```
# Google Driveをマウント
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# GPU確認（任意）
!nvidia-smi

# 依存関係インストール
!pip -q install faster-whisper==1.0.3 ffmpeg-python==0.2.0 pydub==0.25.1

# ffmpeg 本体（音声抽出に使用）
!apt -y install ffmpeg
```

```
Tue Aug 12 06:07:58 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   60C    P8             10W /   70W |       0MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 51.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 70.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 38.6/38.6 MB 68.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.5/16.5 MB 126.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.0/46.0 kB 4.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.8/86.8 kB 7.7 MB/s eta 0:00:00
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ffmpeg is already the newest version (7:4.4.2-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
```

```
INPUT_MP4 = "/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/data/人工知能研究の第一人者らが国会で参考人質疑 松尾豊教授・
```

```
!ffmpeg -y -i "$INPUT_MP4" -ac 1 -ar 16000 -vn -c:a pcm_s16le /content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/audio.
```

```
ffmpeg version 4.4.2-0ubuntu0.22.04.1 Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 11 (Ubuntu 11.2.0-19ubuntu1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.22.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/data/人工知能研究の第一人者
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.76.100
  Duration: 02:45:09.05, start: 0.000000, bitrate: 2034 kb/s
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709), 1920x1080 [SAR 1:1 DAR 16:9], 1898 kb/s,
    Metadata:
      handler_name    : VideoHandler
      vendor_id       : [0][0][0][0]
  Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 128 kb/s (default)
    Metadata:
      handler_name    : ISO Media file produced by Google Inc.
      vendor_id       : [0][0][0][0]
Stream mapping:
  Stream #0:1 -> #0:0 (aac (native) -> pcm_s16le (native))
Press [q] to stop, [?] for help
Output #0, wav, to '/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/audio.wav':
```

```
        Metadata:
          major_brand     : isom
          minor_version   : 512
          compatible_brands: isomiso2avc1mp41
          ISFT            : Lavf58.76.100
        Stream #0:0(eng): Audio: pcm_s16le ([1][0][0][0] / 0x0001), 16000 Hz, mono, s16, 256 kb/s (default)
          Metadata:
            handler_name    : ISO Media file produced by Google Inc.
            vendor_id       : [0][0][0][0]
            encoder         : Lavc58.134.100 pcm_s16le
      size=  309658kB time=02:45:09.06 bitrate= 256.0kbits/s speed= 535x
      video:0kB audio:309658kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.000025%


from faster_whisper import WhisperModel

MODEL_SIZE = "large-v3"     # 精度重視（他: "medium", "large-v2" など）
model = WhisperModel(MODEL_SIZE, device="cuda", compute_type="float16")

segments, info = model.transcribe(
    "/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/audio.wav",
    language="ja",
    vad_filter=True,  # 無音区間で切る
    vad_parameters=dict(min_silence_duration_ms=500),
    condition_on_previous_text=True,
    initial_prompt="これは日本の国会参考人質疑の議事録です。敬体・常用漢字を基本に整えます。"
)

# TXT, SRT, VTT を保存
import math, os, json

os.makedirs("/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out", exist_ok=True)
txt_path = "/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/transcript.txt"
srt_path = "/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/transcript.srt"
vtt_path = "/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/transcript.vtt"
json_path = "/content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/transcript.json"

def to_timestamp(t):
    h = int(t//3600); m = int((t%3600)//60); s = t%60
    return f"{h:02d}:{m:02d}:{s:06.3f}".replace('.', ',')

all_segments = []
with open(txt_path, "w", encoding="utf-8") as ft, ¥
     open(srt_path, "w", encoding="utf-8") as fs, ¥
     open(vtt_path, "w", encoding="utf-8") as fv:

    fv.write("WEBVTT¥n¥n")
    for i, seg in enumerate(segments, start=1):
        all_segments.append(
            dict(id=i, start=seg.start, end=seg.end, text=seg.text.strip())
        )
        # TXT
        ft.write(seg.text.strip() + "¥n")
        # SRT
        fs.write(f"{i}¥n{to_timestamp(seg.start)} --> {to_timestamp(seg.end)}¥n{seg.text.strip()}¥n¥n")
        # VTT
        fv.write(f"{to_timestamp(seg.start).replace(',','.')} --> {to_timestamp(seg.end).replace(',','.')}¥n{seg.text.strip()}

# JSON（タイムスタンプ付き）
with open(json_path, "w", encoding="utf-8") as fj:
    json.dump({"language": info.language, "segments": all_segments}, fj, ensure_ascii=False, indent=2)

print("出力:", txt_path, srt_path, vtt_path, json_path)


出力: /content/drive/MyDrive/AI音声認識/03-国会参考人質疑/out/transcript.txt /content/drive/MyDrive/AI音声認識/03-国会参
```