Personalized Healthcare QA System (Chatbot) - Chatbot UI

Ed:215279167

---

1. Import libraries and model for Chatbot UI

```
#!pip install datasets requests bitsandbytes accelerate peft trl sentencepiece wandb transformers evaluate rouge_score bert-score gra
!pip install requests bitsandbytes wandb transformers gradio langchain-huggingface
```

```
Collecting tomlkit<0.14.0,>=0.12.0 (from gradio)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.15.3)
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->grac
Collecting langchain-core<1.0.0,>=0.3.59 (from langchain-huggingface)
  Downloading langchain_core-0.3.59-py3-none-any.whl.metadata (5.9 kB)
Requirement already satisfied: sentence-transformers>=2.6.0 in /usr/local/lib/python3.11/dist-packages (from langchain-huggingface
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: six>=1.4.0 in /usr/local/lib/python3.11/dist-packages (from docker-pycreds>=0.4.0->wandb) (1.17.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from gitpython!=3.1.29,>=1.0.0->wandb)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (0
Requirement already satisfied: langsmith<0.4,>=0.1.125 in /usr/local/lib/python3.11/dist-packages (from langchain-core<1.0.0,>=0.3
Requirement already satisfied: tenacity!=8.4.0,<10.0.0,>=8.1.0 in /usr/local/lib/python3.11/dist-packages (from langchain-core<1.0
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.11/dist-packages (from langchain-core<1.0.0,>=0.3.59
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3->wandb) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<3->wandb) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3->wandb) (0.4.0
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=2.6.0->langcha
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from sentence-transformers>=2.6.0->langchain-hugg
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch<3,>=2.0->bitsandbytes) (3.4.2)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.11/dist-packages (from torch<3,>=2.0->bitsa
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dist-packages (from torch<3,>=2.0->bitsandbyt
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch<3,>=2.0->bitsandb
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch<3,>=2.0->bitsandbytes)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packages (from torch<3,>=2.0->bitsandbytes) (3.2.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages (from torch<3,>=2.0->bitsandbytes) (1.13.1
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from sympy==1.13.1->torch<3,>=2.0->b
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.11/dist-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.29
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.11/dist-packages (from jsonpatch<2.0,>=1.33->langchain-c
Requirement already satisfied: requests-toolbelt<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from langsmith<0.4,>=0
```

IMPORTANT: Restart Colab runtime after PIP install!!!

