Automated Forced Temporal Alignment of MULTISIMO Transcripts with Speech Signals

by

Kavya Bhadre Gowda

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Dr. Carl Vogel

September 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Kavya Bhadre Gowda

September 8, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Kavya Bhadre Gowda

September 8, 2020

Acknowledgments

I would like to express my special thanks of gratitude to my professor Dr. Carl Vogel as well as our university, which gave me the golden opportunity to undertake this wonderful research work that helped me to explore new areas of Linguistics, for which I am thankful. I would also take this opportunity to thank Dr. Maria Koutsombogera for assisting me by sharing her knowledge related to the MULTISIMO dataset, which helped me to do my research with ease and a better understanding of data.

Lastly, I thank my parents and friends who helped me a lot in finalizing this research work within the limited time frame.

Kavya Bhadre Gowda

University of Dublin, Trinity College
September 2020

Automated Forced Temporal Alignment of MULTISIMO Transcripts with Speech Signals

Kavya Bhadre Gowda, Master of Science in Computer Science

University of Dublin, Trinity College, 2020

Supervisor: Dr. Carl Vogel

This research work explored the various tools and techniques available to perform the time-based alignment of lengthy MULTISIMO audio files, which contain lengthy MULTISIMO audio "Continuous Speech Signals" of approximately 10 minutes to its respective transcripts automatically and accurately. A user-friendly web application was designed and developed to solve this time-based alignment problem, that provides accurate results even with noisy audio inputs and audio containing long pauses. Temporal alignment of audio and transcripts helps linguists to analyze the language at Phonetic and Word level. Analyzing audio signals using the "Forced Aligner Application" that has a higher signal-to-noise ratio, is easy and accurate.

The focus of Forced Aligner Application is to identify the Pronunciations and Syllables in the Speech Signals, which are utilized in training Speech Recognition Systems. This web application is developed by analyzing the existing state-of-the-art tools and techniques in the Forced Alignment area. This application was built using the HTK toolkit and has achieved an accuracy of 77.11% compared to the manual alignment of start and end time of utterances. We have achieved this accuracy by cleaning the transcripts, re-sampling the audio signals, and changing the configurations of the HTK toolkit in the ProsodyLab tool. Our analysis also showed that there is a significant correlation between the end time of the utterances and the mismatch in the alignment.

With Spearman's correlation, it is also proved that the alignment of the start-time of utterances is easier than compared to end-time.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recognizing and detecting speech in a continuous speech processing system is crucial. The performance of such systems and applications is highly dependent on the recognition of the start and end of the utterances through audio segmentation and extraction of words in the audio. The output of Speech Recognition Systems includes the lexical transcripts that can further be used to train their own language-independent speech recognizer (s). Accurate temporal alignment is vital while integrating speech into training data-set for training own acoustic speech recognition language models.

Time aligned audio and transcripts are an important source of inputs to speech recognition, text-to-speech, and dialogue analysis systems, etc. Alignment of audio with the transcripts can be carried out at various levels like word-level, phoneme level, or syllable levels. This alignment process at various levels is mainly carried out by linguists manually during their process of analyzing languages. The manual time alignment of audio signals with the corresponding lexical transcripts at the word or phoneme level is highly expensive and time-consuming. The accuracy of manual alignment is highly dependent on the knowledge and proficiency of the linguists. The alignment produced by one linguist may not be agreed upon by the other linguists. Even the most proficient linguists can time align audio and transcripts accurately but takes too long to align. The manual alignment of audio with the transcripts at the word level takes approximately 10 times more than that of the automated alignment [Goldman, 2011]. Hence our main aim of this research work is to build an automated temporal forced aligner application that replaces human annotation for alignment of

audio and transcripts.

Temporal alignment of utterances with transcripts is very much useful in Language Construction Researches, where the latency of the speaker's response to any stimulus in audio and the temporal alignment of that response is the dependent variable of interest. Hence finding the onset and offset times of utterances is very much important.

1.1 Motivation

The analysis of dialogue is important to know, how various channels of communication align. When people communicate, they not only do with the Linguistic content but also with social signals and back channels. So, in a dialogue when somebody is uttering, the other person will say something or convey some social signals, like Laughter or Visual Gestures. All parties in communication will be doing something or the other. To study how these different channels align, it is important to know where linguistic content starts and ends. Hence the onset and offset times of words aligned with the transcript files need to be accurate. Multimodal channel analysis requires temporal alignment of utterances. The time alignment of audio with transcript is also base for training Automatic Speech Recognizers. This motivated me to take up this research work of automating the alignment of transcripts with speech signals accurately to contribute to the area of Linguistics and Speech Recognition Systems.

1.2 Background

1.2.1 *What is Forced Alignment?*

Transcriber transcribe the audio to the sequence of word transcripts, which then is used by many automatic speech recognition systems to train and build their speech recognition models. The transcripts are converted to speech using text-to-speech synthesis to build an accurate speech recognition system. With transcripts and audio, trying to recognize the utterances require the forced alignment of audio with its lexical annotations. This technique of automating the process of alignment of transcripts with speech signals such that the start time and end time of each utterance are accurate is called Forced Alignment.

1.2.2 *What are Phonemes?*

Phonemes or Phones are entities that represent the Psychological or Cognitive correspondence of speech signals [Baudouin de Courtenay, 1972] and are the main focus of study in the Language Analysis (Linguistics). Each language has it's set of pronunciations which are used during the conversation. For example, North American English Language has approximately 69 Phonemes as listed below.

AA0, AA1, AA2, AE0, AE1, AE2, AH0, AH1, AH2, AO0,
AO1, AO2, AW0, AW1, AW2, AY0, AY1, AY2, EH0, EH1,
EH2, ER0, ER1, ER2, EY0, EY1, EY2, IH0, IH1, IH2, IY0,
IY1, IY2, OW0, OW1, OW2, OY0, OY1, OY2, UH0, UH1,
UH2, UW0, UW1, UW2, B, CH, D, DH, F, G, HH, JH, K,
L, M, N, NG, P, R, S, SH, T, TH, V, W, Y, Z, ZH

Phoneme for the utterance *"I'M"* is *"AH0 M"* or *"AY1 M"*.
The AY0, AY1, AY2 represents the various stress or intensity level in the pronunciation [Kazanina et al., 2018].

1.2.3 *What are Forced Aligners?*

While building an automatic speech recognition system, a forced aligner is used as a tool to efficiently and accurately align the Orthographic (spellings and grammars) transcripts with the audio at Word as well as at phoneme level. Forced Aligners make use of a dictionary file, which contains the pronunciations for each word in the transcript. A word can have multiple pronunciations or phonemes, and the selection of pronunciation is done based on the Probability-Likelihood match with the audio signal [Kempton, 2012]. The Forced Aligners are mainly used by linguists to perform forced alignment and to get the time aligned phonetic transcriptions of utterances in the audio file.

1.2.4 *Terminologies used in the research work*

Praat

Praat is a software and a program by using which we can analyze, manipulate, and synthesize speech signals and also perform various other functionalities [Styler, 2013]. Praat tool is mainly used by linguists for phonetic and language analysis using both audio and lexical transcripts. Evaluation of the Temporal Forced Alignment can be efficiently carried out using Praat, since it provides both read and write functionalities using which the researcher can correct the misaligned areas quickly.

Text Grid

TextGrid is a type of file format that can be read and write using the Praat tool [Styler, 2013]. If a recorded audio file with duration 362.61 seconds need to be aligned with its transcript, the Praat readable TextGrid file format looks as shown in Figure 1.1. The TextGrid file consists of the word start and end-time for the various tiers like phonemes and words. Many more tiers can be added and read using Praat.

Acoustic Models

Acoustic Models are the machine learning models built to effectively perform the time-alignment of transcripts with the audio signals. This speech (Acoustic) model maps the digital audio signals with the list of phones on which they are trained. The performance of these models depends on how well they accurately recognize the speech boundaries and non-speech boundaries.

HTK Toolkit

HTK is a Hidden Markov Model toolkit developed in C language, which provides various tools to build, train, test, and perform analysis on various HMM models. This toolkit supports both Discrete and Gaussian distributions to build and manipulate Hidden State Markov Models (HMMs). Even though Microsoft holds the copyright on HTK which was initially developed at Cambridge University Engineering Department (CUED) [Young et al., 2002], they encourage any contribution made to the source code. Many forced aligner tools are built on top of the Hidden Markov Model Toolkit (HTK)

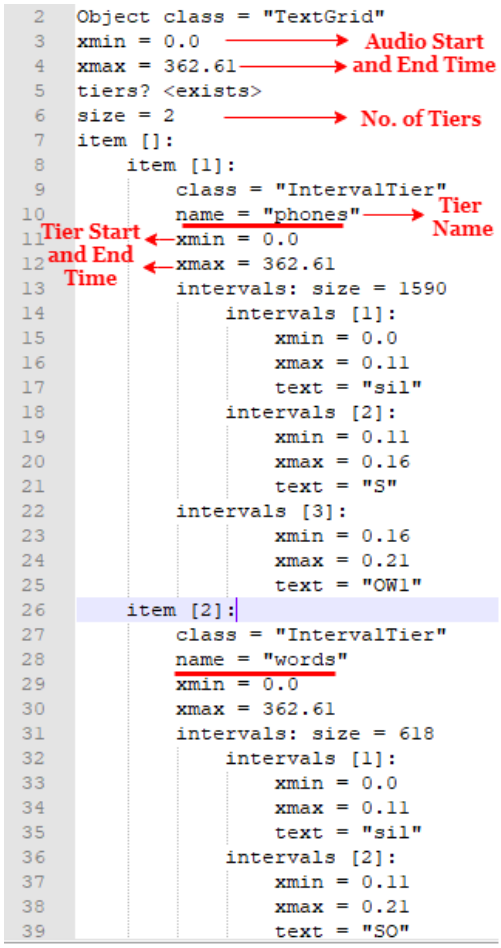

Figure 1.1: Praat readable TextGrid file Format

which is a portable toolkit used to build and manipulate HMM models. Even though it is designed for speech recognition purposes, it serves many other kinds of research.

Hidden State Markov Model (HMM)

HMM is a Statistical Markov Model which changes its state all the time and consists of unobservable hidden states. These are used to model time series in the HTK toolkit internally [Young et al., 2002]. Most of the Forced Aligners are built using Mono-Phone based HMM acoustic models.

MULTISIMO corpus

MULTISIMO corpus called "MULTI-modal and MULTI-party Social Interactions Modelling" is the corpus prepared to model the multiple channels of communication involving behaviors of speakers in dialogue models which are mainly used in the development of speech recognizers [Koutsombogera and Vogel, 2018]. This corpus includes audio, video, and transcripts along with other annotations of speakers involved in the multiple-party social interactions. This data-set involves recordings of sociolinguistic interaction among groups of three people including a facilitator and two students in each recording, where the facilitator is carrying out the quiz and the students are taking part in it. Various annotations for the corpus are available like transcriptions of speech from ELAN and the Transcriber tool, facilitator's feedback data, annotation of laughter, and gesture, etc. This whole data collection is done under an ideal laboratory environment with minimal noise and with the consent of the participants.

