CREATE A CHATBOT USING PYTHON
TEAM MEMBER
311121205011-ASHRUTHA G

Project : Create a chatbot using python

Phase 4: Development Part 2

In this part we will be building the chatbot by integrating it into a web app using Flask

INTRODUCTION

Chatbots have emerged as powerful tools in the world of technology, transforming the way we interact with businesses, services, and even each other. They are computer programs designed to simulate human conversation, providing real-time responses to text or voice inputs. Whether for customer support, information retrieval, or entertainment, chatbots have become an integral part of many online platforms and applications.

Creating a chatbot is an exciting venture that combines elements of artificial intelligence, natural language processing, and user experience design. The primary goal of a chatbot is to engage users in meaningful, efficient, and often personalized conversations, ultimately providing value, assistance, or entertainment.

To load libraries for working with datasets in Python, you typically use a combination of popular libraries, such as Flask,torch. Here's a brief description of how to load these libraries and what they are commonly used for:

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

from transformers import AutoModelForCausalLM, AutoTokenizer
import torch
```

CODE:
We create a html and css file for our front end
#chat.html

```
<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<!DOCTYPE html>
<html>
	<head>
		<title>Chatbot</title>
		<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXo
PkFOJwJ8ERdknLPMO" crossorigin="anonymous">
		<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css"
integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSs
eTk6S+L3BlXeVIU" crossorigin="anonymous">
		<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
		<link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css')}}"/>
	</head>


	<body>
		<div class="container-fluid h-100">
			<div class="row justify-content-center h-100">
				<div class="col-md-8 col-xl-6 chat">
					<div class="card">
						<div class="card-header msg_head">
							<div class="d-flex bd-highlight">
								<div class="img_cont">
									<img
src="https://i.ibb.co/fSNP7Rz/icons8-chatgpt-512.png" class="rounded-circle
user_img">
```

```
                                                    <span
class="online_icon"></span>
                                                </div>
                                                <div class="user_info">
                                                    <span>ChatBot</span>
                                                    <p>Ask me
anything!</p>
                                                </div>
                                            </div>
                                        </div>
                                        <div id="messageFormeight"
class="card-body msg_card_body">


                                        </div>
                                        <div class="card-footer">
                                            <form id="messageArea"
class="input-group">
                                <input type="text" id="text" name="msg" placeholder="Type
your message..." autocomplete="off" class="form-control type_msg" required/>
                                                <div
class="input-group-append">
                                                    <button type="submit"
id="send" class="input-group-text send_btn"><i class="fas
fa-location-arrow"></i></button>
                                                </div>
                                            </form>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>

                        <script>
                            $(document).ready(function() {
                                $("#messageArea").on("submit", function(event) {
                                    const date = new Date();
                                    const hour = date.getHours();
                                    const minute = date.getMinutes();
                                    const str_time = hour+":"+minute;
```

```
                                        var rawText = $("#text").val();

                                        var userHtml = '<div class="d-flex
justify-content-end mb-4"><div class="msg_cotainer_send">' + rawText + '<span
class="msg_time_send">'+ str_time + '</span></div><div
class="img_cont_msg"><img src="https://i.ibb.co/d5b84Xw/Untitled-design.png"
class="rounded-circle user_img_msg"></div></div>';

                                        $("#text").val("");
                                        $("#messageFormeight").append(userHtml);

                                        $.ajax({
                                                data: {
                                                        msg: rawText,
                                                },
                                                type: "POST",
                                                url: "/get",
                                        }).done(function(data) {
                                                var botHtml = '<div class="d-flex
justify-content-start mb-4"><div class="img_cont_msg"><img
src="https://i.ibb.co/fSNP7Rz/icons8-chatgpt-512.png" class="rounded-circle
user_img_msg"></div><div class="msg_cotainer">' + data + '<span
class="msg_time">' + str_time + '</span></div></div>';

$("#messageFormeight").append($.parseHTML(botHtml));
                                        });
                                        event.preventDefault();
                                });
                        });
                </script>

   </body>
</html>
```

#styles.css

```
body,html{
        height: 100%;
        margin: 0;
        background: rgb(105, 112, 138);
```

```
  background: -webkit-linear-gradient(to right, rgb(80, 112, 70), rgb(122, 147, 190), rgb(32, 32, 43));
        background: linear-gradient(to right, rgb(73, 92, 106), rgb(86, 94, 110), rgb(77, 77, 162));
}

