

Configuração do ambiente virtual

Crie um diretório para o seu projeto de *chatbot* em um local de sua escolha no sistema de arquivos da sua máquina. Abra um terminal ou prompt de comando, navegue até o diretório do projeto e crie um ambiente virtual usando o módulo *venv*, conforme a Figura 2:

```
PS C:\> cd .\chatbot\  
PS C:\chatbot> python -m venv venv  
PS C:\chatbot>
```

Figura 2 - Criação do ambiente virtual do Python / Fonte: o autor.

Descrição da Imagem: a figura mostra o comando de criação do ambiente virtual de execução do Python via PowerShell. Fim da descrição.

Ative o ambiente virtual para instalar os pacotes e rodar a aplicação (Figura 3):

```
PS C:\chatbot> .\venv\Scripts\Activate.ps1  
(venv) PS C:\chatbot>
```

Figura 3 - Ativação do ambiente Python / Fonte: o autor.

Descrição da Imagem: a figura mostra o comando de ativação do ambiente virtual do Python via PowerShell. Fim da descrição.

Com o ambiente virtual ativado, você pode instalar o NLTK (Natural Language Toolkit) e outras bibliotecas necessárias para o *chatbot* usando o pip, conforme a Figura 4:

```
(venv) PS C:\chatbot> pip install nltk
Collecting nltk
  Using cached nltk-3.8.1-py3-none-any.whl (1.5 MB)
Collecting click (from nltk)
  Using cached click-8.1.7-py3-none-any.whl (97 kB)
Collecting joblib (from nltk)
  Using cached joblib-1.3.2-py3-none-any.whl (302 kB)
Collecting regex>=2021.8.3 (from nltk)
  Using cached regex-2023.8.8-cp311-cp311-win_amd64.whl (268 kB)
Collecting tqdm (from nltk)
  Using cached tqdm-4.66.1-py3-none-any.whl (78 kB)
Collecting colorama (from click->nltk)
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: regex, joblib, colorama, tqdm, click, nltk
Successfully installed click-8.1.7 colorama-0.4.6 joblib-1.3.2 nltk-3.8.1 regex-2023.8.8 tqdm-4.66.1
```

Figura 4 - Instalação do pacote NLTK / Fonte: o autor.

Descrição da Imagem: a figura mostra o comando de instalação do NLTK via PowerShell e as dependências que estão sendo baixadas. Fim da descrição.

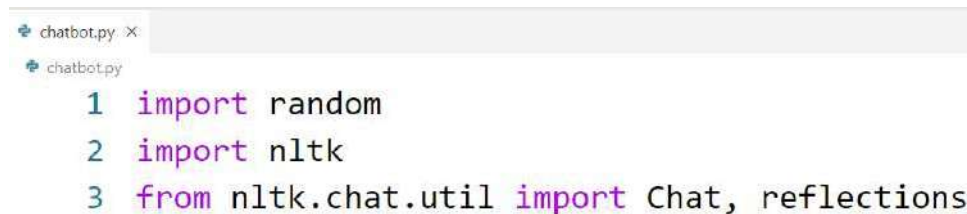
O NLTK requer dados adicionais para funcionar corretamente, então, vamos à instalação, conforme a Figura 5:

```
(venv) PS C:\chatbot> python -m nltk.downloader all
<frozen runpy>:128: RuntimeWarning: 'nltk.downloader' found in sys.modules after import of package 'nltk', but prior to
execution of 'nltk.downloader'; this may result in unpredictable behaviour
[nltk_data] Downloading collection 'all'
[nltk_data] |
[nltk_data] | Downloading package abc to
```

Figura 5 - Instalação dos complementos do pacote NLTK / Fonte: o autor.

Descrição da Imagem: a figura mostra o comando de instalação dos complementos necessários do NLTK via PowerShell. Fim da descrição.

