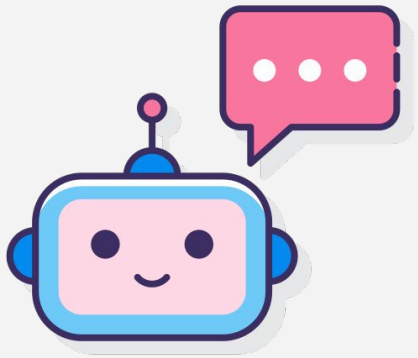
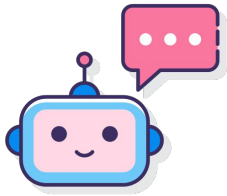


INTELIGÊNCIA ARTIFICIAL
CHATBOT / NLP - NLU

Adilmar Coelho Dantas

Arquitetura e Engenharia



🎯 Ementa

✔Introdução ao conceito de chatbots

✔Introdução a inteligência artificial (NLP)

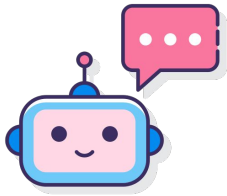
✔Conceitos de intenção/entidade

✔Introdução ao processo de curadoria

✔Introdução aos canais digitais

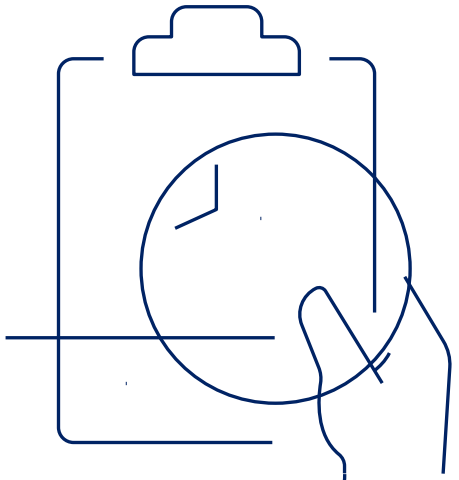
✔Mão na massa! Meu primeiro **ChatBot** em Python.

🎯 Requisitos

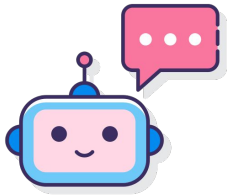


✔Python +3.8 , PIP 3, ou acesso admin para instalar durante o curso para a parte prática.

✔Acessar o google colab: https://colab.google/



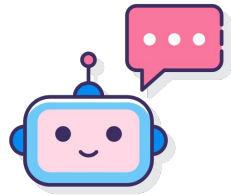
🧩 Definições



- ✔ Um **chatbot** é uma uma aplicação baseada em Inteligência Artificial capaz de manter uma conversa em tempo real por texto ou por voz. No primeiro caso, temos os chatbots de atendimento ao cliente que podemos encontrar em sites de bancos, seguros, viagens, restaurantes etc.No segundo, nos referimos aos famosos assistentes virtuais — Siri, Irene, Cortana ou Alexa que tentam tornar nossa vida mais fácil respondendo às nossas perguntas diariamente.

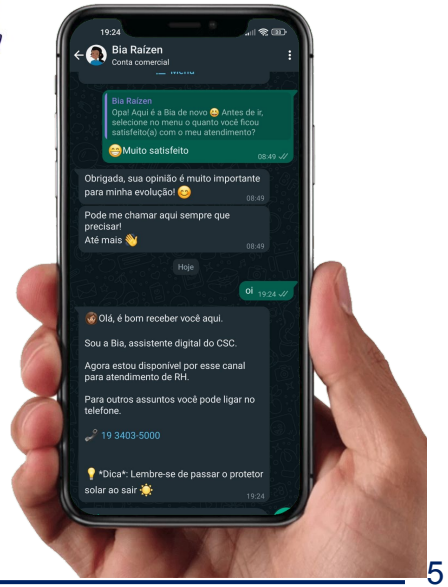
- ✔ Tudo isso é possível por meio de uma área da inteligência artificial aplicada nos chatbot, a NLP do português processamento de linguagem natural (PLN).

🧠 O que é NLP?

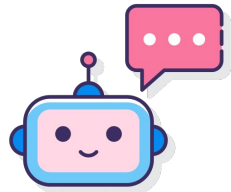


- ✔ O **Processamento de Linguagem Natura**l ou NLP (do inglês Natural Language Processing) é o nome dado aos estudos e às tecnologias relacionados à interação entre computadores e linguagens naturais do homem. Portanto, é uma área ligada à interação entre homens e computadores, por meio dos idiomas falados pelas mais diversas culturas.