The data-set consists of 18 recordings which correspond to approximately 3 hours of total recording time with each recording of length 10 minutes approximately. The access to MULTISIMO corpus is given on acceptance of the licensing terms and conditions for research purposes since all the data is generated by taking consent from all participants under the GDPR act.

1.3 Research Question

This research work addresses the following questions through Forced Aligner Web Application.

- How accurately is the utterance start, and end times are aligned with transcripts using the Forced Aligner Application?
- How accurately the phoneme start and end time are aligned, which will help to understand the variations of phonemes or pronunciations?
- How the Forced Aligner outputs correlate with the start and end time of the utterances?
- Which technique can be used to improve the accuracy of existing state-of-the-art aligner tool?

- How efficient and accurate the Forced Alignment tools are?

The existing Forced Aligners require complete transcripts of the audio files and less noisy audio files. Hence in our research work, we are performing Forced Alignment of the audio file consisting of three speakers, out of which transcripts are fed only for one speaker, and the audio of the other two speakers is considered as noise. This type of analysis of speech as a biological process is very much useful where the speech of an individual speaker is recognized and segmented in a dialogue system.

The function of the Automated Forced Alignment Application is to mainly find the voice from the speech signals and align that with the transcripts. To evaluate our tool, Voice-Activity-Detection packages were used which did not show expected accurate results hence it's discarded, and evaluation of the tool is carried out by comparing manual alignment results with the output of our Forced Alignment Application.

In this research work, the transcriptions and speech files uploaded to the Forced Aligner tool along with the pronunciation dictionary are aligned using the HTK toolkit. This toolkit detects the start and end time of the speech signals and aligns with the transcript's word, phoneme start and end times. The output is in the form of a TextGrid file which helps the user to verify audio, words, and phoneme's temporal alignment in parallel using Praat, and update easily wherever inaccurate results are observed easily. This TextGrid file can be used with Praat, ELAN, or any other TextGrid supporting tools for analysis. The Flow Automated Temporal Alignment Process is depicted in Figure 1.2.

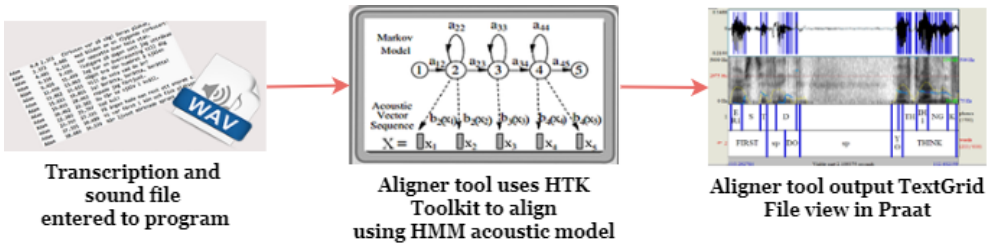

Figure 1.2: Automated Forced Temporal Alignment of Transcripts with Speech Signals

This research work involves investigating variations and change in Languages. The main objective is to automatically align the speech with the transcripts at the word-level and phonetic level. The analysis of the existing state-of-the-art tools and technologies is carried out in the coming chapters followed by improvements.

Chapter 2

Literature Review

2.1 Introduction to Literature Review

Automated alignment of start time and end time of utterances and their respective pronunciations in audio files with their known transcripts is a very important research area where many researchers have achieved significant results. Before going to implement our own automated Forced Aligner, the existing state-of-the-art tools and technologies are reviewed and compared in this chapter to build an automated accurate Forced Aligner of our own. In this research work, Six existing tools are analyzed, and the best solution is carried forward for the improvements. The tools analyzed during our review work are AlignTool, Munich AUtomatic Segmentation System (MAUS), Montreal Forced Aligner (MFA), FAVE-Align, DARLA, and ProsodyLab Aligner. The comparison of the analysis is also presented at the end of the chapter, with the conclusion from our analysis.

2.2 AlignTool

AlignTool, an automatic time-based annotation tool for spoken utterances developed by Prof.Dr. Eva Belke [Schillingmann et al., 2018] using WebMUAS which produces Praat tool readable TextGrid aligned files. AlignTool is a semi-automated open-source alignment tool whose functions are based on voice keys. The inaccurate results can be manually edited through Praat Tool. AlignTool is mainly based on the stimulus

produced by the speakers by finding the "beep" sound or voice-keys in the audio files. AlignTool does audio segmentation and alignment for audios involving beeps, as well as without the beeps. But when this tool installed VMware machine was executed for our data-set involving ∼10 minutes of audio, the tool failed to produce any output because of the absence of beeps in our corpus. AlignTool completely failed to align MULTISIMO corpus. Since, AlignTool uses WebMAUS Web Service to detect voice activity and align them with the transcripts, a further analysis is carried out on the MAUS service.

2.3 Munich AUtomatic Segmentation System

The MAUS system, developed by researchers at Munich was available as CLI and web service to perform the forced alignment process. But now the MAUS is available only as a web service through CLARIN [Strunk et al., 2014]. This web service was developed by statistically modeling pronunciations, using HMM modelling technique. With this technique, various pronunciations are modeled to find the accurate speech segment in the audio. This tool performs accurately if the pronunciations are known for the input transcriptions. Hence MAUS can be used to align audio with transcripts not only for the English language but also for other languages like German, New-Zealand and Australian English, Portuguese, Spanish, Italian, Dutch, Polish, and many more. [Burger et al., 2000] used MAUS on spontaneous German audio corpus and got around 97% accurately time aligned phonemes. MAUS is available now only as a Web Service and WebMAUS is a web interface for the MAUS aligner tool. The input audio and transcripts need to be uploaded through WebMAUS web interface, to a server located in European Union. Since for MULTISIMO corpus, no consent from the interviewee has been taken about uploading data to a third-party server, we continued our research work only on analyzing Command Line Interface (CLI) tools and techniques and improve their accuracy.

2.4 Montreal Forced Aligner

This is an efficient Forced Aligner tool built using the Kaldi toolkit, one of the famous forced alignment toolkit like HTK and Sphinx. This tool supports many other languages apart from English. It is trained in four stages in which, the initial stage makes use of mono-phone models followed by triphone models, LDA+MLLT, and ends with speaker-dependent enhancement to triphone models. At each iteration of model training, the feature extraction is done thoroughly. Instead of this, the acoustic model training for our language and data can be trained by deep neural networks using MFA. The main feature of this tool is the training carried out by considering speaker adaptation during the alignment process. Hence in our corpus, speakers are not properly specified which results in inaccurate alignment. The initial environment setup itself failed because of errors in the MFA package.

2.5 FAVE-Align

This is an automated forced alignment tool developed on top of Penn Phonetics Lab Forced Aligner (P2FA). FAVE-Align is mainly used for finding Out-of-the-Vocabulary (OOV) words present in transcripts for which pronunciations are not available in the dictionary (List of pronunciations). FAVE-Align provides accurate results for transcripts which are approximately time-aligned. The transcription file must include code for Speaker, Speaker Name, Onset time (in seconds), Offset time (in seconds), and Transcription of audio between those times. So, the obtained results are aligned based on the acoustic model and the dictionary supplied on, the word start and end times by adding missing OOV words to dictionaries.

Since our focus is to align based on word and phoneme level and we had transcriptions which are just texts and were not time-aligned, we found this tool inappropriate for our corpus.

2.6 DARLA

DARLA is a web service similar to FAVE but uses MFA with more inputs. This again is a web interface with servers located in the United States. Hence uploading our corpus

to this tool to get temporally aligned results is against ethics. Hence more analysis on this tool is not carried out.

2.7 ProsodyLab Aligner

A ProsodyLab is a Python-based Aligner Tool developed by [Gorman et al., 2011] which is very much similar to Penn Forced Aligner in using the HTK toolkit. But this tool also provides the user with functionality to train their acoustic models with limited unaligned data of around 1 hour audio and its transcriptions along with a dictionary consisting of relevant pronunciations. Using this tool, users can train their own understudied languages with sufficient data. This tool provides pre-trained acoustic mono-phone HMM models trained on data consisting of North American English. It also supports training and building our acoustic models with enough data. During model training, the best time alignments are learned by the model for the data involved in the training. More data including audio transcripts without pre-aligned and pronunciation dictionary, will yield a more accurate model. This is open-source software and can be run on Linux and Mac machines along with Windows using Cygwin. ProsodyLab's initial setup was successful and gave reliable results on trial data. Hence more research work and improvement have been carried out by in this study using this tool.

[DiCanio et al., 2013] in their work has analyzed the alignment of small speech with individual words and their phonetic alignment by considering model trained on different languages using automatic aligner tools like hmalign and p2fa and have found reasonably accurate results. Unlike this, our research work is mainly focused on analyzing continuous speech and improving its accuracy. Even tough Sphinx and Kaldi Forced Alignment toolkits are accurate, they are a bit complicated, and developing using those toolkits are difficult, because of improper documentation. Hence the HTK toolkit-based tool is analyzed and improved with web interface. None of these toolkits have been tested for continuous speech involving multiple speakers but alignment performed only on one speaker's audio and text transcripts.

2.8 Comparison of State-of-the-Art Forced Aligners

The AlignTool and MFA tool failed in the initial environment setup stage itself. Hence these tools are not analyzed further in our research work. Since DARLA and WebMAUS are web applications hosted in the United States and European Countries, uploading MULTISIMO corpus to these applications raises Data Protection issues. Hence these two tools are also not considered for further alignment process with MULTISIMO corpus. Since FAVE-Align requires pre-aligned transcript files and our main focus was to build an accurate automated aligner tool that align transcripts with audio files without any pre-alignment inputs provided.This tool was also not considered for further research work.

ProsodyLab is the only open-source CLI tool which was proved to be efficient in aligning audio files with the transcripts. This tool also had some drawbacks, such as it does not align accurately with noisy, and long pause speech data. It also failed when transcripts for some words in the audio are unavailable. Hence ProsodyLab tool was considered for further analysis with MULTISIMO corpus and improvements. The results of overall analysis is described in Table 2.1

2.9 Conclusion from Literature Review

By analyzing all the existing tools in the area of Forced Alignment and comparing their efficiency, it can be concluded that the ProsodyLab tool is the only available tool that is efficient enough to produce Forced Alignment results. Hence in the coming chapter, various methodologies are followed to analyze this tool and improve the accuracy of the tool, for our MULTISIMO corpus.

