

Let's build an AI-powered API!



April 2022
Tina Zou | Uli Hitzel

Vestiaire
Collective





Help



Playground

Load a preset...

Save

View code

Share

...



I'm giving a talk at a conference. I came up with a title for the session, it is: "Live Hack: Create an AI-based API in 30 Minutes"

The title is not perfect, especially "AI-based API" sounds odd. Here are ten suggestions on how that term can be replaced with something better:

- 1: "Create an API with AI-powered capabilities in 30 minutes"
- 2: "Build an AI-powered API in 30 minutes"
- 3: "Create an API with AI features in 30 minutes"
- 4: "Build an API with AI functionality in 30 minutes"
- 5: "Create an API powered by AI in 30 minutes"

Submit



215

Aspiring Product Manager



NUS FinTech



/in/miaoqinqin

V.

VP Engineering



NUS Fellow



/in/uhitzel



/u1i



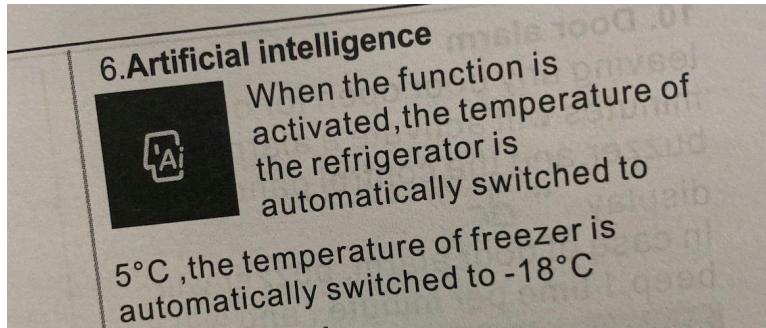
Agenda

- 1. GPT-3 – Artificial Intelligence that also non-tech people can use!**
- 2. Use Case: Content Moderation → Sentiment Analysis**
- 3. Training the Model & Creating and Consuming an API**



Artificial Intelligence

- Marketing Term
- AI vs Machine Learning
- The moment we understand how it works, we no longer think of it as “AI”



PICTURES OF THE FLOATING OF THE WORLD

An Art Collaboration Between Human and AI

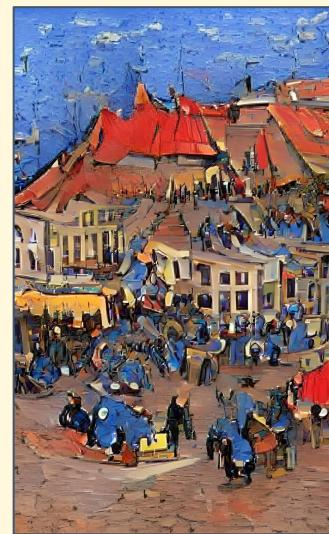
<https://apigeek.net/art>



Claude Monet (b. 1840)
The Lake, 2021



Edward Hopper (b. 1936)
The Storm, 2021



Vincent van Gogh (b. 1853)
The Market, 2021



OpenAI: GPT-3

- “Generative Pretrained Transformer” version 3
- Neural network, leverages deep learning to generate human-like text -> it ‘just’ predicts text
- OpenAI research company: Elon Musk, Reid Hoffman, Sam Altman
- 175 billion trainable parameters, it ‘read’ 45TB of books, Wikipedia, and the internet

[Overview](#)[Documentation](#)[Examples](#)[Playground](#)[Save](#)[View code](#)[Share](#)

...



Prompt

Explain the concept of artificial intelligence and machine learning to a non-technical person in a few sentences.

Artificial intelligence and machine learning are the study and practice of making machines and computer programs that can perform tasks that normally require human intelligence. Machine learning is a subset of artificial intelligence in which computer programs are trained to make predictions from data without being explicitly programmed.

What are the benefits and concerns of artificial intelligence and machine learning?

The benefits of artificial intelligence and machine learning are that they can perform tasks more efficiently and accurately than humans. The downsides are that they may be biased and not able to make decisions that a human could make.