2. Implement base model and imports

First step is to import all necessary libraries and implement base model, to validate it can generate a response from prompt.

```
from google.colab import userdata
from huggingface_hub import login
```

```
from transformers import (
    AutoTokenizer,
    AutoModelForCausalLM,
    BitsAndBytesConfig,
    TrainingArguments,
    logging,
    pipeline
)
from peft import LoraConfig, PeftModel, prepare_model_for_kbit_training, get_peft_model, TaskType
import torch
import wandb
import pandas as pd
import os
import random
import re

# for chatbot UI
import gradio as gr
```

```
base_model_name = "meta-llama/Llama-2-7b-chat-hf"
#model_name = "digitalblue/model_e_merge_v2"
project_name = "ed_medical"
hf_username = "digitalblue"

# define peft adapters saved to huggingface hub
MODEL_A = "digitalblue/ed_medical-2025-04-03_09.11.29" # trained on 200 rows
MODEL_B = "digitalblue/ed_medical-2025-04-05_10.59.54" # trained on 800 rows
MODEL_C = "digitalblue/ed_medical-2025-04-05_11.36.32" # trained on 1600 rows
MODEL_D = "digitalblue/ed_medical-2025-04-06_10.40.29" # trained on 1600 rows with NEFTune
MODEL_E = "digitalblue/ed_medical-2025-04-15_12.43.48" # trained on 8000 rows of pubmedqa only w/ NEFTune
MODEL_F = "digitalblue/ed_medical-2025-05-05_23.51.25" # trained on 9000 row, lower learing rate, 3 epochs
MODEL_G = "digitalblue/ed_medical-2025-05-08_01.23.37" # trained on 9000 rows, NEFTune false
MODEL_H = "digitalblue/ed_medical-2025-05-11_04.05.52" # trained on 900 row, with context, 3 epochs
MODEL_I = "digitalblue/ed_medical-2025-05-12_02.31.47" # trained on 900 rows of MedDialog
```

```
# log into hugging face and wandb
hf_token = userdata.get('HF_TOKEN')
login(hf_token)

wandb_api_key = userdata.get('WANDB_API_KEY')
os.environ["WANDB_API_KEY"] = wandb_api_key
wandb.login()

# Configure Weights & Biases to record against our project
os.environ["WANDB_PROJECT"] = "ed_medical"
os.environ["WANDB_LOG_MODEL"] = "checkpoint" if True else "end"
os.environ["WANDB_WATCH"] = "gradients"
```

```
wandb: Currently logged in as: digitalblue (digitalblue-ai) to https://api.wandb.ai. Use `wandb login --relogin` to force relogin
```

```
# quantisation config to use less memory when loading model
quant_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_compute_dtype=torch.bfloat16,
    bnb_4bit_quant_type="nf4"
)
```

```
model = AutoModelForCausalLM.from_pretrained(
    base_model_name,
    device_map="auto",
    quantization_config=quant_config)
```

```
config.json: 100%  614/614 [00:00<00:00, 30.5kB/s]
model.safetensors.index.json: 100%  26.8k/26.8k [00:00<00:00, 2.56MB/s]
Fetching 2 files: 100%  2/2 [04:45<00:00, 285.89s/it]
model-00002-of-00002.safetensors: 100%  3.50G/3.50G [03:09<00:00, 71.6MB/s]
model-00001-of-00002.safetensors: 100%  9.98G/9.98G [04:45<00:00, 240MB/s]
Loading checkpoint shards: 100%  2/2 [01:23<00:00, 38.30s/it]
generation_config.json: 100%  188/188 [00:00<00:00, 20.1kB/s]
```

```
# initialise tokenizer and pipeline
tokenizer = AutoTokenizer.from_pretrained(base_model_name, token=hf_token)
```

```
tokenizer.pad_token = tokenizer.eos_token
tokenizer.padding_side = "right"
```

```
tokenizer_config.json: 100%  1.62k/1.62k [00:00<00:00, 152kB/s]
tokenizer.model: 100%  500k/500k [00:00<00:00, 16.2MB/s]
tokenizer.json: 100%  1.84M/1.84M [00:00<00:00, 2.12MB/s]
special_tokens_map.json: 100%  414/414 [00:00<00:00, 30.3kB/s]
```

```
#model_g = PeftModel.from_pretrained(model, MODEL_G)
model_i = PeftModel.from_pretrained(model, MODEL_I)
```

```
adapter_config.json: 100%  813/813 [00:00<00:00, 55.2kB/s]
adapter_model.safetensors: 100%  67.1M/67.1M [00:00<00:00, 250MB/s]
```

```
system_prompt = "You are a helpful personalised medical assistant. In your response do not include any personal names and ensure it

def get_model_response(model_x, prompt, chat_history):

  messages = [
    {"role": "system", "content": system_prompt},
  ]
  for turn in chat_history:
    print(f"turn={turn}")
    messages.append(turn)

  messages.append({"role": "user", "content": prompt})

  print(f"messages={messages}")

  inputs = tokenizer.apply_chat_template(messages, return_tensors="pt").to("cuda")
  outputs = model_x.generate(inputs, max_new_tokens=512, temperature=0.1)
  response = tokenizer.decode(outputs[0], skip_special_tokens=False)

  print(f"response={response}")
  answer = response.split("[/INST]")[-1].strip()
  answer = answer.replace('</s>', '') # remove trailing special token if present
  print(f"answer={answer}")
  return answer
```

```
get_model_response(model_i, "Hi, does utility of preliminary bronchoalveolar lavage result in suspected ventilator-associated pneumc
```

