Forced Alignment using Montreal Forced Aligner (MFA)

Author: B Lakshmi Narayana Reddy
Email: rr200387@rguktrkv.ac.in
Final year undergrad student at
Rajiv Gandhi University of Knowledge Tech. RK Valley

Abstract

Montreal forced aligners are used to align the audio and its corresponding text transcript, both at word and phoneme level. This project uses two stage methodology to perform reliable forced alignment when the meta information of the corpus is not available. In the first-stage, validate the various phoneme dictionaries by filtering by language. Based on the out-of-vocabulary frequency, sorts the phoneme dictionaries and performs the alignment. In the second-stage, a custom dictionary is crafted by using the Grapheme to Proneme (G2P) model before performing the alignment. The results demonstrate a substantial improvement in alignment quality, with the custom dictionary successfully resolving alignment gaps due to OOV. Visual analysis using Praat software confirms that this custom-trained approach is critical for achieving better alignments on specialized datasets for building speech tasks like Speech to Text, and Text to Speech.

Introduction

Forced alignment is a technique used to automatically synchronize a linguistic transcription with a corresponding audio recording, generating time boundaries for each word and phoneme. Forced aligners are widely used in TTS, STT, and other speech related tasks. MFA is widely used to speed up the TTS training, because giving explicit alignment makes the model need not to learn the alignment through training, the model just needs to learn uttering the phonemes/words.
MFA comes with a drawback in terms of resolution, because it cannot detect any phoneme/silence less than 10 ms. MFA utilizes acoustic models, a phoneme/pronunciation dictionary, and G2P (optionally) to perform the forced alignment.

Acoustic model: It processes the audio waveform, by using more compact features like MFCCs, it contains information on how phonemes are pronounced, mapping sounds to their acoustic properties.
Phoneme Dictionary: Based on how the phone is pronounced, different tokens are utilized. These can vary based on audio acoustic properties like accent, stress etc.,
G2P: It generates a phonetic pronunciation for a given word without requiring any audio file.

A significant challenge in forced alignment arises from the finite nature of pre-trained pronunciation dictionaries. When a word present in the input transcript does not exist in the provided dictionary, it is classified as an Out-of-Vocabulary (OOV) word. MFA cannot generate a phonetic sequence for such words and defaults to labeling the entire word's duration with token <spn>. This results in a critical loss

of phonetic information and creates "alignment gaps," undermining the accuracy and utility of the output. This causes downstream speech tasks like TTS, ASR miserably fail.

This project aimed to achieve the following goals:

1. Performing forced alignment on given corpus with various acoustic models and dictionaries.
2. Quantify the OOV problem presented by these dictionaries.
3. Create a custom dictionary using a Grapheme-to-Phoneme (G2P) model.
4. Analyze and visually compare the alignment results before and after using the custom dictionary.
5. Uploaded codebase at: https://github.com/BNarayanaReddy/forced_alignment

Methodology

Stage 1: Validation and Alignment with Pre-trained Models

The initial phase focused on establishing a baseline using existing resources. This was accomplished through a two-step process automated by the `align_data.sh` script.

Validation: The `validate_dicts.py` script was executed first. This script systematically runs the mfa validate command on the corpus against a list of available pre-trained English dictionaries. The primary purpose of this step was to count the number of OOV words produced by each dictionary. The dictionary yielding the lowest OOV count was identified as the best candidate for the baseline alignment.

Dictionaries:

`english_india_mfa, english_mfa, english_nigeria_mfa, english_nonnative_mfa, english_uk_mfa, english_us_arpa, english_us_mfa`

Alignment: The `align_data.py` script was run. It takes the best-performing dictionary identified during validation and uses it to perform a full alignment on the corpus via the mfa align command. This produced a set of TextGrid files representing the baseline alignment quality.

