Leveraging AI for Japanese Language Learning Analytics from Video Data

John AOUSSOU
Senior Engineer at GL Navigation Inc.
2023 - present

Last updated: May 5, 2025

1 Introduction

In 2023, GL Navigation sought to explore the potential value hidden within its extensive collection of video recordings. These recordings primarily consisted of one-on-one Japanese language lessons and group discussion sessions involving native teachers and foreign students. The goal was to investigate what insights could be derived from this data using modern data analysis techniques, particularly leveraging advancements in Artificial Intelligence (AI). The key objectives were to establish methods for more objectively assessing student proficiency and progress and to identify opportunities for enhancing the learning experience and feedback mechanisms provided to students.

2 Problem Statement

The primary challenge lay in the unstructured nature of the video data and the lack of consistent, machine-readable metadata. Specifically:

- **Speaker Diarization:** Determining *who spoke when* by segmenting audio based on speaker turns was essential but complicated by varying audio quality. Accurate diarization was a prerequisite for subsequent analysis.
- **Content Transcription:** Converting spoken Japanese (often with non-native accents) into accurate text (Speech-to-Text, STT) was required for analysis.
- **Speaker Identification:** Identifying *which specific person* (teacher vs. student, or individuals across multiple sessions) corresponded to each speaker segment identified by diarization was crucial. This was hampered by the lack of reliable participant metadata and the need to recognize individuals in group settings.
- **Metadata Scarcity:** There was no centralized booking system or database linking videos to specific participants with unique IDs. Information often had to be inferred manually from video titles or recording dates, complicating speaker identification.
- **Scalability:** Processing a large volume of video data for diarization, STT, and identification required significant computational resources and efficient pipelines, especially considering the potential duration of AI tasks.

3 Methodology and Approach

3.1 Initial Prototyping: Diarization and STT

The first step involved building a prototype to assess feasibility using state-of-the-art, open-source models for **speaker diarization** (segmenting speech by speaker turns) and Speech-to-Text (STT). At the time of selection, these free models demonstrated accuracy levels comparable or superior to available commercial alternatives, even when benchmarked against internal data.

Crucially, the prototype's accuracy for both diarization and STT was validated against the company's specific data. While non-native speaker accents occasionally impacted STT performance, the chosen models proved sufficiently robust for the data to yield usable transcriptions and speaker timelines (speaker A spoke from time X to Y, speaker B from Y to Z, etc.).

3.2 Infrastructure Development for Scalability

Initial processing times for the ML models highlighted a bottleneck. To enable faster iteration and handle the potential volume of data, a dedicated workstation was designed and built. The focus was on maximizing computational performance for ML tasks while maintaining cost-effectiveness. The core component selected was a high-performance GPU (**NVIDIA GeForce RTX 4080**) available at the time, providing significant acceleration for model inference.

3.3 Local Batch Processing Workflow

Given the computational demands of the AI models, the initial processing workflow was centered around the dedicated local workstation. While the company's video data resided on `Box`, establishing a reliable, direct interface via its API presented significant challenges during this phase. Consequently, processing primarily involved accessing video files on the local machine (e.g., through synced folders or manual downloads). Batch processing scripts were developed to run on the workstation, automating the application of diarization and STT models to multiple video files efficiently. This local workflow streamlined the transformation from raw video into structured data outputs (diarized transcripts with speaker labels and timestamps), ready for subsequent analysis.

3.4 Speaker Identification using Voice Banks

To address the challenge of identifying who the speakers were (specifically, differentiating teachers from students initially), a **speaker identification** component was developed. This relied on creating a "voice bank":

- **ID Inference:** Custom logic was implemented to parse video titles and recording dates, attempting to infer likely participants for each session.
- **Teacher Voice Bank:** For known teachers, voice samples were manually extracted and stored. During processing of one-on-one lessons, after diarization identified speaker segments, the system attempted to match voices against the teacher bank. If a match was found, that speaker segment was labeled as "Teacher," and the other primary speaker segment was labeled as "Student." This provided the necessary role context.

3.5 Handling Group Discussions: Expanding Speaker Identification

Analyzing group discussions required extending the speaker identification capability to multiple, potentially unknown participants.

