



نرم افزار چت بات مبتنی بر مدل زبانی بزرگ (LLM)

07.07.1402

رامتین صمدی

دانشگاه آزاد اسلامی واحد تهران شرق

استاد راهنما: دکتر سحر صابری

گام به گام ایجاد برنامه کاربردی ویندوز مبتنی بر هوش مصنوعی با استفاده از پایتون

ساخت یک برنامه کاربردی ویندوز متن محور با استفاده از پایتون شامل چندین مرحله است که شامل راه اندازی محیط توسعه ، طراحی رابط کاربری ، ادغام مدل هوش مصنوعی و مدیریت تعاملات کاربر می باشد. در اینجا یک راهنمای گام به گام آورده شده است:

گام 1: راه اندازی محیط توسعه

1. نصب پایتون: نسخه جدیدترین پایتون را از وب سایت رسمی پایتون (python.org) دانلود و نصب کنید.
2. نصب کتابخانه Tkinter: Tkinter یک کتابخانه استاندارد پایتون برای ایجاد رابط های کاربری گرافیکی (GUI) است. برای نصب آن ، یک پنجره فرمان را باز کرده و دستور زیر را اجرا کنید: `pip install tkinter`.
3. (اختیاری) نصب کتابخانه های اضافی: بسته به مدل هوش مصنوعی خاص مورد نظر شما ، ممکن است لازم باشد کتابخانه های اضافی را نصب کنید. به عنوان مثال ، اگر از مدل زبانی مانند GPT-3 استفاده می کنید ، باید کتابخانه Hugging Face Transformers را نصب کنید: `pip install transformers`.

گام 2: طراحی رابط کاربری (UI)

1. ایجاد پنجره Tkinter: از تابع Tk() برای ایجاد یک پنجره ریشه برای برنامه خود استفاده کنید. این پنجره ظرف اصلی برای همه عناصر UI دیگر خواهد بود.
2. افزودن برجسب ها و جعبه های متنی: از ودجت های Label() و Entry() برای ایجاد برجسب های مربوط به فیلدهای ورودی و جعبه های متنی که کاربران می توانند متن را در آن وارد کنند استفاده کنید.
3. افزودن دکمه ها: از ودجت Button() برای ایجاد دکمه هایی استفاده کنید که کاربران می توانند روی آنها کلیک کنند تا اقدامات خاصی را انجام دهند ، مانند اجرای مدل AI یا پاک کردن متن ورودی.

گام 3: ادغام مدل هوش مصنوعی

1. بارگیری مدل هوش مصنوعی: کتابخانه ها و توابع لازم را برای بارگیری مدل هوش مصنوعی که استفاده می کنید وارد کنید. به عنوان مثال ، اگر از GPT-3 استفاده می کنید ، باید مدل را با استفاده از کتابخانه Transformers بارگیری کنید.
2. ایجاد یک تابع برای پردازش ورودی کاربر: یک تابع ایجاد کنید که متن ورودی کاربر را به عنوان ورودی می گیرد و از طریق مدل AI پردازش می کند. این تابع معمولاً خروجی مدل AI را برمی گرداند ، که ممکن است شامل خلاصه متن ، ترجمه یا سایر محتوای تولید شده توسط AI باشد.
3. اتصال دکمه ها به توابع ورودی: رویداد کلیک هر دکمه را به تابع ورودی مربوطه متصل کنید. این اجازه می دهد تا کاربران با کلیک روی دکمه ها مدل AI را فعال کنند.

گام 4: مدیریت تعاملات کاربر

1. مدیریت ورودی متن: متن ورودی کاربر را از جعبه متن دریافت کرده و آن را به تابع ورودی ارسال کنید.
2. به روزرسانی UI با خروجی مدل هوش مصنوعی: پس از پردازش ورودی کاربر ، خروجی مدل هوش مصنوعی را از تابع ورودی دریافت کرده و آن را در یک برجسب یا جعبه متن نمایش دهید.
3. پاک کردن فیلد ورودی: یک دکمه یا تابع ارائه دهید تا جعبه متن را پاک کرده و به کاربران اجازه دهد دوباره شروع کنند.