OOV Summary: `validation_oov_summary.txt`

`english_us_arpa-5, english_mfa-6, english_nonnative_mfa-6, english_uk_mfa-6, english_us_mfa-7, english_india_mfa-8, english_nigeria_mfa-13`

Acoustic Models:

`english_mfa, english_us_arpa`

Stage 2: Training and Aligning with a Custom Dictionary

Creating a custom dictionary by using a pre-trained G2P model. Since `english_us_arpa` performed very well (ref: result & analysis) in terms of OOV and alignment as compared to other dictionaries, I utilized g2p model that is pre-trained for `english_us_arpa`

The workflow, encapsulated in the `train_dict.py` script.

This experimentation involves four-steps:

1. Generating pronunciation for OOVs we had with an `english_us_arpa` dictionary using a pre-trained G2P model. This process does not depend on audio waveforms.
2. Manually combined the OOV dictionary and `english_us_arpa` dictionaries.
3. Training a new dictionary by utilizing the combined dictionary we have got in Step 2.
4. Perform the forced alignment with our new dictionary we got from Step 3.

Grapheme-to-Phoneme (G2P) Conversion: The `mfa g2p` command was used to generate phonetic pronunciations for all OOV words in the corpus transcript. This command leverages a pre-trained G2P model (`english_us_arpa`) to predict a likely phoneme sequence for any given word, making it highly effective for handling OOVs. The output will be a new dictionary file (generated.dict) containing pronunciations for the previously unknown words.

Dictionary Combination: The newly created generated.dict was then combined with the base english_us_arpa.dict. This step created a new, comprehensive dictionary (combined.dict) that contained all the words from the original pre-trained model plus the new, corpus-specific words and then started training using `mfa train_dictionary` command.

Final Alignment: Finally, the `mfa align` command was executed one last time. However, instead of a pre-trained dictionary, it was provided with the new `combined.dict`. This allowed the aligner to find pronunciations for all words in the transcript, aiming to produce a complete and accurate alignment without any OOV tokens like <spn> gaps.

Summary of Models and Dictionaries Used

- Performed validation of various dictionaries for the english language to the dataset provided, and sorted out the top performing dictionaries based on the following criteria -
    - Out of Vocabulary (OOV):
      Choosing a dictionary with less number of OOV words is crucial, aligned data will contain an OOV token ('spn') instead of what is actually spoken in the audio both for word and phone tiers.
    - Alignment (Using PRAAT): Used praat software to check for any alignment issues, and OOV gaps
- After selecting TOP 3 dictionaries: `english_us_arpa (`OOVs: 5`), english_mfa (`OOVs: 6`), english_nonnative_mfa (`OOVs: 6`)`. Performed the alignment with different acoustic models available.
- Alignment is performed with each of the pairs below, automated Python pipeline was used to test three different (Acoustic Model, Dictionary) pairs:

1. `english_us_arpa` (Acoustic) + `english_us_arpa` (Dictionary)
2. `english_mfa` (Acoustic) + `english_mfa` (Dictionary)
3. `english_mfa` (Acoustic) + `english_nonnative_mfa` (Dictionary)