Tools Analyzed	**Results**
AlignTool	Content Failed to Align for MULTISIMO corpus without beeps
MAUS	Only Web Interface available. Hence has Data Privacy issue
MFA	Installation of the Package Failed
FAVE-Align	Pre-Time Aligned input required
DARLA	Only Web Interface available. Hence has Data Privacy issue
ProsodyLab Aligner	Successful in aligning the audio with transcripts and produce TextGrid files

Table 2.1: Forced Aligner tool analysis and results

Chapter 3

Methodology

3.1 Introduction to Methodology

Our research work started with analyzing existing tools and techniques available for the Forced Alignment process. During that study, the ProsodyLab tool was found to be fit to carry out further research work by aligning our MULTISIMO corpus using that tool. Hence the following methodologies are followed to accurately and automatically time-align the MULTISIMO corpus. This chapter includes sections in which, first the existing ProsodyLab tool analysis is specified followed by our implementations. The following sections represent the methodologies followed during our research work.

3.2 Analyzing existing ProsodyLab tool

At this stage the ProsodyLab tool was installed on a Virtual Machine in Azure. Followed by the analysis of the design flow of the tool and the results.

3.2.1 Setup and execution of existing ProsodyLab environment

The ProsodyLab tool can be set up on Linux, Mac OSX, and also on Windows. In our research work, we have set up the complete environment on Azure Virtual machine by installing its pre-dependencies like HTK toolkit, SOX and Python3. Since the

installation of the HTK toolkit and dependencies to execute the ProsodyLab tool has always created problems for users, in our research work we have developed a graphical user interface for the users hosted on the Azure Cloud and can be accessed by any users without any dependencies.

3.2.2 Design

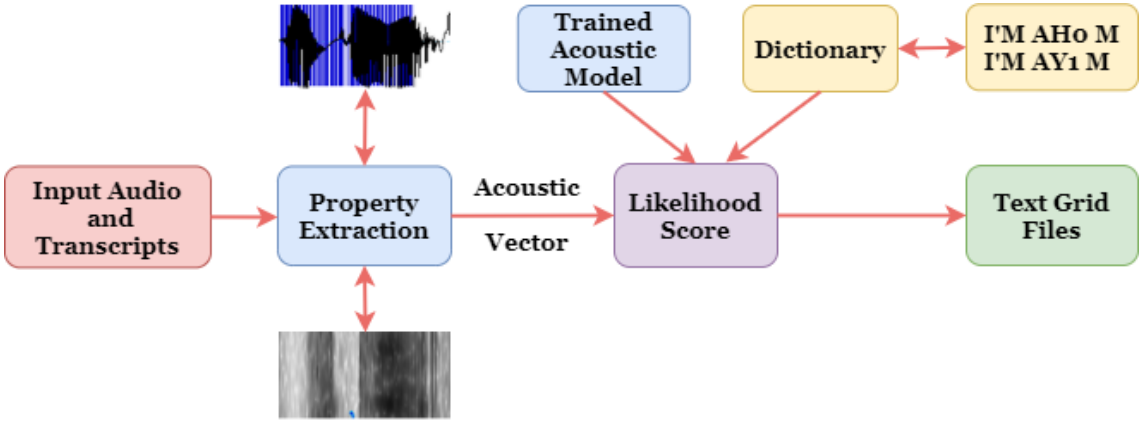

Figure 3.1: ProsodyLab's temporal Forced Alignment design flow

The design of the temporal Forced Aligner tool is as shown in Figure 3.1. On execution of the ProsodyLab CLI tool to train own acoustic model or with a pre-trained model, the audio and transcript files are taken as input. From these files, the various properties of the audio and texts are extracted. From the audio (.wav) files, the "Acoustic Vector" representing the speech signals are extracted. This extraction is done by HTK [Hatala, 2019]. Then the existing acoustic model, along with the dictionary and the files uploaded are estimated to identify the speech, silence, short pause, and the boundaries in continuous speech. Based on the likelihood score of the alignment, the time alignment results are written to Praat viewable TextGrid files.

3.2.3 ProsodyLab tool output analysis for MULTISIMO corpus

The output of the ProsodyLab tool when read using Praat showed inaccurate results at many places as highlighted in Figure 3.2. As visualized in Figure 3.2, the Red Highlight

Mark specifies the manual temporal alignment for the utterance "OK", whereas the output from the ProsodyLab tool misaligned this word and aligned it to time before 1.572 s. Hence the remaining audio and transcripts are also misaligned except at some places decreasing the accuracy of the ProsodyLab tool.

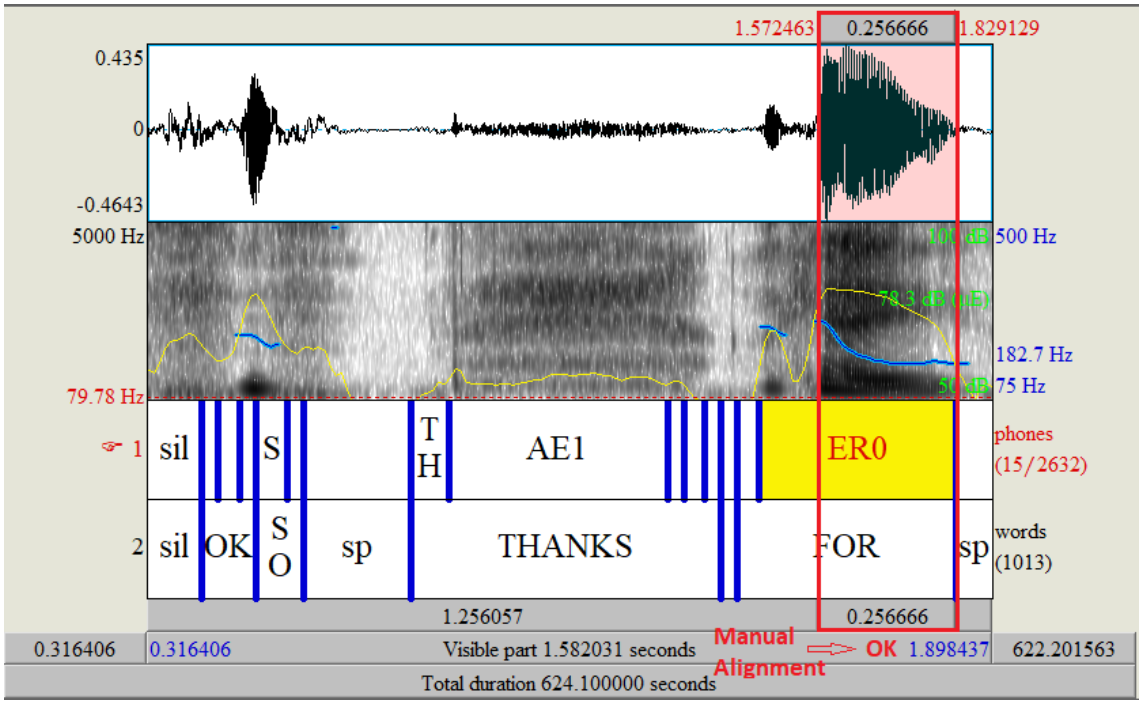

Figure 3.2: Misalignment of the utterance "OK" from the ProsodyLab tool

Following improvements are implemented in our research work after a thorough analysis of the ProsodyLab output, to improve the accuracy of the temporal alignment and delight the linguists as well as other users.

- Prepare and clean the MULTISIMO corpus transcripts by using text normalization and manual transcript correction techniques [Milne, 2015].
- Train own acoustic model using ProsodyLab tool for MULTISIMO corpus.
- Change the configuration of the HCopy and HERest to improve the efficiency of the HTK.
- Re-sample the audio files which are failing in existing tool.

- Build a user-friendly interface for the users to ease their Forced Alignment process.

3.3 Implementation

3.3.1 Data Preparation

Audio (.wav) files:

Testing of the ProsodyLab tool on the MULTISIMO corpus wav audio files was carried out, which was available at MONO.tar.gz in [Koutsombogera and Vogel, 2020a] for our research analysis. This directory consists of 54 mono audio files, 1 wav file for each speaker separated in 18 directories according to sessions. We have considered only the audio files of the Facilitator, where the microphones are placed at the facilitator end and the student's voices are considered as noise and short pause of the facilitator. The audio files are generated at laboratory conditions with less noise and disturbances to get accurate results. The audio files consist of a 48KHz Sample rate and 1 Channel, with a Bit Rate of 768kbps. 48KHz sample rate was re-sampled to 16KHz sample rate using SOX [Milne, 2015], since our acoustic model is trained on the 16KHz sample rate corpus. Each audio file consists of the recording of a facilitator and two students involved in the discussion for approximately 10 to 15 minutes. No pre-processing on audio files is done except re-sampling, since the audio files are generated under ideal normal condition with minimal noise which is considered as clean.

Transcription (.lab) Files:

Using Automated Speech-to-Text (STT) service by IBM Watson and Google we had tried to generate the transcripts. But Google Python API had the limitation on the length of the audio file to be 1 minute, and hence did not use the Google STT service. IBM Watson converted complete speech in the audio file to the Texts, but in our research work, more focus is on the time alignment of only facilitator's speech and transcripts. Hence the transcripts which are available at the MULTISIMO website consisting only facilitator's transcriptions for the facilitator's audio files are used. The transcriptions for the MULTISIMO audio files are available under 'Normalised_Transcripts'

folder inside transcripts and word frequencies.tar.gz in ANNOTATIONS directory [Koutsombogera and Vogel, 2020b]. These transcripts are generated using Transcriber tool by the researchers of MULTISIMO corpus.

All the transcription files are available as txt files in the MULTISIMO website and these files are again text normalised and cleaned to improve the accuracy of the alignment tool.

1. **Automated Text Normalization:** Our Forced Aligner Web Application requires transcriptions in the format specified in Figure 3.3 [Hazen, 2006]. Hence the normal text transcriptions which were available in the MULTISIMO corpus was pre-processed to match the format required by our application. That involved converting the texts in the transcripts to uppercase, removing all the punctuations (like ',' '!', '.', '?','+', '[',']') and removing the words (like 'UHUH', 'SH', 'AHM',' UMHM', 'EHM',' HMM', 'MHMM', 'HM', 'CU') for which no pronunciations are available in the CMU Pronunciation dictionary files, but are annotated manually in the MULTISIMO corpus using ELAN [Lenzo, 2020]. Some words which are in different forms in the dictionary pronunciation files are updated with same words in the .lab files. For example, "THAT'S" word in the transcript file is updated to "THATS" and many more to normalise the dictionary words with the transcripts words. The output of the automated text normalization is as shown in Figure 3.3. This automated Text Normalization is implemented using Python3 automation script.

