

```
!pip install pydub openai-whisper transformers langdetect googletrans==4.0.0-rc1 nltk
!apt install ffmpeg
```

```
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting openai-whisper
  Downloading openai-whisper-20240930.tar.gz (800 kB)
  800.5/800.5 kB 8.1 MB/s eta 0:00:00
```

Double-click (or enter) to edit

```
from pydub import AudioSegment
```

```
audio = AudioSegment.from_file("/content/meeting_audio.mp3", format="mp3")
audio.export("/content/meeting_audio.wav", format="wav")
```

```
import whisper
```

```
model = whisper.load_model("base")
result = model.transcribe("/content/meeting_audio.wav")
transcript = result["text"]
```

```
print(transcript)
```

```
from transformers import pipeline
import torch
```

```
summarizer = pipeline("summarization", model="facebook/bart-large-cnn", device=0 if torch.cuda.is_available() else -1)
max_transcript_length = 1000
if len(transcript) > max_transcript_length:
    transcript = transcript[:max_transcript_length]
    print(f"Transcript truncated to {max_transcript_length} characters.")
if not transcript.strip():
    summary = "Transcript is empty or too short to summarize."
else:
    summary = summarizer(transcript, max_length=150, min_length=30, do_sample=False)[0]['summary_text']
    print("Summary:", summary)
print("Summary:", summary)
```

```
/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 secret
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% 1.58k/1.58k [00:00<00:00, 119kB/s]
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download.
```

```
model.safetensors: 100% 1.63G/1.63G [00:14<00:00, 181MB/s]
```

```
generation_config.json: 100% 363/363 [00:00<00:00, 31.2kB/s]
```

```
vocab.json: 100% 899k/899k [00:00<00:00, 5.34MB/s]
```

```
merges.txt: 100% 456k/456k [00:00<00:00, 14.0MB/s]
```

```
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 17.8MB/s]
```

```
Device set to use cpu
Transcript truncated to 1000 characters.
```

```
Summary: The arm of Springfield acknowledges that we are gathered on ancestral lands. Treaty one territory, traditional territory of the
Summary: The arm of Springfield acknowledges that we are gathered on ancestral lands. Treaty one territory, traditional territory of the
```

```
from transformers import pipeline
```

```
sentiment_pipeline = pipeline("sentiment-analysis")
sentiment = sentiment_pipeline(transcript)
print("Overall Sentiment:", sentiment)
```

➡ No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>)
Using a pipeline without specifying a model name and revision in production is not recommended.

```
config.json: 100% 629/629 [00:00<00:00, 21.9kB/s]
```

Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the 'hf_xet' package.
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download.

```
model.safetensors: 100% 268M/268M [00:02<00:00, 138MB/s]
```

```
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 3.02kB/s]
```

```
vocab.txt: 100% 232k/232k [00:00<00:00, 16.9MB/s]
```

```
Device set to use cpu
```

```
Overall Sentiment: [{'label': 'POSITIVE', 'score': 0.9992223978042603}]
```

```
qa_pipeline = pipeline("question-answering")
```

```
question = "What was discussed in the meeting?"
```

```
qa_result = qa_pipeline(question=question, context=summary)
print("Q&A Result:", qa_result)
```

➡ No model was supplied, defaulted to distilbert/distilbert-base-cased-distilled-squad and revision 564e9b5 (<https://huggingface.co/distilbert/distilbert-base-cased-distilled-squad>)
Using a pipeline without specifying a model name and revision in production is not recommended.

```
config.json: 100% 473/473 [00:00<00:00, 29.7kB/s]
```

Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the 'hf_xet' package.
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download.

```
model.safetensors: 100% 261M/261M [00:01<00:00, 125MB/s]
```

```
tokenizer_config.json: 100% 49.0/49.0 [00:00<00:00, 2.38kB/s]
```

```
vocab.txt: 100% 213k/213k [00:00<00:00, 14.5MB/s]
```

```
tokenizer.json: 100% 436k/436k [00:00<00:00, 23.8MB/s]
```

```
Device set to use cpu
```

```
Q&A Result: {'score': 0.2512718141078949, 'start': 77, 'end': 97, 'answer': 'Treaty one territory'}
```

```
from googletrans import Translator
```

```
translator = Translator()
```

```
translated = translator.translate(summary, dest='fr') # 'fr' for French
```

```
print("Translated Summary:", translated.text)
```

➡ Translated Summary: Le bras de Springfield reconnaît que nous sommes rassemblés sur des terres ancestrales.Traité One Territory, Territoire

```
import smtplib
```

```
from email.message import EmailMessage
```

```
SENDER_EMAIL = "poojitha307.vintha@gmail.com"
```

```
APP_PASSWORD = "gvfz uitg gduh sskb"
```

```
def send_email(summary, recipient):
```

```
    msg = EmailMessage()
```

```
    msg.set_content(summary)
```

```
    msg["Subject"] = "Meeting Summary"
```

```
    msg["From"] = SENDER_EMAIL
```

```
    msg["To"] = recipient
```

```
    try:
```

```
        with smtplib.SMTP_SSL("smtp.gmail.com", 465) as smtp:
```

```
            smtp.login(SENDER_EMAIL, APP_PASSWORD)
```


```
            smtp.send_message(msg)
```

```
            print("Email sent successfully!")
```

```
    except Exception as e:
```

```
        print("Failed to send email:", e)
```

```
summary_text = "Le bras de Springfield reconnaît que nous sommes rassemblés sur des terres ancestrales.Traité One Territory, Territoire trad  
recipient_email = "poojitha307.vintha@gmail.com"  
send_email(summary_text, recipient_email)
```

 Email sent successfully!

Start coding or [generate](#) with AI.