Forced Alignment using Montreal Forced Aligner (MFA)

1. Introduction

Forced alignment is a technique used in speech processing to automatically align a given audio recording with its corresponding text transcription. The main objective of forced alignment is to determine the exact start and end time of each word and phoneme in the speech signal.

This process is useful in many applications such as speech recognition, speech analysis, language learning systems, and phonetic research. Instead of manually labeling audio, forced alignment allows automatic generation of time-aligned annotations.

In this assignment, Montreal Forced Aligner (MFA) is used to perform forced alignment between speech audio files and their transcripts.

---

2. Objective

The objective of this assignment is to:

- Set up the Montreal Forced Aligner environment.
- Perform forced alignment between speech audio and transcripts.
- Generate word-level and phoneme-level boundaries.
- Handle Out-of-Vocabulary (OOV) words using a G2P model.
- Analyze alignment quality before and after OOV handling.

---

3. Dataset Description

The dataset provided consists of:

- A folder containing **audio files (.wav)**
- A folder containing **transcripts (.txt)**

Each audio file has a corresponding transcript file with the same name.
Each transcript contains the exact spoken content of the audio.

Each audio file represents one utterance spoken by a single speaker.

---

4. Tools and Environment

The following tools were used:

- Montreal Forced Aligner (MFA) version 3.3.4
- Praat (for visualization)
- Windows 10
- Miniconda (for environment management)

A separate conda environment was created for MFA to avoid dependency issues.

---

5. Environment Setup and Installation

A new conda environment was created and MFA was installed using conda-forge.

- conda create -n mfa python=3.9
- conda activate mfa
- mamba install -c conda-forge montreal-forced-aligner

Installation was verified using:

mfa version

---

6. Corpus Preparation

All audio files and transcript files were placed into a single folder called:

corpus/

The structure was:

corpus/

├── file1.wav

├── file1.txt

├── file2.wav

├── file2.txt

...

The corpus was validated using:

- mfa validate corpus english_us_arpa

This step checks for format issues, missing files, and text normalization.

---

7. Models Used

The following pretrained models were used:

Acoustic Model

- english_us_arpa

Pronunciation Dictionary

- english_us_arpa

These models are provided by MFA and are suitable for American English speech.

They were downloaded using:

- mfa model download acoustic english_us_arpa
- mfa model download dictionary english_us_arpa

---

8. First Alignment (Before OOV Handling)

The initial forced alignment was performed using:

- mfa align corpus english_us_arpa english_us_arpa aligned --clean --verbose

This generated TextGrid files in the aligned/ folder.

Each TextGrid file contains:

- A word tier
- A phoneme tier

These files show time boundaries for each word and phoneme.

---

9. OOV (Out-of-Vocabulary) Analysis

During alignment, some words were not found in the dictionary. These are called Out-of-Vocabulary (OOV) words.

MFA generated the following files:

- oov_counts_english_us_arpa.txt

- oovs_found_english_us_arpa.txt
- utterance_oovs.txt

Some example OOV words found were:

- dukakis
- politicize
- maffy
- melnicove

These words were skipped in the first alignment because no pronunciation was available.

---

10. OOV Handling using G2P

To handle OOV words, a Grapheme-to-Phoneme (G2P) model was used.

The G2P model generates phoneme sequences automatically from text.

First, the G2P model was downloaded:

- mfa models download g2p english_us_arpa

Then pronunciations were generated:

- mfa g2p oovs_found_english_us_arpa.txt english_us_arpa generated.dict

This produced a file called generated.dict containing phoneme transcriptions for OOV words.

---

11. Dictionary Expansion

The generated pronunciations were merged with the original dictionary:

type english_us_arpa.dict generated.dict > english_us_arpa_fixed.dict

This new dictionary contains both original and OOV words.

---

12. Final Alignment (After OOV Handling)

The alignment was repeated using the expanded dictionary:

- mfa align corpus english_us_arpa_fixed.dict english_us_arpa aligned_fixed --clean --verbose

This generated improved TextGrid files in:

aligned_fixed/

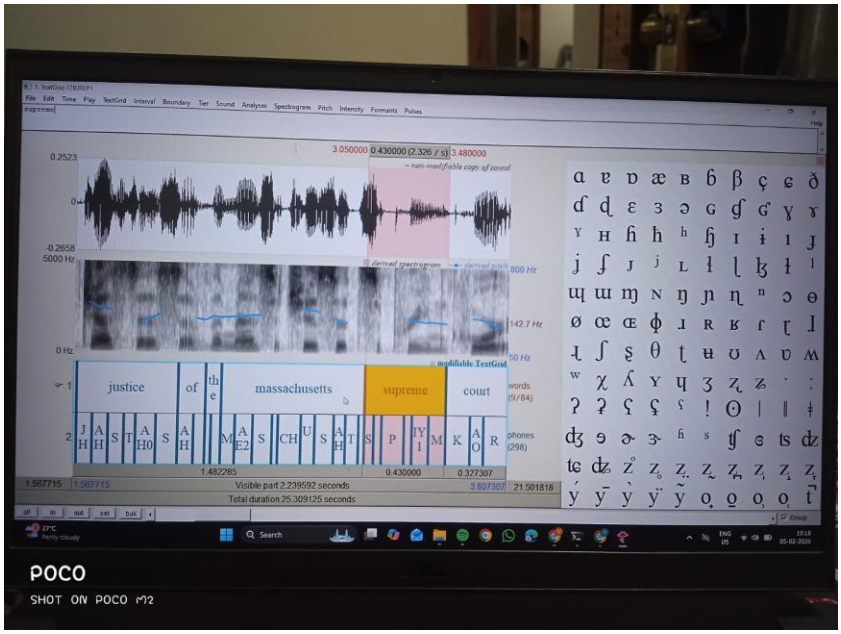
---

13. Sample Alignment Visualization

The generated TextGrid files were opened using **Praat**.

From the visualization, the following were observed:

- Word boundaries aligned correctly with speech.
- Phoneme segmentation matched acoustic transitions.
- Pauses and silence were properly detected.



---

14. Observations: Before vs After OOV Handling

Before OOV Handling

- Several words were missing from the dictionary.
- OOV words were skipped during alignment.

- Some word boundaries were incomplete.

After OOV Handling

- All OOV words were assigned pronunciations.
- All words appeared in final TextGrid files.
- Word and phoneme alignment became complete.
- Overall alignment quality improved significantly.

---

15. Conclusion

In this assignment, a complete forced alignment pipeline was successfully implemented using Montreal Forced Aligner.

The system was able to:

- Automatically align audio and text.
- Generate precise word and phoneme boundaries.
- Handle unknown words using G2P.
- Improve alignment quality through dictionary expansion.

This experiment demonstrates how forced alignment works in real speech processing systems and highlights the importance of pronunciation dictionaries in speech alignment tasks.