```
pip install transformers flask torch datasets nltk pandas
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 12.7 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.4 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 1.9 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 27.2 MB/s eta 0:00:00
Downloading datasets-3.4.1-py3-none-any.whl (487 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 487.4/487.4 kB 13.2 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 7.6 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py311-none-any.whl (143 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 5.9 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 8.7 MB/s eta 0:00:00
Installing collected packages: xxhash, nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    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 datasets-3.4.1 dill-0.3.8 multiprocess-0.70.16 nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvi
```

```
import torch
from transformers import GPT2LMHeadModel, GPT2Tokenizer, T5ForConditionalGeneration, T5Tokenizer, BertForSequenceClass
from transformers import Trainer, TrainingArguments
from flask import Flask, request, jsonify
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize
from datasets import load_dataset

nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
# Load the pre-trained models and tokenizers
chatbot_model = GPT2LMHeadModel.from_pretrained('gpt2')
```

```
chatbot_tokenizer = GPT2Tokenizer.from_pretrained('gpt2')

summarization_model = T5ForConditionalGeneration.from_pretrained('t5-small')
summarization_tokenizer = T5Tokenizer.from_pretrained('t5-small')

sentiment_model = BertForSequenceClassification.from_pretrained('bert-base-uncased')
sentiment_tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secre
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%  665/665 [00:00<00:00, 50.8kB/s]
model.safetensors: 100%  548M/548M [00:03<00:00, 104MB/s]
generation_config.json: 100%  124/124 [00:00<00:00, 9.42kB/s]
tokenizer_config.json: 100%  26.0/26.0 [00:00<00:00, 1.79kB/s]
vocab.json: 100%  1.04M/1.04M [00:00<00:00, 5.14MB/s]
merges.txt: 100%  456k/456k [00:00<00:00, 1.13MB/s]
tokenizer.json: 100%  1.36M/1.36M [00:00<00:00, 3.27MB/s]
config.json: 100%  1.21k/1.21k [00:00<00:00, 63.9kB/s]
model.safetensors: 100%  242M/242M [00:01<00:00, 162MB/s]
generation_config.json: 100%  147/147 [00:00<00:00, 10.0kB/s]
tokenizer_config.json: 100%  2.32k/2.32k [00:00<00:00, 202kB/s]
spiece.model: 100%  792k/792k [00:00<00:00, 1.28MB/s]
tokenizer.json: 100%  1.39M/1.39M [00:00<00:00, 2.26MB/s]
You are using the default legacy behaviour of the <class 'transformers.models.t5.tokenization_t5.T5Tokenizer'>. This is expected, and si
config.json: 100%  570/570 [00:00<00:00, 46.1kB/s]
model.safetensors: 100%  440M/440M [00:02<00:00, 149MB/s]
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-uncased and are newly initiali
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 2.66kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 13.3MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 20.0MB/s]
```

```
# Example datasets (small subset for testing)
chatbot_data = [
    ("How can I reset my password?", "To reset your password, click on 'Forgot password' at the login screen."),
    ("Where can I find my order status?", "You can find your order status in your account under 'Order History'.")
]

summarization_data = [
    ("The global climate change issue is becoming more critical each day due to the rising levels of CO2.", "Summary o
]

sentiment_data = [
    ("I love this product, it's amazing!", 1),  # 1 for positive sentiment
    ("This is the worst purchase I've made.", 0)  # 0 for negative sentiment
]

# Convert to pandas DataFrame
chatbot_df = pd.DataFrame(chatbot_data, columns=["Question", "Answer"])
summarization_df = pd.DataFrame(summarization_data, columns=["Text", "Summary"])
sentiment_df = pd.DataFrame(sentiment_data, columns=["Text", "Sentiment"])
```

```
def fine_tune_chatbot(chatbot_data):
    # Add padding token to the tokenizer
    chatbot_tokenizer.pad_token = chatbot_tokenizer.eos_token # Use eos_token as pad_token

```

```
    # Tokenize inputs and labels with padding and truncation
    encodings = chatbot_tokenizer([q[0] for q in chatbot_data],
                                  text_pair=[q[1] for q in chatbot_data],  # Tokenize input-output pairs together
                                  return_tensors="pt",
                                  padding=True,
                                  truncation=True)

    # Ensure labels have the same shape as inputs and are on the same device
    encodings["labels"] = encodings["input_ids"].clone().detach()
    encodings.to(chatbot_model.device)  # Move all tensors to the model's device

    # Set labels to -100 where padding tokens are present in the input_ids
    encodings["labels"] = torch.where(encodings["attention_mask"] == 0, -100, encodings["labels"])

    # Forward pass
    outputs = chatbot_model(**encodings)  # Pass the encodings dictionary directly
    loss = outputs.loss
    return loss

fine_tune_chatbot(chatbot_data)
```

```
tensor(2.7128, grad_fn=<NllLossBackward0>)
```

```
def fine_tune_summarization(summarization_data):
    inputs = summarization_tokenizer(summarization_data["Text"].tolist(), return_tensors="pt", padding=True, truncatio
    labels = summarization_tokenizer(summarization_data["Summary"].tolist(), return_tensors="pt", padding=True, trunca
    outputs = summarization_model(input_ids=inputs["input_ids"], labels=labels)
    loss = outputs.loss
    return loss

# Fine-tune the summarization model (simplified for demonstration)
fine_tune_summarization(summarization_df)

