Title : MFA (Montreal Forced Aligner)

Introduction
MFA is a widely used open source tool for forced alignment.
Forced alignment is the process of aligning an audio(speech signal) with its corresponding transcript (text) where we determine the exact start and end time of each word and phoneme in the audio.

Why Montreal Forced Aligner (MFA) is Used
- Generates precise word and phoneme level time boundaries.
- It uses pronunciation dictionaries , Acoustic models ,Hidden Markov Models.
- Supports many languages and regional accents.
- Generates textgrid files as an output which contains:
    - Intervals
    - name(words/phonemes)
    - xmin (start time of word / phonemes)
    - xmax(end time of word / phonemes)
    - text(Label)

MFA Environment Setup

Operating System : Windows 10 (64-bit)
Python version   : Python 3.10 (Installed and managed via conda)
Conda Environment: mfa (created to isolate MFA dependencies and avoid package conflicts)
Montreal Forced Aligner Version:3.3.9 (Installed via conda using the official MFA distribution)

Installation of Montreal Forced Aligner:

Installation Method
MFA was installed using conda (mini conda),which
Step1: Download Miniconda
- Download from https://www.anaconda.com/docs/getting-started/miniconda/main
- Install it normally

Step 2: Create MFA Environment
This keeps MFA isolated and avoids conflicts
- Command : conda create -n mfa python=3.10 -y
    - Here we are creating a new virtual environment called “mfa” and installing Python version 3.10 inside the environment.
    - -y automatically answers “yes” to all the prompts.
- Command: conda activate mfa
    - In the terminal we need to see (mfa) , which means the environment is active.

Step 3: Install Montreal Forced Aligner
- Command : conda install -c conda-forge montreal-forced-aligner -y
- Command : mfa version

Step 4 : Download Required Models
MFA needs three core components to work

- Acoustic model
    - Command : mfa model download acoustic english_us_arpa
        - Maps audio signals -> phonemes
        - Trained on American English Speech
        - Uses ARPAbet phoneme set
- Dictionary
    - Command: mfa model download dictionary english_us_arpa
        - Maps words -> phoneme sequences
        - Must match the same phoneme set as the acoustic model
        - It is fully compatible with the pretrained English acoustic model provided by MFA.
- G2P Model (Grapheme-to-Phoneme)
    - Command : mfa model download g2p english_us_arpa
        - Generates pronunciation for out of vocabulary words.
        - Converts spelling to phonemes

Data Preprocessing
Data set has two folders : audio (.wav files) and transcripts(.txt) files

- Transcripts have 6 text files,In that 3 files have an extension .TXT and other 3 files have .txt.
    - So,first I converted .TXT files to .txt files by opening folder location inside command prompt and command given :ren *.TXT *.txt(windows)
        - This converts all the .TXT files to .txt in the given folder
- After the conversion of all files into .txt files ,I converted transcripts(consists of multiple lines) into paragraph level text suitable for forced alignment .
    - This I have done using Python Script (para.py) and saved to output in para folder
    - In this we do sentence filtering which removes empty lines and then converted sentence to paragraphs and white space normalization which removes multiple spaces.
- The output .txt files inside the para folder have punctuation marks inside it so ,I removed all the punctuation marks.
    - This I have done using python script (punct.py) and saved the output in the no_punct folder.
    - The text is converted into tokens and punctuations(even apostrophes) are removed at starting and ending of the tokens.
    - After this we save normalized of .txt files inside the no_punct folder
- Now the transcript we give as an input to MFA is .txt files of no_punct folder
- At last transcripts should contain
    - Each .txt file must contain only the spoken words
    - No Time stamps
    - No Punctuation marks
    - No Speaker labels
    - Uppercase is recommended

- But MFA does not accept separate .wav and .txt folders i.e both must be in the same folder and need to be placed in the folder(named mfa_corpus) where each .wav file should be followed by its .txt file (File names must match exactly).
    - We can do this by manually or using python script.
    - I have done this manually by cut ->copy->pasting of .txt files with respect to their .wav files and placing them in the mfa_corpus folder.
- The mfa_corpus will be the input dataset for MFA.

Forced Alignment Execution

- I had installed python dependencies inside the command prompt forextracting TextGrids
    - Command: pip install praatio
- I had written a python script to run MFA alignment (main.py)
    - Input
        - mfa_corpus (.wav+.txt pairs)
        - Pre-trained MFA Models
            - Dictionary: english_us_arpa
            - Acoustic model: english-us-arpa
        - MFA tool to perform forced alignment
- After this in the command prompt we run python main.py where we can see the output files generated inside the mfa_output folder.
    - Output
        - TextGrid files inside the output folder mfa_output
            - Folder contains time-aligned word and phoneme boundaries
            - Word Interval Files that has
                - Word starting time
                - Word ending time
                - The word label
            - Phoneme Interval Files
                - Phoneme start time
                - Phoneme end time
                - Phoneme label

Installation of praat software

Praat is a speech analysis tool used for viewing and inspecting .wav + .TextGrid files which are generated by MFA.