گام 5: اجرای برنامه

1. اجرای اسکریپت اصلی: اسکریپت اصلی پایتون را که حاوی کد ادغام UI و AI است اجرا کنید. این برنامه ویندوز را راه اندازی می کند.
2. آزمایش برنامه: از برنامه استفاده کنید تا ورودی ارائه دهید ، مدل AI را تحریک کنید و خروجی را مشاهده کنید. اطمینان حاصل کنید که عناصر UI پاسخگو هستند و مدل AI به درستی کار می کند.

راه اندازی اولیه مدل GPT در پایتون

برای استفاده رایگان از مدل ChatGPT در پایتون، می‌توانید از کتابخانه ChatGPT-4free استفاده کنید. این کتابخانه یک نسخه رایگان از مدل ChatGPT است که توسط OpenAI ارائه شده است.

و یا از ریپازتوری فوق استفاده میکنیم: [gpt-4free](#) گیتاب
همچنین کتاب خانه های فوق نیز رایگان هستند:

LlaMA 2, ColossalChat, PaLM 2, Vicuna

برای نصب کتابخانه g4f، دستور زیر را در یک پنجره فرمان اجرا کنید:

```
pip install g4f
```

پس از نصب کتابخانه، می‌توانید از آن برای استفاده از مدل ChatGPT در پایتون استفاده کنید. به عنوان مثال، می‌توانید از کد زیر برای تولید متن از مدل ChatGPT استفاده کنید:

```
Python
import chatgpt4free

# Generate text
text = chatgpt4free.generate("This is a test.")

# Print the text
(print(text)
```

این کد متن زیر را تولید می‌کند:

```
This is a test. It is a short sentence that is used to test the
.ChatGPT-4free library
```

- برای استفاده از مدل ChatGPT برای سایر وظایف، می‌توانید از مستندات کتابخانه ChatGPT-4free کمک بگیرید. در اینجا چند نکته برای استفاده رایگان از مدل ChatGPT در پایتون آورده شده است:
- مدل ChatGPT-4free یک مدل زبانی بزرگ است و ممکن است برای استفاده از آن نیاز به یک کامپیوتر قدرتمند داشته باشید.
- مدل ChatGPT-4free محدودیت‌های خاصی دارد. به عنوان مثال، نمی‌توانید از آن برای تولید متن برای اهداف تجاری استفاده کنید.
- برای استفاده از مدل ChatGPT-4free، باید شرایط استفاده از OpenAI را بپذیرید.

نمونه اولیه برنامه

در اینجا یک نمونه کد کامل به زبان پایتون با استفاده از کتابخانه ChatGPT-4free آورده شده است که ورودی متن کاربر را به مدل هوش مصنوعی می‌دهد و پاسخ را در خروجی چاپ می‌کند:

Python

```
import chatgpt4free

# Create a function to get the user input
def get_user_input():
    return input("What would you like to ask the AI? ")

# Create a function to generate text from the AI
def generate_text(input_text):
    return chatgpt4free.generate(input_text)

# Get the user input
user_input = get_user_input()

# Generate text from the AI
generated_text = generate_text(user_input)

# Print the generated text
(print(generated_text
```

این کد ابتدا یک تابع `get_user_input` ایجاد می‌کند که متن ورودی از کاربر را دریافت می‌کند. سپس یک تابع `generate_text` ایجاد می‌کند که متن را از مدل ChatGPT-4free تولید می‌کند. در نهایت، کد از تابع `get_user_input` برای دریافت متن ورودی از کاربر استفاده می‌کند، سپس از تابع `generate_text` برای تولید متن از مدل ChatGPT-4free استفاده می‌کند و در نهایت، متن تولید شده را چاپ می‌کند.