[Generate](#)

133

Potential Use Cases

- **Fraud Detection & Anti-Money Laundering:** identifying suspicious behavior based on historical data
- **Sentiment Analysis:** understanding customer sentiment and moderate content accordingly
- **Generate Insights from Unstructured Data:** identify trends and customer preferences



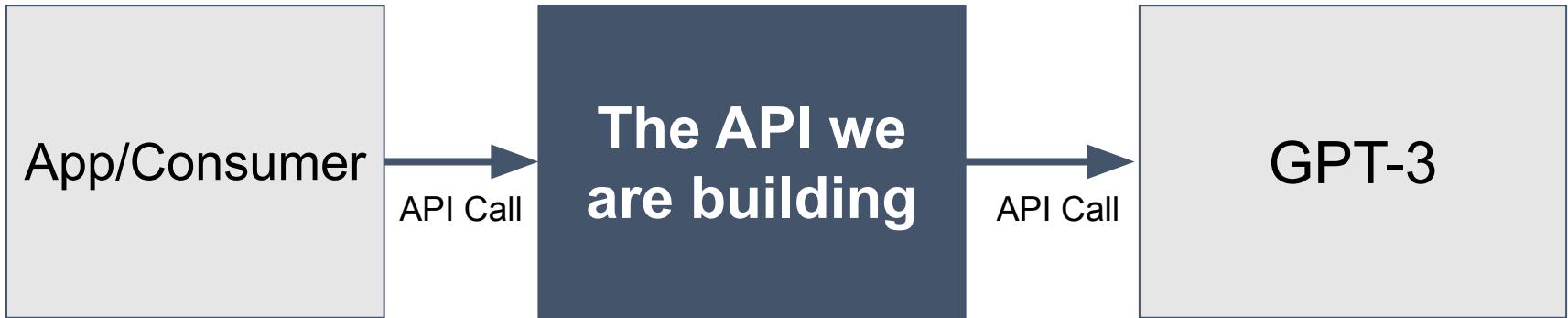
Content Moderation



*“Thanks for shopping
with us! Please write a
quick review about your
experience with us”*

Our API says your customer is:	Based on this, you may decide to:
:	→ publish review immediately, promote for other shoppers to see
:	→ trigger a customer service agent to get in touch & try to resolve
:	→ block the review, send notice
/ :	→ acknowledge & derive insights how to improve your service quality!

Let's Make an {"API"} !



POST https://api.mybackend.com/v1/sentiment_analysis

Body: e.g. "I love the product a lot, thanks for the great service!!"

→ **Response: positive, negative, neutral, upset**

Our Steps

1. Train the Model →
OpenAI Playground
2. Generate Python /
JavaScript Code
3. Add API wrapper

1010011010010001010100111010101100101001101010010011011011001
111001010110100110100100010101001110101011001010011010100110101
0011000010111001010110100110100100010101001110101011001010011001
001101110100101001101001000101010011101010110010100110101011001
0100110000101110010101101001101000101010011101010110010100110101
010011011010011000010111001010110100110100010001010100101010011
1001101010010011011101010100110100100010101001110101011001010011
001001101110100110000 101110010 01011
01000101010011101 01011
10010011011101 01011
00101011010 0110100 010
11101010 11 00101001101 010
110101 0100 11010010001 01010
01011 0010100 1101010 0100110
0011 000010111 11010 00101010110110101011010101101010110101011010
1010010001010100111010101100101001101010010011011101001101101001
0101101001101001000101010011101010110010100110101001001101001001
11010010001010100111010101100101001101010010011011101001101101001
10101101001101001000101010011101010110010100110101001101010100100
0001011100101011010011010010001010100111010101100101001100101001001
1110101010011010010001010100111010101100101001101010010011010100100
0000101110010101101001101001000101010011101010110010100110010100100
0111010011000010111001010110100110010001010100111010101101010011010
1010010011011101001010011010010001010100100111010101100101001100101
110111010011000010111001010110100110010001010100111010101001110101
10101001001101110100110000101110010101101001100100010101001110101001
11001010011010100100110111010101001101001000101010011101010100111010
1101010010011011101001100001011100101011010011001000101010011001001001
01100101001101010010011011101001100001011100101011010011001001101001001

Analyzing sentiment from user input. The possible results are:
happy, neutral or upset:



Text: the product is terrible I want my money back!!!

Sentiment: upset

Text: loved it, will always buy again thanks!

Sentiment: happy

Text: everything ok thank you.

