ChatInterface

October 20, 2024

[17]:

```
import gradio as gr
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer

# Load the pre-trained DialoGPT model and tokenizer (replace with your trained␣
 ↪model path if needed)
#model_name = "microsoft/DialoGPT-small"  # or path to your trained model
model_name = r'./fine_tuned_dialoGPT/fine_tuned_dialoGPT_epoch3_step3600.pt'
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

# Move model to GPU if available
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model.to(device)

# Function to generate a response using the trained DialoGPT model
def generate_response(message, history):

    print("Me: {}".format(message))
    # Tokenize the input and convert it to input_ids
    new_input_ids = tokenizer.encode(message + tokenizer.eos_token,␣
 ↪return_tensors='pt').to(device)

    # Generate a response from the model
    response_ids = model.generate(new_input_ids, max_length=1000,␣
 ↪pad_token_id=tokenizer.eos_token_id)

    # Decode the generated response
    response = tokenizer.decode(response_ids[:, new_input_ids.shape[-1]:][0],␣
 ↪skip_special_tokens=True)
    print("ChatFlix: {}".format(response))

    # return response, history  # Return both response and updated history
    return response


# Chat Interface in Gradio
```

```
interface = gr.ChatInterface(
    fn=generate_response,
    title="ChatFlix using DialoGPT",
    description="A chatbot based on the DialoGPT model, fine-tuned for␣
  ↪multi-turn conversations.",
    examples=[{"text": "Hello", "files": []}],
    multimodal=False
)

interface.launch()
```

```
C:\Users\mthiruma\AppData\Local\Programs\Python\Python312\Lib\site-
packages\gradio\components\chatbot.py:228: UserWarning: The 'tuples' format for
chatbot messages is deprecated and will be removed in a future version of
Gradio. Please set type='messages' instead, which uses openai-style 'role' and
'content' keys.
  warnings.warn(
```

```
* Running on local URL:  http://127.0.0.1:7873

To create a public link, set `share=True` in `launch()`.
```

```
<IPython.core.display.HTML object>
```

[17]:

```
Me: hello
ChatFlix: Hi
Me: how are you doing
ChatFlix: I'm here
Me: how's it going
ChatFlix: I'm good
Me: what are your plans
ChatFlix: I'm here
Me: wanna go out?
ChatFlix: I'm here
```