

Guilherme de Abreu – 222220287

Kaique Fernandes – 222210114

[Link do vídeo](#)

Figura com o Diálogo proposto

```
[nltk_data] C:\Users\Guilherme\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data] C:\Users\Guilherme\AppData\Roaming\nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!
2025-02-17 19:56:08.440145: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Modelo carregado com sucesso.
Bem-vindo ao Chatbot de suporte ao PIPE!
Você: Quanto posso receber no PIPE?
1/1 ██████████ 0s 91ms/step
Chatbot: Fase 1: destinada à demonstração da viabilidade técnica da pesquisa, com duração de até 9 meses. [financiamento]
Você: Qual é o valor do financiamento?
1/1 ██████████ 0s 56ms/step
Chatbot: Os valores e condições específicas de financiamento podem variar; recomenda-se consultar as normas atualizadas no site da FAPESP. [financiamento]
Você: O PIPE oferece bola?
1/1 ██████████ 0s 55ms/step
Chatbot: O PIPE oferece financiamento em diferentes fases: [financiamento]
Você: Como o projeto é avaliado?
1/1 ██████████ 0s 60ms/step
Chatbot: A avaliação das propostas considera critérios como: grau de inovação tecnológica, viabilidade técnica e econômica, potencial de mercado e capacidade da equipe executora. [critérios_
avaliacao]
Você: Quais são os critérios de avaliação?
1/1 ██████████ 0s 58ms/step
Chatbot: Detalhes completos sobre os critérios de avaliação podem ser encontrados nas normas do PIPE no site da FAPESP. [critérios_avaliacao]
Você: Quais documentos são necessários para o PIPE?
1/1 ██████████ 0s 55ms/step
Chatbot: É importante consultar as normas específicas do PIPE disponíveis no site da FAPESP para obter detalhes completos sobre a documentação necessária. [documentos]
Você: O que preciso enviar para o PIPE?
1/1 ██████████ 0s 62ms/step
Chatbot: Os principais documentos exigidos incluem: plano de pesquisa detalhado, orçamento completo, informações sobre a empresa e a equipe de pesquisa. [documentos]
Você: Que documentos devo incluir na proposta?
1/1 ██████████ 0s 61ms/step
Chatbot: Os principais documentos exigidos incluem: plano de pesquisa detalhado, orçamento completo, informações sobre a empresa e a equipe de pesquisa. [documentos]
Você: █
```

Dependências

O chatbot depende das seguintes bibliotecas:

- numpy
- json
- pickle
- nltk
- ssl
- random
- keras
- tensorflow

Para garantir que todos os pacotes do NLTK estejam disponíveis, o script faz o download de:

- punkt
- wordnet

- omw-1.4
- punkt_tab

Estrutura do Código

Inicialização do Chatbot

A classe ChatBot é responsável por criar e gerenciar o modelo de IA. Na inicialização, ela define:

- Uma lista de palavras-chave (words)
- Classes de intenções (classes)
- Lista de documentos processados (documents)
- Lista de intenções (intents)
- O modelo (model)
- Palavras ignoradas (ignore_words)
- Um lematizador para normalização de palavras (lemmatizer)

Criação do Modelo (createModel)

1. Carrega os dados do arquivo intents.json.
2. Tokeniza e processa os padrões de cada intenção.
3. Gera um conjunto de treinamento com:
 - Vetores Bag of Words (BoW)
 - Saídas categóricas correspondentes
4. Cria um modelo de rede neural com keras, incluindo:
 - Camada de entrada
 - Duas camadas densas ocultas com relu e Dropout
 - Camada de saída com softmax
5. Compila e treina o modelo usando o otimizador SGD.
6. Salva o modelo treinado em chatbot_model.keras.

Carregamento do Modelo (loadModel)

Se um modelo treinado já existir, ele é carregado a partir de `chatbot_model.h5`. O script também carrega as palavras e classes previamente salvas nos arquivos `words.pkl` e `classes.pkl`.

Processamento de Entrada e Predição

1. A função `clean_up_sentence` tokeniza e lematiza a entrada do usuário.
2. `bow` transforma a frase em um vetor BoW.
3. `predict_class` passa esse vetor pelo modelo para prever a intenção correspondente.
4. `getResponse` retorna uma resposta apropriada baseada na intenção detectada.

Interface de Usuário

O chatbot é executado em um loop que:

1. Aguarda uma entrada do usuário.
2. Prediz a intenção.
3. Exibe uma resposta.
4. Se a intenção for "despedida", finaliza o programa.

Uso

1. Execute o script.
2. Interaja com o chatbot digitando perguntas.
3. O chatbot responderá com base nas intenções definidas em `intents.json`.
4. Para encerrar, digite uma despedida reconhecida pelo modelo.