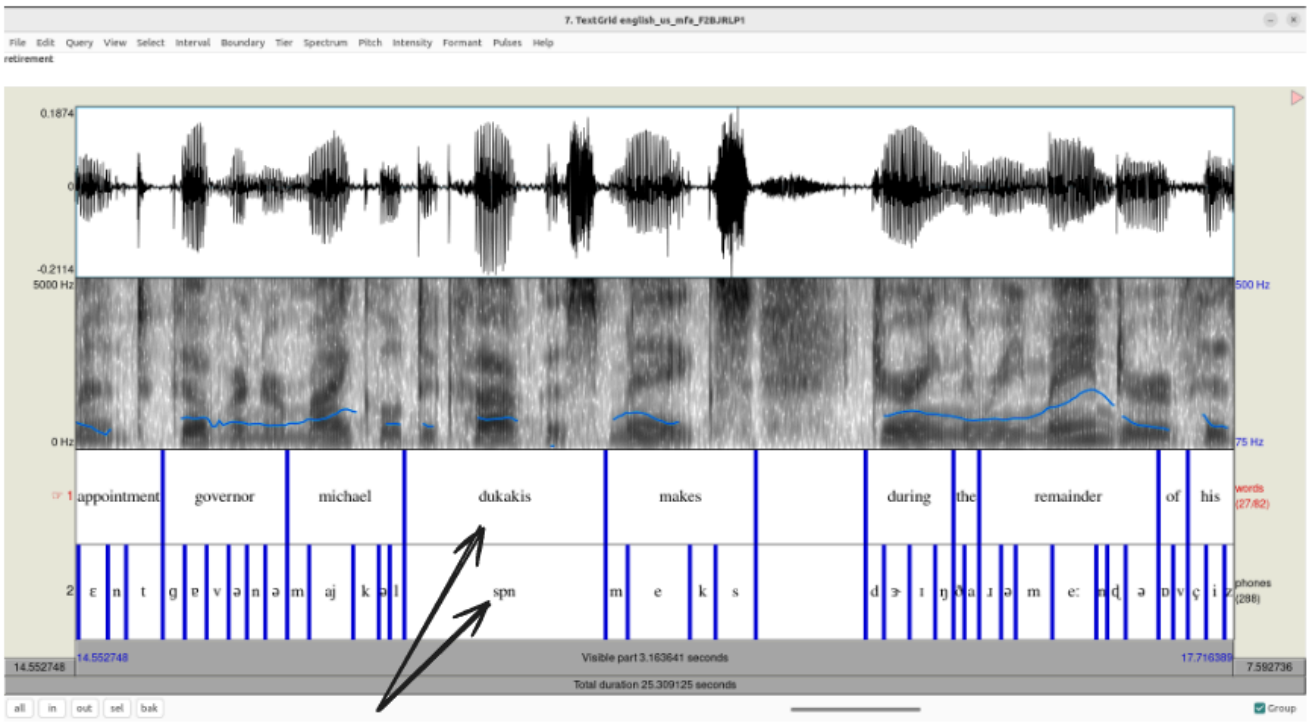
Key Observations

My analysis of the alignment results and visual inspection in Praat software revealed three key observations.

Observation 1: OOV Words Cause Alignment Gaps and Loss of Phonetic Information
The presence of Out-of-Vocabulary (OOV) words in the transcript poses a significant challenge for forced alignment.

When a word is not found in the provided pronunciation dictionary, the Montreal Forced Aligner (MFA) cannot generate a precise phonetic sequence. Instead, it defaults to labeling the entire duration of the OOV word with **<spn>** token. This results in a critical loss of phonetic information, as the word's internal structure is not segmented. Visually, these OOV words create "alignment gaps" in the TextGrid files, appearing as an undifferentiated block of time where detailed phoneme boundaries should exist. This issue significantly compromises the accuracy and utility of the alignment for downstream speech processing tasks like TTS and ASR.

