Report: Assignment-1

Forced Alignment Using Montreal Forced Aligner (MFA)

Applicant: Syed Khaja Fareed Uddin

1. Objective

The objective of this assignment was to implement a complete forced alignment pipeline using Montreal Forced Aligner (MFA). Forced alignment automatically aligns speech audio with its corresponding transcript at both word and phoneme levels by identifying the start and end times of each unit in the speech signal.

In addition to running alignment, the assignment required identification and handling of Out of Vocabulary (OOV) words and implementation of a systematic solution.

2. Environment Setup

A dedicated Conda environment was created and Montreal Forced Aligner was installed using:

```
conda create -n aligner montreal-forced-aligner
conda activate aligner
```

The following pretrained models were downloaded:

```
mfa model download acoustic english_us_arpa
mfa model download dictionary english_us_arpa
mfa model download g2p english_us_arpa
```

The English US ARPA acoustic model and pronunciation dictionary were used throughout the experiments.

3. Dataset Preparation

The dataset consisted of:

- 6 audio files in WAV format
- 6 corresponding transcript files

The dataset was used exactly as provided. No modifications were made to the audio recordings. The transcript files were changed from .txt to .lab format.

Audio and transcript files were organized in a directory structure compatible with MFA requirements.

4. Baseline Forced Alignment

Baseline alignment was performed using the pretrained dictionary:

```
mfa align dataset english_us_arpa english_us_arpa aligned_baseline
--clean --overwrite
```

Validation was performed using:

```
mfa validate dataset english_us_arpa english_us_arpa
```

Baseline OOV statistics were:

- 29 OOV word types
- 61 total OOV tokens

This indicated that several words in the transcripts were not present in the pronunciation dictionary.

5. OOV Handling Strategy

A structured approach was implemented to handle OOV words.

Step 1: OOV Extraction
A Python script was developed to extract words from transcripts that were not present in the base dictionary.

Step 2: Pronunciation Generation using G2P

```
mfa g2p oov_words.txt english_us_arpa generated_oov.dict
```

The pretrained G2P model generated phoneme sequences for missing words.

Step 3: Dictionary Extension
The generated pronunciations were appended to the base dictionary to create extended_dictionary.dict

Step 4: Re-Alignment

```
mfa align dataset extended_dictionary.dict english_us_arpa
aligned_oov_fixed --clean --overwrite
```

This significantly reduced OOV occurrences.

6. Numeric Token Normalization

During iterative validation, numeric tokens were identified as OOV:

- 1971
- 300
- 35
- 800

Pronunciation dictionaries do not contain numeric digits, so numeric normalization was implemented. These tokens were converted to their spoken forms:

- 1971 became nineteen seventy one
- 300 became three hundred
- 35 became thirty five
- 800 became eight hundred

This preprocessing step further reduced OOV counts and improved alignment stability.

7. OOV Reduction Summary

OOV counts across stages were:

1. Initial baseline:
    a. 29 OOV types
    b. 61 OOV tokens
2. After dictionary extension:
    a. 16 OOV types
    b. 25 OOV tokens
3. After numeric normalization and refinement:
    a. 12 OOV types
    b. 15 OOV tokens
4. Final stage:
    a. 1 OOV type
    b. 1 OOV token

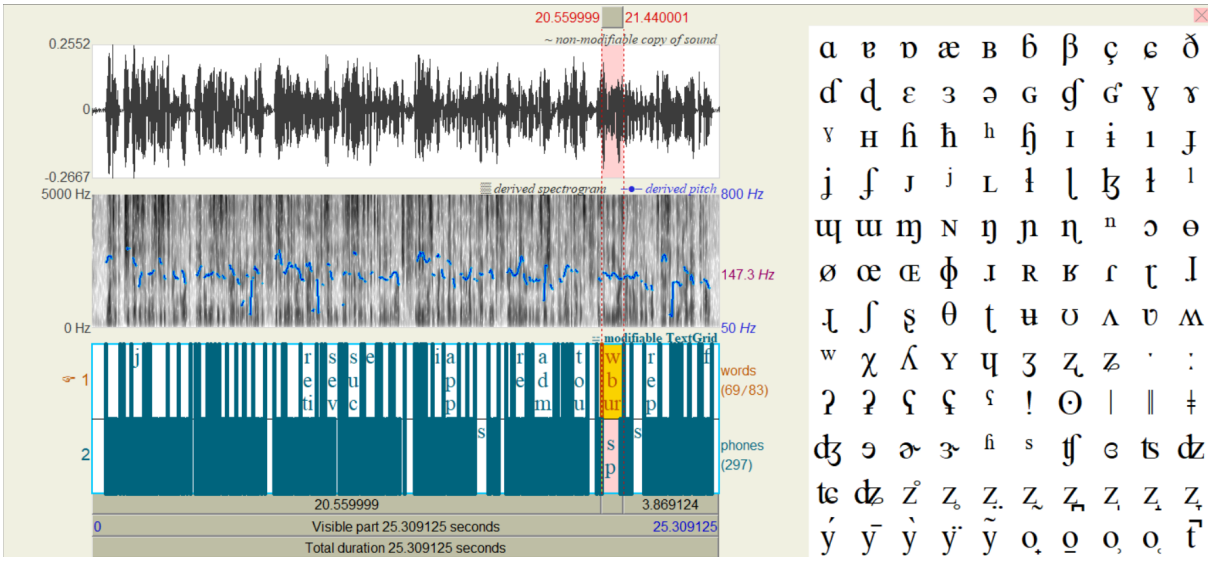
The OOV count was substantially reduced, demonstrating successful implementation of OOV handling.

