

[Home](#) [AI](#) [How to Train an AI Chatbot With Custom Knowledge Base Using ChatGPT API](#)

# How to Train an AI Chatbot With Custom Knowledge Base Using ChatGPT API



Arjun Sha - Last Updated: March 14, 2023 4:45 pm



In our earlier article, we demonstrated how to [build an AI chatbot with the ChatGPT API](#) and assign a role to personalize it. But what if you want to train the AI on your own data? For example, you may have a book, financial data, or a large set of databases, and you wish to search them with ease. In this article, we bring you an easy-to-follow tutorial on how to train an AI chatbot with your custom knowledge base with LangChain and ChatGPT API. We are deploying LangChain, GPT Index, and other powerful libraries to train the AI chatbot using OpenAI's Large Language Model (LLM). So on that note, let's check out how to train and create an AI Chatbot using your own dataset.

## Train an AI Chatbot With Custom Knowledge Base Using ChatGPT API, LangChain, and GPT Index (2023)

In this article, we have explained the steps to teach the AI chatbot with your own data in greater detail. From setting up tools and software to training the AI model, we have included all the instructions in an easy-to-understand language. It is highly recommended to follow the instructions from top to down without skipping any part.

[Table of Contents](#) ▼

### Notable Points Before You Train AI with Your Own Data

1. You can train the AI chatbot on any platform, whether **Windows, macOS, Linux, or ChromeOS**. In this article, I'm using Windows 11, but the steps are nearly identical for other platforms.

## YOU MIGHT LIKE

### NEWS

**Vodafone Idea Launches AI-Powered Chatbot on WhatsApp**

### NEWS

**Indian Railways Will Now Use AI Chatbots To Answer Your Queries**



easily train and create a Q&A AI chatbot in a few minutes. If you followed our previous ChatGPT bot article, it would be even easier to understand the process.

3. Since we are going to train an AI Chatbot based on our own data, it's recommended to **use a capable computer** with a good CPU and GPU. However, you can use any low-end computer for testing purposes, and it will work without any issues. I used a Chromebook to train the AI model using a book with 100 pages (~100MB). However, if you want to train a large set of data running into thousands of pages, it's strongly recommended to use a powerful computer.

4. Finally, the data set **should be in English** to get the best results, but according to OpenAI, it will also work with popular international languages like French, Spanish, German, etc. So go ahead and give it a try in your own language.

## Set Up the Software Environment to Train an AI Chatbot

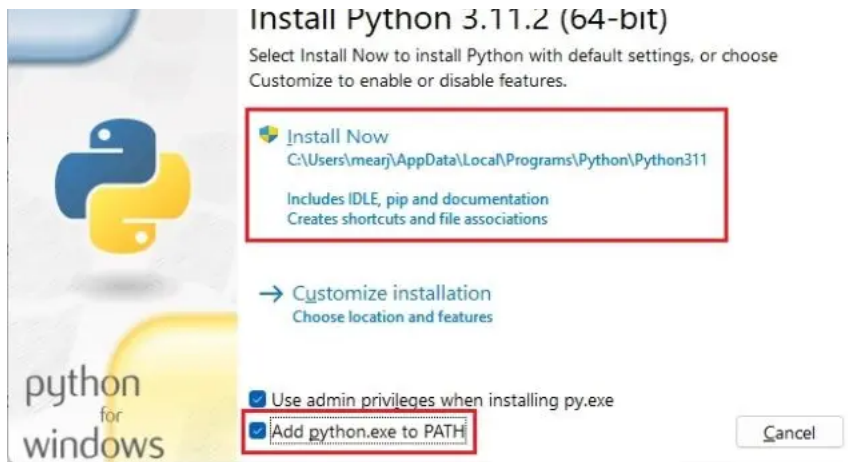
Like our previous article, you should know that Python and Pip must be installed along with several libraries. In this article, we will set up everything from scratch so new users can also understand the setup process. To give you a brief idea, we will install Python and Pip. After that, we will install Python libraries, which include OpenAI, GPT Index, Gradio, and PyPDF2. Along the process, you will learn what each library does. Again, do not fret over the installation process, it's pretty straightforward. On that note, let's jump right in.

### Install Python

1. First off, you need to install Python (Pip) on your computer. Open [this link](#) and download the setup file for your platform.

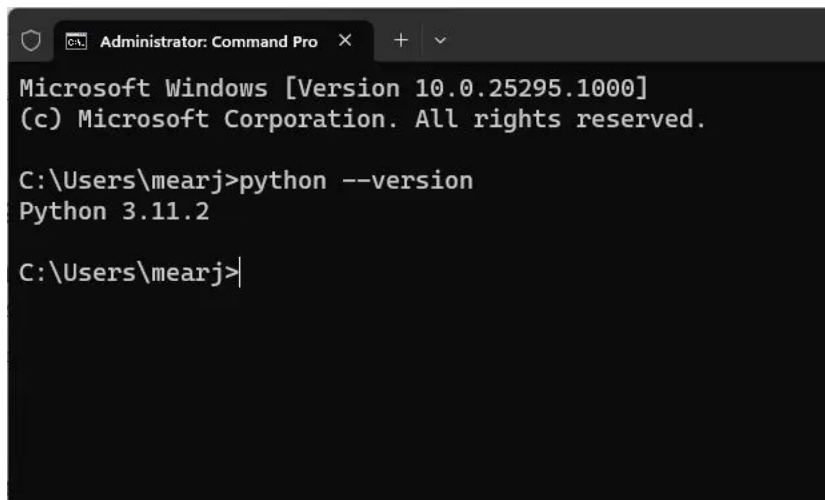


2. Next, run the setup file and make sure to enable the checkbox for **“Add Python.exe to PATH.”** This is an extremely important step. After that, click on “Install Now” and follow the usual steps to install Python.



3. To check if **Python is properly installed**, open the Terminal on your computer. I'm using Windows Terminal on Windows, but you can also use Command Prompt. Once here, run the below command below, and it will output the Python version. On Linux and macOS, you may have to use `python3 --version` instead of `python --version`.

```
python --version
```



## Upgrade Pip

When you install Python, Pip is installed simultaneously on your system. So let's upgrade it to the latest version. For those unaware, **Pip is the package manager for Python**. Basically, it lets you install thousands of Python libraries from the Terminal. With Pip, we can install OpenAI, gpt\_index, gradio, and PyPDF2 libraries. Here are the steps to follow.

