Assembly-first Speaker Tagging Pipeline (Colab Guide)

Updated: 2025-11-11 01:13

Goal

- Start from raw audio in Colab.
- STT with Whisper (word timestamps).
- Diarization with Senko (fast path) or NVIDIA NeMo (alt path).
- Merge STT words ↔ speaker segments (max-overlap).
- (Optional) Role/behavior tagging (host, requester, backchanneler...).
- Export JSON + SRT; keep outputs privacy-friendly.

1) Colab Runtime

- Runtime → Change runtime type → GPU (T4/A100 OK).
- Ensure ffmpeg available (Colab default).

2) Project Structure

- /content/audio/meeting.m4a # input
- /content/work/outputs/ # JSON/SRT artifacts
- /content/work/cache/ # temp files

3) Install (Core)

```
!pip -q install --upgrade openai-whisper faster-whisper torchaudio librosa soundfile pandas numpy
```

4) STT (Whisper)

```
import whisper
model = whisper.load_model("medium")  # or "large-v3"
res = model.transcribe("/content/audio/meeting.m4a", language="ko", word_timestamps=True)
# Persist words: list of dicts with keys 'text', 'start', 'end'
```

5) Diarization (Option A: Senko, recommended fast path)

```
!pip -q install "git+https://github.com/narcotic-sh/senko.git"
import senko
d = senko.Diarizer(device="auto", warmup=True, quiet=False)
wav = "/content/work/cache/meeting_16k.wav"  # use ffmpeg to resample mono 16k
result = d.diarize(wav)
# result['merged_segments']: list of segments with start/end/speaker_id
```

6) Diarization (Option B: NVIDIA NeMo)

```
!pip -q install nemo_toolkit[asr]
# Use the official Colab or ClusteringDiarizer config; export RTTM/JSON.
```

7) Merge (Max-Overlap)

- For each word (start,end), find speaker segment with maximum time overlap.
- Group into utterances per speaker; join words.

8) Role/Behavior Tagging (optional)

- Features: turns per speaker, avg turn length, question/request markers, backchannel ratio, topic similarity (sentence embeddings).
- Heuristics → labels: Host, Primary Contributor, Requester, Backchanneler, Contrarian, Returnee, Off-topic.

9) Outputs

- final_transcript.json: list of items {speaker, start, end, text, role?}
- meeting.srt: [index]\nHH:MM:SS,ms --> HH:MM:SS,ms\n[speaker/role] text
- diagnostics.csv: per-speaker stats

10) Using AssemblyAI diarization text (existing file)

- Write a parser that extracts (speaker_label, start, end, text) → standard 'turns' schema.
- You can skip running diarization and still test Merge + Role tagging with Whisper words + parsed turns.

11) QA & Thresholds

- Role confidence tau_role (0.6–0.8) sweep on a dev clip.
- For short interjections (“■/■/■”), inherit neighbor speaker.