8. Alignment Inspection and Analysis Using Praat

The generated TextGrid files were inspected using Praat software. Two versions were analyzed:

aligned_baseline
aligned_oov_fixed

8.1 Baseline Alignment Analysis

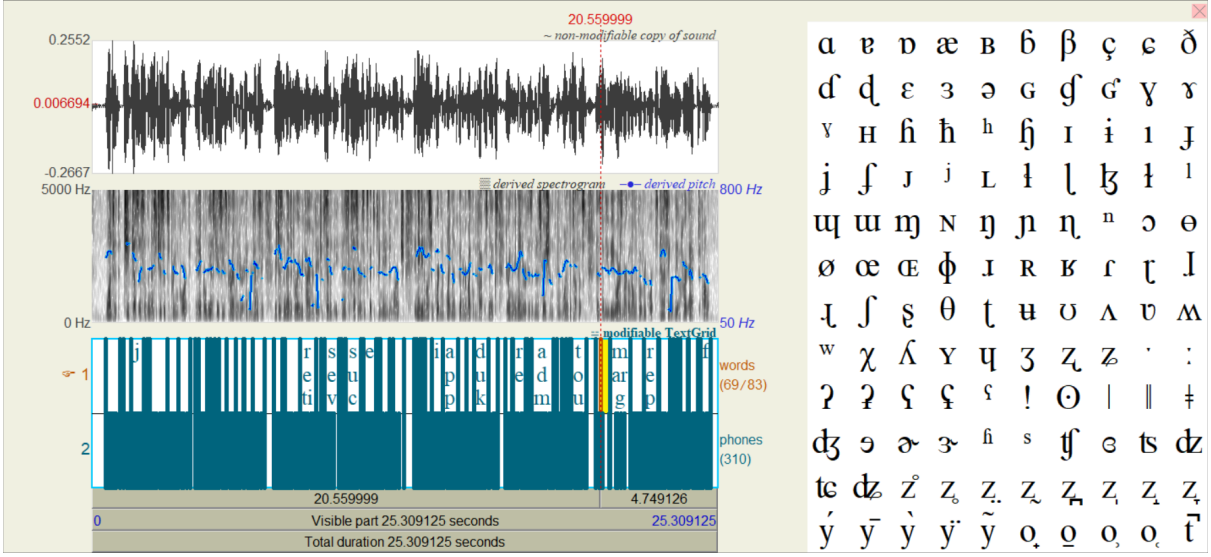


In the baseline alignment:

- Some words corresponding to OOV tokens were skipped or poorly segmented.
- Silence intervals appeared in regions where lexical items were expected.
- Word boundaries showed slight timing offsets in OOV regions.
- Phoneme segments in affected regions showed inconsistent durations.

In certain areas, phoneme boundaries did not correspond clearly to acoustic transitions, indicating degraded alignment quality in those segments.

8.2 OOV-Fixed Alignment Analysis



After dictionary extension and normalization:

- Previously skipped words were correctly aligned.
- Word boundaries matched waveform transitions more accurately.
- Silence modeling improved in corrected regions.
- Phoneme segmentation became more stable and consistent.
- Vowel regions aligned closely with peaks in acoustic energy.
- Consonant boundaries appeared temporally sharper and more precise.

Alignment quality improved noticeably in regions previously affected by OOV words.

8.3 Word-Level Boundary Observations

In the corrected alignment:

- Word start times aligned with rising acoustic energy.
- Word end times aligned with amplitude drops or transitions.
- Artificial silence intervals were reduced.

8.4 Phoneme-Level Observations

Phoneme segmentation after correction showed:

- Clear vowel durations corresponding to formant energy.
- Consistent consonant boundaries.
- More natural duration distributions.
- Reduced abrupt boundary shifts.

Minor residual issues included:

- Very short phoneme segments in fast speech regions.
- Slight boundary shifts in low-energy consonants.

Overall alignment quality improved significantly after OOV handling.

9. Technical Understanding

Forced alignment works by matching the spoken audio with its corresponding transcript. The aligner uses a pronunciation dictionary to convert words into phoneme sequences and then determines where each word and phoneme occurs in time within the audio signal.

If a word is not present in the dictionary, the aligner cannot assign phonemes to it properly. This can lead to skipped words, incorrect boundaries, or silence segments in place of actual speech. Therefore, handling Out of Vocabulary words is important to improve alignment quality.

This assignment demonstrated that improving the pronunciation dictionary and properly preparing transcripts directly improves word and phoneme boundary accuracy in the final alignment.

10. Repository Contents

The public GitHub repository includes:

1. Dataset
2. Baseline alignment outputs in TextGrid format
3. OOV-corrected alignment outputs in TextGrid format
4. OOV extraction script
5. Numeric normalization script
6. Extended dictionary
7. README with setup and execution instructions
8. Assignment Report

GitHub Repository:
https://github.com/skfareeduddin/mfa_forced_alignment

11. Conclusion

This assignment successfully implemented a complete forced alignment pipeline using Montreal Forced Aligner.

A systematic OOV handling strategy was designed and implemented using:

- G2P-based pronunciation generation
- Dictionary extension
- Numeric normalization

The OOV count was reduced from 29 types to 1 through iterative refinement. Alignment inspection using Praat confirmed noticeable improvements in word and phoneme boundary accuracy after OOV handling.