1. Open the Terminal of your choice on your computer. I'm using the [Windows Terminal](#), but you can also use Command Prompt. Now, run the below command to **update Pip**. Again, you may have to use `python3` and `pip3` on Linux and macOS.

```
python -m pip install -U pip
```



```
C:\Users\mearj>python -m pip install -U pip
Requirement already satisfied: pip in c:\users\mearj\appdata\local\programs\python\python311\lib\site-packages (22.3.1)
Collecting pip
  Downloading pip-23.0.1-py3-none-any.whl (2.1 MB)
    2.1/2.1 MB 3.7 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3.1
    Uninstalling pip-22.3.1:
      Successfully uninstalled pip-22.3.1
  Successfully installed pip-23.0.1

C:\Users\mearj>
```

2. To **check if Pip was properly installed**, run the below command. It will output the version number. If you get any errors, follow our dedicated guide on [how to install Pip on Windows](#) to fix PATH-related issues.

```
pip --version
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.1344]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mearj>pip --version
pip 23.0.1 from C:\Users\mearj\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)

C:\Users\mearj>
```

## Install OpenAI, GPT Index, PyPDF2, and Gradio Libraries

Once we have set up Python and Pip, it's time to install the essential libraries that will help us train an AI chatbot with a custom knowledge base. Here are the steps to follow.

1. Open the Terminal and run the below command to **install the OpenAI library**. We will use it as the LLM (Large language model) to train and create an AI chatbot. And we will also import the LangChain framework from OpenAI. Note that, Linux and macOS users may have to use `pip3` instead of `pip`.

```
pip install openai
```



```
C:\Users\mearj>pip install openai
Collecting openai
  Downloading openai-0.27.0-py3-none-any.whl (70 kB)
    70.1/70.1 kB 1.3 MB/s eta 0:00:00
Collecting requests>=2.20
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
    62.8/62.8 kB 3.3 MB/s eta 0:00:00
Collecting tqdm
  Downloading tqdm-4.65.0-py3-none-any.whl (77 kB)
    77.1/77.1 kB 4.2 MB/s eta 0:00:00
Collecting aiohttp
  Downloading aiohttp-3.8.4-cp311-cp311-win_amd64.whl (317 kB)
    317.2/317.2 kB 2.8 MB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.1.0-cp311-cp311-win_amd64.whl (96 kB)
    96.7/96.7 kB 5.8 MB/s eta 0:00:00
Collecting idna<4,>=2.5
  Downloading idna-3.4-py3-none-any.whl (61 kB)
    61.5/61.5 kB 1.1 MB/s eta 0:00:00
```

2. Next, let's **install GPT Index**, which is also called LlamaIndex. It allows the LLM to connect to the external data that is our knowledge base.

```
pip install gpt_index
```

```
Administrator: Command Pro
C:\Users\mearj>pip install gpt_index
Collecting gpt_index
  Downloading gpt_index-0.4.27-py3-none-any.whl (240 kB)
    240.6/240.6 kB 1.8 MB/s eta 0:00:00
Collecting dataclasses-json
  Downloading dataclasses_json-0.5.7-py3-none-any.whl (25 kB)
Collecting langchain
  Downloading langchain-0.0.108-py3-none-any.whl (374 kB)
    374.6/374.6 kB 5.9 MB/s eta 0:00:00
Requirement already satisfied: numpy in C:\users\mearj\appdata\local\programs\python\python311\lib\site-packages (from gpt_index) (1.24.2)
Collecting tenacity<9.0.0,>=8.2.0
  Downloading tenacity-8.2.2-py3-none-any.whl (24 kB)
Requirement already satisfied: openai>=0.26.4 in c:\users\mearj\appdata\local\programs\python\python311\lib\site-packages (from gpt_index) (0.27.0)
Requirement already satisfied: pandas in c:\users\mearj\appdata\lo
```

3. After that, **install PyPDF2** to parse PDF files. If you want to feed your data in PDF format, this library will help the program read the data effortlessly.

```
pip install PyPDF2
```



```
C:\Users\mearj>pip install PyPDF2
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
    232.6/232.6 kB 1.8 MB/s eta 0:00:00
```

4. Finally, **install the Gradio library**. This is meant for creating a simple UI to interact with the trained AI chatbot. We are now done installing all the required libraries to train an AI chatbot.

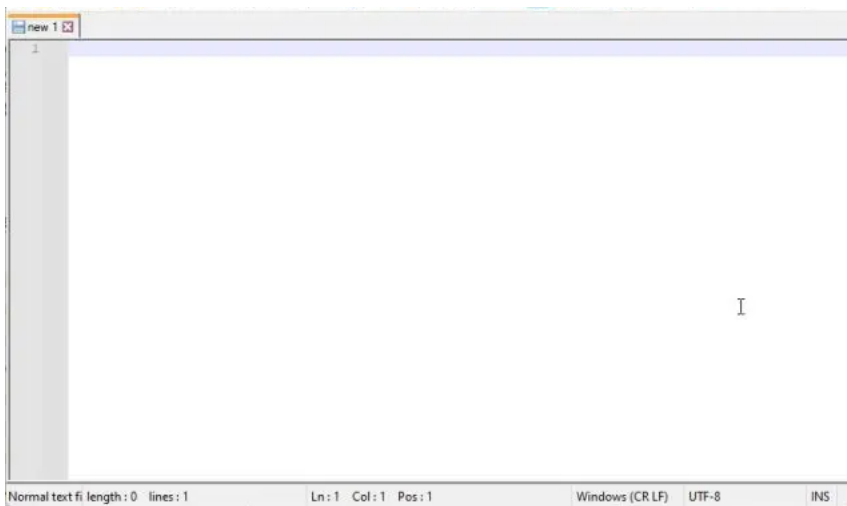
```
pip install gradio
```

```
Administrator: Command Prompt
C:\Users\mearj>pip install gradio
Collecting gradio
  Downloading gradio-3.20.0-py3-none-any.whl (14.3 MB)
    4.8/14.3 MB 1.5 MB/s eta 0:00:07
```

### Download a Code Editor

Finally, we need a code editor to edit some of the code. On Windows, I would recommend **Notepad++** ([Download](#)). Simply download and install the program via the attached link. You can also use VS Code on any platform if you are comfortable with powerful IDEs. Other than VS Code, you can install Sublime Text ([Download](#)) on macOS and Linux.


