

Customer Support Website Using OpenAi API

Machine Learning
By Gangzhaorige Li





Table of Contents

1. Introduction
2. Design
3. Implementation
4. Test
5. Conclusion
6. Enhancement



Introduction

What is the project about?

The objective of this project is to leverage widely-used OpenAI API models in the development of a customer support website. The website's primary function will be to respond to customer inquiries regarding their website.



Text embedding

'text-embedding-ada-002'

OpenAi's text embedding measure the relatedness of text string

1. Search organizes results by relevance to a query.
2. Clustering groups similar text strings.
3. Recommendations suggest items with related text.
4. Anomaly detection finds outliers with little similarity.
5. Diversity measurement analyzes similarity distributions.
6. Classification categorizes text by its most similar label



Open API Model

``gpt-3.5-turbo-instruct``

GPT-3.5-turbo-instruct marks a significant advancement in enhancing user interactions with its models.

It was specifically trained to overcome issues present in earlier models, providing clearer and more precise answers.

This versatility makes it well-suited for a wide range of applications, catering to both tech-savvy users and those less familiar with technology.



Design Structure of the project.


1. Backend Framework: Flask and Node.js
 - a. For handling server side logic.
2. Integrated API: Open AI
3. Call python scripts from JavaScript.
 - a. To reduce the work of rewriting same code from one language to another.



What would we be doing?

1. Choose a website to crawl and save the data as csv file.
2. Create and apply embedding using **“text-embedding-ada-002”** from the crawled csv file.
3. Answer questions using **“gpt-3.5-turbo-instruct”** model.
4. Display the result on the UI.

Implementation: Flask



```

✓ flask
  > processed
  > static
  > templates
  > text
  ⚙ .env
  🔒 .gitignore
  📄 app.py
  📄 crawl.py
  📄 embed.py
  📄 init_api.py
  ⓘ README.md
  ≡ requirements.txt

```

Project Structure:

1. Processed: Contains csv files for embedding and crawling.
2. Text: Contains raw text that was crawled.
3. App.py: Main application.
4. Crawl.py: crawls a specific website.
5. Embed.py: create embedding and save it as csv file.
6. Requirements.txt: libraries that is needed to to run the project.



Requirements for projects. Flask

1. Make sure to have python installed.
2. To avoid unnecessary libraries installed on root of the system.
 - a. Recommended to have a virtual environment setup.
3. Getting an Open API Key



Setup the project.

1. Activate your virtual environment
 - a. command: `'workon your_environment_name'`
2. Install required packages
 - a. command: `'pip3 install -r requirements.txt'`
3. Use `'pip3 list'` to see installed packages on your environment. Note: the right image only contains portion of the installed packages.
4. Use `'flask run'` to start the project.
5. Setup your your Open API Key in .env file.

```
FLASK_APP=app
FLASK_ENV=development
OPENAI_API_KEY=your_api_key
```

Package	Version
annotated-types	0.6.0
anyio	4.2.0
beautifulsoup4	4.12.3
blinker	1.7.0
bs4	0.0.2
certifi	2023.11.17
charset-normalizer	3.3.2
click	8.1.7
contourpy	1.2.0
cycler	0.12.1
distro	1.9.0
Flask	3.0.0
fonttools	4.47.2
h11	0.14.0
httpcore	1.0.2
httpx	0.26.0
idna	3.6
itsdangerous	2.1.2
Jinja2	3.1.3



Crawling data from website.

Inside crawl.py specify the domain and full_url of the website to crawl.

```
domain = "about.fb.com"  
full_url = "https://about.fb.com/news/2021/10/facebook-company-is-now-meta/"
```

Use the following command to start crawling: 'crawl.py'

```
🍏 > ~/C/generative_ai/flask > on 🐱 main !1 python3 crawl.py  
https://about.fb.com/news/2021/10/facebook-company-is-now-meta/  
https://about.fb.com/news/2024/01/investing-in-privacy  
https://about.fb.com/news/category/data-and-privacy/  
https://about.fb.com/news/2023/05/integrity-and-transparency-reports-q1-2023  
https://about.fb.com/news/category/integrity-security/  
https://about.fb.com/news/2023/11/how-meta-is-planning-for-elections-in-2024  
https://about.fb.com/news/tag/coordinated-inauthentic-behavior  
https://about.fb.com/news/2022/04/metasp-adversarial-threat-report-q1-2022
```