- ✔ Podemos resumir esse conhecimento como sendo “ensinar sistemas a interpretar e reagir as “linguagens humanas”.



🧪Como ela funciona?



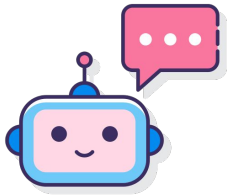
- ✔ **Morfológico:** se concentra em distinguir os diferentes tipos de palavras (verbos, substantivos, preposições, etc.) e suas variações (gênero, número, tempo, etc.).

- ✔ **Sintático:** seleciona algumas frases de outras e analisa as partes que as compõem (sujeito, verbo, predicado) para assim poder retirar seu sentido.

- ✔ **Semântico:** analisa o significado não só das palavras individuais, mas também das frases das quais elas fazem parte e do discurso em seu conjunto.

- ✔ **Pragmático:** encarrega-se de encontrar a intenção do texto dependendo de seu contexto, permitindo diferenciar fatores tais como ironia, ambiguidade ou estado de ânimo.

🧪Onde ela é aplicada?



✔ ***Classificação de documentos**: A tarefa de classificar grandes quantidades de documentos conforme seu assunto ou estilo pode ser agilizada com sistemas de PLN.*

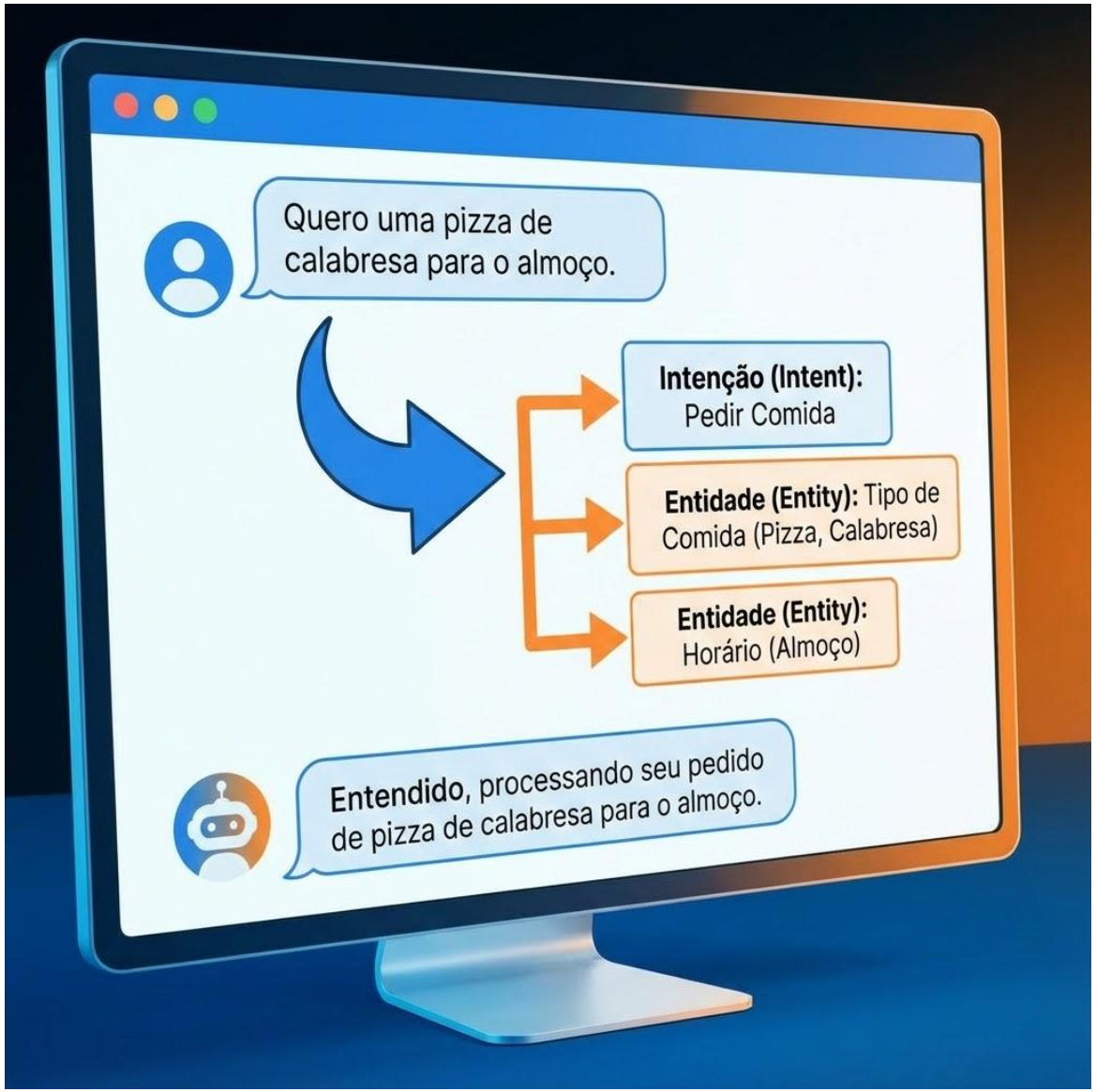
✔ ***Análise de sentimentos e da opinião**: Os comentários nas redes sociais sobre produtos ou serviços são muito relevantes para as empresas e os sistemas de PLN permitem retirar informações relevantes.*