For ChromeOS, you can use the excellent **Caret** app ([Download](#)) to edit the code. We are almost done setting up the software environment, and it's time to get the OpenAI API key.



## Get the OpenAI API Key For Free

Now, to train and create an AI chatbot based on a custom knowledge base, we need to get an API key from OpenAI. The API key will allow you to use OpenAI's model as the LLM to study your custom data and draw inferences. Currently, OpenAI is offering **free API keys with \$5 worth of free credit** for the first three months to new users. If you created your OpenAI account earlier, you may have free \$18 credit in your account. After the free credit is exhausted, you will have to pay for the API access. But for now, it's available to all users for free.

1. Head to [platform.openai.com/signup](https://platform.openai.com/signup) and **create a free account**. If you already have an OpenAI account, simply log in.




### Create your account


Please note that phone verification is required for signup. Your number will only be used to verify your identity for security purposes.

Continue

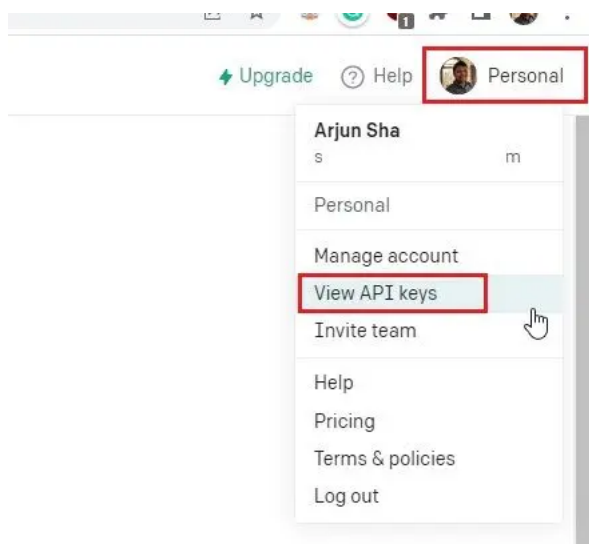
Already have an account? [Log in](#)

OR

 Continue with Google

 Continue with Microsoft Account

2. Next, click on your profile in the top-right corner and select **"View API keys"** from the drop-down menu.



3. Here, click on “**Create new secret key**” and copy the API key. Do note that you can’t copy or view the entire API key later on. So it’s strongly recommended to copy and paste the API key to a Notepad file immediately.

## API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we’ve found has leaked publicly.

SECRET KEY	CREATED	LAST USED	
sk-...kelz	Mar 3, 2023	Mar 4, 2023	🗑
sk-...m01H	Mar 4, 2023	Mar 6, 2023	🗑

[+ Create new secret key](#)

4. Also, **do not share or display the API key** in public. It’s a private key meant only for access to your account. You can also delete API keys and create multiple private keys (up to five).

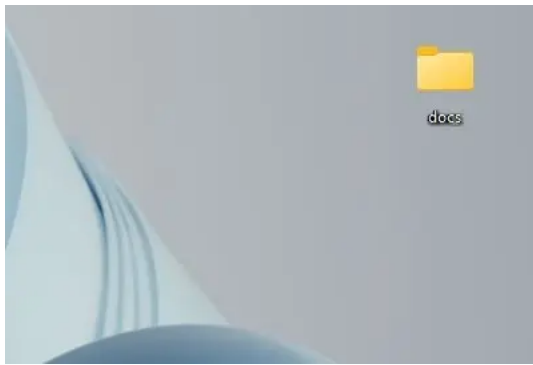
## Train and Create an AI Chatbot With Custom Knowledge Base

Now that we have set up the software environment and got the API key from OpenAI, let’s train the AI chatbot. Here, we will use the “**text-davinci-003**” model instead of the latest “gpt-3.5-turbo” model because Davinci works much better for text completion. If you want, you can very well change the model to Turbo to reduce the cost. With that out of the way, let’s jump to the instructions.

### Add Your Documents to Train the AI Chatbot

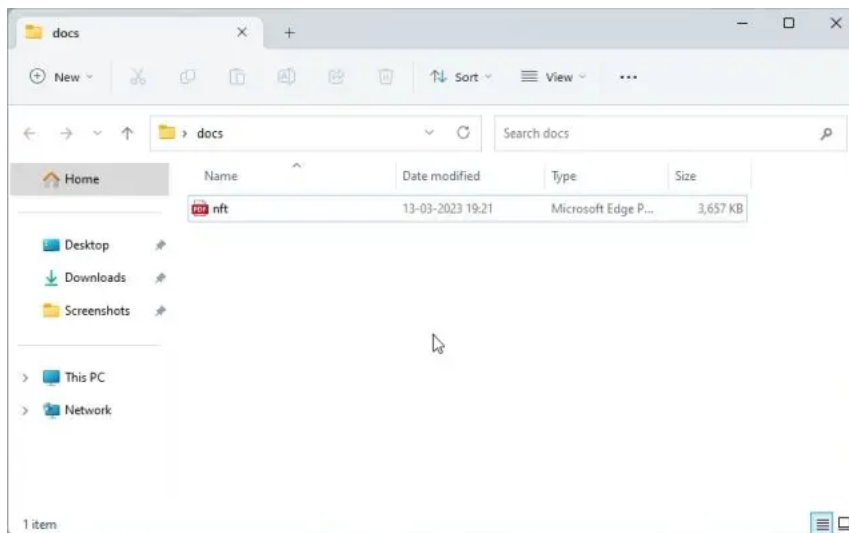
1. First, **create a new folder called docs** in an accessible location like the Desktop. You can choose another location as well according to your preference. However, keep the folder name `docs`.