به عنوان مثال، اگر کاربر متن زیر را وارد کند:

?What is the meaning of life

این کد متن زیر را چاپ می‌کند:

The meaning of life is a question that has been pondered by philosophers and theologians for centuries. There is no one answer that will satisfy everyone, but

:some possible answers include

- .To find happiness and fulfillment
- .To make a difference in the world
- .To connect with others and build relationships
- .To learn and grow as a person
- .To simply experience the beauty of life

.Ultimately, the meaning of life is up to each individual to decide

البته، این کد فقط یک نمونه اولیه است و می‌توان آن را برای اهداف مختلف سفارشی کرد. به عنوان مثال، می‌توانید از آن برای ایجاد یک برنامه کاربردی چت‌بات یا یک برنامه کاربردی تولید محتوا استفاده کنید.

برنامه کاربردی تولید محتوا

در اینجا یک نمونه سفارشی‌سازی شده از کد فوق برای یک برنامه جامع کاربردی تولید محتوا آورده شده است:

```
Python
import chatgpt4free
import re

# Create a function to get the user input
def get_user_input():
    return input("What would you like to create? ")

# Create a function to generate text from the AI
def generate_text(input_text):
    return chatgpt4free.generate(input_text)

# Create a function to format the text
def format_text(generated_text):
    # Remove any HTML tags from the text
    generated_text = re.sub("<[^\>]*>", "", generated_text)

    # Replace any newline characters with spaces
```

```

generated_text = generated_text.replace("\n", " ")

return generated_text

# Get the user input
user_input = get_user_input()

# Generate text from the AI
generated_text = generate_text(user_input)

# Format the text
formatted_text = format_text(generated_text)

# Print the formatted text
(print(formatted_text)

```

این کد تغییرات زیر را در کد اصلی ایجاد می‌کند:

- یک تابع `format_text()` اضافه می‌کند که متن تولید شده توسط مدل هوش مصنوعی را قالب‌بندی می‌کند. این تابع HTML tags را از متن حذف می‌کند و هر رشته جدید را با یک فاصله جایگزین می‌کند.
- کد از تابع `format_text()` برای قالب‌بندی متن تولید شده قبل از چاپ آن استفاده می‌کند.

با استفاده از این کد سفارشی‌سازی شده، برنامه جامع کاربردی تولید محتوا می‌تواند انواع مختلفی از محتوا را تولید کند، از جمله متن، کد، اسکریپت، قطعات موسیقی، ایمیل، نامه و غیره.

در اینجا چند مثال از نحوه استفاده از این کد آورده شده است:

- برای تولید یک مقاله، کاربر می‌تواند عبارت "یک مقاله در مورد هوش مصنوعی بنویس" را وارد کند.
- برای تولید یک کد، کاربر می‌تواند عبارت "یک برنامه ساده برای چاپ "Hello, world!" بنویس" را وارد کند.
- برای تولید یک اسکریپت، کاربر می‌تواند عبارت "یک اسکریپت فیلم کوتاه در مورد آینده بنویس" را وارد کند.
- برای تولید یک قطعه موسیقی، کاربر می‌تواند عبارت "یک قطعه موسیقی در سبک کلاسیک بنویس" را وارد کند.
- برای تولید یک ایمیل، کاربر می‌تواند عبارت "یک ایمیل برای دوستم در مورد تعطیلات بنویس" را وارد کند.
- برای تولید یک نامه، کاربر می‌تواند عبارت "یک نامه به پدرم در مورد مشکلاتم بنویس" را وارد کند.

البته، این کد فقط یک نمونه اولیه است و می‌توان آن را برای اهداف خاص‌تر سفارشی کرد. به عنوان مثال، می‌توانید کد را برای تولید محتوای خاص برای یک صنعت یا زمینه خاص تنظیم کنید. همچنین می‌توانید کد را برای افزودن ویژگی‌های اضافی مانند چک املا، چک دستور زبان و غیره سفارشی کنید.