✔ ***Comparação de textos**: Os sistemas de PNL permite encontrar padrões em textos e localizar coincidências entre eles, o que facilita a detecção de plágios e o controle de qualidade.*

✔ ***Anonimização de documentos**: Através de sistemas PNL podem ser processados documentos para identificar e eliminar as menções de dados pessoais, certificando assim a privacidade das pessoas e instituições.*

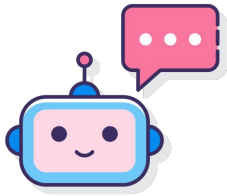
✔

🧪 Intenção e Entidades



- ✔ **Você deve estar se perguntando**
- ✔ **“Mas como o chatbot ou agente consegue entender o que é questionado a ele?”**
- ✔ **A resposta é: através do processo de treinamento, onde é ensinado ao mesmo como identificar as entidades (sujeitos) e as intenções (estímulos/perguntas) feitas para o Chatbot.**

Intenção e Entidades



✔ **Intenção:**

É o estímulo do enviado ao chatbot no qual o usuário aguarda uma resposta.

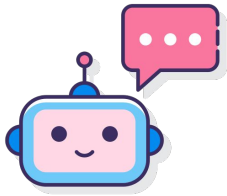
Ex: “Desbloquear senha da VPN**”, “Desbloquear senha** de Rede**”, “Baixar** o holerite de Março**”**

✔ **Entidade:**

Uma entidade pode ser considerada uma unidade de informação que representa um certo tipo de assunto do mundo real, como um número de telefone, CEP, cidade ou mesmo o nome de uma pessoa.

Ex: “Desbloquear senha da **VPN”,** “Desbloquear senha de **Rede”**

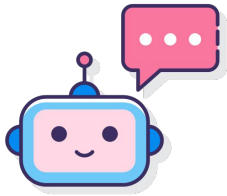
🧪 Curadoria



A curadoria é o meio que possibilita a identificação de “estímulos” que a solução de chatbot ainda não é **capaz de compreender** e a partir dessa análise evoluir essa maturidade das respostas, principalmente nos bot’s de NLP (Natural Language Processing) – toda intenção precisa ter um número considerável de frases de treinamento para não dar conflito com as demais.

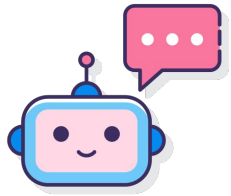
Além disso esse processo de curadoria envolve reuniões com as áreas de negócio no que chamamos de imersão, onde o curador tem como objetivo **levantar todas as oportunidades de fluxos** conversacionais e realizar o desenho desses fluxos para que ele seja passado para o time de tecnologia responsável pela sua implementação.

📍 Etapas de curadoria



- ✔ *Etapas de um curador(a) de Chatbot*
- ✔ *Reuniões com o negócio;*
- ✔ *Desenho dos fluxos conversacionais;*
- ✔ *Levantamento dos estímulos (frases, palavras chaves) do chatbot;*
- ✔ *Repasse com o time responsável pelo desenvolvimento;*
- ✔ *Validação das implementações – testes funcionais;*
- ✔ *Validação junto ao negócio;*
- ✔ *Acompanhamento contínuo*
- ✔ *Análise de interação, atendimentos e procura de oportunidades de melhoria da solução.*