Step 1: Download praat from https://www.fon.hum.uva.nl/praat/
- for windows click praat6459_win-intel64.zip

Step 2: Extract praat
- Inside the extracted folder we see praat.exe

Step 3: Launch praat
- We see two windows
    - Praat objects
    - Praat windows

Now we can use Praat software for the analysis

Analysis of the output

- For analysing the output :
    - In praat objects window ->click open we see a column : Read from file
    - Select both audio.wav from audio folder and TextGrid file from the mfa_output folder (output stored after mfa alignment).
    - Click on view and edit
- We see word tier and Phoneme tier and time boundaries
- Identification of Phoneme and Word Boundary Alignment
    - Top tier- words
    - Bottom tier- phones
- Each words in the top tier are made up several phonemes in the phone tier
- The phonemes for a word lie completely inside the word boundary.
- Silent regions (empty labels or sp) appears between words phonemes
- Observation of Errors or Mismatches in Alignment
    - Phoneme tier shows many small segments which appear dense vertical lines i.e single phoneme is split into tiny parts instead of a clean segment.
    - Sometimes ,phonemes start slightly before or after the wordboundary
        - Small timing differences (few milli seconds)

OOV (Out -Of Vocabulary) words
OOV words are the words which are present in the .txt files but does not exist in the pronunciation dictionary.

How OOV words are handled
- Identification of OOV words
    - First , audio files were validated using MFA. During validation ,the words of the .txt files were compared with the pronunciation dictionary.
    - Make sure the path in the command prompt should be folder that should contain the mfa_corpus (.wav+ .txt files)
    - Command : mfa validate mfa_corpus english_us_arpa english_us_arpa
- Generally after validation we see 3 files :
    - oovs_found_english_us_arpa.txt
        - This file is given to G2P
        - This contains a list of all OOV words
        - This contain one word per line
        - Each words appears only once
    - Utterance_oovs.txt
        - Tells which audio file contains which OOV word.
    - Oov_counts_english_us_arpa.txt
        - Frequency of each OOV word occurring.

- Locating the OOV word list
    - The problem I faced here is I couldn’t find where the OOV files are saved.
    - The pronunciation dictionary is in different location where these OOV list is are saved .

    - So to locate the files I used command : dir C:\Users\ranga\Documents\MFA /s /b | findstr oov
        - \Users\ranga\Documents\MFA is my MFA folder location.
    - By using this command we can see all the files related to the name oov
    - After finding the location of oovs_found_english_us_arpa.txt I have copied this file to copy in my desired folder or destination path english_us_arpa by using command : copy "SOURCE_PATH" "DESTINATION_PATH"

- G2P pronunciation generation
    - Command : mfa g2p "DESTINATION_PATH"\oovs_found_english_us_arpa.txt" english_us_arpa oov_pronunciations.dict
    - Takes each OOV word from oovs_found_english_us_arpa.txt
    - Uses the english_us_arpa G2P model
    - Generates phoneme pronunciations
    - Output is saved in a dictionary file named oov_pronunciations.dict
    - To view the generated pronunciation command : type oov_pronunciations.dict
- Lexicon Update
    - The original pronunciation dictionary english_us_arpa.dict was copied from MFA folder into the current working directory and given new name as final_dictionary.dict
        - Command : copy “ english_us_arpa.dict location” final_dictionary.dict
    - After this we need to merge OOV pronunciations into the final dictionary
        - Command : type oov_pronunciations.dict >> final_dictionary.dict
        - This gives complete pronunciation lexicon
    - Now the dictionary contains
        - Original dictionary words
        - Previously unknown OOV words
    -
- Forced Alignment
    - Command : mfa align mfa_corpus final_dictionary.dict english_us_arpa "output folder location" --clean
        - Input
            - mfa_corpus (.wav +.txt files)
            - Final pronunciation lexicon
            - Acoustic model : english_us_arpa
        - Output :
            - Text grid files inside the output folder named output_OOV
                - Folder contains time-aligned word and phoneme boundaries
                - Word Interval Files that has
                    - Word starting time
                    - Word ending time
                    - The word label
                - Phoneme Interval Files
                    - Phoneme start time
                    - Phoneme end time
                    - Phoneme label