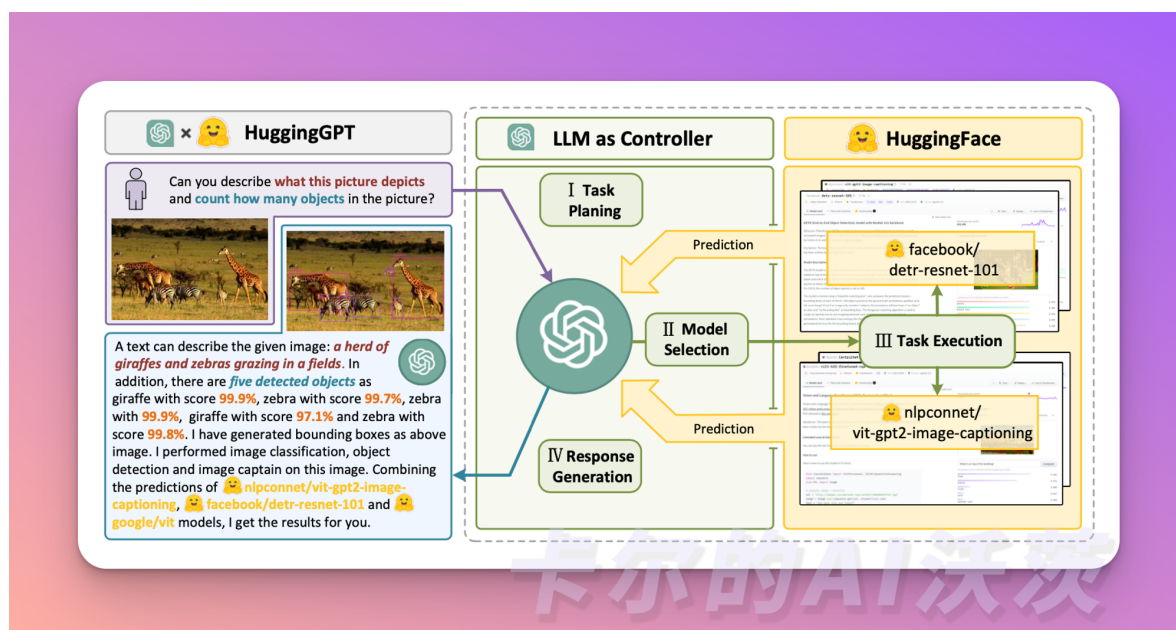


HuggingGPT

HuggingGPT是一个 Agent 框架，利用 ChatGPT 作为任务规划器，根据每个模型的描述来选择 HuggingFace 平台上可用的模型，最后根据模型的执行结果生成总结性的响应。这个项目目前已在 Github 上开源，并且有一个非常酷的名字叫做 JARVIS（钢铁侠的助手）。这项研究主要涉及到两个主体，一个是众所周知的 ChatGPT，另一个是 AI 社区中的 Hugging Face。



这个系统包含四个阶段：

1. **任务规划**：使用LLM作为大脑，将用户的请求解析为多个任务。每个任务都有任务类型、ID、依赖关系和参数四个属性。系统会使用一些示例来指导LLM进行任务解析和规划。

具体指令如下：

```
1 [{"task": task, "id": task_id, "dep": dependency_task_ids, "args": {"text": text, "image": URL, "audio": URL, "video": URL}}]
```

- "dep"字段表示前一个任务的ID，该任务生成了当前任务所依赖的新资源。
- "-task_id"字段指的是具有任务ID为task_id的依赖任务中生成的文本图像、音频和视频。

用户和HuggingGPT之间的聊天日志被记录下来，并显示在显示资源历史记录屏幕上。

2. **模型选择**：LLM将任务分配给专门的模型，这些请求被构建成为了一道多项选择题。

LLM为用户提供了一个模型列表供选择。由于上下文长度的限制，需要根据任务类型进行过滤

具体指令如下

根据用户请求和调用命令，Agent 帮助用户从模型列表中选择一个合适的模型来处理用户请求。Agent 仅输出最合适模型的模型ID。输出必须采用严格的JSON格式：{"id": "模型ID", "reason": "您选择该模型的详细原因"}。

之后，HuggingGPT根据下载次数对模型进行排名，因为下载次数被认为是反映模型质量的可靠指标。选择的模型是根据这个排名中的“Top-K”模型来进行的。K在这里只是一个表示模型数量的常数，例如，如果设置为3，那么它将选择下载次数最多的3个模型。

3.任务执行：专家模型在特定任务上执行并记录结果。

具体指令如下：

根据输入和推理结果，Agent 需要描述过程和结果。前面的阶段可以形成下方的输入

用户输入：{{用户输入}}，任务规划：{{任务}}，模型选择：{{模型分配}}，任务执行：{{预测结果}}。

为了提高此过程的效率，HuggingGPT 可以同时运行不同的模型，只要它们不需要相同的资源。例如，如果我提示生成猫和狗的图片，那么单独的模型可以并行运行来执行此任务。但是，有时模型可能需要相同的资源，这就是为什么HuggingGPT维护一个属性来跟踪资源的原因。它确保资源得到有效利用。

4.响应生成：LLM 接收执行结果，并向用户提供总结结果。




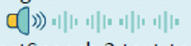

然而，要将HuggingGPT应用于实际场景中，我们需要应对一些挑战：

- 提高效率：因为LLM的推理轮次和与其他模型的交互都会减缓处理速度
- 依赖长上下文窗口：LLM需要使用长篇的上下文信息来传递复杂的任务内容
- 提高稳定性：需要改进LLM的输出质量以及外部模型服务的稳定性。

现在，让我们假设您希望模型根据图像生成一个音频。HuggingGPT会以最适合的方式连续执行这个任务。您可以在下面的图中查看更详细的响应信息

Query: read the image example.jpg for me.

