

CREATE A CHATBOT IN PYTHON

Team Leader:

962821106033 : JERRIK PRAYWIN RAJ J

Team Members:

962821106004 : ABINANDH R S

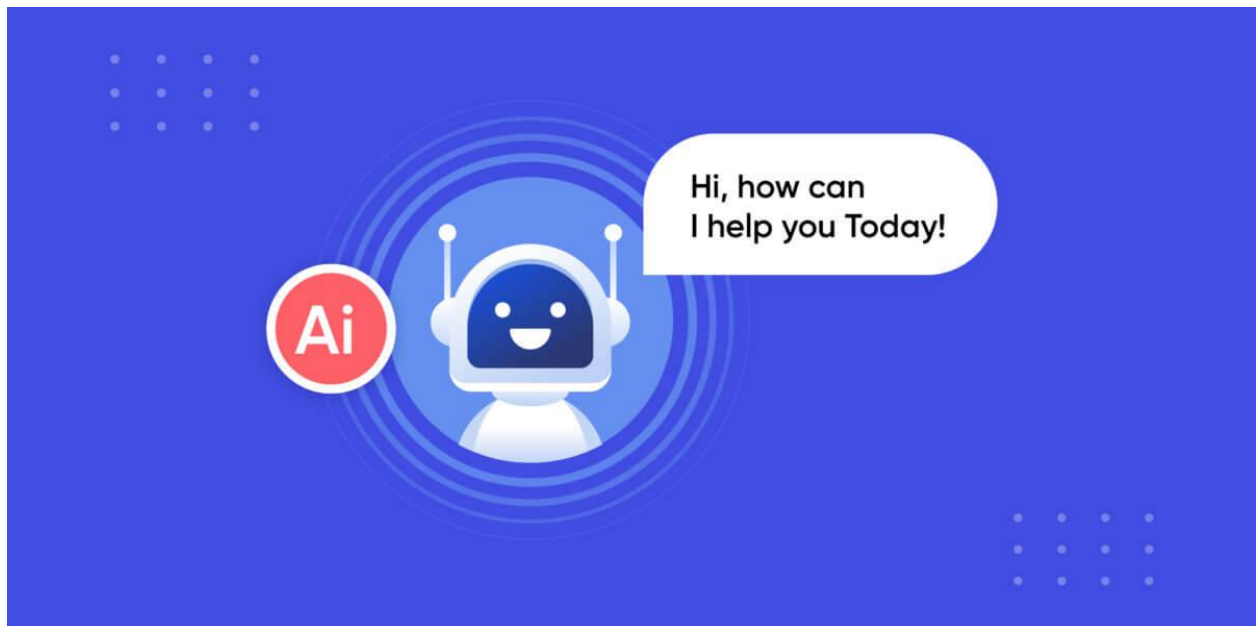
962821106014 : ARSHATH SHARIEF S

962821106017 : BALASUBRAMANIAN K

962821106018 : BASKARAN K

Phase - 4: DEVELOPMENT PART 2

TOPIC: CREATE A CHATBOT



INTRODUCTION:

- ❖ Chatbots are used in various applications across industries, including customer support, e-commerce, healthcare, and education. They are employed to streamline processes, provide instant responses, and enhance user experiences. Chatbots can be simple rule-based systems that follow predefined scripts, or more advanced AI-driven models that learn and adapt from user interactions, becoming more proficient over time.
- ❖ Feature selection is the process of identifying and selecting the most relevant features from a dataset to improve the performance of a machine learning model. Feature engineering for a chatbot involves the process of selecting, creating, and preparing relevant input data to improve the chatbot's natural language understanding and generation capabilities. It is a critical step in building a chatbot that can interact effectively with users.
- ❖ Model training is a process in machine learning and artificial intelligence where a computational model, typically a mathematical algorithm or a neural network, is "taught" to perform a specific task or make predictions based on input data.
- ❖ Model evaluation for a chatbot is the process of assessing how well the chatbot performs in understanding user inputs and generating appropriate responses. It helps determine the quality of the chatbot's interactions and whether it meets its intended goals.

These are the points we are going to discuss in this phase to develop our chatbot further.

Overview of the process:

The following is an overview of the process of building a house price prediction model by feature selection, model training, and evaluation:

1. Prepare the Data
2. Perform feature engineering
3. Train the Model
4. Evaluate the Model
5. Deploy the Model

DATA SET:

We are using the dataset provided in the problem. They provide some basic dialogues in question and answer which are commonly used conversation in day-to-day life.

Dataset Link: <https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot>

Feature Engineering:

- **Text Processing:** This includes tokenization, removing stop words, stemming, and lemmatization to break down and clean the text input for analysis.
- **Word Embeddings:** Word embeddings are a technique used to represent words as continuous vectors in a high-dimensional space, capturing semantic relationships between words. One popular method for creating word embeddings is Word2Vec. You can use Python and libraries like Gensim to create word embeddings using Word2Vec.
- **Named Entity Recognition (NER):** Named Entity Recognition (NER) is a natural language processing (NLP) technique that focuses on identifying and classifying

named entities within a text. Named entities are words or phrases that represent specific, named objects or entities such as names of people, organizations, locations, dates, percentages, monetary values, and more. NER is crucial for tasks like information extraction, sentiment analysis, question answering, and chatbot development.