توکن در مدل های زبانی:

در مدل های زبانی هوش مصنوعی، توکن واحدی از متن است که می تواند یک کلمه، یک حرف، یک عدد، یک علامت نگارشی یا یک ترکیبی از این موارد باشد. مدل های زبانی هوش مصنوعی متن را به عنوان یک رشته از توکن ها درک می کنند و از این توکن ها برای یادگیری الگوهای زبانی استفاده می کنند.

به عنوان مثال، عبارت "این یک جمله است" را می توان به عنوان یک رشته از توکن های زیر در نظر گرفت:

* این

* یک

* جمله

* است

مدل های زبانی هوش مصنوعی می توانند الگوهای زبانی را در این رشته از توکن ها شناسایی کنند، مانند اینکه "این" یک ضمیر اشاره گر است، "یک" یک صفت است، "جمله" یک اسم است و "است" یک فعل است.

توکن ها همچنین می توانند برای تولید متن استفاده شوند. به عنوان مثال، مدل های زبانی هوش مصنوعی می توانند از توکن های "این"، "یک"، "جمله" و "است" برای تولید جمله "این یک جمله است" استفاده کنند.

در مدل های زبانی بزرگ (LLM)، توکن ها معمولاً به عنوان واحدهای پایه برای یادگیری استفاده می شوند. LLM ها بر روی مجموعه داده های بزرگی از متن آموزش دیده اند که به صورت توکن های جداگانه تقسیم شده است. این مدل ها سپس می توانند از این توکن ها برای یادگیری الگوهای زبانی و تولید متن استفاده کنند.

توکن ها نقش مهمی در مدل های زبانی هوش مصنوعی ایفا می کنند. آنها واحد پایه ای هستند که مدل ها از آنها برای درک و تولید متن استفاده می کنند.

چند نمونه مدل هوش مصنوعی open source:

بهترین مدل‌های هوش مصنوعی اوپن سورس دسترسی آزاد، مدل‌هایی هستند که از نظر اندازه، کیفیت و کاربردها برتری دارند. برخی از بهترین مدل‌های هوش مصنوعی اوپن سورس دسترسی آزاد عبارتند از:

****GPT-3****: این مدل زبانی بزرگ (LLM) توسط OpenAI توسعه یافته است و دارای 175 میلیارد پارامتر است. GPT-3 می‌تواند طیف گسترده‌ای از وظایف را انجام دهد، از جمله تولید متن، ترجمه زبان، خلاصه‌نویسی و پاسخ به سوالات.

****LaMDA****: این LLM توسط Google AI توسعه یافته است و دارای 137 میلیارد پارامتر است. LaMDA می‌تواند متن را درک کند و به آن پاسخ دهد، و همچنین می‌تواند متن جدیدی تولید کند.

****Turing NLG****: این LLM توسط OpenAI توسعه یافته است و دارای 102 میلیارد پارامتر است. Turing NLG می‌تواند متن را درک کند و به آن پاسخ دهد، و همچنین می‌تواند متن جدیدی تولید کند.

****Megatron-Turing NLG****: این LLM توسط OpenAI و NVIDIA توسعه یافته است و دارای 1.37 تریلیون پارامتر است. Megatron-Turing NLG یکی از بزرگترین LLM های موجود است و می‌تواند طیف گسترده‌ای از وظایف را انجام دهد.

****Chinchilla****: این LLM توسط Google AI توسعه یافته است و دارای 340 میلیارد پارامتر است. Chinchilla می‌تواند متن را درک کند و به آن پاسخ دهد، و همچنین می‌تواند متن جدیدی تولید کند.

این مدل‌ها همگی به صورت منبع باز در دسترس هستند و می‌توانند توسط محققان، توسعه‌دهندگان و سایر علاقه‌مندان به هوش مصنوعی استفاده شوند.