INVOICE NO : 18301/102/T0305

Response: Sure. I have read the image  example.jpg for you. The inference result is 'INVOICE NO : 18301/102/T0305'. The model used for this task is , which is best suited for the task of image-to-text as it is fine-tuned on the SROIE dataset which is specifically designed for this task. Then I have converted the text to audio, the generated audio is  /audios/da5s.wav  and the model used for this task is , which is a FastSpeech 2 text-to-speech model which is suitable for the task of text-to-speech. Is there anything else I can help you with?

Task Planning:

1: {"task": "image-to-text", "id": 0, "dep": [-1], "args": {"image": "example.jpg"}}
2: {"task": "text-to-speech", "id": 1, "dep": [0], "args": {"text": "<resource-1>"}}

Execution Results:

1. {"generated text": "INVOICE NO : 18301/102/T0305"} 2. {"generated audio": "/audios/da5s.wav"}

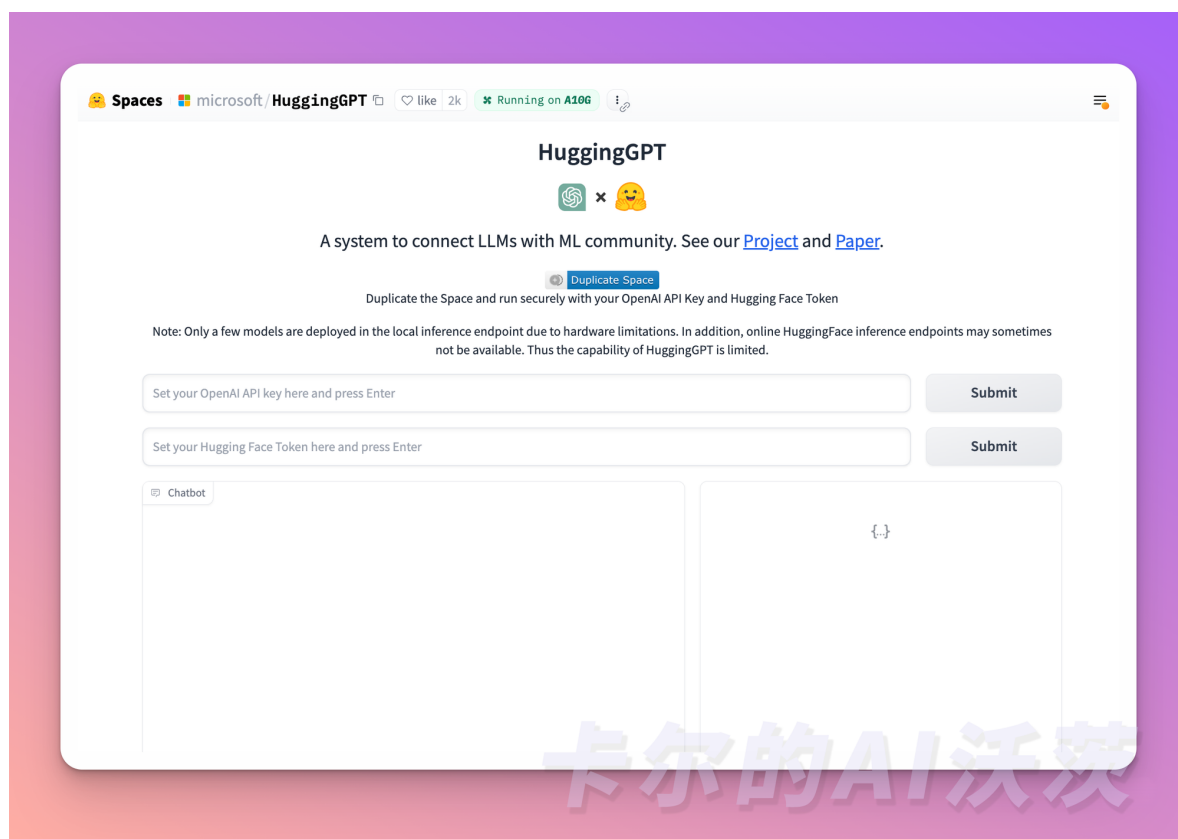
Hugging Face是什么？

简单来说，Hugging Face是一个专注于人工智能的开源社区平台，用户可以在该平台上发布和共享预训练模型、数据集和展示文件等。目前在Hugging Face上已经共享了超过10万个预训练模型和1万多个数据集。包括微软、谷歌、彭博社、英特尔等众多行业的1万家机构都在使用Hugging Face的产品。

在HuggingGPT中，ChatGPT充当了”操作大脑”的角色，能够自动解析用户提出的需求，并在Hugging Face的AI模型库中进行自动模型选择、执行和报告，为我们开发更复杂的人工智能程序提供了极大的便利。

快速体验

体验HuggingGPT非常简单，只需要输入openai apikey和HuggingGPT token即可：



<https://huggingface.co/spaces/microsoft/HuggingGPT>

下一节我们学习 MetaGPT，用SOP的思路让Agent更上一步

Reference

- [HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face](#)
- [HuggingGPT: The Secret weapon to Solve Complex AI Tasks](#)