2. **Manual Text Normalization of Transcripts:** The automated pre-processed .lab files are again read using ELAN tool to manually prepare the accurate transcript file for the speeches. Since the accuracy of any forced aligner tool depends on the accuracy of the transcripts for the speeches [Johnson et al., 2018]. In this, we again manually pre-processed and removed some words for which proper pronunciations are not available and missing-words are inserted, incorrect words are updated manually.

Dictionary (.dict) file:

The ProsodyLab tool provides a sample of North American English Dictionary (.dict) file which consists of pronunciations for words taken from CMU Pronunciation Dic-

M003_S20_audio_M_C.lab

```
SO HELLO THANKS VERY MUCH FOR COMING HERE TODAY AH WE'RE GOING TO PLAY A QUIZ I'M
GOING TO ASK YOU THREE SURVEY QUESTIONS THAT WERE EH PREVIOUSLY POSED TO A GROUP OF
HUNDRED PEOPLE YOU HAVE TO COME UP WITH THE THREE MOST POPULAR ANSWERS AND AFTER
TALKING TO EACH OTHER YOU YOU NEED TO COLLABORATE IN ORDER TO ORDER THESE ANSWERS
IN TERMS OF POPULARITY IS THAT CLEAR ARE YOU READY FOR THE FIRST QUESTION YEAH NO NO
JUST YEAH YEAH YEAH YEAH SO NAME A PUBLIC PLACE WHERE YOU'RE LIKELY TO CATCH A COLD
OR A FLU BUG MHM OR ANYTHING OR A COLD PERFECT YEAH PERFECT PERFECT SO YOU HA YOU
FOUND TWO ALREADY YOU FOUND THE SCHOOL AND HOSPITAL SO ONE MORE WHEN ITS YOU MENTIONED
PUBLIC TRANSPORT ITS ITS ITS ABOUT PUBLIC TRANSPORT SO WHAT WHICH MEANS YEAH OR PLANE
YEAH YEAH GUYS YOU FOUND ALL THREE YEAH SO CAN YOU RANK THOSE ANSWERS NOW PLEASE DO
YOU REMEMBER WHICH ONES THEY ARE ITS SCHOOL YEAH OK PERFECT SO WHICH ONE IS FIRST YOU
THINK YOU HAVE TO RANK THEM FROM THE MOST POPULAR TO THE LEAST POPULAR AH SO YOU YOUR
ANSWERS ARE HOSPITAL AND THEN AND THEN YOU WERE ALMOST THERE NOW THE FIRST ONE IS SCHOOL
YEAH YEAH THEN ITS HOSPITAL YEAH MHM TEACHERS AS WELL AND THE THIRD ANSWER WAS PLANE
APPARENTLY AH BUT STILL AH THATS GOOD AH ARE YOU READY FOR THE SECOND ONE EH NAME AN
INSTRUMENT YOU CAN FIND IN A SYMPHONY ORCHESTRA THATS PERFECT YOU GOT IT YEAH YEAH
PERFECT THATS THE SECOND ONE YEAH GREAT GUYS PERFECT SO YOU HAVE ALL THREE SO WE HAVE
VIOLIN EH DRUM AND CELLO AND CAN YOU ORDER THEM PLEASE NOW IN TERMS OF POPULARITY SO
WHAT DO YOU THINK PERFECT THATS IT THATS CORRECT WELL DONE GUYS VERY GOOD YEAH I THINK
SO EH THIRD QUESTION THE LAST QUESTION NOW NAME SOMETHING EXCUSE ME THAT PEOPLE CUT
EXCUSE ME CUT WHAT DO THEY CUT CAN BE ANYTHING YEAH THAT THEY CUT PEOPLE WOULD CUT YEAH
PAPER IS ONE YEAH VERY GOOD HAIR VERY GOOD YOU HAVE TWO SO ONE MORE YOU'RE CLOSE F MEAT
PERFECT YOU HAVE ALL THREE ANSWERS GREAT AND WOULD YOU ORDER THEM NOW PLEASE ARE YOU SURE
YEAH AH NO THE FIRST ONE IS HAIR THE MOST POPULAR ANSWER WAS EH WAS HAIR ACTUALLY YEAH AND
THEN PAPER AND THE LAST ONE WAS MEAT YOU DID GREAT WELL DONE OOP BUT EH THANKS VERY MUCH FOR
COMING I HOPE YOU ENJOYED IT YEAH THAT WAS IT AH THANK YOU
```

Figure 3.3: Transcripts after Text Normalization process

tionary [Gorman et al., 2020]. To train own languages including own pronunciations, the particular word, and its phonemes need to be included in the .dict file as shown in Figure 3.4. In our Forced Aligner application, the eng.dict file is provided for the MULTISIMO corpus used and through the user interface the user can even upload the updated dictionary files easily.

The dictionary file format is as specified in Figure 3.4, in which each word and its Phonemes are specified with space after the word and also single spaces in between phonemes. For a similar word if various pronunciations are available, then both pronunciations need to be included in separate lines inside the .dict file. The order of the word and its pronunciation must be sorted as per python sorting order and if not sorted, the tool will throw the Out-of-the-Vocabulary (OOV.txt) file consisting of missing words. For the MULTISIMO corpus, the .dict file has been created manually by adding missing words to the dictionary file and is available under the root directory

as eng.dict file [Gowda, 2020]. The user is given the option to upload their updated dictionary file through the user interface as well.

```
54332  I AY1
54333  I'D AY1 D
54334  I'ERS AY1 ER0 Z
54335  I'LL AY1 L
54336  I'M AH0 M
54337  I'M AY1 M
54338  I'VE AY1 V
54339  I. AY1
54340  I.S AY1 Z
54341  IA IY1 AH0
54342  IACOBELLI IY0 AA2 K OW0 B EH1 L IY0
54343  IACOBELLIS IY0 AA2 K OW0 B EH1 L IH0 S
54344  IACOBUCCI IY0 AA0 K OW0 B UW1 CH IY0
```

Figure 3.4: English dictionary file(eng.dict) consisting of words and pronunciations

Configuration (.yaml) file:

The configuration file gives settings and standard instructions to be used by the HTK while training HMM models. This file usually consists of phoneme-set which is the list of phonemes or characters used in the dictionary file, for which Single HMM definitions are trained for the corpus along with few other settings. Since our MULTISIMO corpus consists of English Language, the configuration file includes all other HTK settings along with phoneme-set of 69 English phonemes. The user can upload their own configuration file from our Web interface as well. This configuration file helps in aligning speech signals to respective transcriptions. The phoneme-set is based on the CMU Pronunciation dictionary since pronunciations for transcriptions follow the same. Hence the configuration file also includes ARPAbet phoneme-set convention. [Ma, 2012] Training the own language model for different languages requires this file to be updated with new configurations and phonemes on which the new language model has to be trained. For our research work, we reused the configuration config.yaml file provided by ProsodyLab and tried with various configuration changes to get the accurate alignment for our corpus through the User interface (Figure 3.5). When training our language model for MULTISIMO corpus, we changed the TARGETRATE value to 200000, since the ProsodyLab tool was failing because of the mismatch in

the TARGETRATE. We also trained our model without re-sampling and changing the audio files sample-rate to 48KHz to check the performance of the model.

```
HCompV: {F: 0.01}
HCopy: {CEPLIFTER: 22, ENORMALIZE: T, NUMCEPS: 12, NUMCHANS: 20, PREEMCOEF: 0.97,
  SOURCEFORMAT: WAVE, SOURCEKIND: WAVEFORM, TARGETKIND: MFCC_D_A_0, TARGETRATE: 100000.0,
  USEHAMMING: T, WINDOWSIZE: 250000.0}
HERest: {CEPLIFTER: 22, ENORMALIZE: T, NUMCEPS: 12, NUMCHANS: 20, PREEMCOEF: 0.97,
  TARGETKIND: MFCC_D_A_0, TARGETRATE: 100000.0, USEHAMMING: T, WINDOWSIZE: 250000.0}
HVite: {SFAC: 5}
URL: http://prosodylab.org/tools/aligner/
authors: Kyle Gorman
citation: 'K. Gorman, J. Howell, and M. Wagner. 2011. Prosodylab-Aligner: A tool for
  forced alignment of laboratory speech. Canadian Acoustics, 39(3), 192-193.'
epochs: 10
language: English
phoneset: [AA0, AA1, AA2, AE0, AE1, AE2, AH0, AH1, AH2, AO0, AO1, AO2, AW0, AW1, AW2,
  AY0, AY1, AY2, EH0, EH1, EH2, ER0, ER1, ER2, EY0, EY1, EY2, IH0, IH1, IH2, IY0,
  IY1, IY2, OW0, OW1, OW2, OY0, OY1, OY2, UH0, UH1, UH2, UW0, UW1, UW2, B, CH, D,
  DH, F, G, HH, JH, K, L, M, N, NG, P, R, S, SH, T, TH, V, W, Y, Z, ZH]
pruning: [250, 100, 5000]
samplerate: 16000
```

Figure 3.5: Configuration file (config.yaml) consisting of HMM model settings

3.3.2 Creating own acoustic model using ProsodyLab

MULTISIMO audio and normalized transcripts are used to train our own HTK HMM acoustic model definitions. Since we had audio files of approximately 1 hour of speech signals in total, which included 3 different facilitator's speech for the alignment to be tested. ProsodyLab tool was used to train our own HMM model and its design is as followed.

For training our own pronunciations, the HMM model produces an acoustic model output, which was then used by Forced Aligner application to align the label files with .wav files. Since ProsodyLab is a Command-Line-Interface the training of our model was executed by running ProsodyLab python script in the Azure Virtual Machine terminal where the tool was set up. While executing the tool, the configuration file which had settings as in Figure 3.5, English dictionary file (eng.dict), the path to the directory containing audio, .lab transcript files, and the file to which our model has to write to (the output .zip file name) are entered. In our research work, we have used MULTISIMO corpus for both training and testing purpose.

During the process of training own acoustic language model, the ProsodyLab (align.py) python script first verified if any audio or equivalent transcript files used in the training were missing. If for any speech file, the transcripts with the same name were missing in .lab format, then the program terminated. The tool then looked at the pronunciations for the words in transcripts with that in the dictionary file. If any OOV words were present in the transcript files, but were absent in the dictionary file, then the program terminated by writing those words to a text file. These steps help the Linguists to properly train the model with correct data.

ProsodyLab training process includes three stages as in Figure 3.6.