- **Inefficiencies of Initial Manual Approach:** An initial workflow relied on manually created spreadsheets provided by contractors, which mapped timestamps to specific speakers for voice sample extraction. While this method could function after manual correction, the inconsistent formatting and accuracy of contractor inputs made the process highly inefficient and labor-intensive, necessitating frequent manual intervention to ensure data validity before voice samples could be reliably extracted.

- **Voice Sample Collection and Group Identification:** To enable speaker identification in group discussions, a preparatory process was required to build the student voice bank. Before analyzing a group discussion video, the participants expected in that session needed to be known. For each anticipated student participant, a prior one-on-one lesson video featuring only that specific student and a known teacher was *manually located* in the archive. The system then processed this selected one-on-one video: leveraging the existing teacher voice bank to identify the teacher's segments, it automatically extracted the voice sample of the other speaker (identified as the target student). These pre-collected student voice samples were added to the voice bank. Subsequently, during the analysis of the actual group discussion video, the speaker identification model used this expanded voice bank (containing teachers and the relevant pre-identified students) to match diarized voice segments and identify each participant by their voice signature.

- **Handling Unidentified Speakers:** If a distinct speaker's voice segments during diarization did not match any sample in the expanded voice bank, the system would assign a generic label (e.g., `speaker_1`, `speaker_2`) instead of a specific name. This fallback allowed the analysis pipeline to complete successfully and provide diarization results even when some participants could not be specifically identified.

3.6 Linguistic Analysis

Once accurate transcripts with identified speaker labels were obtained, linguistic analysis was performed using `MeCab`, a standard Japanese morphological analyzer, along with other calculations. Key metrics, now attributable to specific individuals or roles, were extracted:

- Speaker turn duration and frequency (e.g., which student spoke the most).

- Vocabulary richness analysis per participant.

- Speaking rate analysis (e.g., measuring speed in syllables or morae per unit of time).

These metrics demonstrated a reliable correlation with assessed student Japanese proficiency levels, validating their potential for educational feedback.

4 Transition to Production

As the demand for analysis grew, the limitations of the initial prototype became apparent. It was susceptible to breaking with edge cases in the messy data and difficult to integrate seamlessly with the company's existing, non-database data management practices.

Recognizing the proven value and the need for a more robust solution, a proposal was made to develop the prototype into a production-ready internal **web application**. Subsequently:

- **Team Building and Mentorship:** Two junior programmers and one graduate student were initially hired on a part-time basis to contribute to development and research, respectively. Through dedicated mentoring and hands-on project involvement, they played a crucial role in refactoring the codebase, enhancing robustness, and building a stable application architecture. Reflecting their growth and the project's success, these three initial hires have since transitioned into full-time roles within the company.

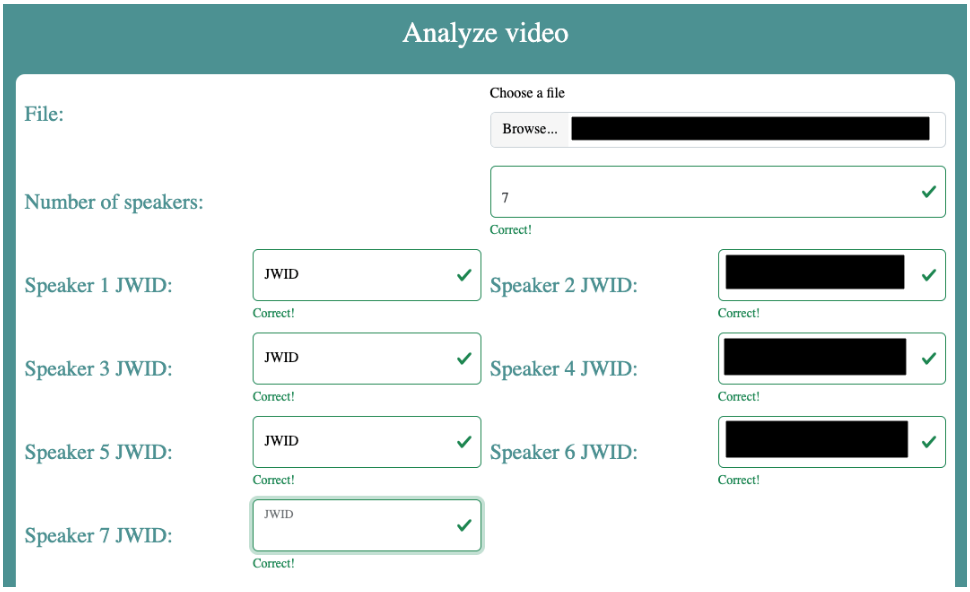
- **Technology Stack:** The application was developed using **Kotlin Multiplatform** to ensure code reusability and future-proof the system for potential expansion to mobile platforms.

