DATE : 12.11.2024

**CUSTOMER SUPPORT CHATBOT**  
 ORDER ASSISTANCE

**Student:**

AKHILA M (22CS004)

CHANTHURU S R (22CS013)

DHILIP M (22CS024)

HANUSHREE M (22CS029)

**ABOUT:**

**Key Features for Customer Support:**

**Natural Conversation**: The chatbot is designed with a friendly prompt template, ensuring user-centric and approachable responses.

**Model Integration:** It uses multiple models (Mistral and Llama 2) to process and generate accurate answers for user queries.

**Dynamic Query Handling**: The chatbot can address general inquiries, troubleshoot issues, and provide assistance on topics like product details or policies.

**Order Assistance:**

* The chatbot supports order-related tasks:
* Tracking orders: “Where is my order?”
* Processing returns: “How do I return my item?”
* Order updates: “Can I change my delivery address?”
* Responses are generated dynamically, ensuring personalized interactions.

The chatbot interface allows users to send queries and receive real-time responses from the model. This modular approach makes it adaptable for e-commerce platforms, enabling smooth customer support for both general and order-specific needs.

from ctransformers import AutoModelForCausalLM

llm = AutoModelForCausalLM.from\_pretrained("TheBloke/Mistral-7B-Instruct-v0.1-GGUF", model\_file="mistral-7b-instruct-v0.1.Q4\_K\_M.gguf", model\_type="mistral", gpu\_layers=50)

print(llm("AI is going to"))

from ctransformers import AutoModelForCausalLM

async def callback(contents: str):

    llms = {}

    if "mistral" not in llms:

        llms["mistral"] = AutoModelForCausalLM.from\_pretrained(

            "TheBloke/Mistral-7B-Instruct-v0.1-GGUF",

            model\_file="mistral-7b-instruct-v0.1.Q4\_K\_M.gguf",

            gpu\_layers=1,

        )

    llm = llms["mistral"]

    response = llm(contents, stream=True, max\_new\_tokens=1000)

    message = ""

    for token in response:

        message += token

        yield message

import panel as pn

from langchain.chains import LLMChain

from langchain.llms import CTransformers

from langchain.prompts import PromptTemplate

pn.extension()

# Model configurations

MODEL\_KWARGS = {

    "llama": {

        "model": "TheBloke/Llama-2-7b-Chat-GGUF",

        "model\_file": "llama-2-7b-chat.Q5\_K\_M.gguf",

    },

    "mistral": {

        "model": "TheBloke/Mistral-7B-Instruct-v0.1-GGUF",

        "model\_file": "mistral-7b-instruct-v0.1.Q4\_K\_M.gguf",

    },

}

llm\_chains = {}

# Template for the chatbot prompt

TEMPLATE = """<s>[INST] You are a friendly chat bot who's willing to help answer the user:

{user\_input} [/INST] </s>

"""

async def callback(contents: str, user: str, instance: pn.chat.ChatInterface):

    """Callback function to process user input and return model responses."""

    config = {"max\_new\_tokens": 256, "temperature": 0.5}

    responses = {}

    # Loop through models and get individual responses

    for model\_name, model\_kwargs in MODEL\_KWARGS.items():

        if model\_name not in llm\_chains:

            instance.placeholder\_text = (

                f"Downloading {model\_name}, this may take a few minutes,"

                f"or longer, depending on your internet connection."

            )

            # Initialize the LLMChain for the model

            llm = CTransformers(\*\*model\_kwargs, config=config)

            prompt = PromptTemplate(template=TEMPLATE, input\_variables=["user\_input"])

            llm\_chain = LLMChain(prompt=prompt, llm=llm)

            llm\_chains[model\_name] = llm\_chain

        # Get the response from the model

        response = await llm\_chains[model\_name].apredict(user\_input=contents)

        responses[model\_name] = response

    # Send individual responses to the chat interface

    for model\_name, response in responses.items():

        instance.send(response, user=f"{model\_name.capitalize()} Response", respond=False)

# Set up the chat interface

chat\_interface = pn.chat.ChatInterface(callback=callback, placeholder\_threshold=0.1)

# Initial message to the user

chat\_interface.send(

    "Send a message to get individual replies from Llama 2 and Mistral (7B)!",

    user="System",

    respond=False,

)

chat\_interface.servable()