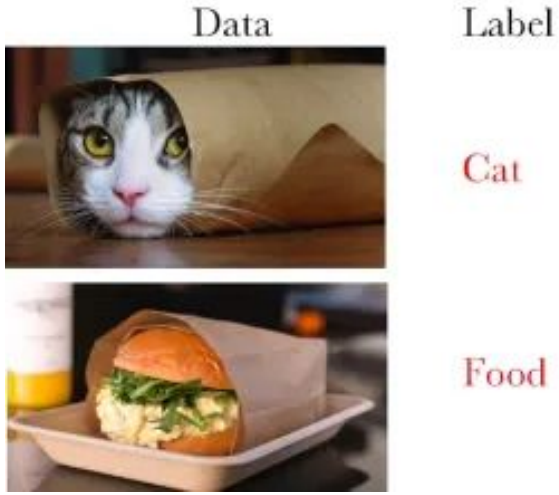
📍 Treinamento: Supervisionado



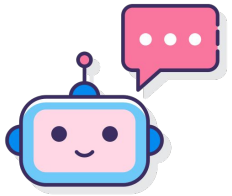
- ✔ **Treinamento supervisionada:** Nesse modelo passamos um conjunto de dados rotulados, recebendo assim o feedback do que é e não é correto, tendo que aprender apenas o mapeamento entre os labels e os dados.

Nesse modelo o chatbot é treinado por um conjunto de dados com seus respectivos rótulos.

- ✔ **Vantagens**: Refinamento de dados;
- ✔ **Desvantagens**: Tempo para processar grandes volumes de dados, por se tratar de um processo manual.



📍 Treinamento: Não Supervisionado

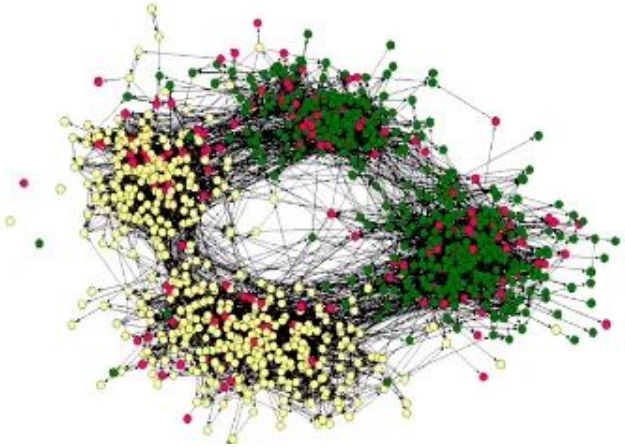


- ✔ **Treinamento semi supervisionado:** Nesse modelo além dos dados supervisionados passamos um conjunto de dados **sem rótulos**, além de não receber nenhum **feedback** sobre o que é ou não correto, tendo que aprender por si só, sendo o mais indicado uma vez que nem sempre temos a base bem formada com rótulos.

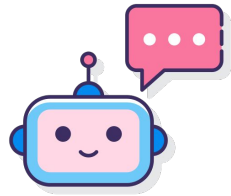
- ✔ **Treinamento não supervisionado**: Nesse modelo o chatbot é treinado por um conjunto de dados com seus respectivos rótulos e com o auxílio de técnicas de inteligência artificial essas informações são atualizadas de maneira automática ou para monitoramento.

- ✔ **Vantagens**: Refinamento de dados; treinamento de grande volumes de dados; Redução de tempo.

- ✔ **Desvantagens**: Gastos com recursos computacionais.