Sentiment: neutral

Text:

```
1 import os
2 import openai
3 import json
4 from flask import Flask, request
5
6 openai.api_key = os.getenv("OPENAI_API_KEY")
7
8 def analyze_sentiment(input_text):
9
10     response = openai.Completion.create(
11         engine="text-davinci-002",
12         prompt="analyzing sentiment from user input. The possible results are: positive,negative,neutral,upset\n\n"
13         "text:the product is terrible I want my money back!!!\nsentiment:upset\n\n"
14         "text:loved it, will always buy again\n\n"
15         "thanks!\nsentiment:positive\n\n"
16         "text:everything ok thank you\n\n"
17         "sentiment:neutral\n\n"
18         "text:wasn't very happy about the delay but overall ok\n\n"
19         "sentiment:negative\n\n"
20         "text:" + input_text + "\nsentiment:",
21         temperature=0.7,
22         max_tokens=256,
23         top_p=1,
24         frequency_penalty=0,
25         presence_penalty=0
26     )
27
28     return(response["choices"][0]["text"])
29
30 app = Flask(__name__)
31 @app.route('/sentiment', methods=['POST'])
32 def apirequest():
33
34     if not request.json or not 'text' in request.json:
35         print(request.json)
36         abort(400)
37     sentiment=analyze_sentiment(request.json['text'])
38     return json.dumps({'sentiment': sentiment})
39
40 app.run(port=8080)
```

Python Code



Consuming the API – Postman

The screenshot shows the Postman application interface. At the top, it displays the URL `localhost:8080/sentiment`. Below the URL, there's a dropdown menu set to `POST` and another dropdown showing the same URL. To the right of these are `Save`, `Edit`, and `Send` buttons. The main navigation bar includes `Params`, `Authorization`, `Headers (9)`, `Body` (which is currently selected), `Pre-request Script`, `Tests`, and `Settings`. Below the navigation bar, there are several radio buttons for selecting the body type: `none`, `form-data`, `x-www-form-urlencoded`, `raw`, `binary`, `GraphQL`, and `JSON` (which is selected). A `Beautify` button is also present. The `Body` section contains a code editor with the following JSON payload:

```
1 {"text": "What an amazing session! I am very happy to be at API Days :)"}]
```

Below the body section, the response status is shown as `200 OK` with a duration of `2.22 s` and a size of `179 B`. There are `Save Response` and `Copy` buttons. The bottom of the interface shows tabs for `Body`, `Cookies`, `Headers (4)`, and `Test Results`, with `Body` being the active tab. Below these tabs are buttons for `Pretty`, `Raw`, `Preview`, `Visualize`, and `HTML`. The response body under the `Body` tab is:

```
1 {"sentiment": "positive"}
```



`curl -X POST -H
"Content-Type:application/json" -d '{"text":"I
love the product"}' localhost:8080/sentiment`

Consuming the API – Frontend

The screenshot shows a web application interface. On the left, a terminal window displays a text file with sentiment analysis results:

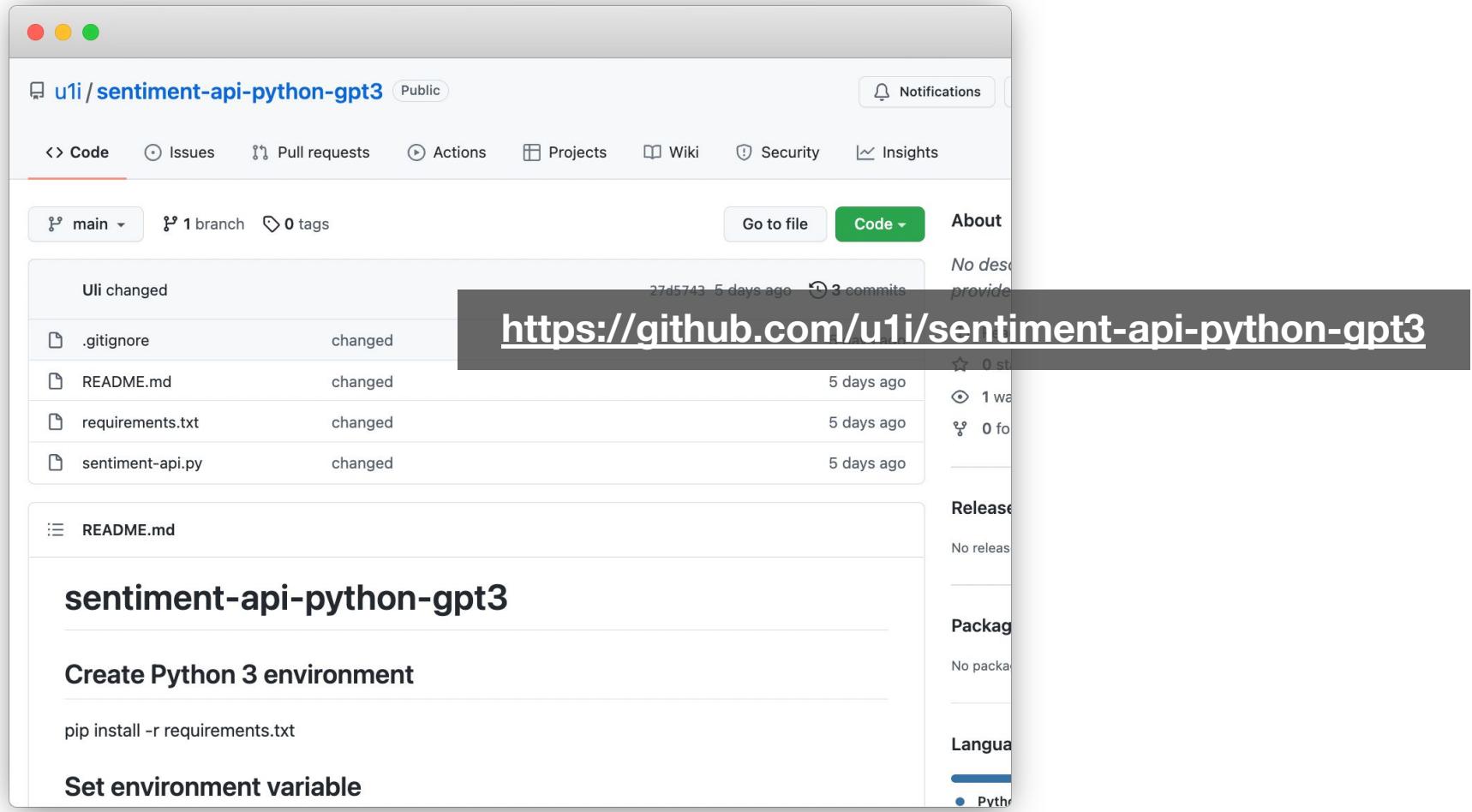
```
neutral###  
###  
neutral###  
The necklace is beautiful, I ordered a send one for my sister right away.  
###  
happy###  
###  
happy###  
[]
```

Below the terminal is a code editor showing the same text, with "Prettier" and "Submit" buttons at the bottom.

The main content area features a "Thank you for shopping with us!" message with a background image of a gift box. It includes a text input field containing the sentence from the terminal, followed by a "Submit" button.

A separate modal window on the right displays the response: "I'm glad you enjoying shopping with us!" above an illustration of a person riding a horse.

<https://github.com/tinazou-github/sentiment-api-js-gpt3>

A screenshot of a GitHub repository page for "sentiment-api-python-gpt3". The repository is public and has 1 branch and 0 tags. The code tab is selected. A modal window is open, displaying the URL <https://github.com/u1i/sentiment-api-python-gpt3>. The repository contains files: .gitignore, README.md, requirements.txt, and sentiment-api.py, all last changed 5 days ago. The README.md file shows the project title "sentiment-api-python-gpt3" and instructions to "Create Python 3 environment" with the command "pip install -r requirements.txt". The repository has 0 stars, 1 watch, 0 forks, and no releases.

u1i / **sentiment-api-python-gpt3** Public

Code Issues Pull requests Actions Projects Wiki Security Insights

main · 1 branch · 0 tags

Uli changed

27d5743 · 5 days ago · 3 commits

<https://github.com/u1i/sentiment-api-python-gpt3>

.gitignore changed

README.md changed

requirements.txt changed

sentiment-api.py changed

README.md

sentiment-api-python-gpt3

Create Python 3 environment

```
pip install -r requirements.txt
```

Set environment variable

No description or README.

0 stars · 1 watch · 0 forks

Releases

No releases

Packages

No packages

Languages

Python

Your Turn!

- Try out GPT-3:
<https://openai.com/api/>
- Enjoy some AI art:
<https://apigeek.net/art/>
- Let's connect on LinkedIn!