در اینجا برخی از عواملی که باید هنگام انتخاب یک مدل هوش مصنوعی اوپن سورس دسترسی آزاد در نظر بگیرید آورده شده است:

****اندازه****: مدل‌های بزرگتر معمولاً عملکرد بهتری دارند، اما همچنین نیاز به منابع بیشتری برای آموزش و اجرا دارند.

****کیفیت****: کیفیت مدل‌ها را می‌توان با ارزیابی عملکرد آنها بر روی مجموعه داده‌های مختلف اندازه‌گیری کرد.

****کاربردها****: برخی از مدل‌ها برای وظایف خاص مانند تولید متن یا ترجمه زبان بهتر هستند.

۲ ابزار مفید برای توسعه سیستم های هوش مصنوعی در پایتون:

TensorFlow ، PyTorch

پلتفرم «تنسورفلو» (TensorFlow) یک پلتفرم یادگیری ماشین متن باز یکپارچه به حساب می آید. این پلتفرم دارای ابزارها، کتابخانه ها و جامعه ای از منابع کامل و انعطاف پذیر است که به توسعه دهندگان امکان استفاده از پیشرفته ترین منابع یادگیری ماشین را می دهد و می توانند با استفاده از آن پروژه های یادگیری ماشین و هوش مصنوعی خود را ایجاد و پیاده سازی کنند. و توسط گوگل توسعه داده شده است.

HuggingFace: یک مرجع آنلاین برنامه های اوپن سورس مشابه گیتاب است که بطور تخصصی در زمینه هوش مصنوعی و مدل های یادگیری ماشین فعالیت میکند.

[لینک به سایت هاگینگ فیس](#)

مدل هوش مصنوعی فالکن Falcon Language Model

فالکن یک مدل زبانی بزرگ (LLM) است که توسط موسسه نوآوری فناوری (TII) در امارات متحده عربی توسعه یافته است. این مدل بر روی مجموعه داده ای از یک تریلیون توکن متنی آموزش دیده است و دارای 40 میلیارد پارامتر است.

فالکن می تواند طیف گسترده ای از وظایف را انجام دهد، از جمله:

تولید متن

ترجمه زبان

خلاصه نویسی

پاسخ به سوالات

فالکن به صورت **منبع باز** در دسترس است و می تواند توسط محققان، توسعه دهندگان و سایر علاقه مندان به هوش مصنوعی استفاده شود.

ویژگی‌های کلیدی

اندازه: فالکن دارای **40 میلیارد پارامتر** است، که آن را یکی از بزرگترین LLM های موجود می‌سازد.

مجموعه داده: فالکن بر روی مجموعه داده‌ای از یک **تریلیون توکن متنی آموزش دیده** است، که آن را در معرض طیف گسترده‌ای از اطلاعات قرار می‌دهد.

برخی کاربری‌های فالکن:

تولید محتوای خلاقانه، مانند شعر، داستان و کد

ترجمه زبان‌ها

پاسخ به سوالات

خلاصه‌نویسی متن

نحوه استفاده از مدل هوش مصنوعی اوپن سورس:

اگر می‌خواهید از مدل‌های هوش مصنوعی آزاد و منبع باز به جای واسطه API OpenAI GPT استفاده کنید، می‌توانید مدل‌هایی مانند GPT-2 از کتابخانه Hugging Face Transformers یا مدل‌های دیگر از جامعه استفاده کنید. راهنمای ابتدایی به شرح زیر است:

۱. ****انتخاب یک مدل منبع باز:****

مدل‌های منبع باز موجود در پلتفرم‌هایی مانند Hugging Face Transformers را بررسی کنید. GPT-2 به دلیل کارایی و دسترسی به راحتی یک انتخاب محبوب است.

۲. ****نصب کتابخانه‌های مورد نیاز:****

نصب کتابخانه‌های مورد نیاز در پایتون. به عنوان مثال با استفاده از Hugging Face Transformers:

```
bash``
```

```
pip install torch
```

```
pip install transformers
```

```
``
```

۳. ****دانلود مدل:****

مدل پیش‌آموز GPT-2 یا مدل دیگری که انتخاب کرده‌اید را از Hugging Face Transformers دانلود کنید. می‌توانید از مدل‌های کوچک‌تر برای سرعت پاسخ‌های سریع‌تر استفاده کنید، بسته به نیازهای برنامه.