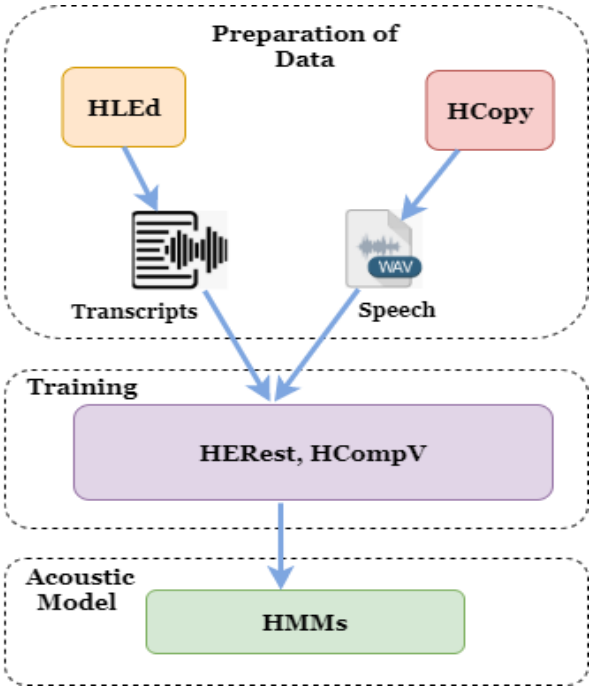

Figure 3.6: Building own acoustic language model design

Preparation of Data:

Building an acoustic model takes place in various stages involving data preparation, training, and finally model creation. [Thomas et al., 1998]. For each phoneme in the configuration file MFCC model is built, since alignment is done at both word and

phoneme level. These models are based on Monophone Gaussian Mixture consisting of 39 MFCC (Mel Frequency Cepstral Coefficients). For our corpus, we tried using both MFCC_D_A_0 and MFCC_O_D_A 'TARGETKIND' and found no significant variations in the acoustic model output. Before starting the model training, data preparation was carried out. ProsodyLab training HMM model script requires a path to the folder containing the audio file and the transcript files based on which model is trained and built. Once the tool reads all the .wav files and .lab files in that folder, it internally calls the HLEd and HCopy toolkit of HTK to process the transcript files and audio files.

- **HLEd:** HLEd mainly manipulate the .lab files. This toolkit merges all the transcript files into a single integrated file. During this label file manipulation process three separate '.mlf' files are created.

  - Word.mlf consists of merged transcript file words as each word in one line.
  - Phone.mlf consists of each phone in a single line same as the word.mlf.
  - Taskdictionary.mlf is the merge of word.mlf and phone.mlf aligned parallely.

  Using HLEd even 'SIL' (Silence) and 'Sp'(Short Pause) are also added in .mlf files that depict the non-speech areas in transcripts.

- **HCopy:** Recognition of speech by HTK takes place through the transformation of speech signals to vector representation of the words. This transformation of speech from analog signals to the acoustic vector involves the following stages.

  - Conversion of analogue speech signals to digital signals.
  - Splitting of digital signals to overlapping frames of 10 ms in order to analyse the speech and non-speech signals.
  - Creation of acoustic vectors for each signal frame using MFCC.

  The preparation of audio files includes parameterizing the analog speech waveforms into a feature vector sequence. ProsodyLab tool uses HCopy which derives MFCCs vectors from FFT-based log spectra. This tool uses a configuration file uploaded while executing ProsodyLab, where all the parameters and its values are specified for training the model as per Figure 3.7.

```
SOURCEKIND: WAVEFORM
SOURCEFORMAT: WAVE
TARGETRATE: 100000.0
TARGETKIND: MFCC_D_A_0
WINDOWSIZE: 250000.0
PREEMCOEF: 0.97
USEHAMMING: T
ENORMALIZE: T
CEPLIFTER: 22
NUMCHANS: 20
NUMCEPS: 12
```

Figure 3.7: Configuration parameters used by HCopy and HERest tools

Training:

The output from the data preparation stage is used in training the HMM model. The ProsodyLab training starts by initializing the flat start for mono-phones which after four iterations in a single round are then submitted for model estimation. Flat start means initializing all phoneme models with identical value along with assigning global speech mean and variance values to state means and variances using the HCompV tool. HERest is the main tool of the HTK toolkit which supports parallel execution, and requires less computation. After the initial model estimation, HERest is used to perform embedded training over a complete training set. HERest first performs Baum-Welch re-estimation of HMM phone models at a time, where for each word the high likely phone models are appended and then the forward-backward algorithm of HMM is used. This is used to compute the State means, variances, occupations, etc for each HMM phone sequence. After finishing the processing of all data, the computed statistics are further used to re-estimate the HMM state parameters. A tied state Sp (Short Pause) model is inserted when doing the second round of model estimation. In the next round, the data is aligned to identify the most likely pronunciation amongst the pronunciations in the dictionary which has various pronunciations for the same word followed by the final round of model estimation.

The training process starts with single Gaussian flat context-independent phoneme

models and then iterates to become context-dependent by using various Gaussian distribution components. The HERest and HVite are used to adapt the model for speakers while using a small amount of training data. So, in our project, since we had very little data for each speaker, which was of around 30 minutes to 50 minutes, this training own model gave inaccurate and unreliable alignments. Building context-dependent models always require a large amount of accurate data.

Acoustic model generation:

The HMM acoustic model after training consists of hmmdefs and macros files as in Figure 3.8 and Figure 3.9. The models trained with large data-set can be used to align any data accurately of that language which has the same phones.

From this trained model, the accurate and high probable alignment is computed by finding the start and end times of the resulting word and phonemes. The resulting output is written to TextGrid files that can be analyzed through Praat.

```
1  ~o
2  <STREAMINFO> 1 39
3  <VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
4  ~v "varFloor1"
5  <VARIANCE> 39
6   7.614123e-01 5.564770e-01 6.510779e-01 6.822374e-01 6.669589e-01
    6.252755e-01 5.641341e-01 4.775682e-01 4.213330e-01 3.489751e-01
    3.304530e-01 2.477210e-01 9.907007e-01 3.078064e-02 2.260722e-02
    2.316771e-02 2.807352e-02 2.922030e-02 2.852915e-02 2.823248e-02
    2.701484e-02 2.413328e-02 2.090735e-02 1.881183e-02 1.520878e-02
    2.821474e-02 4.325620e-03 3.592001e-03 3.536991e-03 4.540271e-03
    4.910931e-03 4.898836e-03 4.960256e-03 4.838547e-03 4.347010e-03
    3.860147e-03 3.442934e-03 2.809061e-03 4.257455e-03
```

Figure 3.8: Snippet of Macros HMM model file consisting of global speech variances

The results of training our own HMM models for the MULTISIMO dataset using the ProsodyLab tool were highly inaccurate because of the very less number of training audio and transcript files. Hence the ProsodyLab Direct Alignment is analyzed and improved to get more accurate results.

```
10  ~h "B"
11  <BEGINHMM>
12  <NUMSTATES> 5
13  <STATE> 2
14  <MEAN> 39
15   4.896167e-02 3.541217e+00 3.219681e+00 9.104447e-01
     -5.735697e-02 -2.283369e+00 -4.667334e+00 -3.599202e+00
     -2.387662e+00 -1.556020e+00 -2.037103e+00 -2.896153e+00
     4.924449e+01 -4.436838e-01 7.221255e-01 3.617127e-01
     8.099178e-01 7.756979e-01 5.415033e-01 7.237800e-01 9.774724e-02
     -2.172609e-02 -1.563613e-01 -6.195498e-02 -3.005130e-02
     -1.460977e+00 -3.629116e-01 -5.519280e-01 -1.599029e-01
     -4.314187e-01 -3.591531e-01 -2.810814e-01 -8.553025e-02
     1.792924e-01 1.511019e-01 8.500999e-02 6.434925e-02 1.306216e-01
     8.394083e-01
16  <VARIANCE> 39
17   3.587453e+01 3.467567e+01 3.582057e+01 3.347437e+01 3.250980e+01
     3.651354e+01 3.123032e+01 2.715129e+01 2.840481e+01 2.586110e+01
     2.650769e+01 1.890066e+01 3.386187e+01 1.576215e+00 2.026930e+00
     1.724173e+00 2.243353e+00 1.915424e+00 2.124386e+00 2.181717e+00
     2.041437e+00 2.064710e+00 1.869656e+00 1.637504e+00 1.378601e+00
     2.643441e+00 2.260599e-01 3.311397e-01 3.405963e-01 4.171974e-01
     3.893666e-01 4.169949e-01 4.424548e-01 4.213894e-01 4.140867e-01
     3.795394e-01 3.294593e-01 2.802288e-01 4.296079e-01
18  <GCONST> 1.114826e+02
19  <STATE> 3
20  <MEAN> 39
21   -4.539155e+00 2.476229e+00 2.568129e+00 1.315827e+00
     1.030823e+00 -1.134083e+00 -1.960452e+00 -1.681930e+00
     -1.222486e+00 -8.714221e-01 -1.323531e+00 -1.642802e+00
     4.595878e+01 -9.060119e-01 -1.501232e+00 -3.627147e-01
     -8.808589e-01 -6.900488e-01 -8.198233e-01 -1.483240e-01
     2.788869e-01 4.869705e-01 1.954952e-01 2.190261e-01 3.040045e-01
     2.377116e+00 3.779463e-01 -5.068546e-01 -8.239527e-02
     -6.860350e-01 -6.543548e-01 -4.422665e-01 -4.870821e-01
     -1.065148e-01 -3.611135e-02 6.084453e-02 6.583720e-03
```

Figure 3.9: Snippet of Acoustic HMM model definition for 'B' Phone

3.3.3 Developing Web Interface

ProsodyLab lacks a user-friendly web interface that would help linguists to use it much easily and comfortably. [Johnson et al., 2018] in their research work has also mentioned the trouble of using the ProsodyLab command-line interface tool. Hence in our research work, we have provided a web interface to help even the linguists without any programming experience to align their data-set without complicated HTK tools and Python installation.

Python Flask API

A Python Flask API project is created which allowed us to build a web application. Flask is a web framework and consisting of various modules and libraries which in turn helped us to develop a web interface for our Aligner without concentrating on protocols. This is mainly based on the Jinja2 template engine and Web Interface Gateway Interface (WSGI). Forced Aligner web interface has been developed for ProsodyLab with changes to make it accessible by the web users. A Flask web application was developed and deployed in the Azure Virtual Machine and the code is available in the Github [Gowda, 2020] to host the application in any other environment with pre-dependencies of Python, HTK toolkit, and also SOX.

This web application can be used by any user irrespective of his/her programming proficiency, unlike using the ProsodyLab tool. The User Interface can be used only to align the audio files with the transcripts. The user interface is not available for training own acoustic models for different languages. But, this interface can be used to align the files of languages other than North American English also by using ProsodyLab tool to create own acoustic model and our web application to quickly align. For aligning understudied and other languages, the user is provided with the option to upload his config file on which the model has to be trained. The user can even choose their dictionary file containing words and their pronunciation. If any words are missing during the alignment process an OOV.txt file is generated as shown in Figure 3.13. The user can update the dictionary file based on the missing pronunciation for the word in the OOV file and restart the alignment process.

The MULTISIMO corpus audio and transcript aligner web application is as shown in Figure 3.10.