2. Next, move the documents you wish to use for training the AI inside the “docs” folder. You can **add multiple text or PDF files** (even scanned ones). If you have a large table in Excel, you can import it as a CSV or PDF file and then add it to the “docs” folder. You can even add SQL database files, as explained in this [Langchain AI tweet](#). I haven’t tried many file formats besides the mentioned ones, but you can add and check on your own. For this article, I am adding one of my articles on [NFT](#) in PDF format.

***Note:*** If you have a large document, it will take a longer time to process the data, depending on your CPU and GPU. In addition, it will quickly use your free OpenAI tokens. So in the beginning, start with a small document (30-50 pages or < 100MB files) to understand the process.



## Make the Code Ready

1. Now, **launch Notepad++** (or your choice of code editor) and paste the below code into a new file. Once again, I have taken great help from [armrrs on Google Colab](#) and tweaked the code to make it compatible with PDF files and create a Gradio interface on top.

```
from gpt_index import SimpleDirectoryReader,
GPTListIndex, GPTSimpleVectorIndex, LLMPredictor,
PromptHelper
from langchain import OpenAI
import gradio as gr
```



```
os.environ["OPENAI_API_KEY"] = 'Your API Key'

def construct_index(directory_path):
    max_input_size = 4096
    num_outputs = 512
    max_chunk_overlap = 20
    chunk_size_limit = 600

    prompt_helper = PromptHelper(max_input_size,
    num_outputs, max_chunk_overlap,
    chunk_size_limit=chunk_size_limit)

    llm_predictor =
    LLMPredictor(llm=OpenAI(temperature=0.7,
    model_name="text-davinci-003", max_tokens=num_outputs))

    documents =
    SimpleDirectoryReader(directory_path).load_data()

    index = GPTSimpleVectorIndex(documents,
    llm_predictor=llm_predictor,
    prompt_helper=prompt_helper)

    index.save_to_disk('index.json')

    return index

def chatbot(input_text):
    index =
    GPTSimpleVectorIndex.load_from_disk('index.json')
    response = index.query(input_text,
    response_mode="compact")
    return response.response

iface = gr.Interface(fn=chatbot,
                     inputs=gr.inputs.Textbox(lines=7,
    label="Enter your text"),
                     outputs="text",
                     title="Custom-trained AI Chatbot")

index = construct_index("docs")
iface.launch(share=True)
```

2. This is what the **code looks like** in the code editor.

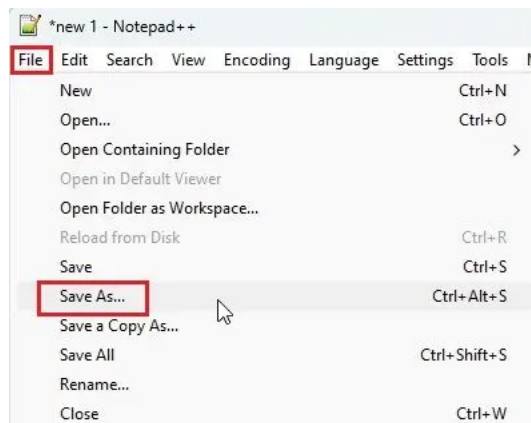


```

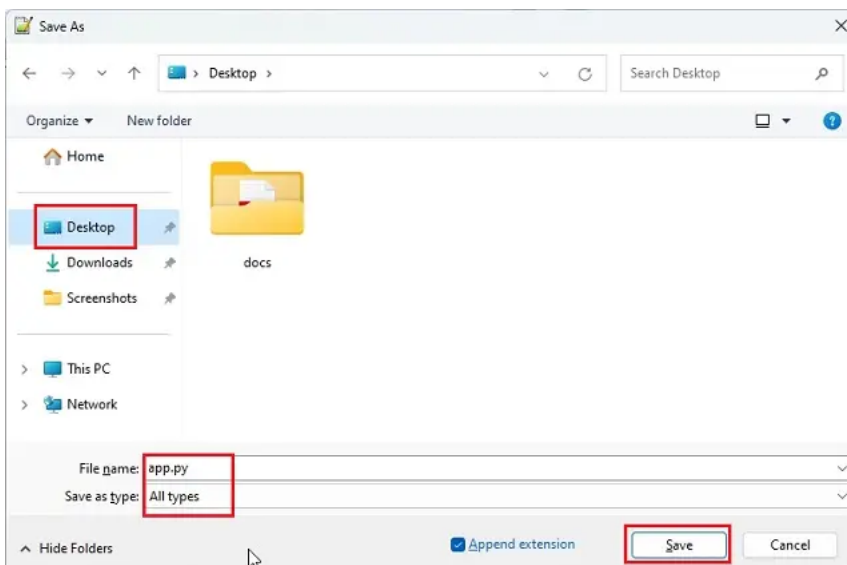
1 from gpt_index import SimpleDirectoryReader, GPTListIndex, GPTSimpleVectorIndex,
2   LLMPredictor, PromptHelper
3 from langchain import OpenAI
4 import gradio as gr
5 import sys
6 import os
7
8 os.environ["OPENAI_API_KEY"] = 'Your API Key'
9
10 def construct_index(directory_path):
11     max_input_size = 4096
12     num_outputs = 512
13     max_chunk_overlap = 20
14     chunk_size_limit = 600
15
16     prompt_helper = PromptHelper(max_input_size, num_outputs, max_chunk_overlap,
17     chunk_size_limit=chunk_size_limit)
18
19     llm_predictor = LLMPredictor(llm=OpenAI(temperature=0.7,
20     model_name="text-davinci-003", max_tokens=num_outputs))
21
22     documents = SimpleDirectoryReader(directory_path).load_data()
23
24     index = GPTSimpleVectorIndex(documents, llm_predictor=llm_predictor,
25     prompt_helper=prompt_helper)

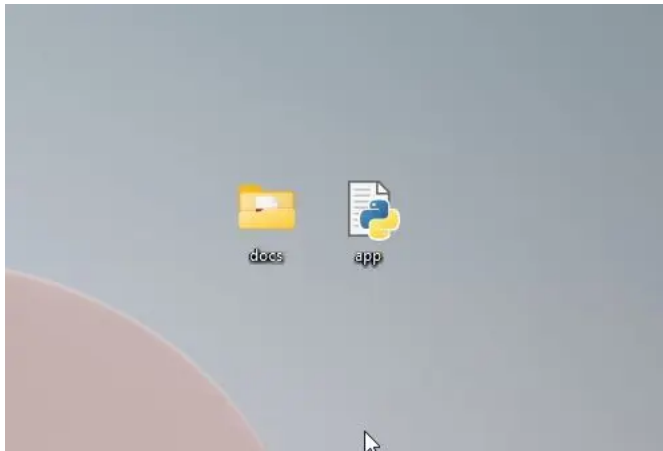
```

3. Next, click on “File” in the top menu and select “Save As...” from the drop-down menu.



4. After that, set the file name **app.py** and change the “Save as type” to “All types” from the drop-down menu. Then, save the file to the location where you created the “docs” folder (in my case, it’s the Desktop). You can change the name to your liking, but make sure **.py** is appended.