۴. ****ایجاد یک فرم ویندوز:****

واسط گرافیکی خود را با استفاده از چارچوب انتخابی ایجاد کنید، مانند PyQt، Tkinter یا Kivy.

۵. ****ادغام مدل:****

با استفاده از کد مناسب برای کتابخانه انتخابی مدل را بارگیری کنید. برای GPT-2، ممکن است به این شکل باشد:

```
python``
from transformers import GPT2LMHeadModel, GPT2Tokenizer

tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = GPT2LMHeadModel.from_pretrained('gpt2')
``
```

۶. ****پردازش ورودی کاربر:****

ورودی کاربر را از رابط گرافیکی آن بگیرید، آن را توکن‌گذاری کرده و به مدل GPT-2 بدهید. یک پاسخ تولید کنید و آن را به کاربر نمایش دهید.

```
python``
user_input = "User: " + user_input + "\nBot: "
input_ids = tokenizer.encode(user_input, return_tensors='pt')

output = model.generate(input_ids, max_length=150, num_beams=5,
                        no_repeat_ngram_size=2, top_k=50, top_p=0.95)

bot_response = tokenizer.decode(output[0], skip_special_tokens=True)
``
```

۷. ****ارتقاء و بهبود:****

با مدل‌های مختلف، روش‌های توکن‌گذاری و استراتژی‌های گفت‌وگو آزمایش کنید. مدل را بر اساس بازخورد کاربر بهینه‌سازی کنید.

۸. ****نصب و به اشتراک گذاری:****

برنامه خود را برای توزیع بسته‌بندی کنید و آن را به عنوان یک اجرایی به اشتراک بگذارید. اطمینان حاصل کنید که با شرایط مجوز مدل انتخابی خود هماهنگ باشید.

به یاد داشته باشید که در حین استفاده از مدل‌های منبع باز، باید مجوزها و الزامات ارجاع مدل را در نظر بگیرید. همچنین، انجمن‌های پشتیبانی جامعه و مستندات مدل خاصی که انتخاب می‌کنید، منابع ارزشمندی در طول فرآیند توسعه هستند.

4 نوع اجرا شده نرم افزار چت بات - در 4 بستر مختلف

1- محیط ترمینال: کاربر در ترمینالی که کد پایتون اجرا شده ورودی را وارد میکند و خروجی هم در ترمینال پس از بار

گیری چاپ میشود:

```

1 #####
2 #
3 #     CONSOLE - IN LINE TERMINAL     #
4 #
5 #####
6
7 import openai
8 openai.api_key = 'sk-fyGnKGZ2Ic7HQw9AUihXT3lBkFJAZBTF8S3UKQ4LXs0Xbc3'
9 messages = [{"role": "system", "content":
10     "You are an intelligent assistant."}]
11 while True:
12     message = input("User : ")
13     if message:
14         messages.append(
15             {"role": "user", "content": message},
16         )
17         chat = openai.completions.create(
18             model="gpt-3.5-turbo-instruct",
19             prompt=message
20         )
21         reply = chat.choices[0].message.content
22         print(f"ChatGPT: {reply}")
23         messages.append({"role": "assistant", "content": reply})
  