.chat{
        margin-top: auto;
        margin-bottom: auto;
}
.card{
        height: 500px;
        border-radius: 15px !important;
        background-color: rgba(0, 0, 0, 0.139) !important;
}
.contacts_body{
        padding:  0.75rem 0 !important;
        overflow-y: auto;
        white-space: nowrap;
}
.msg_card_body{
        overflow-y: auto;
}
.card-header{
        border-radius: 15px 15px 0 0 !important;
        border-bottom: 0 !important;
}
.card-footer{
border-radius: 0 0 15px 15px !important;
        border-top: 0 !important;
}
.container{
        align-content: center;
}
.search{
        border-radius: 15px 0 0 15px !important;
        background-color: rgba(0,0,0,0.3) !important;
        border:0 !important;
        color:white !important;
}
```

```
.search:focus{
        box-shadow:none !important;
  outline:0px !important;
}
.type_msg{
        background-color: rgba(0,0,0,0.3) !important;
        border:0 !important;
        color:white !important;
        height: 60px !important;
        overflow-y: auto;
}
        .type_msg:focus{
        box-shadow:none !important;
  outline:0px !important;
}
.attach_btn{
        border-radius: 15px 0 0 15px !important;
        background-color: rgba(0,0,0,0.3) !important;
        border:0 !important;
        color: white !important;
        cursor: pointer;
}
.send_btn{
        border-radius: 0 15px 15px 0 !important;
        background-color: rgba(0,0,0,0.3) !important;
        border:0 !important;
        color: white !important;
        cursor: pointer;
}
.search_btn{
        border-radius: 0 15px 15px 0 !important;
        background-color: rgba(0,0,0,0.3) !important;
        border:0 !important;
        color: white !important;
        cursor: pointer;
}
.contacts{
        list-style: none;
        padding: 0;
}
```

```
.contacts li{
	width: 100% !important;
	padding: 5px 10px;
	margin-bottom: 15px !important;
}
.active{
	background-color: rgba(0,0,0,0.3);
}
.user_img{
	height: 70px;
	width: 70px;
	border:1.5px solid #f5f6fa;

}
.user_img_msg{
	height: 40px;
	width: 40px;
	border:1.5px solid #f5f6fa;

}
.img_cont{
	position: relative;
	height: 70px;
	width: 70px;
}
.img_cont_msg{
	height: 40px;
	width: 40px;
}
.online_icon{
	position: absolute;
	height: 15px;
	width:15px;
	background-color: #4cd137;
	border-radius: 50%;
	bottom: 0.2em;
	right: 0.4em;
	border:1.5px solid white;
}
.offline{
```

```
        background-color: #c23616 !important;
}
.user_info{
        margin-top: auto;
        margin-bottom: auto;
        margin-left: 15px;
}
.user_info span{
        font-size: 20px;
        color: white;
}
.user_info p{
        font-size: 10px;
        color: rgba(255,255,255,0.6);
}
.video_cam{
        margin-left: 50px;
        margin-top: 5px;
}
.video_cam span{
        color: white;
        font-size: 20px;
        cursor: pointer;
        margin-right: 20px;
}
.msg_cotainer{
        margin-top: auto;
        margin-bottom: auto;
        margin-left: 10px;
        border-radius: 25px;
        background-color: rgb(82, 172, 255);
        padding: 10px;
        position: relative;
}
.msg_cotainer_send{
        margin-top: auto;
        margin-bottom: auto;
        margin-right: 10px;
        border-radius: 25px;
        background-color: #58cc71;
```

```
        padding: 10px;
        position: relative;
}
.msg_time{
        position: absolute;
        left: 0;
        bottom: -15px;
        color: rgba(255,255,255,0.5);
        font-size: 10px;
}
.msg_time_send{
        position: absolute;
        right:0;
        bottom: -15px;
        color: rgba(255,255,255,0.5);
        font-size: 10px;
}
.msg_head{
        position: relative;
}
#action_menu_btn{
        position: absolute;
        right: 10px;
        top: 10px;
        color: white;
        cursor: pointer;
        font-size: 20px;
}
.action_menu{
        z-index: 1;
        position: absolute;
        padding: 15px 0;
        background-color: rgba(0,0,0,0.5);
        color: white;
        border-radius: 15px;
        top: 30px;
        right: 15px;
        display: none;
}
.action_menu ul{
```

