

```

import gradio as gr

import torch

from transformers import AutoTokenizer, AutoModelForCausalLM


# Load model and tokenizer

model_name = "ibm-granite/granite-3.2-2b-instruct"

tokenizer = AutoTokenizer.from_pretrained(model_name)


model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)


if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token


def generate_response(prompt, max_length=512):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)
    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id

```

```

    )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)

    response = response.replace(prompt, "").strip()

    return response


def concept_explanation(concept):

    prompt = f"Explain the concept of {concept} in detail with examples:"

    return generate_response(prompt, max_length=800)


def quiz_generator(concept):

    prompt = (

        f"Generate 5 quiz questions about {concept} with different question types "

        f"(multiple choice, true/false, short answer). "

        f"At the end, provide all the answers in a separate ANSWERS section:"

    )

    return generate_response(prompt, max_length=1000)


# Create Gradio interface

with gr.Blocks() as app:

    gr.Markdown("# 🧠 Educational AI Assistant")

    with gr.Tabs():

        with gr.TabItem("Concept Explanation"):

            concept_input = gr.Textbox(label="Enter a concept", placeholder="e.g., machine learning")

            explain_btn = gr.Button("Explain")

            explanation_output = gr.Textbox(label="Explanation", lines=10)

```

```
explain_btn.click(concept_explanation, inputs=concept_input,  
outputs=explanation_output)
```

```
with gr.TabItem("Quiz Generator"):
```

```
    quiz_input = gr.Textbox(label="Enter a topic", placeholder="e.g., physics")
```

```
    quiz_btn = gr.Button("Generate Quiz")
```

```
    quiz_output = gr.Textbox(label="Quiz Questions", lines=15)
```

```
    quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)
```

```
app.launch(share=True)
```