```

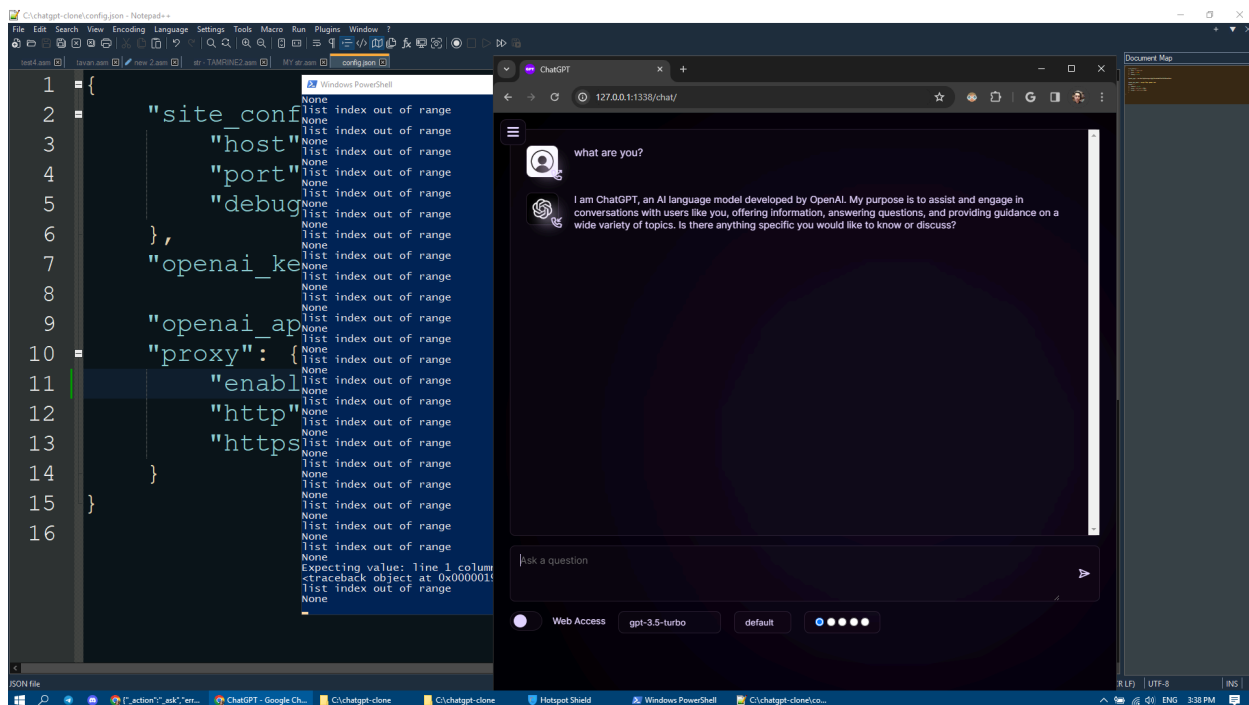
Run: projectbasic x

"E:\Project Files\Aichat\venv\Scripts\python.exe" "E:/Project Files/Aichat/projectbasic.py"