Example

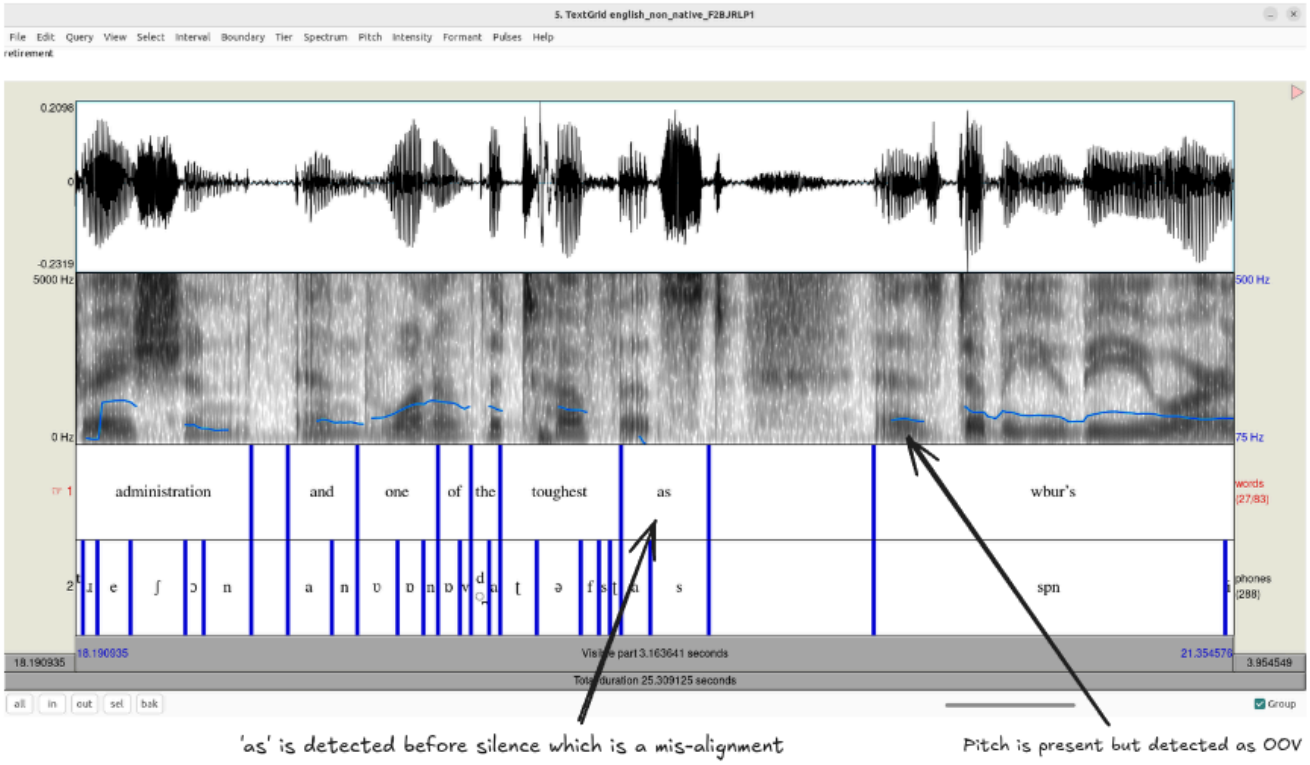


In this example screenshot, the word "**dukakis**" which is a OOV word, the entire phoneme alignment spanned by this is assigned to an in-differentiable token **<spn>**