- **Step 1. Re-sample audio files:** In the first step of forced alignment process using our web application, users are provided with an option to choose their audio (.wav file/s) and re-sample them to 16KHz sample rate. To re-sample audio from the different sample rates of 41-48 KHz to 16KHz (standard sample rate supported by HTK tool kit), we are using the SOX tool. When the user uploads the audio files and clicks on the "Re-Sample" button, our application will call the SOX tool and re-sample the file to 16KHz (Figure 3.10). ProsodyLab tool and failing to Re-sample the audio files. Hence we have implemented this re-sampling in our web application as the first step in Forced Alignment process, to get accurate alignments.

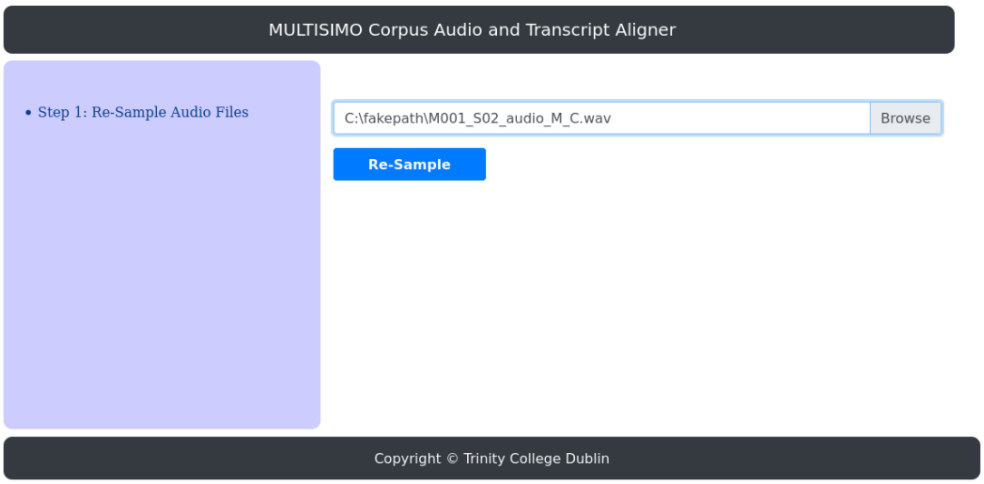

Figure 3.10: Re-sample MULTISIMO corpus using Flask web application

- **Step 2. Upload transcripts and align:** After the re-sampling of the uploaded audio files to 16KHz, the user is given option to upload all the transcription .lab files for the audio file, config.yaml file (containing configurations of the trained acoustic model) and .dict file (dictionary file) as in Figure 3.11. On click of the "Align Files" button, the Forced Alignment process starts and if any missing pronunciations are found, then list of words are available to download by users as in Figure 3.13 or else the TextGrid files with time aligned data will be available to download as .zip file consisting of model alignment score as well (Figure 3.12).

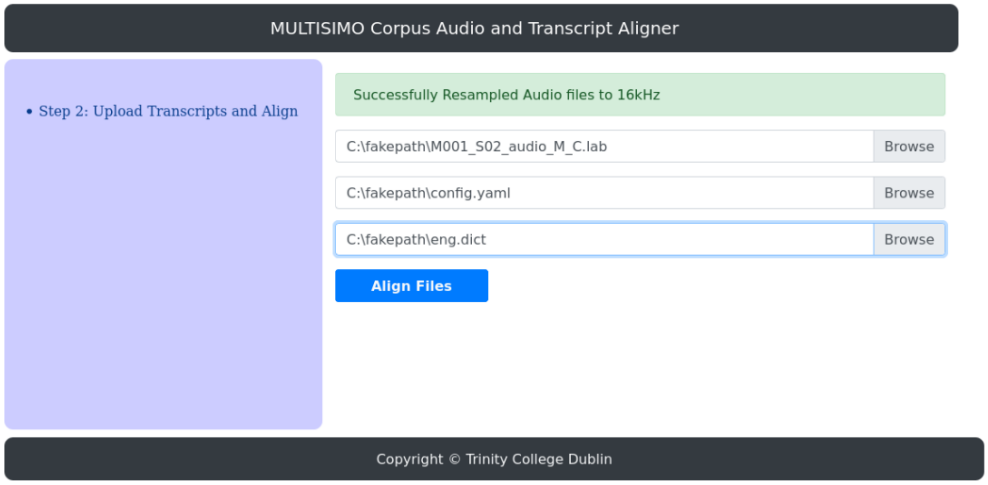

Figure 3.11: Upload files for the alignment

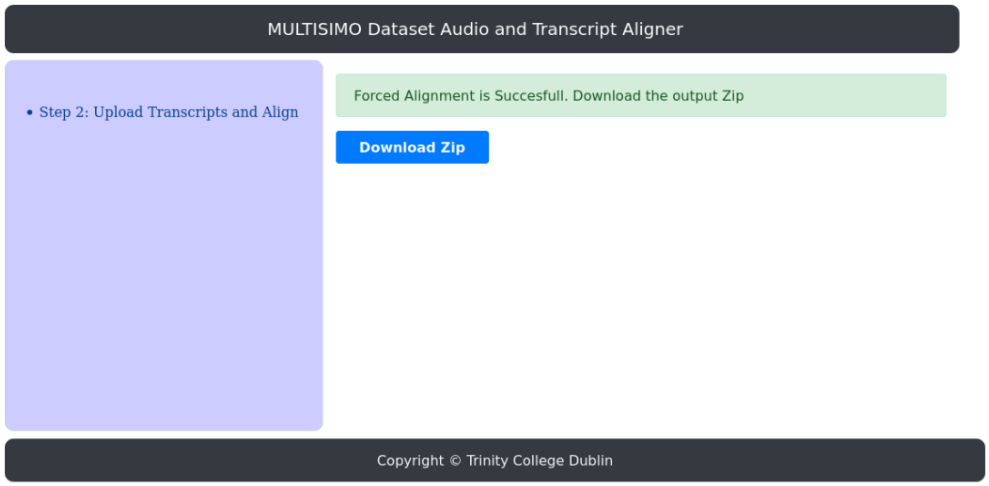

Figure 3.12: Download TextGrid aligned output Zip

This web interface is not specific to the MULTISIMO corpus and can be used for any continuous speech and/or transcripts for which time-based alignment is required.

3.3.4 Forced Alignment using our web application

Once the acoustic training model is available for the language analysis, that model is used to align audio with transcripts in the form of a two-layered time aligned TextGrid file which can be viewed through Praat. The Forced Alignment is a process that

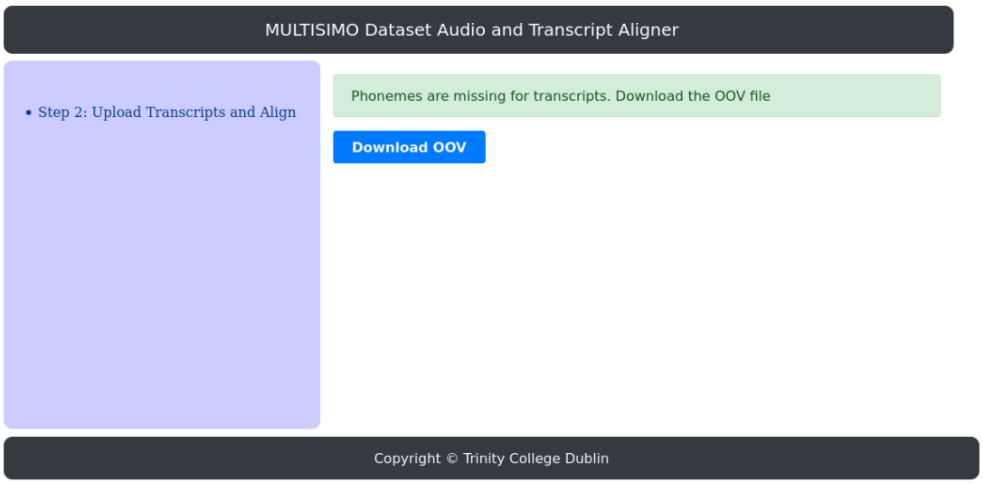

Figure 3.13: Download the OOV missing word file

happens when all the necessary required files are uploaded to the HTK tool. As in Figure 3.14, our web application takes an audio file as input and re-sample it to 16KHz sample rate. In the next step, the uploaded configuration .yaml file will be replaced with the existing trained model configuration file to include some model parameter changes. Then the .config, .lab, and .dict files along with audio files are processed using the HTK toolkit. This involves the following steps.

Preparing Label Files:

The uploaded .lab files are verified against the .dict file for pronunciation match, and new .lab and .mlf files are created as follows.

1. Words read from the transcript lab files are written to separate word lab files.

2. Pronunciation for each word in that file is fetched from the .dict file and written to the 'phonelab' file. If pronunciations are missing, then those words are added to the new .txt file to report as OOV and program is exited. Else, the word list is made by running through the HTK HDMan tool which includes SIL in 'phonelist' and 'taskdict'. Finally, the word list, phone list, and task dictionary are re-estimated through the HLEd tool.

Preparing Audio Files:

The re-sampled audio file from the web interface is executed using the HCopy tool to extract the audio feature vectors.

Align using HVite:

The HVite perform the final round of alignment by estimating the probability of each vector segment in the audio with the pronunciation and then aligning the transcripts with the speech.

The HVite chooses the pronunciation with the highest probability of a match. The HVite outputs the pronunciations, words, and their boundaries to a file which is then put into TextGrid programmatically.

Produce two-tiered Praat TextGrid file:

The output of the alignment process is the two-layered TextGrid file consisting of "words" and "phones" layers. The silence is labeled as "sil" and included at the start and end of the audio file. Short Pause is labeled as "sp" and included between the words when there is a pause between speaker responses. The "phones" layer consists of pronunciations. This TextGrid file is available for the user to download as a .zip file and is used to analyze phones and perform other language analysis by reading those TextGrid and audio files in Praat or ELAN tool.

3.4 Conclusion from methodologies followed and implementations

In the next chapter, implementation of our research work is evaluated and interpreted whether our application replaces the human alignments or not. The implementation of web application and its alignments are evaluated to determine the accuracy of our application by comparing with the manual alignments. The evaluation of our implementation and the results can be found in the next chapter.