Agora que você tem o ambiente configurado, vamos iniciar a criação do *chatbot*. Crie um arquivo Python no diretório do projeto (por exemplo, *chatbot.py*). Para esta etapa, pode-se utilizar o Visual Studio Code. Escrever o código do seu *chatbot* usando a biblioteca NLTK, para isso, vamos começar importando as dependências conforme a Figura 6.



```

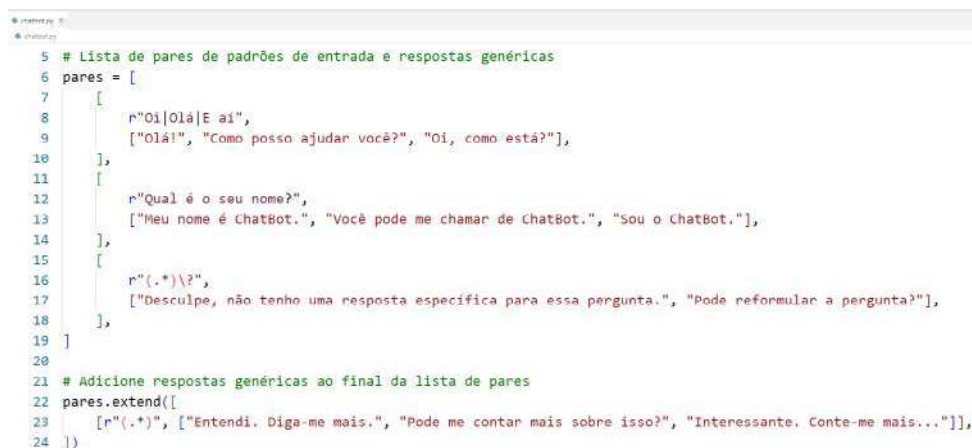
1 import random
2 import nltk
3 from nltk.chat.util import Chat, reflections

```

Figura 6 - Imports do *chatbot* / Fonte: o autor.

Descrição da Imagem: a figura mostra a importação das bibliotecas para criação do *chatbot*. Fim da descrição.

Após isso, vamos criar a base de dados do nosso *chatbot*. Esta base será criada considerando as possibilidades cujo usuário irá se cumprimentar, pode perguntar o nome do *chatbot* ou qualquer outra pergunta. Além disso, pode ter respostas genéricas, conforme a Figura 7.



```

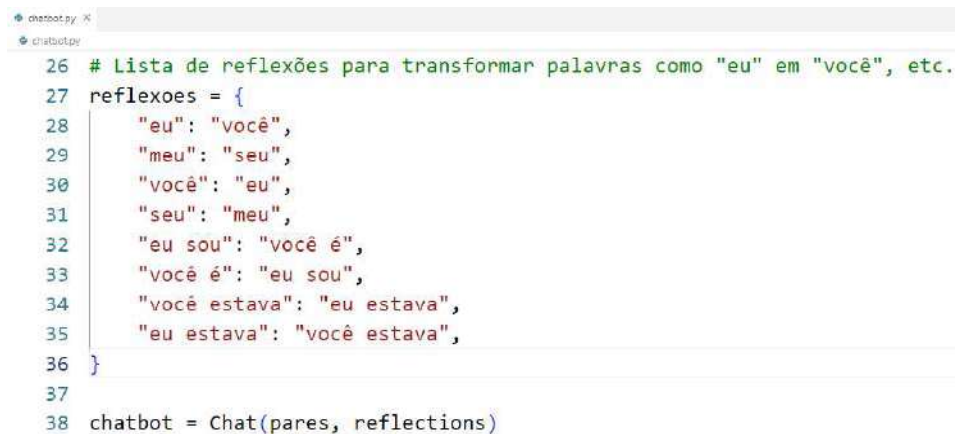
5 # Lista de pares de padrões de entrada e respostas genéricas
6 pares = [
7     [
8         r"Oi|Olá|E aí",
9         ["Olá!", "Como posso ajudar você?", "Oi, como está?"]],
10    ],
11    [
12        r"Qual é o seu nome?",
13        ["Meu nome é ChatBot.", "Você pode me chamar de ChatBot.", "Sou o ChatBot."],
14    ],
15    [
16        r"(.*)\?",
17        ["Desculpe, não tenho uma resposta específica para essa pergunta.", "Pode reformular a pergunta?"]],
18    ],
19 ]
20
21 # Adicione respostas genéricas ao final da lista de pares
22 pares.extend([
23     [r"(.*)", ["Entendi. Diga-me mais.", "Pode me contar mais sobre isso?", "Interessante. Conte-me mais..."]],
24 ])

```

Figura 7 - Base de conhecimento do *chatbot* / Fonte: o autor.

Descrição da Imagem: a figura mostra a criação da base de dados do *chatbot*. Fim da descrição.