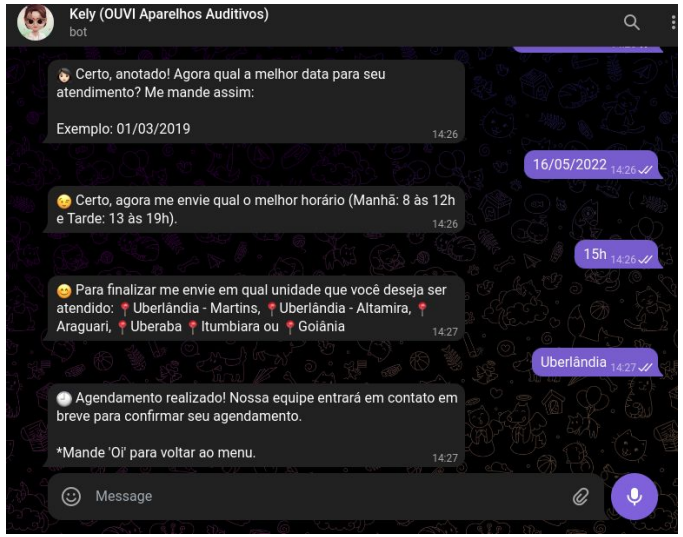


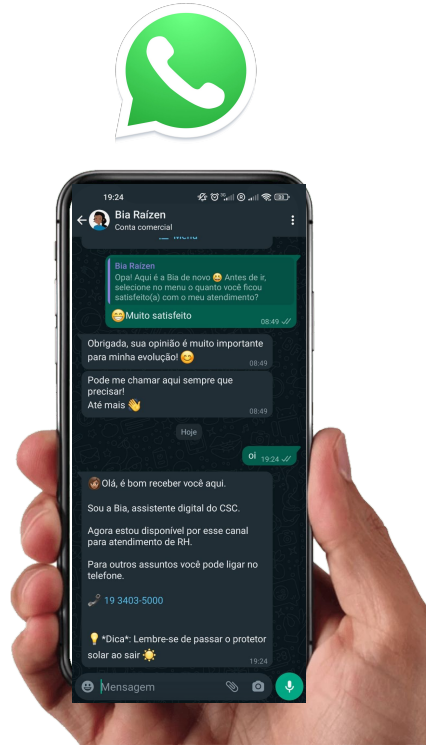
📍 Canais Digitais



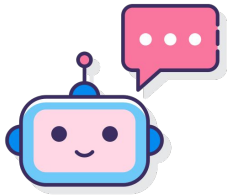
- ✔ **Canais digitais:** É denominado canal digital todo e qualquer meio pelo qual é possível enviar e receber informações com o chatbot.

- ✔ **Exemplo de canais digitais**: WebChat, Telegram, Teams, WhatsApp, Messenger,etc.





📍 Prática: NLTK



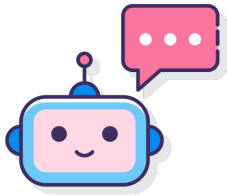
- ✔ Nesse tutorial iremos aprender mais sobre a biblioteca NLTK para o processamento natural de linguagem.

- ✔ O NLTK irá ajudá-lo com tudo, desde dividir sentenças de parágrafos, dividir palavras, reconhecer a parte da fala dessas palavras, destacar os principais assuntos e, depois, até mesmo ajudar sua máquina a entender o conteúdo do texto.

- ✔ Nesse treinamento, vamos abordar o campo da mineração de intenções, comparando a similaridade de sentenças em banco de dados como, por exemplo, em um atendimento realizado por canal digital.

- ✔ O primeiro passo é instalar a biblioteca por meio do comando: **pip install nltk**

📍 Prática: NLTK



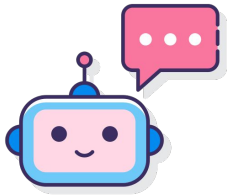
- ✔ Após a instalação precisamos entender alguns pacotes, como corpus, **lexicon**, **tokenizers**, **chunkers** da NLTK que iremos utilizar nos exercícios práticos a seguir.

- ✔ **Corpus**: corpo do texto como, por exemplo, uma coleção de revistas.

- ✔ **Lexicon**: conjunto de palavras e seus significados, exemplo um dicionário.

- ✔ **Tokenizers**: cada palavra de uma frase é um “token” ou “entidade”dividida baseada em regras específicas.

📍 Prática: Google Colab

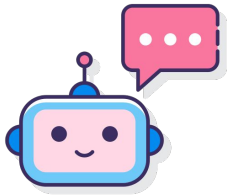


- ✔ Com o objetivo de deixar a parte prática um pouco mais simples, vamos utilizar uma ferramenta chamada Google Colab.

- ✔ https://colab.research.google.com

- ✔ O Google Colab é um ambiente virtual gratuito mantido pela Google, voltado para o desenvolvimento em Python para ciência de dados, ML e aprendizagem em geral.