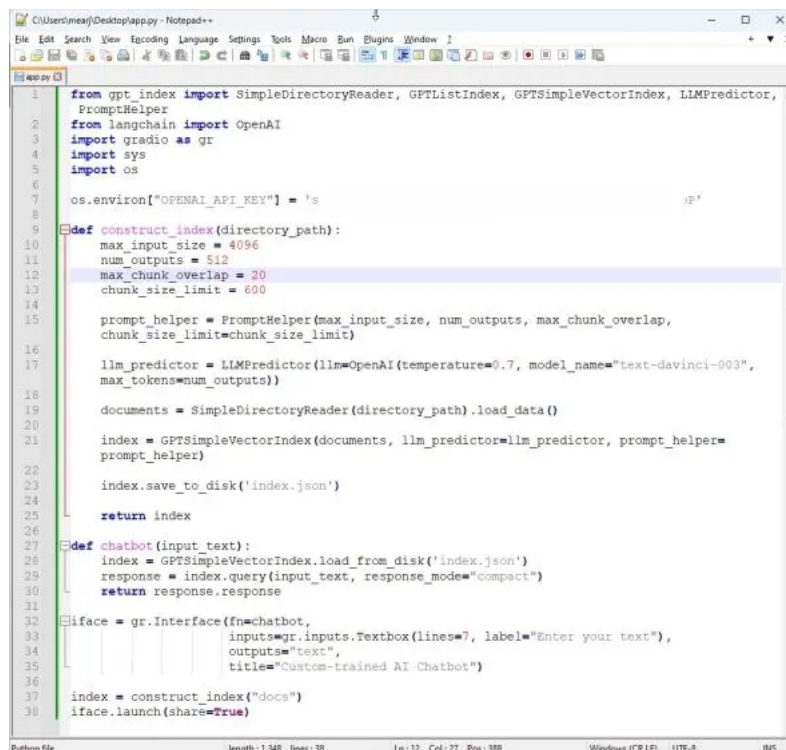
6. Come back to the code again in Notepad++. Here, replace **Your API Key** with the one generated on OpenAI's website above.

```
from gpt_index import SimpleDirectoryReader, GPTListIndex, GPTSimpleVectorIndex, LLMPredictor, PromptHelper
from langchain import OpenAI
import gradio as gr
import sys
import os

os.environ["OPENAI_API_KEY"] = 'sk-'

def construct_index(directory_path):
    max_input_size = 4096
    num_outputs = 512
    max_chunk_overlap = 20
    chunk_size_limit = 600
```

7. Finally, press “**Ctrl + S**” to save the code. You are now ready to run the code.

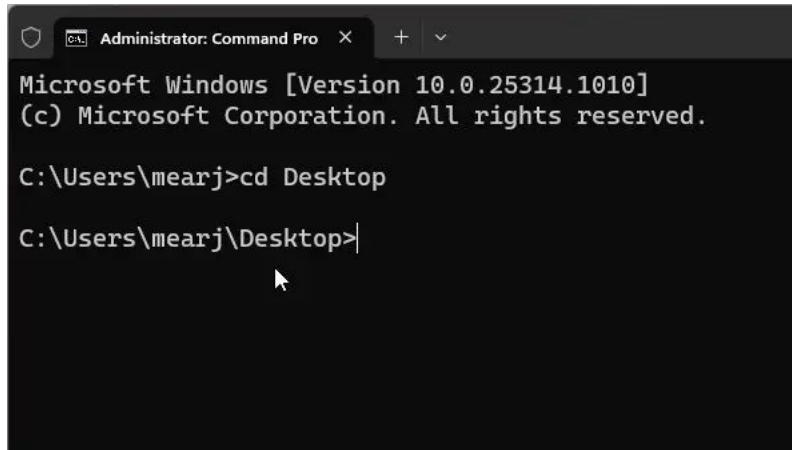


## Create ChatGPT AI Bot with Custom Knowledge Base



that location via the Terminal.

```
cd Desktop
```



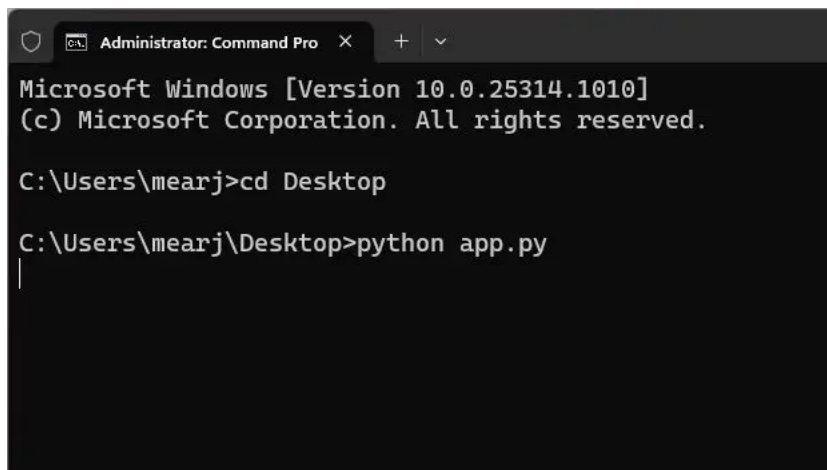
```
Administrator: Command Pro
Microsoft Windows [Version 10.0.25314.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mearj>cd Desktop

C:\Users\mearj\Desktop>
```

