```
# Import necessary modules

import nltk

import re

from nltk.chat.util import Chat,reflections


# Download NLTK data

nltk.download('punkt')

nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
True
```

```
# Define patterns and responses

pairs = [

    [r"my name is (.*)", ["Hello %1, how can I assist you today?",]],

    [r"hi|hey|hello", ["Hello, how can I help you?", "Hey there! What can I do for you?",

                       "Hi! How can I assist you today?"]],

    [r"what is your name?", ["I am a chatbot created to assist you. You can call me Chatb

    [r"how are you?", ["I'm a bot, so I don't have feelings, but I'm here to help you!",]

    [r"can you help me with (.*)", ["Sure, I can help you with %1. Please provide more de

    [r"sorry (.*)", ["It's okay. How can I assist you?",]],

    [r"thank you|thanks", ["You're welcome!", "No problem!", "Happy to help!"]],

    [r"quit", ["Bye! Have a great day!", "Goodbye!"]],

    [r"best place (.*)", ["Hyderabad","Banglore","Vizag", "Mumbai"]],

    [r"golden|temple (.*)", ["It is in amritsar"]],

    [r"(.*)", ["I'm sorry, I don't understand that. Can you rephrase?",

               "Could you please elaborate on that?"]]
```

```
# Define the chatbot class

class RBChatbot:

    def _init(self, pairs): # Changed _init to _init_

        self.chat = Chat(pairs, reflections)



    def respond(self, user_input):

        return self.chat.respond(user_input)



# Initialize the chatbot

chatbot = RBChatbot(pairs)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-577406ea0a41> in <cell line: 0>()
     17 # Initialize the chatbot
     18
---> 19 chatbot = RBChatbot(pairs)

TypeError: RBChatbot() takes no arguments
```

Next steps: Explain error

```
# Define the chatbot class

class RBChatbot:

    def __init__(self, pairs): # Changed _init to __init__

        self.chat = Chat(pairs, reflections)



    def respond(self, user_input):

        return self.chat.respond(user_input)



# Initialize the chatbot

chatbot = RBChatbot(pairs)


# Function to chat with the bot
```

```
def chat_with_bot():

    print("Hi, I'm your chatbot. Type 'quit' to exit.")

    while True:

        user_input = input("You: ")

        if user_input.lower() == 'quit':

            print("Chatbot: Bye! Have a great day!")

            break

        response = chatbot.respond(user_input)

        print(f"Chatbot: {response}")



# Start chatting with the bot

chat_with_bot()
```

```
Hi, I'm your chatbot. Type 'quit' to exit.
You: hi
Chatbot: Hi! How can I assist you today?
You: my name is raj
Chatbot: Hello raj, how can I assist you today?
You: best place to visit
Chatbot: Banglore
You: ok  bye
Chatbot: I'm sorry, I don't understand that. Can you rephrase?
You: quit
Chatbot: Bye! Have a great day!
```

```
!pip install transformers torch
```

```
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127
```

```
from transformers import BertForQuestionAnswering, BertTokenizer
import torch
# Load pre-trained BERT model and tokenizer
model_name = "deepset/bert-base-cased-squad2"  # BERT trained on SQuAD 2.0
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertForQuestionAnswering.from_pretrained(model_name)
```

```
Some weights of the model checkpoint at deepset/bert-base-cased-squad2 were not used
- This IS expected if you are initializing BertForQuestionAnswering from the checkpoi
- This IS NOT expected if you are initializing BertForQuestionAnswering from the chec
```

Generate | a slider using jupyter widgets | Close

```
# Define the context (passage) and question
context = """The Eiffel Tower is a wrought-iron lattice tower on the Champ de Mars in Par
```

```
It is named after the engineer Gustave Eiffel, whose company designed and built the tower
question = "Who designed the Eiffel Tower?"
# Tokenize the input
inputs = tokenizer(question, context, return_tensors="pt")
# Get model output (start and end index of answer)
with torch.no_grad():
    outputs = model(**inputs)
    start_scores = outputs.start_logits
    end_scores = outputs.end_logits


# Get the most probable answer span
start_idx = torch.argmax(start_scores)
end_idx = torch.argmax(end_scores) + 1
# Decode the answer
answer = tokenizer.convert_tokens_to_string(tokenizer.convert_ids_to_tokens(input
print(f"Answer: {answer}")
```

```
Answer: Gustave Eiffel
```

```
context1 ="""Coronavirus disease 2019 (COVID-19, also known as SARS-2) is a conta

caused by the coronavirus SARS-CoV-2. In January 2020, the disease spread worldwi

resulting in the COVID-19 pandemic."""

question1 = "what is the cause for coronavirus?"

# Tokenize the input

inputs = tokenizer(question1, context1, return_tensors="pt")
with torch.no_grad():
    outputs = model(**inputs)
    start_scores = outputs.start_logits
    end_scores = outputs.end_logits


# Get the most probable answer span
start_idx = torch.argmax(start_scores)
end_idx = torch.argmax(end_scores) + 1
# Decode the answer
answer = tokenizer.convert_tokens_to_string(tokenizer.convert_ids_to_tokens(input
print(f"Answer: {answer}")
```

```
Answer: SARS - CoV - 2
```

```
Start coding or generate with AI.
```