📍 Prática: Token, Tokenizar

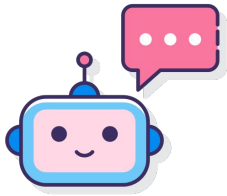


- ✔ Essas são algumas das palavras que vocês irão ouvir muito na área de NLP, e outras que virão mais adiante. Então para começar vamos Tokenizar uma frase e ver como isso funciona.

```
1  from nltk.tokenize import sent_tokenize, word_tokenize  #import lib nltk
2
3  texto  = "Oi Adilmar tudo bem ? Estou com problemas na service desk pode me ajudar a reso
4  texto2 = "Sim claro! me informe por gentileza seu usuário, senha e token."
5
6  #tokenize texto,texto2
7  print(sent_tokenize(texto))
8  print(sent_tokenize(texto2))
```

token.py hosted with ❤ by **GitHub** **view raw**

📍 Prática: Stop Words

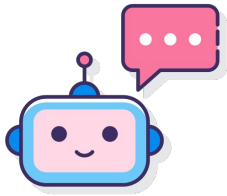


- ✔ Você vai notar que ele trouxe pontuação como palavras, nosso próximo passo agora será tratar esses elementos como “**stop words**” ou palavras inúteis ou de parada.

```python
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from string import punctuation

example_sent = "Oi Adilmar tudo bem ? Estou com problemas na service desk pode me ajudar

stop_words = set(stopwords.words('portuguese') + list(punctuation))

word_tokens = word_tokenize(example_sent)

filtered_sentence = [w for w in word_tokens if not w in stop_words]

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

#print(word_tokens)
print(filtered_sentence)
```

stop_words.py hosted with ❤ by GitHub view raw

📍 Prática: Cosine Similarity



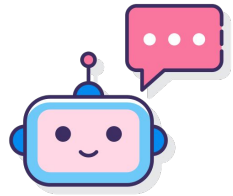
- ✔ Agora vamos imaginar seguinte situação “Você tem uma base de conversas de atendimento da sua empresa e gostaria de agrupá-las pelo mesmo assunto “intenção”, claro da pra fazer isso na mão mas já imaginou fazer isso com > 100k de sentenças, vamos ver agora como utilizar (NLP) para resolver esse problema.

- ✔ Para comparar sentenças precisamos de uma função para **vetorizar** essas frases e compará-las por meio de uma **função matemática como cossenos** .

- ✔ Bom, para isso teremos que importar outras bibliotecas e vetorizar nossas frases conforme o código a seguir.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

📍 Prática: Cosine Similarity

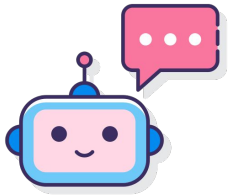


✔ Agora você pode usar esse conhecimento para pegar uma base de dados grande e agrupar os diálogos conforme as similaridades as separando em **intenções em comum**.

```python
import nltk, string
from sklearn.feature_extraction.text import TfidfVectorizer

stemmer = nltk.stem.porter.PorterStemmer()
remove_punctuation_map = dict((ord(char), None) for char in string.punctuation)

def stem_tokens(tokens):
    return [stemmer.stem(item) for item in tokens]

'''removendo pontuação'''
def normalize(text):
    return stem_tokens(nltk.word_tokenize(text.lower().translate(remove_punctuation_map)

vectorizer = TfidfVectorizer(tokenizer=normalize)

def cosine_sim(text1, text2):
    tfidf = vectorizer.fit_transform([text1, text2])
    return ((tfidf * tfidf.T).A)[0,1]


print (cosine_sim('eu tenho um celular da y', 'eu tenho um celular da y'))
print (cosine_sim('um celular', 'um celular da marca x é meu'))
print (cosine_sim('um celular', 'uma maquina de lavar'))
```

cosine.py hosted with ❤ by GitHub view raw

📍 Prática: Construindo meu primeiro chatbot



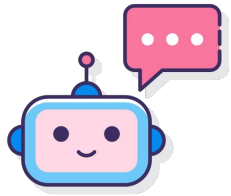
- ✔ Para construção do chatbot será utilizado a biblioteca do Python chamada **Chatterbot**. O Chatterbot usa **Machine Learning** para aprender a partir das interações dos usuários, ou seja, quanto mais você "conversa" com o chatbot, mais ele irá aprender.

- ✔ O Chatterbot é uma biblioteca que utiliza diversos processos de Machine Learning (ML) para gerar respostas automáticas para o input do usuário.

- ✔ Essa solução não vem treinado, ou seja, não sabe interagir numa conversa. O que ele faz é "tomar nota" de todas as conversas que ele tem e assinalar as respostas para cada uma das perguntas.