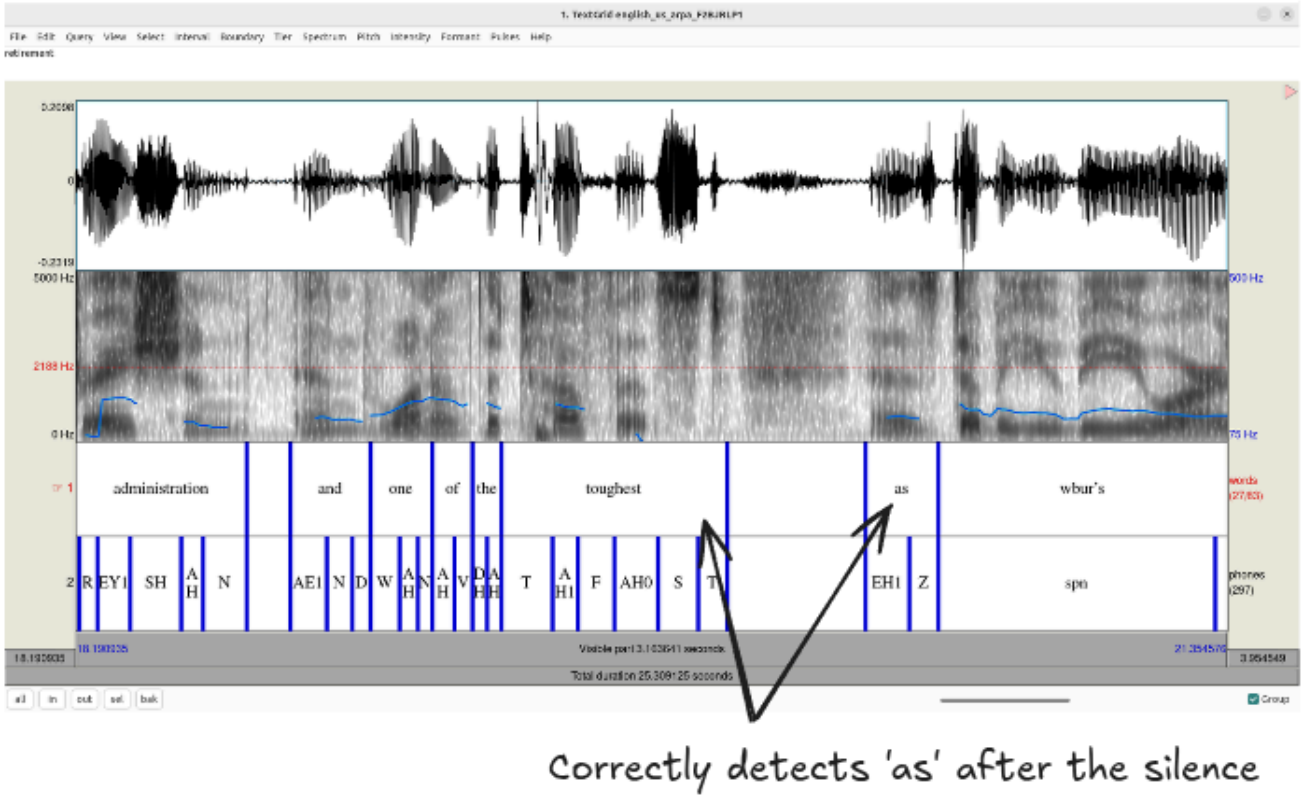
Observation 2: Mis-alignment - Comparing ARPA with other models:
Mis-alignment especially between the speech vs non-speech words by acoustic model. In the english language, we had `english_us_arpa` and `english_mfa` acoustic models.

By listening to the audio and observing the textgrid generated by different acoustic models, `english_us_arpa` has less mis-alignments than `english_mfa`