- **Cloud Infrastructure and Pipeline:** The backend infrastructure was built on AWS cloud services. It utilizes AWS Lambda for API requests and lightweight processing, while compute-intensive machine learning tasks (like diarization, STT, and speaker identification) run on Amazon SageMaker EC2 instances. Recognizing that some AI modules can be slow, the architecture operates as an asynchronous, **trigger-based pipeline**. The completion of one processing stage (e.g., video upload, diarization) automatically triggers the next relevant step. This event-driven approach efficiently manages potentially long-running tasks, optimizes resource utilization, and ensures system responsiveness.

- **Development Practices:** To ensure robust development and deployment, separate development and production environments were established. A CI/CD pipeline was integrated with the Git repository, enabling automated builds, testing, and deployments triggered simply by pushing code updates to designated branches.

- **Team Expansion:** Recently, an additional part-time team member has been brought on board to further support ongoing development and research efforts.

5 Results and Impact



Analyze video

File: Choose a file
Browse... [redacted]

Number of speakers: 7 ✓
Correct!

Speaker 1 JWID: JWID ✓
Correct!

Speaker 2 JWID: [redacted] ✓
Correct!

Speaker 3 JWID: JWID ✓
Correct!

Speaker 4 JWID: [redacted] ✓
Correct!

Speaker 5 JWID: JWID ✓
Correct!

Speaker 6 JWID: [redacted] ✓
Correct!

Speaker 7 JWID: JWID ✓
Correct!

Figure 1: video analysis UI

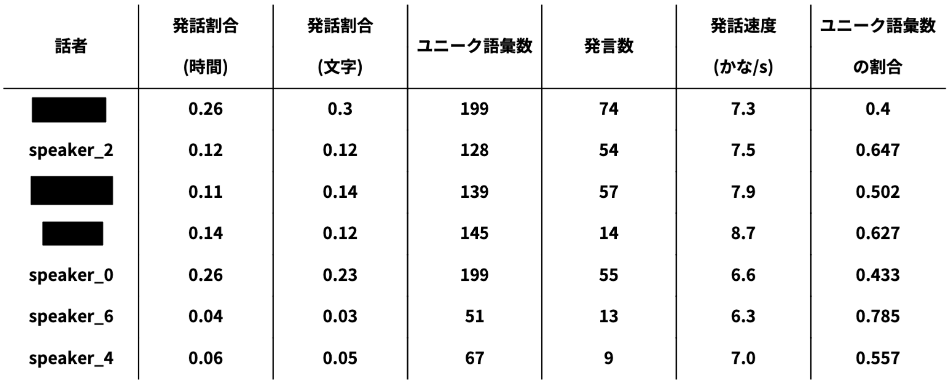
The project successfully transitioned from an exploratory phase into a functional internal tool with a growing dedicated team.

- A production-grade **web application**, built with Kotlin Multiplatform and hosted on AWS using a trigger-based pipeline architecture, performing automated diarization, STT, speaker identification, and analysis of Japanese lesson videos was developed and deployed internally.

- The application provides reliable, automated metrics correlated with student language proficiency, offering valuable insights for the company's educational services.

- The developed system overcame significant data management and speaker identification hurdles through custom logic and voice banking techniques.