```
        list-style: none;
        padding: 0;
        margin: 0;
}
.action_menu ul li{
        width: 100%;
        padding: 10px 15px;
        margin-bottom: 5px;
}
.action_menu ul li i{
        padding-right: 10px;
}
.action_menu ul li:hover{
        cursor: pointer;
        background-color: rgba(0,0,0,0.2);
}
@media(max-width: 576px){
        .contacts_card{
        margin-bottom: 15px !important;
}
}
```

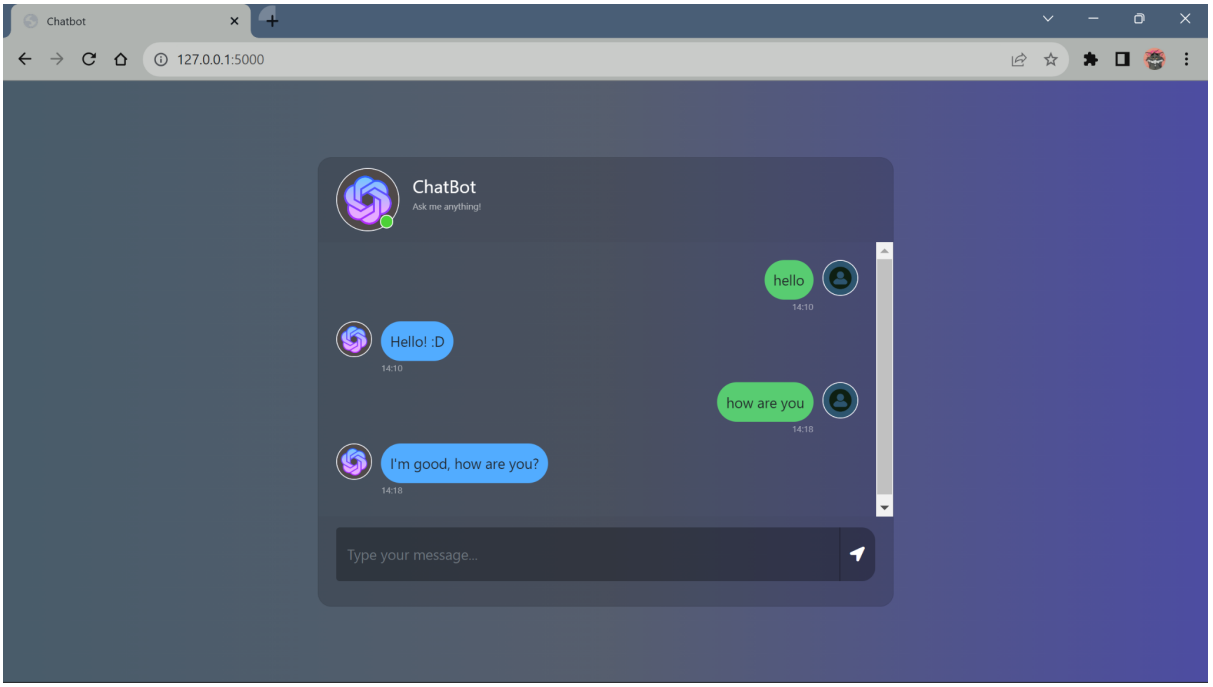
Then we create a python file which includes Flask for the chatbot implementation

#app.py

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


from transformers import AutoModelForCausalLM, AutoTokenizer
import torch


tokenizer = AutoTokenizer.from_pretrained("microsoft/DialoGPT-medium")
model = AutoModelForCausalLM.from_pretrained("microsoft/DialoGPT-medium")


app = Flask(__name__)

@app.route("/")
```

```
def index():
    return render_template('chat.html')


@app.route("/get", methods=["GET", "POST"])
def chat():
    msg = request.form["msg"]
    input = msg
    return get_Chat_response(input)


def get_Chat_response(text):
    for step in range(5):
        # encode the new user input, add the eos_token and return a tensor in Pytorch
        new_user_input_ids = tokenizer.encode(str(text) + tokenizer.eos_token,
return_tensors='pt')

        # append the new user input tokens to the chat history
        bot_input_ids = torch.cat([chat_history_ids, new_user_input_ids], dim=-1) if step
> 0 else new_user_input_ids

        # generated a response while limiting the total chat history to 1000 tokens,
        chat_history_ids = model.generate(bot_input_ids, max_length=1000,
pad_token_id=tokenizer.eos_token_id)

        # pretty print last ouput tokens from bot
        return tokenizer.decode(chat_history_ids[:, bot_input_ids.shape[-1]:][0],
skip_special_tokens=True)
if __name__ == '__main__':
    app.run()
```

And finally our chatbot would look like this



CONCLUSION:

In summary, our Python chatbot project with Flask has been a successful demonstration of the power of conversational AI and web development. By combining natural language processing capabilities and a user-friendly web interface, we've created a chatbot that can assist users, answer their questions, and engage in meaningful conversations. This project lays a strong foundation for future enhancements, including multi-language support, domain-specific knowledge integration, and even machine learning for improved intent recognition and personalization. As we continue to collect and analyze user interactions and feedback, we are committed to refining the chatbot's performance and ensuring an exceptional user experience. The possibilities for further advancements in this exciting field are vast, and we look forward to harnessing these opportunities to create even more intelligent and responsive chatbot solutions.