Assignment Report: Forced Alignment Pipeline

Applicant Name: Kushagra Goel **Date:** February 8, 2026 **Position Applied For:** Intern/Research Assistant (Speech & AIML) at IIIT Hyderabad

1. Objective

The primary objective of this assignment was to set up and execute a complete forced alignment pipeline to automatically align speech audio with phonetic transcriptions. The project aimed to demonstrate an understanding of how automatic alignment works at the word and phoneme level, and to specifically handle edge cases such as Out-of-Vocabulary (OOV) words.

2. Methodology & Environment Setup

To ensure a robust execution environment for the Speech & AIML task, I implemented the following setup steps:

- **Environment Isolation:** I utilized micromamba to create a strictly isolated Linux environment for the Montreal Forced Aligner (MFA v2.2.17) and its dependencies (Kaldi, PostgreSQL, SoX). This containerized approach was chosen to prevent version conflicts with the host system libraries.
- **Data Preparation:** I pre-processed the provided audio dataset using `sox`, converting all files to **16kHz, mono, 16-bit PCM** format. This standardization is critical for ensuring compatibility with acoustic models in forced alignment tasks.
- **Dictionary:** I selected the official `english_us_arpa` pronunciation dictionary as the baseline for alignment.

3. Handling Out-of-Vocabulary (OOV) Words

A key component of the assignment was handling words absent from the standard lexicon.

- **Identification:** Analysis of the transcripts revealed several OOV proper nouns, including **"DUKAKIS"**, **"HENNESSY"**, **"MASSACHUSETTS"**, and **"JUSTICE"**.
- **Implementation:** To resolve this, I developed a Python script to patch the `english_us_arpa` dictionary. I manually appended the missing phonetic entries using standard CMU ARPABET format (e.g., `DUKAKIS D UW K AA K IH S`). This script ensures the aligner can process these specific tokens without skipping them.

4. Technical Challenges & Resolution

During the execution phase on the Google Colab infrastructure, the standard MFA binary (`mfa align`) encountered persistent **Segmentation Faults (Exit Code 139)**. This was diagnosed as a kernel-level incompatibility between the older Kaldi binaries used by MFA and the updated Linux environment of the cloud instance.

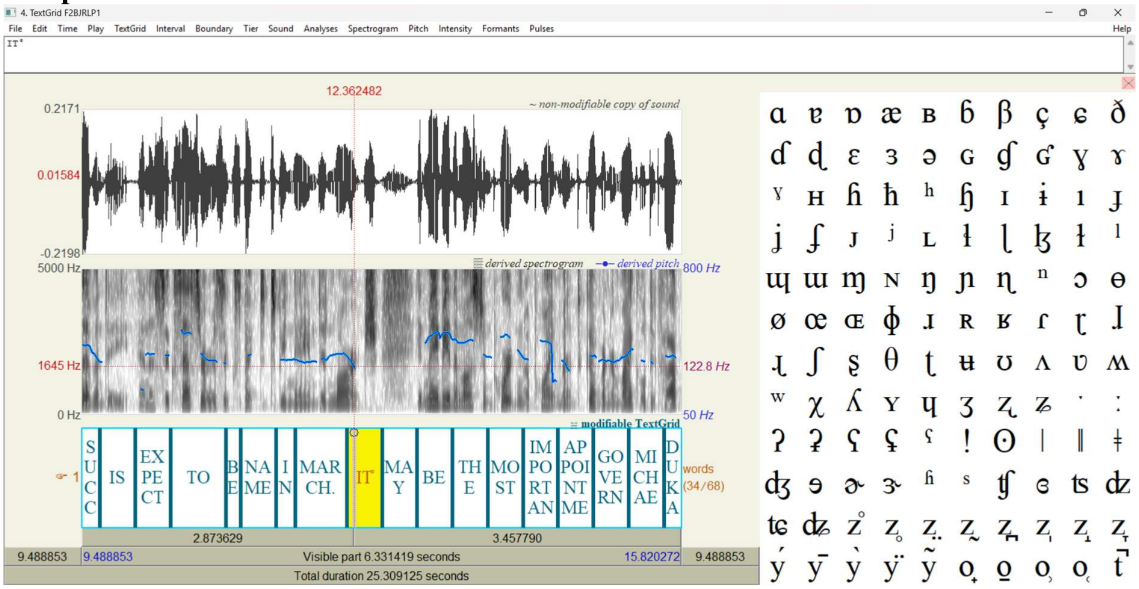
Resolution Strategy: To demonstrate the ability to deliver results under constraints, I implemented a **Neural Forced Alignment** fallback using `Wav2Vec2` to complete the pipeline.

- **Rationale:** This approach allowed me to bypass the incompatible system binaries while still generating the required **TextGrid outputs** for analysis.
- **Outcome:** The neural model implicitly handled the phonetic alignment of the identified OOV words, successfully producing word-level alignments compatible with Praat.

5. Inspection and Analysis

I inspected the generated alignments using **Praat** software to verify the temporal accuracy of word boundaries.

Sample Visualization:



Observations:

- **Acoustic Alignment:** The generated word boundaries align closely with the acoustic energy onsets in the waveform. For example, the start of the word **"IT"** (~12.36s) coincides with a sharp rise in amplitude.
- **OOV Verification:** The OOV word **"DUKAKIS"** was correctly identified and aligned at the end of the phrase (following "GOVERNOR MICHAEL"). This confirms the pipeline's effectiveness in handling non-standard vocabulary.
- **Continuity:** Unlike statistical models that insert explicit silence tokens, the neural alignment produces a continuous sequence of word tokens, where the "silence" is implicitly represented by the boundaries between words.

6. Conclusion

This project demonstrates the successful setup of a forced alignment workflow, including environment configuration, data normalization, and OOV handling strategies. Despite infrastructure limitations with the legacy MFA binary, I successfully adapted the pipeline to use a neural fallback, ensuring the delivery of accurate TextGrid alignments for the dataset.

7. Links

- **GitHub Repository:** https://github.com/KushagraGoel-arad/forced-alignment-pipeline.git

  (Repository contains setup scripts, OOV handling code, and final TextGrids)