- Technical leadership was demonstrated through infrastructure design (local workstation and asynchronous AWS backend), ML pipeline development (diarization, STT, identification), adherence



話者	発話割合 (時間)	発話割合 (文字)	ユニーク語彙数	発言数	発話速度 (かな/s)	ユニーク語彙数 の割合
[redacted]	0.26	0.3	199	74	7.3	0.4
speaker_2	0.12	0.12	128	54	7.5	0.647
[redacted]	0.11	0.14	139	57	7.9	0.502
[redacted]	0.14	0.12	145	14	8.7	0.627
speaker_0	0.26	0.23	199	55	6.6	0.433
speaker_6	0.04	0.03	51	13	6.3	0.785
speaker_4	0.06	0.05	67	9	7.0	0.557

Figure 2: example of metrics calculated

to modern development practices (CI/CD), and the successful hiring and mentoring of an initial part-time team, fostering their growth into full-time employees and subsequently expanding the team.

The web application is currently in active use within the company, supported by a dedicated team, and contributing to data-driven improvements in language education.

6 Conclusion

This project showcases the successful application of AI/ML techniques to derive meaningful insights from unstructured video data in an educational context. Starting with an open-ended request, challenges related to data quality, speaker diarization, speaker identification, and processing scale were systematically addressed through prototyping, custom infrastructure development (utilizing an RTX 4080 GPU locally and an asynchronous, trigger-based AWS Lambda/SageMaker pipeline in the cloud), and innovative data handling and voice banking strategies. The initiative culminated in a valuable internal production web application, developed using Kotlin Multiplatform, managed via CI/CD, and deployed on AWS. Furthermore, it demonstrated the ability to lead technical projects from conception through deployment while building and nurturing a technical team, successfully developing initial part-time hires into full-time contributors and continuing to expand the team's capabilities.

7 Future Work

Building on the success of the current system and the established team, several avenues for future development are envisioned:

- **Team Development and Leadership:** Continue mentoring the team members, with a specific focus on developing leadership skills within the group to ensure long-term project ownership, sustainability, and innovation capacity.
- **Infrastructure Enhancement - Data Storage Migration:** Transition the data storage strategy away from reliance solely on Box folders and inferred metadata. The proposed architecture involves migrating raw media files (videos, extracted audio) to **Amazon S3** for scalable object storage, while utilizing **MongoDB** for storing metadata, analysis results, participant information, and voice bank data. This migration aims to significantly reduce manual data handling (especially related to participant identification and voice sample management), improve data integrity and accessibility, streamline the processing pipeline, enhance query capabilities, and increase overall system scalability for future growth.

- **Evaluate AI Model Hosting vs. Managed APIs:** The initial prototype ran AI models directly on the dedicated local workstation, which was feasible for development. The current production system hosts these models on `Amazon SageMaker EC2` instances. While the application serves internal users, the processing volume is non-trivial, and obtaining results can sometimes be slow due to the need to scale these compute instances in order to minimize cost. Consequently, we plan to evaluate the trade-offs of migrating certain core AI tasks (particularly STT, potentially diarization) from our self-hosted models on SageMaker to managed cloud AI service APIs (e.g., `AWS Transcribe`, `Google Cloud Speech-to-Text`). This investigation will compare factors like processing speed (latency), accuracy on our specific data, ease of maintenance (removing the need to manage model infrastructure), and overall cost-effectiveness at our current and projected usage levels.

- **Leveraging Large Language Models (LLMs):** Explore the integration of Large Language Models (LLMs) to unlock new analytical capabilities beyond the current statistical metrics. Potential applications include generating automated summaries of lesson interactions, performing more nuanced sentiment analysis on student or teacher speech, providing AI-driven pedagogical feedback suggestions based on conversational patterns, or enabling natural language querying of the analyzed lesson data.

- **Explore Multimodal Analysis (Emotion Recognition):** Expand the system's capabilities into multimodal analysis by incorporating the visual data stream from video recordings. Leveraging my personal knowledge and expertise in computer vision, we plan to investigate techniques for analyzing participant facial expressions and potentially other visual cues during group discussions. The objective is to automatically infer emotional states (such as engagement, confusion, or confidence), providing richer insights into participant experience and group interaction dynamics than is possible through audio analysis alone.