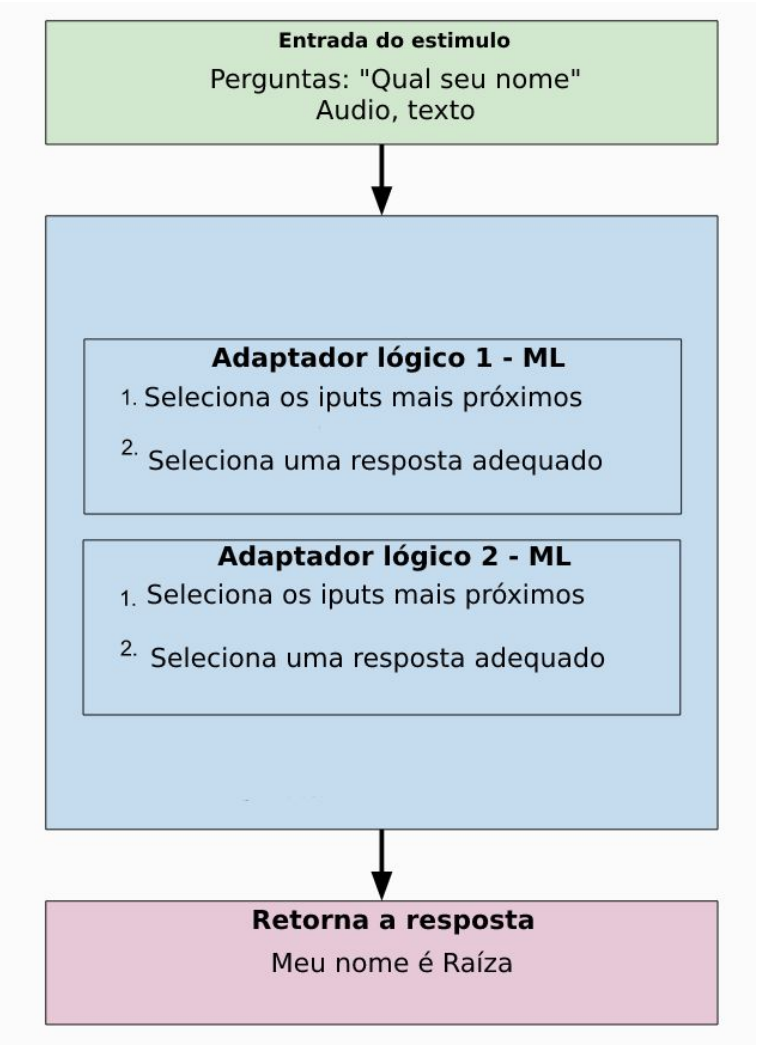
- ✔ A vantagem disso é que ele pode ser usado em qualquer idioma e em qualquer contexto.

📍 Prática: Construindo meu primeiro chatbot

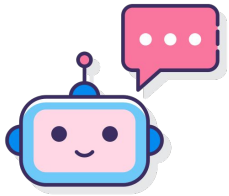


- ✔ Para essa atividade prática, você pode escolher fazer um bot que irá funcionar como assistente pessoal, um atendente de Uma pizzaria, etc. No exemplo aqui iremos fazer um **chatbot** assistente da CNP Seguradora.

- ✔ Para a construção do nosso chatbot, vamos começar com a criação de um arquivo **.txt**, contendo as **intenção de saudação** tais como: **bom dia, oi, boa tarde, qual seu nome?, etc**.



📍 Prática: Construindo meu primeiro chatbot



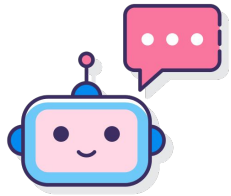
- ✔ No segundo arquivo iremos colocar assuntos específicos do qual o seu chatbot irá tratar, por exemplo, assuntos relacionados a CNP Seguradora.

- ✔ Finalizando a construção dos dois arquivos acesse o **Google Colab** e comece um novo projeto.

```
pi
Olá!
eai
Oi, tudo bem?
opa
Olá! tudo bem?
Bom dia
Bom dia, como posso te ajudar?
Boa tarde
Bom tarde, como posso te ajudar?
Boa noite
Bom noite, como posso te ajudar?
boa noite
```

```
Como surgiu a Raízen
A Raízen é uma joint venture, criada a par
Qual o propósito da Raízen
Realizamos agora olhando o futuro. - Ampli
Qual ano de fundação da raízen
A Raízen foi fundada em 2011
Qual o ano do IPO da raízen
2021
Quem é o presidente da Raízen
O presidente da Raízen é Ricardo Mussa.
Quantas pessoas trabalham na raízen
Pouco mais de 40 mil pessoas.
Qual os segmentos de negócio da raízen
```