Antes de associarmos essa base com o *chatbot*, podemos definir reflexões para transformar o sujeito da frase e aí sim vincular essa base ao *chatbot*, conforme a Figura 8.



```

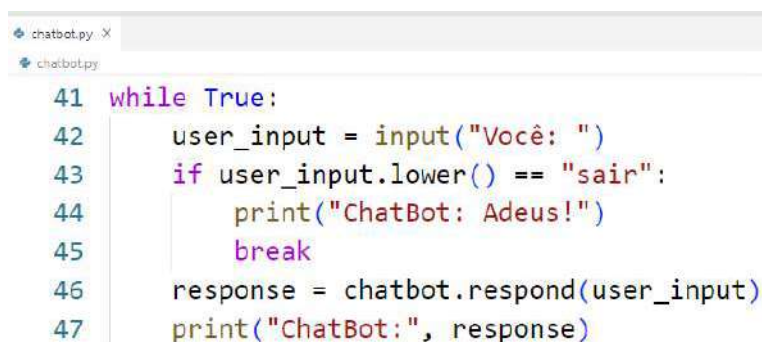
26 # Lista de reflexões para transformar palavras como "eu" em "você", etc.
27 reflexoes = {
28     "eu": "você",
29     "meu": "seu",
30     "você": "eu",
31     "seu": "meu",
32     "eu sou": "você é",
33     "você é": "eu sou",
34     "você estava": "eu estava",
35     "eu estava": "você estava",
36 }
37
38 chatbot = Chat(pares, reflections)

```

Figura 8 - Reflexões e criação do *chatbot* / Fonte: o autor.

Descrição da Imagem: a figura mostra a criação do *chatbot* e das reflexões. Fim da descrição.

Por fim, vamos criar uma função em loop capturando o que o usuário informou e apresentando as respostas do *chatbot*, conforme a Figura 9.



```

41 while True:
42     user_input = input("Você: ")
43     if user_input.lower() == "sair":
44         print("ChatBot: Adeus!")
45         break
46     response = chatbot.respond(user_input)
47     print("ChatBot:", response)

```

Figura 9 - Função de loop do *chatbot* / Fonte: o autor.

Descrição da Imagem: a figura mostra a função de loop do *chatbot*, em que aguarda a entrada do usuário e imprime a saída do bot. Fim da descrição.

Agora é só testar via terminal (Figura 10).

```
(venv) PS C:\chatbot> python.exe .\chatbot.py
Olá! Eu sou um chatbot. Digite 'sair' para encerrar a conversa.
Você: Oi
ChatBot: Oi, como está?
Você: Bem e você?
ChatBot: Desculpe, não tenho uma resposta específica para essa pergunta.
Você: Eu estou feliz
ChatBot: Pode me contar mais sobre isso?
Você: Eu ganhei um presente
ChatBot: Interessante. Conte-me mais...
Você: sair
ChatBot: Adeus!
```

Figura 10 – Conversando com o *chatbot* / Fonte: o autor.

Descrição da Imagem: a figura mostra um exemplo de conversa entre o usuário e o *chatbot* que foi criado no exemplo. Fim da descrição.

DISCUTIR DESAFIOS E CONSIDERAÇÕES ÉTICAS

A tecnologia de PLN tem avançado significativamente nas últimas décadas, abrindo portas para uma ampla gama de aplicações e possibilidades. No entanto, também trazem preocupações importantes que merecem atenção à medida que os sistemas de PLN se tornam mais poderosos e difundidos.

A capacidade dos sistemas de PLN de entender e gerar linguagem humana tem aplicações impressionantes, desde *chatbots*, que fornecem suporte ao cliente, até assistentes virtuais, que automatizam tarefas complexas. No entanto, essa mesma capacidade levanta questões cruciais sobre privacidade, viés, responsabilidade e o impacto social da tecnologia.

Em primeiro lugar, a privacidade é uma preocupação fundamental quando se trata de PLN. À medida que os sistemas processam grandes volumes de dados de texto, eles podem inadvertidamente acessar informações pessoais e confidenciais. A coleta e o uso indevidos de dados levantam preocupações significativas, especialmente quando os sistemas de PLN são usados em aplicações que envolvem a comunicação sensível do usuário e estão respaldadas pela Lei Geral de Proteção de Dados (LGPD).