```

```
Passing a tuple of `past_key_values` is deprecated and will be removed in Transformers v4.48.0. You should pass an instance of `EncoderD
tensor(4.7516, grad_fn=<NllLossBackward0>)
```

```
def fine_tune_sentiment_analysis(sentiment_data):
    inputs = sentiment_tokenizer(sentiment_data["Text"].tolist(), return_tensors="pt", padding=True, truncation=True)
    labels = torch.tensor(sentiment_data["Sentiment"].tolist())
    outputs = sentiment_model(input_ids=inputs["input_ids"], labels=labels)
    loss = outputs.loss
    return loss

# Fine-tune the sentiment model (simplified for demonstration)
fine_tune_sentiment_analysis(sentiment_df)
```

```
tensor(0.7194, grad_fn=<NllLossBackward0>)
```

```
app = Flask(__name__)

# Chatbot endpoint
@app.route('/chatbot', methods=['POST'])
def chatbot():
    input_text = request.json['text']
    tokens = chatbot_tokenizer(input_text, return_tensors='pt')
    response_ids = chatbot_model.generate(tokens['input_ids'])
    response = chatbot_tokenizer.decode(response_ids[0], skip_special_tokens=True)
    return jsonify({'response': response})

# Summarization endpoint
@app.route('/summarize', methods=['POST'])
def summarize():
    input_text = request.json['text']
    inputs = summarization_tokenizer(input_text, return_tensors='pt', padding=True, truncation=True)
    summary_ids = summarization_model.generate(inputs["input_ids"])
    summary = summarization_tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return jsonify({'summary': summary})
```

```
20
# Sentiment analysis endpoint
@app.route('/sentiment', methods=['POST'])
def sentiment():
    input_text = request.json['text']
    inputs = sentiment_tokenizer(input_text, return_tensors='pt', padding=True, truncation=True)
    outputs = sentiment_model(**inputs)
    sentiment_score = torch.argmax(outputs.logits, dim=1).item()
    sentiment = 'Positive' if sentiment_score == 1 else 'Negative'
    return jsonify({'sentiment': sentiment})

if __name__ == '__main__':
    app.run(debug=True)
```

```
 * Serving Flask app '__main__'
 * Debug mode: on
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with stat
```

```
!pip install flask flask-ngrok

```

```
Requirement already satisfied: flask in /usr/local/lib/python3.11/dist-packages (3.1.0)
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: Werkzeug>=3.1 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.6)
Requirement already satisfied: itsdangerous>=2.2 in /usr/local/lib/python3.11/dist-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.9 in /usr/local/lib/python3.11/dist-packages (from flask) (1.9.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from flask-ngrok) (2.32.3)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2>=3.1.2->flask) (3.0.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (2025.1.31)
Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Installing collected packages: flask-ngrok
Successfully installed flask-ngrok-0.0.25
```

```
from flask import Flask, request, jsonify
from flask_ngrok import run_with_ngrok  # Import ngrok for tunneling

app = Flask(__name__)
run_with_ngrok(app)  # Enables ngrok

@app.route("/", methods=["GET"])
def home():
    return "Flask app is running!"

@app.route("/chatbot", methods=["POST"])
def chatbot():
    data = request.json
    user_input = data.get("text", "")
    response = {"reply": f"You said: {user_input}"}
    return jsonify(response)

if __name__ == "__main__":
    app.run()
```

```
The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/usr/local/lib/python3.11/dist-packages/requests/adapters.py", line 667, in send
    resp = conn.urlopen(
           ^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/urllib3/connectionpool.py", line 841, in urlopen
    retries = retries.increment(
              ^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/urllib3/util/retry.py", line 519, in increment
    raise MaxRetryError(_pool, url, reason) from reason  # type: ignore[arg-type]
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
urllib3.exceptions.MaxRetryError: HTTPConnectionPool(host='localhost', port=4040): Max retries exceeded with url: /api/tunnels (Cause

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3.11/threading.py", line 1045, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.11/threading.py", line 1401, in run
    self.function(*self.args, **self.kwargs)
  File "/usr/local/lib/python3.11/dist-packages/flask_ngrok.py", line 70, in start_ngrok
    ngrok_address = _run_ngrok()
                    ^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/flask_ngrok.py", line 35, in _run_ngrok
    tunnel_url = requests.get(localhost_url).text  # Get the tunnel information
                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/requests/api.py", line 73, in get
    return request("get", url, params=params, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/requests/api.py", line 59, in request
    return session.request(method=method, url=url, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/requests/sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/requests/sessions.py", line 703, in send
    r = adapter.send(request, **kwargs)
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/requests/adapters.py", line 700, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPConnectionPool(host='localhost', port=4040): Max retries exceeded with url: /api/tunnels (Ca
```