

LLM App Collection

Jiayi Yao

8.

Contents

- Openai API call
 - “openai.ChatCompletion.create”
- Ultimate goal
 - prompt reusing
- App collection
 - repos with “openai.ChatCompletion.create”
- Prompt collection
 - Find reusable prompt templates
- What you guys can do

Openai API call

- `openai.ChatCompletion.create`
- Useful resources
 - [official documentation](#)
 - [playground](#), not chatgpt
 - [prompt engineering guide](#)

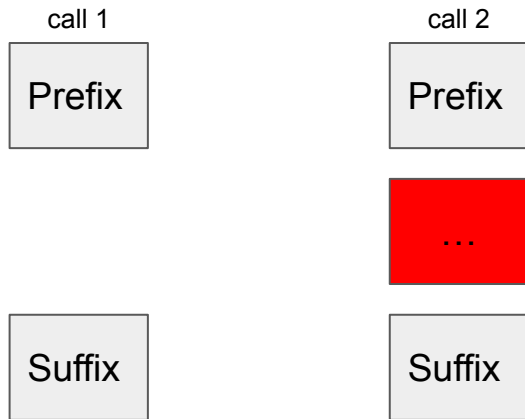
```
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")

response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {
            "role": "user",
            "content": ""
        }
    ],
    temperature=1,
    max_tokens=256,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
)
```

Ultimate Goal: Prompt Reusing

- Naive example:
 - call 1: Hello, GPT! My name is Jiayi.
 - call 2: Hello **Translation** GPT! My name is Jiayi.
 - only “values” corresponding to suffix are reusable



Ultimate Goal: Prompt Reusing

- [Real world example](#)

App Collection

- Goal
 - collect github repos that contain “[openai.ChatCompletion.create](#)”
- Tool
 - [Github RestAPI](#) ✓
 - Scraper ✗
- Problem
 - The RestAPI only returns the top 1000 results
- Method
 - `path` restriction ✓
 - other keyword restrictions ✗
- [Current implementation](#)

App Collection - Filtering

- Goal
 - filter out “uninteresting” repos
 - “uninteresting”: repos that do not use any prompt template
 - [tutorials](#)
 - [vector database](#)
- Method
 - same as prompt collection

Prompt Collection

- Goal
 - Extract prompt template
- Methods
 - static analysis
 - instrumentation + dynamic
- Current implementation

What you guys can do...

- Improve app collection
 - as many as possible
- Improve prompt collection
 - as accurate as possible
- Reimplement 0-GPT on other open-source llms
 - current implementation: [llama-2](#) (most popular)
 - other open-source llms: [opt](#), [gpt-neox](#)*