- **Intent Classification:** Implement machine learning or deep learning models to categorize user intents based on features like word frequencies, TF-IDF, or word embeddings.
- **Contextual Information:** Maintain a conversation history to understand user queries within the context of the ongoing conversation.
- **User Profile and History:** Personalize responses by considering user profiles and past interactions if available.
- **Sentiment Analysis:** Analyze the sentiment of user messages to adapt responses accordingly.
- **Dialogue State Tracking:** If the chatbot conducts multi-turn conversations, use features to track the current state of the dialogue.
- **Knowledge Integration:** Incorporate external data sources or knowledge graphs to enrich responses.

Model training:

This is a process in machine learning and artificial intelligence where a computational model, typically a mathematical algorithm or a neural network, is "taught" to perform a specific task or make predictions based on input data. During training, the model adjusts its internal parameters to minimize a predefined error or loss function, making its predictions or classifications more accurate and aligned with the desired outcomes.

Key components of model training include:

- **Training Data:** A dataset containing examples of input data and their corresponding target values or labels. The model learns from this data and uses it to adjust its parameters.
- **Loss Function:** A function that quantifies the difference between the model's predictions and the actual target values. The goal during training is to minimize this loss, which guides the model in the right direction.
- **Optimization Algorithm:** A method for adjusting the model's internal parameters to reduce the loss. Gradient descent is a common optimization algorithm used in training neural networks.
- **Epochs and Batches:** Training is typically organized into iterations known as epochs, where the entire training dataset is processed. Data is often divided into smaller batches for efficiency.
- **Hyperparameter Tuning:** Parameters like learning rate, batch size, and the model's architecture are fine-tuned to improve the training process and final model performance.

The outcome of model training is an optimized model that, ideally, can make accurate predictions or classifications on new, unseen data. This process is at the core of supervised learning, where the model learns from labeled data, and it's also fundamental in other machine learning paradigms such as reinforcement learning and deep learning.

MODEL EVALUATION:

Model evaluation for a chatbot is the process of assessing how well the chatbot performs in understanding user inputs and generating appropriate responses. It helps determine the quality of the chatbot's interactions and whether it meets its intended goals. Here's a brief explanation of model evaluation for chatbots:

- **Test Data:** Prepare a dataset separate from the one used for training. This dataset should include user queries and the expected chatbot responses.

- **Metrics:** Define evaluation metrics to measure the chatbot's performance. Common metrics for chatbot evaluation include:
 - **Accuracy:** Measures the percentage of correct responses.
 - **BLEU Score:** Evaluates the quality of generated text by comparing it to reference text.
 - **F1 Score:** Combines precision and recall to assess the model's overall performance.
 - **Perplexity:** Measures how well the model predicts the next word in a sequence.
 - **Human Evaluation:** Involve human assessors who rate the quality and relevance of chatbot responses.

- **Quantitative Evaluation:** Use the defined metrics to assess the chatbot's performance on the test dataset. This quantitative evaluation provides an objective measure of how well the chatbot is doing.
- **Qualitative Evaluation:** Additionally, consider qualitative aspects such as the appropriateness, coherence, and naturalness of the responses. This can be achieved through human evaluation or manual review.
- **Error Analysis:** Analyze the errors or shortcomings of the chatbot. This helps in understanding specific areas where the chatbot may be failing and where improvements are needed.
- **A/B Testing:** If possible, conduct A/B testing to compare the chatbot's performance with different versions or approaches. This allows for direct comparison and real-world user feedback.
- **Iterative Improvement:** Based on the evaluation results, make necessary adjustments to the chatbot's model, features, and data. Reiterate the training and evaluation process to improve performance.
- **User Feedback:** Continuously collect user feedback and integrate it into the model's training and improvement cycle. User feedback is invaluable for making a chatbot more user-friendly and effective.

- **Benchmarking:** Compare your chatbot's performance to industry benchmarks or other similar chatbots to gauge how it ranks in terms of quality.

Model evaluation is an ongoing process, and it's essential for ensuring that the chatbot remains effective and aligns with its intended objectives. Both quantitative and qualitative evaluations are crucial for understanding the chatbot's strengths and weaknesses and for making necessary enhancements.