📍 Prática: Instalando e importando bibliotecas



Instalando biblioteca

```
pip install chatterbot==1.0.4
```

Definindo versão

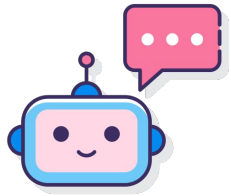
Realizando import das libs

```
from chatterbot import ChatBot
chatbot = ChatBot("raiza")

from chatterbot.trainers import ListTrainer

import re
```

Importando a lib instalada

Definindo o nome do chatbot

📍 Prática: Criando array de treinamento

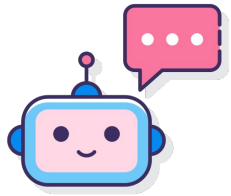


```
conversation = [
    "oi",
    "Olá!",
    "eai",
    "Olá eu sou a Raíza!",
    "opa",
    "Olá! tudo bem?",
    "bem e você?",
    "tudo bem?",
    "bem e você?",
    "Estou otima obrigado!",
    "O que você faz",
    "Eu sou o chatbot da Raízen",
    "Que legal",
    "Muito obrigado!",
    "qual seu nome?",
    "Raíza",
    "bem",
    "bem e você?",
    "Que bom que está bem!",
    "você mora onde",
    "Na internet"
]
```

Definindo o array

Preenchendo o mesmo com perguntas e respostas

📍 Prática: Criando arquivo de treinamento



Treinamento por meio de arquivo

```
1 f = open('/content/conversas.txt','r')
2
3 conversation = []
4
5 for line in f:
6     m = re.search('(Q:|A:)?(.+)', line)
7     if m:
8         conversation.append(m.groups()[1])
9
10 print(conversation)
```

Definindo o caminho e abrindo o arquivo

Criando o array para armazenar as intenções

Regex para captura linha a linha

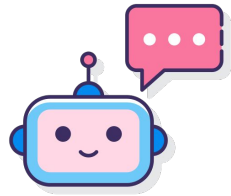
Adicionando as linhas ao nosso array

Treinando o chatbot

```
1 trainer = ListTrainer(chatbot)
2 trainer.train(conversation)
```

Criando a instrução de treinamento e treinando

📍 Prática: Construindo meu primeiro chatbot



```
1 while True:
2     pergunta = input("👤Usuário: ")
3     resposta = chatbot.get_response(pergunta)
4     #print(resposta.confidence)
5     if float(resposta.confidence) > 0.5:
6         print('🤖Raíza: ', resposta)
7     else:
8         print('🤖Raíza: Ainda não sei responder esta pergunta')
```

Loop para iniciar a conversa

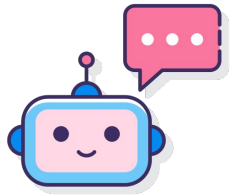
Envia o estímulo para o chatbot

Verifica acurácia da classificação

Envia resposta ao usuário

```
👤Usuário: Bom dia
🤖Raíza:  Bom dia, como posso te ajudar?
👤Usuário: O que você faz?
🤖Raíza:  Eu sou o chatbot da Raízen
👤Usuário: Qual o ano de fundação da Raízen?
🤖Raíza:  A Raízen foi fundada em 2011
👤Usuário:
```

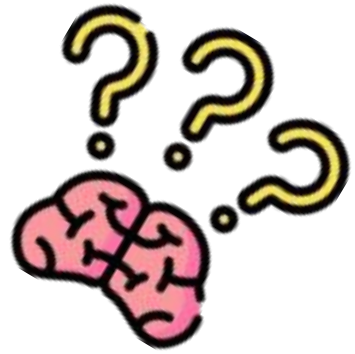
📍 Prática: Curiosidades



Quanto mais o chatbot for utilizado **mais inteligente** ele irá ficar, como estamos usando uma solução de ML, o mesmo possui algoritmos específicos para realizar **novos treinamentos** a cada nova interação.

Isso pode ser facilmente percebido nos arquivos dos projetos: são criados arquivos .sqli contendo as interações para serem utilizadas em novos treinamentos.

Você pode criar dentro do seu chatbot, operações logicas como chamadas de api de previsão do tempo, etc.



Agora que você já sabe como criar seu próprio chatbot, você pode pesquisar como integrá-lo em algum **canal digital** ou aguardar até a parte dois desse mini curso :)

Obrigado!