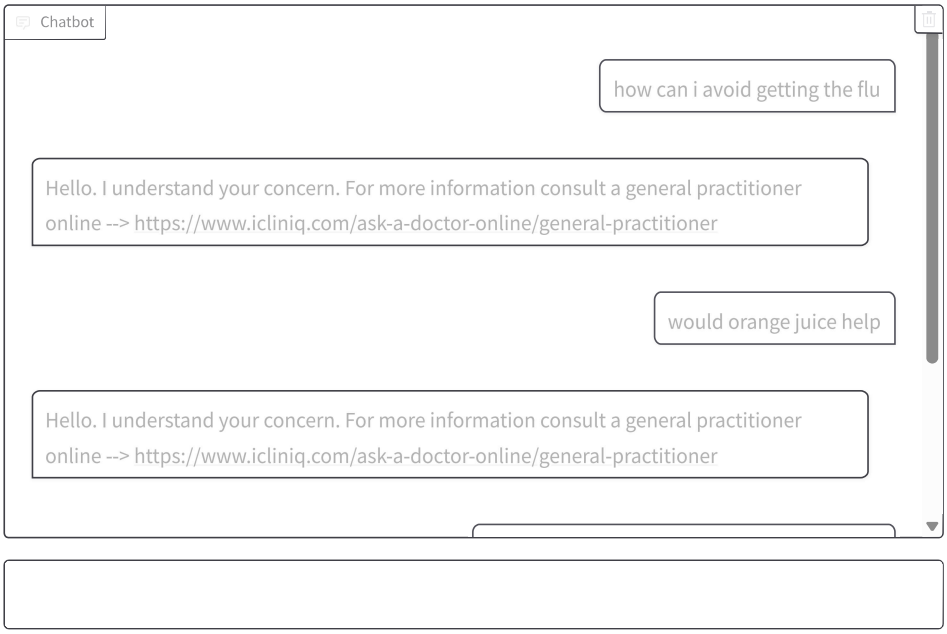
```
messages=[{'role': 'system', 'content': 'You are a helpful personalised medical assistant. In your response do not include any perso
response=<s> [INST] <<SYS>>
You are a helpful personalised medical assistant. In your response do not include any personal names and ensure it is detailed. Do r
<</SYS>>

Hi, does utility of preliminary bronchoalveolar lavage result in suspected ventilator-associated pneumonia? [/INST] Hello. I have re
answer=Hello. I have read your query. For further information consult a pulmonologist online --> https://www.icliniq.com/ask-a-docto
'Hello. I have read your query. For further information consult a pulmonologist online --> https://www.icliniq.com/ask-a-doctor-onl
ine/pulmonologist '
```

```

def respond(message, chat_history):
  try:
    # Format prompt for Llama 2
    print(f"chat_history={chat_history}")
    print(f"message={message}")

    # Get model output
    model_output = get_model_response(model_i, message, chat_history)
    print(f"model_output = {model_output}")
    print("--------")

    return model_output


  except Exception as e:
    print(f"Error in respond(): {str(e)}")
    return "Error occurred", chat_history
```

```
force_dark_mode = """
function refresh() {
    const url = new URL(window.location);
    if (url.searchParams.get('__theme') !== 'dark') {
        url.searchParams.set('__theme', 'dark');
        window.location.href = url.href;
```

```
    }
}
"""

chatbot_ui = gr.ChatInterface(
    fn=respond,
    type="messages",
    flagging_mode="never",
    js=force_dark_mode)


chatbot_ui.launch(debug=True)
```

```
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to work, sharing must be enabled. Automatica

Colab notebook detected. This cell will run indefinitely so that you can see errors and logs. To turn off, set debug=False in launch
* Running on public URL: https://563757c4598215d38e.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working
```




```
chat_history=[]
message=how can i avoid getting the flu
messages=[{'role': 'system', 'content': 'You are a helpful personalised medical assistant. In your response do not include any perso
response=<s> [INST] <<SYS>>
You are a helpful personalised medical assistant. In your response do not include any personal names and ensure it is detailed. Do n
<</SYS>>

how can i avoid getting the flu [/INST] Hello. I understand your concern. For more information consult a general practitioner online
answer=Hello. I understand your concern. For more information consult a general practitioner online --> https://www.icliniq.com/ask
model_output = Hello. I understand your concern. For more information consult a general practitioner online --> https://www.icliniq
--------
chat_history=[{'role': 'user', 'metadata': None, 'content': 'how can i avoid getting the flu', 'options': None}, {'role': 'assistant
message=would orange juice help
turn={'role': 'user', 'metadata': None, 'content': 'how can i avoid getting the flu', 'options': None}
turn={'role': 'assistant', 'metadata': None, 'content': 'Hello. I understand your concern. For more information consult a general pr
messages=[{'role': 'system', 'content': 'You are a helpful personalised medical assistant. In your response do not include any perso
response=<s> [INST] <<SYS>>
You are a helpful personalised medical assistant. In your response do not include any personal names and ensure it is detailed. Do n
<</SYS>>

how can i avoid getting the flu [/INST] Hello. I understand your concern. For more information consult a general practitioner online
answer=Hello. I understand your concern. For more information consult a general practitioner online --> https://www.icliniq.com/ask
model_output = Hello. I understand your concern. For more information consult a general practitioner online --> https://www.icliniq
--------
chat_history=[{'role': 'user', 'metadata': None, 'content': 'how can i avoid getting the flu', 'options': None}, {'role': 'assistant
message=ok what if I have a problem with a blood clot
turn={'role': 'user', 'metadata': None, 'content': 'how can i avoid getting the flu', 'options': None}
turn={'role': 'assistant', 'metadata': None, 'content': 'Hello. I understand your concern. For more information consult a general pr
turn={'role': 'user', 'metadata': None, 'content': 'would orange juice help', 'options': None}
turn={'role': 'assistant', 'metadata': None, 'content': 'Hello. I understand your concern. For more information consult a general pr
messages=[{'role': 'system', 'content': 'You are a helpful personalised medical assistant. In your response do not include any perso
response=<s> [INST] <<SYS>>
You are a helpful personalised medical assistant. In your response do not include any personal names and ensure it is detailed. Do n
<</SYS>>

how can i avoid getting the flu [/INST] Hello. I understand your concern. For more information consult a general practitioner online
answer=Hello. I understand your concern. For more information consult a general practitioner online --> https://www.icliniq.com/ask
model_output = Hello. I understand your concern. For more information consult a general practitioner online --> https://www.icliniq
--------
```

```
# clear gpu memory
torch.cuda.empty_cache()
torch.cuda.reset_max_memory_allocated()
```

```
/usr/local/lib/python3.11/dist-packages/torch/cuda/memory.py:391: FutureWarning: torch.cuda.reset_max_memory_allocated now calls tor
  warnings.warn(
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.