2. Now, run the below command. Linux and macOS users may have to use `python3` .

```
python app.py
```

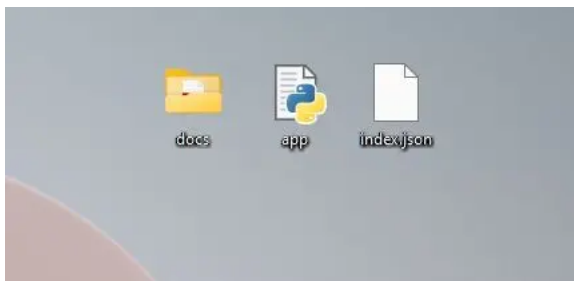


```
Administrator: Command Pro
Microsoft Windows [Version 10.0.25314.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mearj>cd Desktop

C:\Users\mearj\Desktop>python app.py
|
```

3. Now, it will **start analyzing the document** using the OpenAI LLM model and start indexing the information. Depending on the file size and your computer's capability, it will take some time to process the document. Once it's done, an **"index.json"** file will be created on the Desktop. If the Terminal is not showing any output, do not worry, it might still be processing the data. For your information, **it takes around 10 seconds to process a 30MB document**.

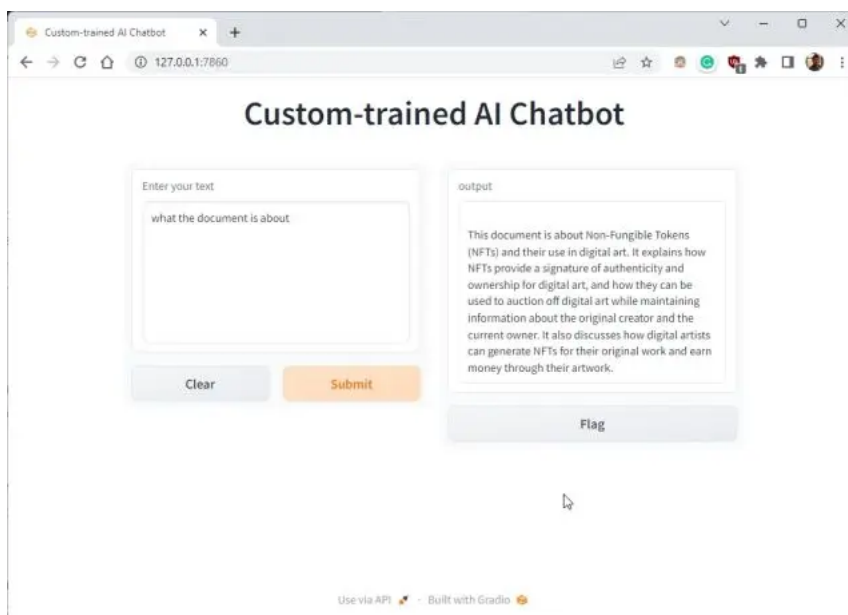


4. Once the LLM has processed the data, you will get a few warnings that can be safely ignored. Finally, at the bottom, you will find a **local URL**. Copy it.

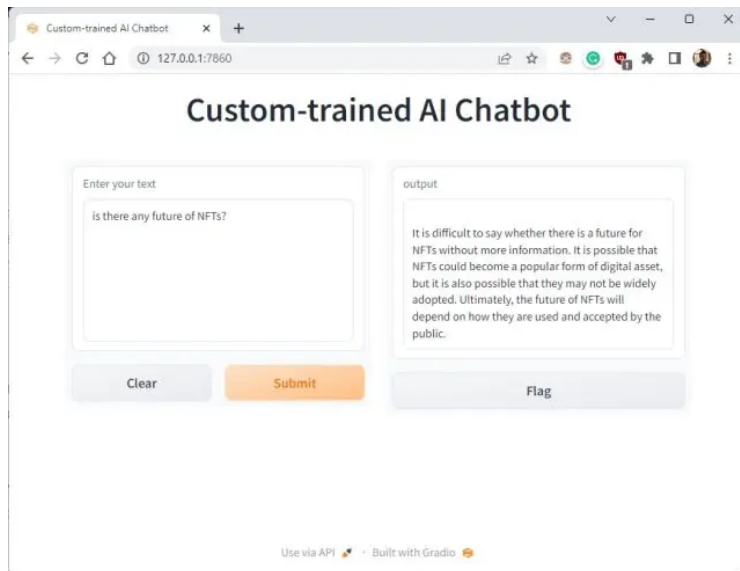
```
Administrator: Command Prompt
nd will not be supported in the future, please import your component from
gradio.components
warnings.warn(
C:\Users\mearyj\AppData\Local\Programs\Python\Python311\Lib\site-packages\
gradio\deprecation.py:40: UserWarning: 'optional' parameter is deprecated
, and it has no effect
warnings.warn(value)
C:\Users\mearyj\AppData\Local\Programs\Python\Python311\Lib\site-packages\
gradio\deprecation.py:40: UserWarning: 'numeric' parameter is deprecated,
and it has no effect
warnings.warn(value)
INFO:root:> [build_index_from_documents] Total LLM token usage: 0 tokens
INFO:root:> [build_index_from_documents] Total embedding token usage: 336
6 tokens
Running on local URL: http://127.0.0.1:7860
Running on public URL: https://572deefed895ae17e2.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU u
pgrades (NEW!), check out Spaces: https://huggingface.co/spaces
```

5. Now, paste the copied URL into the web browser, and there you have it. Your custom-trained ChatGPT-powered AI chatbot is ready. To start, you can ask the AI chatbot **what the document is about**.