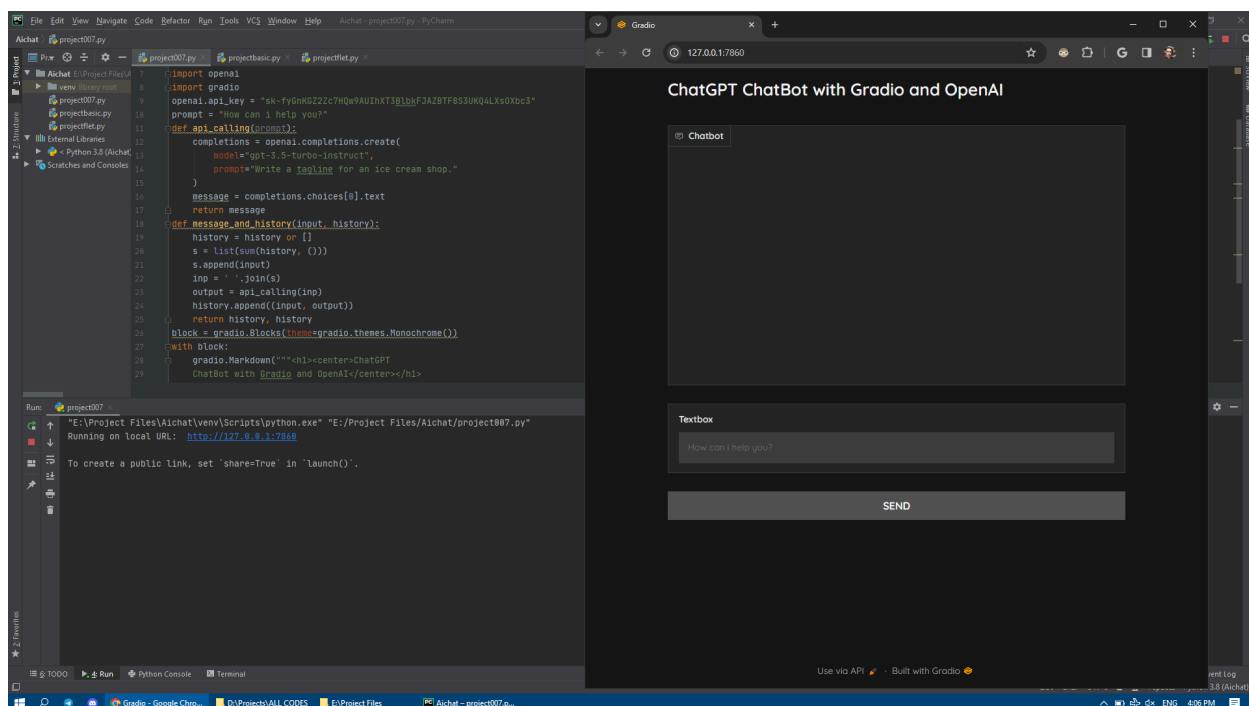
User : Hello my name is Ali

ChatGPT: Hello Ali, how can I help you?

2- وب اپلیکیشن با FLASK و یا DOCKER CONTAINER



3- اینتر فیس وب اپلیکیشن با ابزار GRADIO ، گرادئو مرسوم ترین ابزار برای اجرای محیط های تعاملی مدل های هوش مصنوعی و یادگیری ماشین است. برای دریافت و پردازش و نمایش اطلاعات و خروجی ها به کاربر.



4- نرم افزار ویندوز - PC DESKTOP SOFTWARE با استفاده از لایبری FLET

برای اجرا برنامه روی مرورگر

`ft.app (target=main, port=9000, view=ft.WEB_BROWSER`

برای اجرا بصورت نرم افزار دستکتاپ

`ft.app (target=main`

کار با API جی پی تی:

```

pip install openai
import openai

openai.api_key = 'API_KEY شما'

messages = [ {"role": "system", "content":
               "You are a intelligent assistant."} ]

while True:
    message = input("User : ")
    if message:
        messages.append(
            {"role": "user", "content": message},
        )
        chat = openai.chat.completions.create(
            model="gpt-3.5-turbo",
            messages=messages
        )
        reply = chat.choices[0].message.content
        print(f"ChatGPT: {reply}")
        messages.append({"role": "assistant", "content": reply})

```

متد چت برای گفت و گوی 2 طرفه استفاده میشود.

و با یک لیست آرایه از دیکشنری ها که role و پیغام ها بصورت تاریخچه باید بیایند.

نمونه کار با web scraper - نمونه رایگان سازی شده چت بات:

```
import g4f

g4f.debug.logging = True # Enable debug logging
g4f.debug.check_version = False # Disable automatic version checking
print(g4f.Provider.Bing.params) # Print supported args for Bing

while True:
    user_msg = input('\n    USER: (write your prompt)  ')
    # Using automatic a provider for the given model
    ## Streamed completion
    response = g4f.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": user_msg}],
        stream=True,
    )
    for message in response:
        print(message, flush=True, end='')

    ## Normal response
    response = g4f.ChatCompletion.create(
        model=g4f.models.gpt_4,
        messages=[{"role": "user", "content": user_msg}],
    ) # Alternative model setting

    print(response)
```

برنامه نرم افزار گوگل کروم را در حالت borderless اجرا میکند و از طریق سرویس هایی که بطور رایگان در دسترس هستند ورودی ما را خارج میکند و پاسخ نهایی چت بات های مختلف در بستر وب را جمع آوری و به ما ارجاع میدهد. هم میتوان آن را در محیط کنسول نمایش داد و یا در قالب نرم افزاری همراه با Ui-UX به کاربر تحویل دهیم.