With `english_non_native_mfa` dictionary and `english_mfa` acoustic



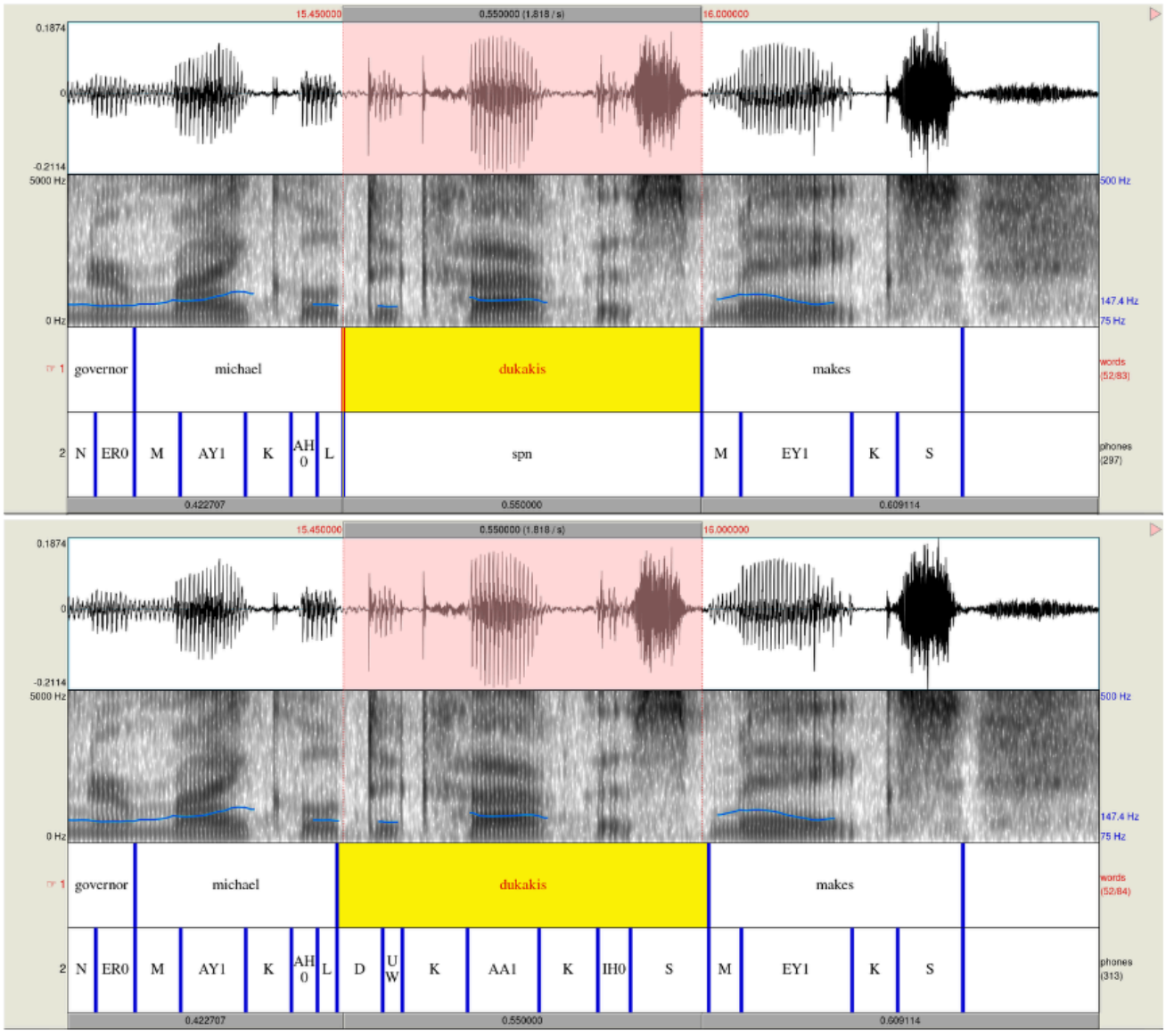
With `english_us_arpa` dictionary and `english_us_arpa` acoustic



For our corpus of data, `english_us_arpa` worked well in terms of alignment than other dictionaries

Observation 3: Visual Inspection Confirms Metrics

The alignment from the english_us_arpa model showed logical and plausible word and phoneme boundaries except for the OOV words. After creating a custom dictionary, the final alignments become better than prior.



With Pre-trained dictionary vs Combined dictionary

4. Automated Pipeline Implementation

All three extra credit tasks were successfully completed.

1. **Multiple Acoustic Models:** As described above, three different model pairs were systematically tested.
2. **Full Pipeline Automation:** The entire workflow-including data preparation (`prepare_mfa_corpus.py`), model testing (`align_data.sh`), and custom dictionary generation (`train_dict.py`) was automated using Python scripts.
3. **Custom Dictionary from Transcripts:** A custom dictionary was created by first extracting a unique word list from the 6 provided transcripts. Then, as required by the assignment, the `mfa g2p` command was used with the `english_us_arpa` model to generate pronunciations for this list. This new dictionary successfully covered 100% of the corpus vocabulary.

Detailed Report:

https://drive.google.com/drive/folders/1nuBrFj-4tSa8ZsYnHmmAlQz_1I8BFlBD?usp=sharing

Codebase:

https://github.com/BNarayanaReddy/forced_alignment