6. You can ask further questions, and the ChatGPT bot will **answer from the data you provided** to the AI. So this is how you can build a custom-trained AI chatbot with your own



7. You can also copy the **public URL** and share it with your friends and family. The link will be live for 72 hours, but you also need to keep your computer turned on since the server instance is running on your computer.

```
nd will not be supported in the future, please import your component from
gradio.components
warnings.warn(
C:\Users\mearyj\AppData\Local\Programs\Python\Python311\Lib\site-packages\
gradio\deprecation.py:40: UserWarning: 'optional' parameter is deprecated
, and it has no effect
warnings.warn(value)
C:\Users\mearyj\AppData\Local\Programs\Python\Python311\Lib\site-packages\
gradio\deprecation.py:40: UserWarning: 'numeric' parameter is deprecated,
and it has no effect
warnings.warn(value)
INFO:root:> [build_index_from_documents] Total LLM token usage: 0 tokens
INFO:root:> [build_index_from_documents] Total embedding token usage: 336
6 tokens
Running on local URL: http://127.0.0.1:7860
Running on public URL: https://75fe664f415ef7b696.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU u
pgrades (NEW!), check out Spaces: https://huggingface.co/spaces
```

8. To **stop the custom-trained AI chatbot**, press “Ctrl + C” in the Terminal window. If it does not work, press “Ctrl + C” again.



```
INFO:root:> [build_index_from_documents] Total embedding token usage: 336
6 tokens
Running on local URL:  http://127.0.0.1:7860
Running on public URL: https://75fe664f415ef7b696.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU u
pgrades (NEW!), check out Spaces: https://huggingface.co/spaces
INFO:root:> [query] Total LLM token usage: 76 tokens
INFO:root:> [query] Total embedding token usage: 4 tokens
INFO:root:> [query] Total LLM token usage: 94 tokens
INFO:root:> [query] Total embedding token usage: 4 tokens
INFO:root:> [query] Total LLM token usage: 94 tokens
INFO:root:> [query] Total embedding token usage: 6 tokens
INFO:root:> [query] Total LLM token usage: 755 tokens
INFO:root:> [query] Total embedding token usage: 6 tokens
Keyboard interruption in main thread... closing server.
Killing tunnel 127.0.0.1:7860 <-> https://75fe664f415ef7b696.gradio.live

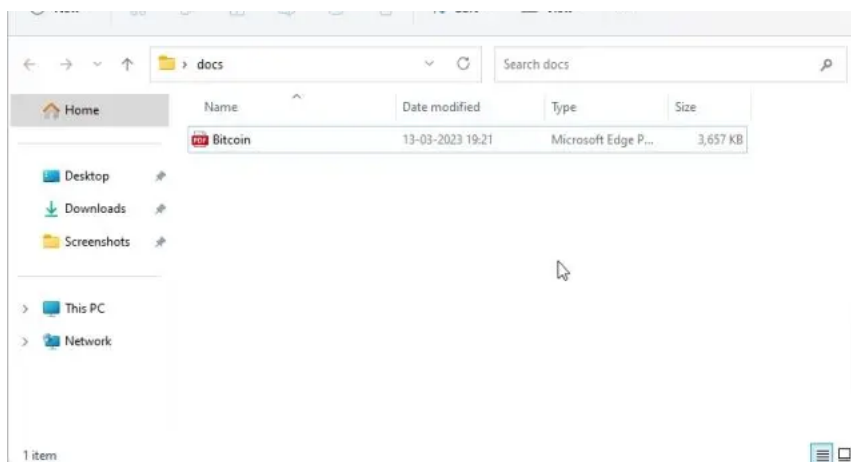
C:\Users\mearyj\Desktop>
```

9. To **restart the AI chatbot** server, simply move to the Desktop location again and run the below command. Keep in mind, the local URL will be the same, but the public URL will change after every server restart.

```
python app.py
```

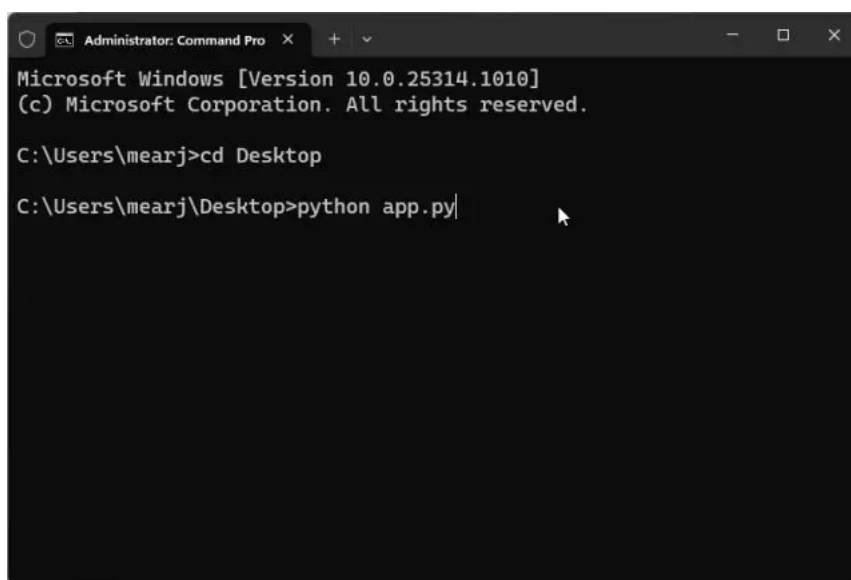
10. If you want to train the AI chatbot with **new data**, delete the files inside the “docs” folder and add new ones. You can also add multiple files, but feed information on the same subject otherwise you may get an incoherent response.





11. Now, **run the code again** in the Terminal, and it will create a new “index.json” file. Here, the old “index.json” file will be replaced automatically.

```
python app.py
```



12. To keep track of your tokens, head over to OpenAI’s online [dashboard](#) and check how much free credit is left.



13. Lastly, you don't need to **touch the code** unless you want to change the API key or the OpenAI model for further customization.

## Build a Custom AI Chatbot Using Your Own Data

So this is how you can train an AI chatbot with a custom knowledge base. I have used this code to train the AI on medical books, articles, data tables, and reports from old archives, and it has worked flawlessly. So go ahead and create your own AI chatbot using OpenAI's Large Language Model and ChatGPY. Anyway, that is all from us. If you are looking for the [best ChatGPT alternatives](#), head to our linked article. And to [use ChatGPT on your Apple Watch](#), follow our in-depth tutorial. Finally, if you are facing any kind of issues, do let us know in the comment section below. We will definitely try to help you out.

