Summary Report of Work Completed on Day 20

□ **Speaker Expertise Recognition:**□ **Speaker Expertise Recognition:**

- Focused on identifying and attributing audio segments to specific speakers based on their expertise, ensuring that each speaker’s contributions were properly recognized.
- Used speaker diarization techniques with tools like pyannote.audio to detect speaker boundaries and label speaker segments accurately.
- Applied custom models or pre-trained ones to classify the expertise of the speakers by analyzing their speaking style, content knowledge, or past interactions.

□ **Transcript Generation for Large Audio Files:**

- Worked on breaking large audio files into smaller, manageable segments to generate more accurate transcripts.
- Encountered issues with maintaining context and continuity between segments while generating transcripts, leading to incomplete or fragmented outputs.
- Applied strategies like:
  - **Overlap Segments:** Ensured there was some overlap between adjacent segments to maintain contextual integrity.
  - **Batch Processing:** Split the audio file into batches and used parallel processing to improve speed and efficiency.
  - **Whisper and Other Models:** Used Whisper or similar models for transcription, optimizing them to handle longer audio durations effectively.

□ **Optimization Challenges:**

- Despite creating segments, the generation of transcripts for large files was still slow or produced incomplete results.
- Encountered memory management issues when processing long files and tried to optimize by handling audio chunks in memory-efficient ways.

- Focused on identifying and attributing audio segments to specific speakers based on their expertise, ensuring that each speaker’s contributions were properly recognized.
- Used speaker diarization techniques with tools like pyannote.audio to detect speaker boundaries and label speaker segments accurately.
- Applied custom models or pre-trained ones to classify the expertise of the speakers by analyzing their speaking style, content knowledge, or past interactions.

□ **Transcript Generation for Large Audio Files:**

- Worked on breaking large audio files into smaller, manageable segments to generate more accurate transcripts.

Encountered issues with maintaining context and continuity be□ **Speaker Expertise Recognition:**

- Focused on identifying and attributing audio segments to specific speakers based on their expertise, ensuring that each speaker’s contributions were properly recognized.

- Used speaker diarization techniques with tools like pyannote.audio to detect speaker boundaries and label speaker segments accurately.
- Applied custom models or pre-trained ones to classify the expertise of the speakers by analyzing their speaking style, content knowledge, or past interactions.

□ **Transcript Generation for Large Audio Files:**

- Worked on breaking large audio files into smaller, manageable segments to generate more accurate transcripts.
- Encountered issues with maintaining context and continuity between segments while generating transcripts, leading to incomplete or fragmented outputs.
- Applied strategies like:
  - **Overlap Segments:** Ensured there was some overlap between adjacent segments to maintain contextual integrity.
  - **Batch Processing:** Split the audio file into batches and used parallel processing to improve speed and efficiency.
  - **Whisper and Other Models:** Used Whisper or similar models for transcription, optimizing them to handle longer audio durations effectively.

□ **Optimization Challenges:**

- Despite creating segments, the generation of transcripts for large files was still slow or produced incomplete results.
- Encountered memory management issues when processing long files and tried to optimize by handling audio chunks in memory-efficient ways.

- tween segments while generating transcripts, leading to incomplete or fragmented outputs.

- Applied strategies like:

  - **Overlap Segments:** Ensured there was some overlap between adjacent segments to maintain contextual integrity.

  - **Batch Processing:** Split the audio file into batches and used parallel processing to improve speed and efficiency.

  - **Whisper and Other Models:** Used Whisper or similar models for transcription, optimizing them to handle longer audio durations effectively.

□ **Optimization Challenges:**

- Despite creating segments, the generation of transcripts for large files was still slow or produced incomplete results.

- Encountered memory management issues when processing long files and tried to optimize by handling audio chunks in memory-efficient ways.