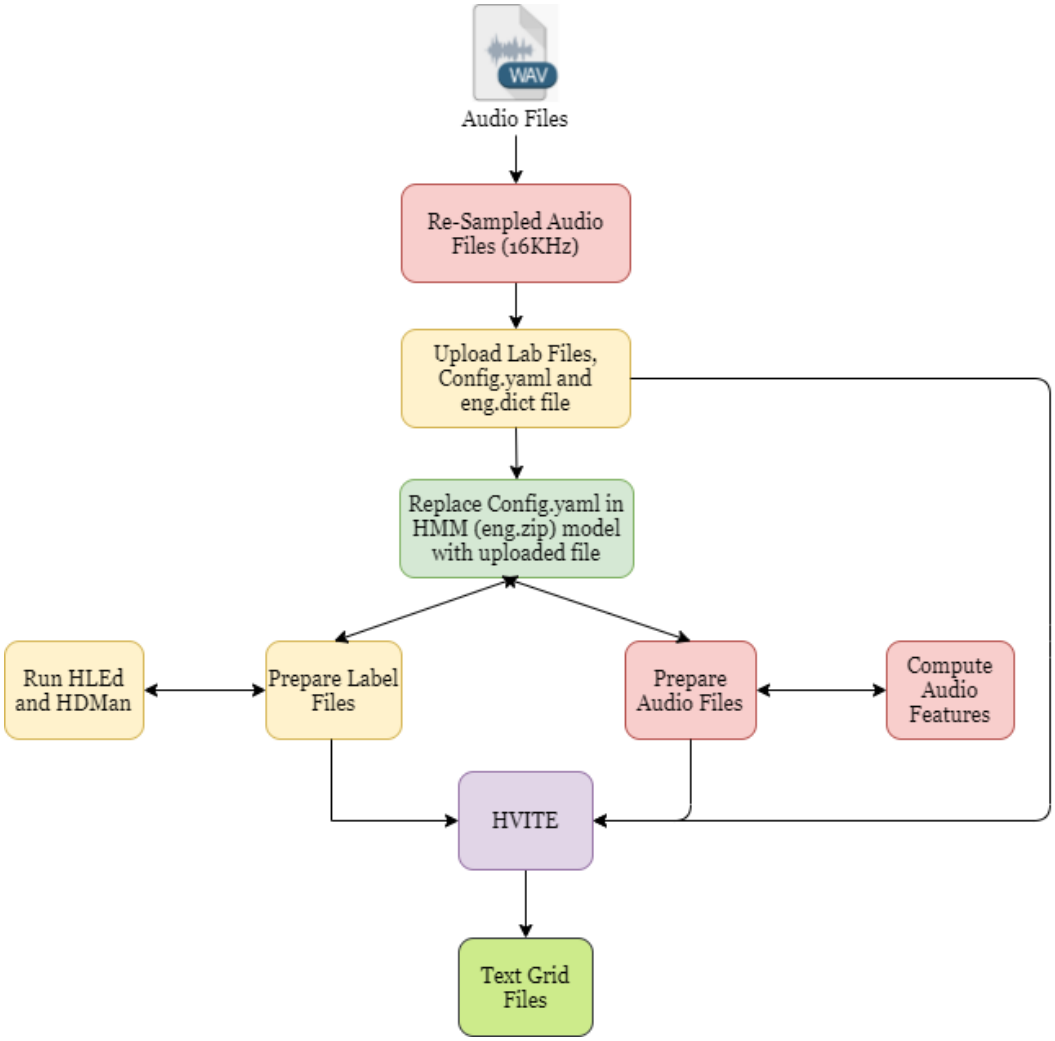

Figure 3.14: Forced Alignment process using web interface

Chapter 4

Evaluation

4.1 Introduction to Evaluation and its outcome

In this chapter, our automated temporal Forced Alignment Web Application implemented is evaluated, by comparing our alignment results with the manual alignment results to find the accuracy of the application. The evaluation is also carried out to find out the correlation between the alignment start-time and end-time using Spearman's Rank Correlation technique. The results of the evaluation are presented at the end of the chapter.

4.2 Selection of corpus for Evaluation

Even though we carried out the Alignment Process on complete MULTISIMO corpus, the evaluation of our Forced Aligner Web Application is done on randomly selected MULTISIMO audio and transcript file that is "`M003_S20_audio_M_C.wav`" file and "`M003_S20_audio_M_C.lab`" file. The audio file is available with the same name at [Koutsombogera and Vogel, 2020a], but the respective Text-Normalised transcript file is renamed to be the same as that of the audio and is available at [Gowda, 2020]. The speech and the transcript in this test corpus consist of the main speaker as the facilitator. The audio includes 3 speeches, one is of the facilitator, and the other two of the students. In this research work, we are mainly focused on analyzing the speech of one speaker in a continuous speech signal file. Hence facilitator's speech is considered

as significant speech and the other speech as short Pause or silence of the facilitator. The transcript is also available only for the Facilitator. The transcripts do not include any start and end times at word or phone level. The length of the test audio file is 362.61 seconds ($\sim 6.03 minutes$).

4.3 Forced Aligner Evaluation

Two different Forced Aligner outputs are evaluated to verify the quality of the Forced Alignment tool for MULTISIMO test corpus;

4.3.1 Forced Alignment Tool Evaluation for our own MULTISIMO corpus Acoustic Model

The ProsodyLab tool is used to build our own acoustic model for the complete MULTISIMO corpus. We trained our own acoustic model using the ProsodyLab tool training module and then aligned the test corpus using our web application. The snippet of the Evaluation results, when visualized through Praat, is as depicted in Figure 4.1. This visually proves that the results are highly inaccurate and most of the words are misaligned.

In the Figure 4.1 we can see a top layer containing Speech Signals and a bottom layer containing "phones" and "words" tiers representing test transcripts. As seen in the Figure 4.1 only "SO", "HELLO", "THANKS" transcript words are aligned with the audio signals whereas the audio file contains [SO HELLO THANKS VERY MUCH FOR COME HERE] these words in that window.

Only the "SO" word is time aligned accurately, whereas remaining all other words are misaligned. Green Colour text and line represent the actual time alignments in the Figure 4.1.

Evaluation Outcome of our own acoustic model

The visual inspection of the TextGrid output for our model trained on MULTISIMO corpus shows that the results are highly inaccurate and misaligned. Even by trying the various model training configuration like changing the "TARGETKIND" of the HMM

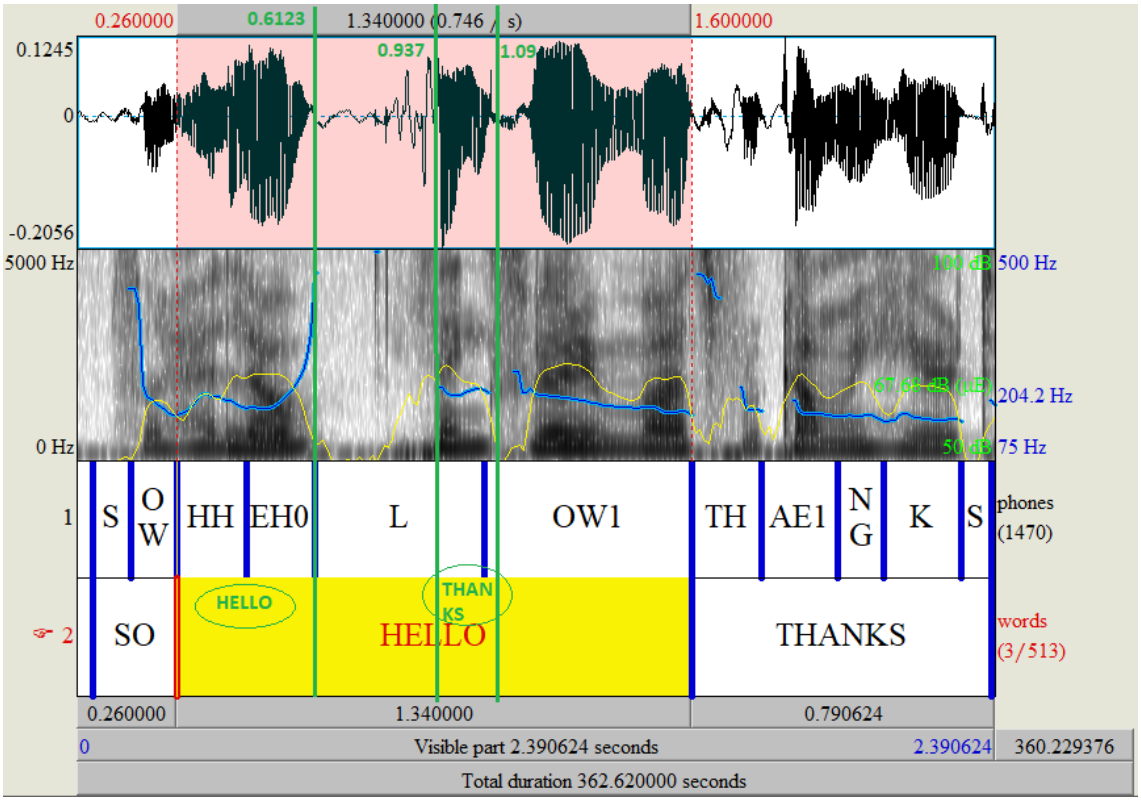

Figure 4.1: Snippet of misalignment of start and end time of words for our own acoustic model

model to MFCC_0_D_A instead of MFCC_D_A_0 resulted in the same misalignment output. The misalignment is mainly observed because of the following reasons as per our analysis;

- The MULTISIMO audio and transcripts used during the model training are insufficient for the Forced Alignment process to be carried out using that model.
- ProsodyLab tool requires a minimum of 1 hour of audio and clean transcripts. Whereas, MULTISIMO corpus even though contains a total of more than 1 hour of corpus, because of the various speakers, accurate model building is difficult using MULTISIMO corpus.
- The acoustic model is inaccurate because it's trained only on the facilitator's speech and transcripts, whereas the corpus includes other speaker's speech as well which are considered as Noise during training own model.

Hence MULTISIMO corpus is insufficient to build our own acoustic language model.

4.3.2 Forced Alignment Web Application Evaluation using ProsodyLab North American English Language acoustic model

Our Automated Temporal Forced Alignment Web Application is used with a pre-trained North American English Language acoustic model. This model produces alignments of high-quality [Gorman et al., 2011]. This model is trained on TIMIT corpus which includes high-quality speech signals and hence this model can be used by any other Forced Aligners without building their own model [Mohamed et al., 2009].

The output of our Web Application when visualized in Praat gives almost accurate alignment of 2 ms of difference in some areas which is acceptable even for manual alignment as per [Johnson et al., 2018]. The output from our Forced Alignment Web Application is as shown in Figure 4.2. When our output is analyzed manually using Praat, at most of the places the start-time and end-time of word utterances match with the manual alignment as in Figure 4.2.