TAGS

Chatbots

chatGPT

featured

OpenAI

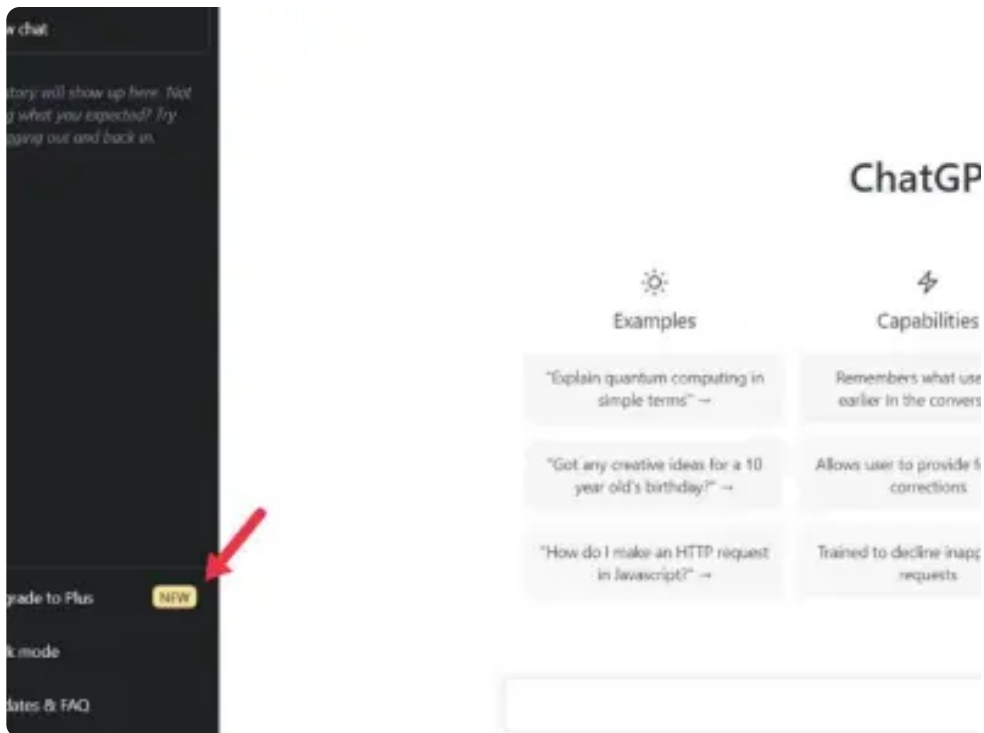
9 Comments

## RECOMMENDED ARTICLES

---



How to Set up and Use ChatGPT in Linux Terminal



ChatGPT's Subscription Plan Can Now Be Bought in India



Jackery Solar Generator 3000 Pro with Ultra-Fast Charging Goes on Sale from March 27th



OpenAI GPT-4: Multimodal, New Features, Image Input, How to Use & More



## How to Get Access to GPT-4 Right Now!



**Jio Plus Is Your Solution When You Need Everything**

9 COMMENTS



**Jo** Mar 18, 2023 at 10:52 am

Thanks for the free source of codes. I tried it, it worked!

But I have a few questions which need your advice please as I couldn't find them in the post.

1. On training with my custom data by calling the “construct\_index” function, will any of my data be stored on the OpenAPI Cloud?
2. What will be contained in the index.json file? How will this file be used in the chatbot Q&A process?



4. On this:

INFO:gpt\_index.token\_counter.token\_counter:> [query] Total embedding token usage: 10 tokens

INFO:gpt\_index.token\_counter.token\_counter:> [query] Total LLM token usage: 598 tokens

Is it the imported OpenAPI library on my server or the OpenAI Cloud which calculate this token usage?

5. Noted that the token used will be synced to the OpenAI portal dashboard in every few minutes, so this program has to run with internet connectivity?

Thanks much in advance! Your advice will be very helpful to me!

Reply



**Chris** Mar 18, 2023 at 4:06 am

Excellent helpful article thank you Arjun. Instead of using a laptop, can this be hosted on a server and on a cpanel hosted site?

Reply



**shone** Mar 17, 2023 at 8:54 pm

Hi Arjun, that's a great article, thanks for sharing that. I got a concern here, if you got any idea for that, that would be great,

If we are using custom data to train this AI mode, where is the data store for custom data? I mean will custom data be saved into open ai server? or the data will only keep on our server.

Reply



**Arjun Sha** Mar 17, 2023 at 9:25 pm

No, the data is stored locally in the index.json file. However, OpenAI LLM is used to draw inferences from the dataset. OpenAI has also said that data submitted to the company through the API won't be used for its AI training.

Reply



**Tobias** Mar 17, 2023 at 12:27 am

I have problems with training the model. Can you share a dataset that should work?

Reply



**Rahul Nagpure** Mar 15, 2023 at 11:28 am

Very nice article. I could setup everything withing 10 mins on my mac.  
Can we provide URLs instead of PDF or test documents ?

Reply



**Arjun Sha** Mar 15, 2023 at 3:36 pm

You can actually use wget to download files automatically using URLs. You will have to add "wget URL" before you run the construct\_index function.

Reply

## LEAVE A REPLY

### Your Comment

Comment:

### Your Name

Name:\*

[Post Comment](#)

## REVIEWS

---



### MSI Titan GT77 HX 13V Review: Desktop-Grade Performance for the Price of a Car

8.6

Last year, MSI launched the Titan GT77 with the Intel Core i9-12900HX and the RTX 3080 Ti Laptop GPU, and it was the most powerful gaming laptop on the face of the planet. It was the heaviest of heavy hitters [...]



### iPhone 14 Plus Review: The Less Noticed Big Guy!

8.6

It's been a few months since the iPhone 14 series launched, and it has been well established that it is the year of the Pro models. But if you intend to go for the frilly Pros, it amounts to shell [...]





## Wondershare Filmora 12 Review: A Cross-Platform Video Editor for Budding Creators

Wondershare has been developing some of the best software and tools to simplify our life and creative endeavors for the past several years. Particularly, Wondershare Filmora 12 has received numerous accolades. It was the recipient of the Video Editing Leader award [...]

[CONTACT US](#)[ADVERTISE](#)[ABOUT US](#)

© Beebom Media Private Limited