DEPLOYING MODEL:

Python-Integration with website:

The Python chat-bot program is integrated with Flask Library

Coding and Process:

In chat-bot the basic concept to compare questions and answers and give it to the user

```
import csv
from flask import Flask, request, render_template

app = Flask(__name__)

# Read destinations from CSV
destinations = {}
with open('output.csv', 'r', newline='', encoding='utf-8') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        destinations[row['Question'].lower()] = row['Answer']

@app.route('/')
def index():
    return render_template('index.html')
```



```
@app.route('/get_destination', methods=['POST'])
def get_destination():
if __name__ == '__main__':
    app.run(debug=True)
```

To integrate this python code with a website we need a HTML- (Hyper Text Mark-up Language) code which is the outline for all websites

```
<!DOCTYPE html>
<html>
<head>
    <title>Ultron </title>
</head>
<body>
    <h1>Hi I'm Ultron </h1>
    <p>Welcome TONY STARK:</p>
    <p>What's your request</p>
    <form action="/get_destination" method="post">
        <input type="text" name="user_input">
        <input type="submit" value="Ask">
    </form>
    {% if response %}
        <p>{{ response }}</p>
    {% endif %}
</body>
</html>
```

To converse with the user we need a set of data of pre-set questions and answers

	A	B
1	Question	Answer
2	hi, how are you doing?	i'm fine. how about yourself?
3	i'm fine. how about yourself!?	i'm pretty good. thanks for asking.
4	i'm pretty good. thanks for asking.	no problem. so how have you been?
5	no problem. so how have you been?	i've been great. what about you?
6	i've been great. what about you?	i've been good. i'm in school right now.
7	i've been good. i'm in school right now.	what school do you go to?
8	what school do you go to?	i go to pcc.
9	i go to pcc.	do you like it there?
10	do you like it there?	it's okay. it's a really big campus.
11	it's okay. it's a really big campus.	good luck with school.
12	good luck with school.	thank you very much.
13	how's it going?	i'm doing well. how about you?
14	i'm doing well. how about you?	never better, thanks.
15	never better, thanks.	so how have you been lately?
16	so how have you been lately?	i've actually been pretty good. you?
17	i've actually been pretty good. you?	i'm actually in school right now.
18	i'm actually in school right now.	which school do you attend?
19	which school do you attend?	i'm attending pcc right now.
20	i'm attending pcc right now.	are you enjoying it there?
21	are you enjoying it there?	it's not bad. there are a lot of people there.
22	it's not bad. there are a lot of people there	good luck with that.
23	good luck with that.	thanks.
24	how are you doing today?	i'm doing great. what about you?
25	i'm doing great. what about you?	i'm absolutely lovely, thank you.
26	i'm absolutely lovely, thank you.	everything's been good with you?
27	everything's been good with you?	i haven't been better. how about yourself?
28	i haven't been better. how about yourself	i started school recently.
29	i started school recently.	where are you going to school?

To Execute the code we need python interpreter and also it has to act as a development server but to work as a proper website we need a WSGI Server

```
Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\Travel>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 111-564-874
127.0.0.1 - - [24/Oct/2023 15:47:04] "GET / HTTP/1.1" 200 -
```

This development server works in the localhost of a system in the port:5000. Thus the execution of the program in the localhost port <http://127.0.0.1:5000> is given by

Hi I'm ULTRON , Welcome to the World of AI

Greetings TONY(T.S):

Ask Me Anything

i've been good. i'm in school right now.

CONCLUSION:

In conclusion, the development and integration of a chatbot into a website using Flask is a multi-faceted process that involves several key components: feature engineering, model training, model evaluation, and seamless integration. Each of these components is essential to create a functional and effective chatbot that can provide value to users. Here's a summary of the key points:

- ❖ **Feature Engineering:** Feature engineering is the initial step in preparing the chatbot for understanding and generating natural language responses. It involves tasks like text preprocessing, intent classification, context management, sentiment analysis, and the integration of external knowledge sources. Effective feature engineering is essential for enhancing the chatbot's understanding and responsiveness.
- ❖ **Model Training:** Model training is the process of teaching the chatbot's underlying AI model to make accurate predictions or generate relevant responses based on the features and data. In this context, you might use machine learning or deep learning models to train the chatbot. Training involves selecting the right architecture, optimization, and hyperparameter tuning to achieve high performance.
- ❖ **Model Evaluation:** Model evaluation is a critical step to ensure that the chatbot performs well on unseen data. It involves testing the chatbot using a separate dataset, assessing its accuracy, and considering both quantitative and qualitative aspects of its performance. Evaluation metrics and human

feedback help in understanding the model's strengths and weaknesses.

- ❖ **Integration with Flask:** The integration of the chatbot into a website using Flask allows users to interact with it seamlessly. Flask is a web framework for Python that provides a platform for developing web applications. By integrating the chatbot into a Flask-based website, you make the chatbot accessible to a broader audience and enhance the user experience.

The creation of a chatbot for a website using Flask involves a combination of feature engineering, model training, model evaluation, and website integration. This comprehensive approach ensures that the chatbot can effectively understand user queries and provide valuable responses. The iterative nature of these steps allows for continuous improvement, as user feedback and real-world data can be used to refine the chatbot's performance and make it a valuable asset on the website.