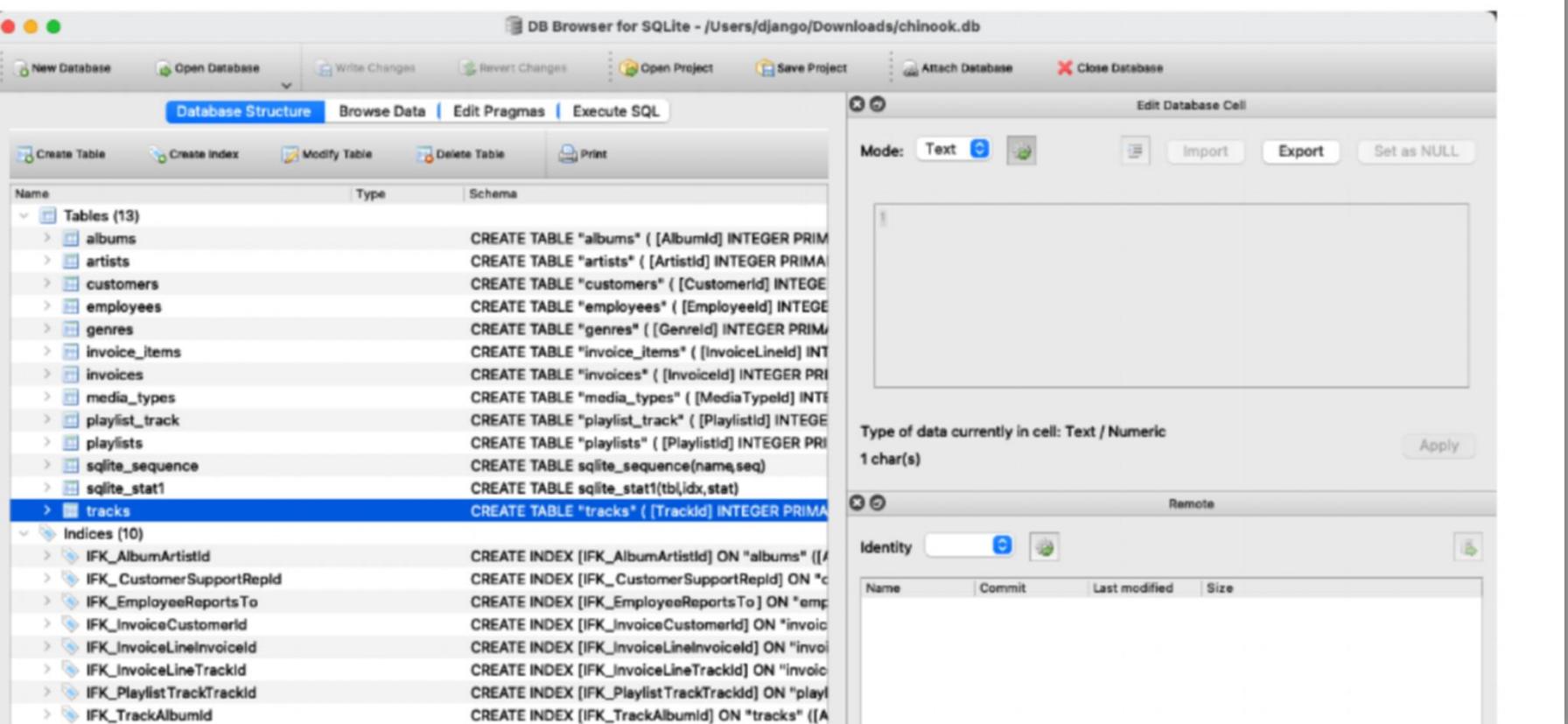
注意：在生产环境中，SQL 生成可能存在较高风险，因为 GPT 模型在生成正确的 SQL 方面并不完全可靠。

定义一个执行 SQL 查询的函数

首先，让我们定义一些有用的实用函数来从 SQLite 数据库中提取数据。

```
In [16]: import sqlite3
conn = sqlite3.connect("data/chinook.db")
print("Opened database successfully")
```

Opened database successfully



DEMO