Creating Embedding

Use `'python3 embed.py'` to create embedding. `embedding.csv` should be created inside processed folder.

```
Apple > ~/C/generative_ai/flask > on main !1 python3 embed.py  
Creating embedding.  
Done.
```



Requirements for projects. NodeJs

1. Make sure to have nodejs installed.
2. Make sure the python environment was setup correctly.
 - a. Since we will be calling python scripts from nodeJs



Calling python scripts. (Python script)

```
# python_launched_from_nodejs.py
question = sys.argv[1]
print(answer_question(df, question=question))
sys.stdout.flush()
```

1. Get the question from the arguments.
2. Use OpenAI Model to answer the question.
3. Print the result.



Calling python scripts. (Javascript)

1. Using `child_process` module to execute the `answer.py` and pass the question as argument.
2. Make sure to call this on the server side. Since `child_process` does not support browser.

```
var spawn = require("child_process").spawn;
var process = spawn('python', ["answer.py", question]);

return new Promise((resolve) => {
  process.stdout.on('data', (data) => {
    res.status(200).json({ result: data.toString('utf8') });
  });
  process.stderr.on("data", (x) => {
    process.stderr.write(x.toString());
  });
  process.on("exit", (code) => {
    resolve(code);
  });
});
```



Running Flask App

To start the project 'flask run'

```
Apple > ~/C/generative_ai/flask > on main !2 flask run
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production
SGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [27/Jan/2024 14:09:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [27/Jan/2024 14:09:32] "GET /static/main.css HTTP/1.1" 200 -
127.0.0.1 - - [27/Jan/2024 14:09:32] "GET /static/dog.png HTTP/1.1" 200 -
```



Customer Support

About Meta

Enter your question

Generate response



Running NodeJs App

To start the project 'npm run dev'

```
Apple > ~/C/generative_ai/node_js > on main !1 npm run dev  
  
> openai-quickstart-node@0.1.0 dev  
> next dev  
  
ready - started server on 0.0.0.0:3000, url: http://localhost:3000  
info - Loaded env from /Users/ganzorig/Code/generative_ai/node_js/.env  
event - compiled client and server successfully in 164 ms (150 modules)
```



Customer Support

About Meta

Enter your question

Generate response



Testing

Provide a question inside the input box.



Customer Support

About Meta

Who is the CEO of Meta?

Generate response

Mark Zuckerberg



Customer Support

About Meta

What is the article about?

Generate response

The article is about the company Meta and its focus on building the metaverse through immersive experiences like augmented and virtual reality. It also mentions the company's annual Connect conference, new tools for building in the metaverse, and changes in the company's financial reporting.



Customer Support

About Meta

How many apples are there?

Generate response

I don't know.



Conclusion

Complete project can be found below:

<https://github.com/gangzhaorige/ML-OPENAi-CustomerSupport>

Accuracy:

1. The accuracy of the answers relies significantly on the questions I formulated using both my personal data and information gathered from the website.
2. I believe that the precision is primarily influenced by the quality of the data supplied by the website.

Cost:

1. To mitigate excessive charges, we can minimize data usage during the development phase. (Ex: Crawled only a single page)
2. Utilize actual data only when the entire project progresses as planned. In this particular project, the embedding text can be executed just once to economize.

Final Cost

The cost: GPT 3.5 Turbo +
Embedding Model

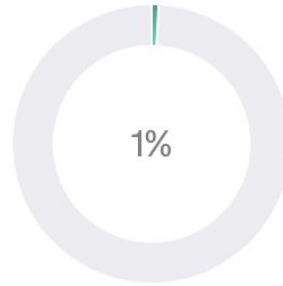
The cost includes multiple
testing on different pages.
Mostly single page.

Embedding cost is far lower
than the GPT 3.5 Turbo Model.

0.19\$ on GPT 3.5 Turbo

0.01\$ on Embedding

Monthly Bill Jan 1 - 31



\$0.20

/ \$25.00 limit

[Increase limit](#)

Credit Grants USD



\$0.20 / \$25.00

AVAILABLE
FROM

STATE

BALANCE

EXPIRATION
DATE



References:

General information about OpenAi embedding:

<https://platform.openai.com/docs/guides/embeddings/what-are-embeddings>

How to call python script from Nodejs:

<https://www.geeksforgeeks.org/run-python-script-node-js-using-child-process-spawn-method/>

Web QA Embedding:

<https://platform.openai.com/docs/tutorials/web-qa-embeddings>