Hence evaluation is carried out on this output to evaluate the quality and accuracy of our Forced Aligner Web Application. Evaluation is carried out by comparing the automated temporal alignment results with the manual temporal alignment results at the word level. The manual alignment for the test corpus at the word level is found out by listening to the Audio file and aligning the words in the transcripts with the audio using Praat. The start and end time for the manually aligned corpus is recorded in the Evaluation excel as shown in Figure 4.3. The start and end time of the words in the test corpus through our automated Forced Alignment process is also recorded in the same excel. The mismatch between manual and automated alignment in 'Column G' is found out by calculating the difference in start times and end times between both types of alignment. If there is any difference in either of the times it is marked as a mismatch and the '0' value is assigned for the particular mismatch word. 'Column I' represents the difference in end times of the two alignments and 'Column H' represents the difference in start times. 'Duration-Delta' is the delta difference of the word pronunciation times (i.e (D3-C3)-(H3-G3)). 'KBG-Match' values are similar to 'Column G'. 'Automatic-Duration' is the total duration taken by the utterance when aligned automatically

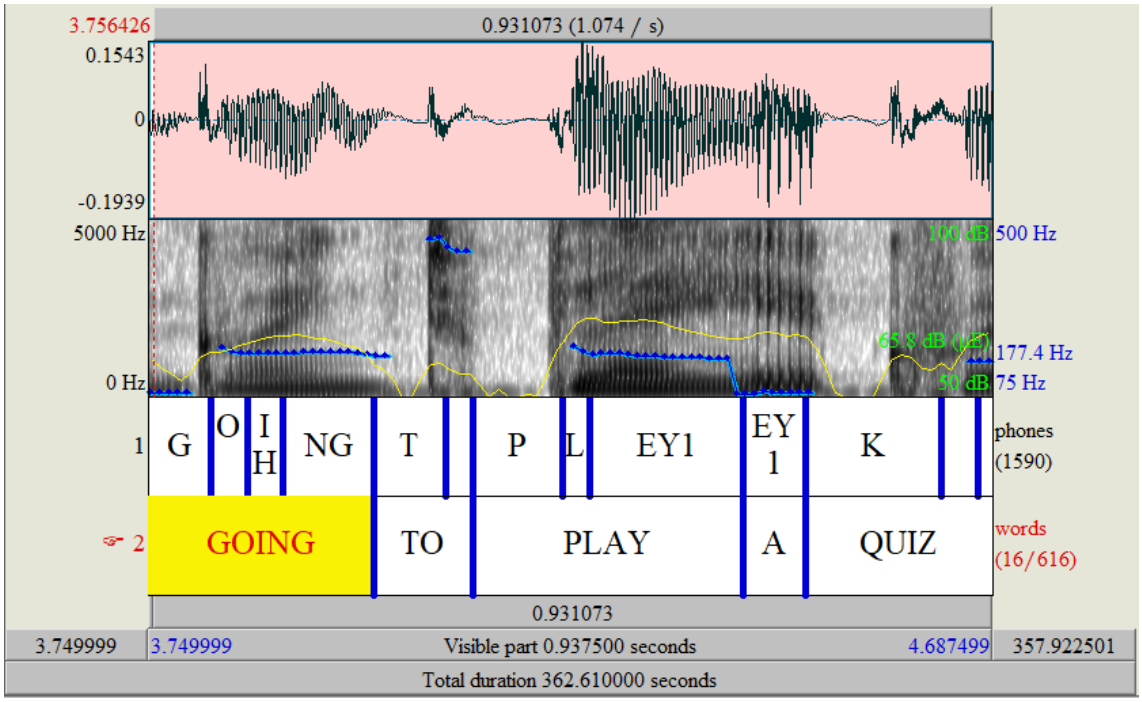

Figure 4.2: Accurate Forced Alignment output of our Web Application view in Praat

and 'Manual-Duration' is the same utterance duration when estimated manually. This Evaluation data is used to find the correlation and also the accuracy of our Aligner web interface output.

The quality of the Forced Aligner web application is evaluated by finding the Gross Errors in the manual and automated alignments as follows;

Mismatch in start and end times among two types of alignments:

This evaluation helps in finding out the accuracy of the system.We have used the R tool to find out the matches between two types of alignments using the below formula in R, after reading the Evaluation output excel available at [Gowda, 2020] into R project.

DA25 <- read.csv("Evaluation-v1.5.csv", header=TRUE, sep=",")
with(DA25, xtabs(~KBG.Match))

The output of this evaluation is as depicted in Table 4.1.

From this, it can be concluded that there are more matches than mismatches. Hence the accuracy of our Forced Aligner for the test corpus is calculated as follows ;

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Aligner Tool Output			Manual Alignment Output			Evaluation						
2	Words	Start Time(ms)	End Time(ms)	Words	Start Time(ms)	End Time(ms)	Mismatch between manual and automatic(Match=1, Mismatch=0)	Start-Time-Delta	End-Time-Delta	Duration-Delta	KBG-Match	Automatic-Duration	Manual-Duration
199	PLANE	90	90.66	PLANE	90	90.66	1	0	0	0	1	0.66	0.66
200	YEAH	90.66	90.96	YEAH	90.66	90.96	1	0	0	0	1	0.3	0.3
201	sp	90.96	91.21	sp	90.96	91.21	1	0	0	0	1	0.25	0.25
202	YEAH	91.21	91.7	YEAH	91.21	91.7	1	0	0	0	1	0.49	0.49
203	GUYS	91.7	92.02	GUYS	91.7	92.02	1	0	0	0	1	0.32	0.32
204	sp	92.02	92.12	sp	92.02	92.12	1	0	0	0	1	0.1	0.1
205	YOU	92.02	92.29	YOU	92.02	92.29	1	0	0	0	1	0.27	0.27
206	FOUND	92.29	92.6	FOUND	92.29	92.6	1	0	0	0	1	0.31	0.31
207	ALL	92.6	92.76	ALL	92.6	92.76	1	0	0	0	1	0.16	0.16
208	THREE	92.76	93.3	THREE	92.76	93.3	1	0	0	0	1	0.54	0.54
209	sp	93.3	93.78	sp	93.3	93.78	1	0	0	0	1	0.48	0.48
210	YEAH	93.78	94.59	YEAH	93.78	94.59	1	0	0	0	1	0.81	0.81
211	SO	94.59	96.25	SO	94.59	99.01	0	0	-2.76	-2.76	0	1.66	4.42
212	sp	96.25	97.11	sp	99.01	99.16	0	-2.76	-2.05	0.71	0	0.86	0.15
213	CAN	97.11	97.79	CAN	99.16	99.35	0	-2.05	-1.56	0.49	0	0.68	0.19
214	sp	97.79	99.01	sp	99.35	99.38	0	-1.56	-0.37	1.19	0	1.22	0.03
215	YOU	99.01	99.47	YOU	99.38	99.47	0	-0.37	0	0.37	0	0.46	0.09
216	RANK	99.47	99.73	RANK	99.47	99.73	1	0	0	0	1	0.26	0.26

Figure 4.3: Evaluation data snippet containing Manual and Automated alignment results

KBG.Match	
0	1
141	475

Table 4.1: Count of Matches (1) and Mismatches (0)

Accuracy of our Forced Aligner = (475/(141+475))*100 = 77.11%

Evaluating the mismatch between Manual and Automatic duration with respect to longer event duration:

To perform this evaluation the mean of the Manual Duration is grouped on 'KBG.Match' column results and the Automatic Duration is also grouped on 'KBG.Match' as shown below.

with(DA25, tapply(Manual.Duration, list(KBG.Match), mean))

KBG.Match	
0	1
0.9433333	0.4886947

Table 4.2: Mean of Manual Duration for matching utterances

with(DA25, tapply(Automatic.Duration, list(KBG.Match), mean))

KBG.Match	
0	1
0.9450355	0.4886947

Table 4.3: Mean of Automatic Duration for matching utterances

As expected from Table 4.2 and Table 4.3, there is no difference in duration in the cases of matched annotations.

Finding Correlation between the difference in start and end times when there is mismatch:

Using Spearman's Rank Correlation rho:

This is a non-parametric measurement of strength and direction of association that exists between two variables and is measured on an ordinal scale denoted by rho. By using this method the correlation was found out between Delta start and Delta end time and the results are as shown below;

with(DA25[DA25$KBG.Match= =0,],summary((Start.Time.Delta-End.Time.Delta)))

Results:
S = 78161, p-value < 2.2e-1)
Alternative hypothesis: true rho is not equal to 0
sample estimates:

rho
0.8326964

From the above results, it seems to be a greater magnitude of a mismatch at the end times than with start times as rho is greater than '0'.

Spearman's Rank Correlation rho:

The Spearman's Rank Correlation method is applied to find the correlation rank between start-time delta with automatic and manual duration as well as end-time delta

with automatic and manual duration.

with(DA25,cor.test(End.Time.Delta,Automatic.Duration,method="spearman"))
rho = -0.006019836; S = 39191897; p-value = 0.8815

with(DA25,cor.test(End.Time.Delta,Manual.Duration,method="spearman"))
rho = 0.05809468; S = 36694163; p-value = 0.1498

with(DA25,cor.test(Start.Time.Delta,Manual.Duration,method="spearman"))
rho = 0.1728568; S = 32223333; p-value = 1.603e-05

with(DA25,cor.test(Start.Time.Delta,Automatic.Duration,method="spearman"))
rho = -0.01564049; S = 39566692; p-value = 0.6984

Note:
If rho = +1 then perfect association of ranks.
If rho = 0 then no association between ranks.
If rho = -1 then negative association of ranks.
If rho is approximately equal to 0, then weaker association between ranks.

Hence rho of 0.05 and -0.006 signifies that there is a weaker association of end times, whereas some association is there with the manual start time delta. From the evaluation results, it's observed that only the difference in start times between manual and automatic annotation has a significant correlation with the duration. The correlation is with the manual duration and is weak.

4.4 Evaluation Outcomes

- The accuracy of our Automated Forced Aligner Web Application is evaluated to be approximately 77.11%.
- Evaluating mismatch between Manual and Automatic duration proves that there is a mismatch when there is a longer event duration.
- It is also proved that in the matched annotations the delta difference in duration is not more than 2 ms.

- From the Spearman's Rank Correlation analysis it's proved that there is a greater mismatch at the end times than with the start times of utterances.
- Spearman's Correlation analysis states that, there exists a correlation between the difference in start time with manual alignment.

4.5 Conclusion from Evaluations and its outcomes

In this chapter, we evaluated the accuracy of our alignment and it's correlation with the manual alignment using Spearman's Correlation Rank technique. It can be concluded from this chapter that our tool is approximately 77.11% accurate in aligning audio signals with their transcripts and most of the alignment is observed while aligning the end of the speech with its transcripts. The conclusion of our research work along with future improvements are specified in the following chapter.

Chapter 5

Conclusion

5.1 Introduction to Conclusion

In this chapter, we are concluding our research work on "Automated Forced Temporal Alignment of MULTISIMO Transcripts with Speech Signals" by providing an overall summary of our work, along with the future scope for improvements.

5.2 Summary of the research work

In this research work, we analyzed state-of-the-art Forced Alignment tools which automatically align the audio with the transcripts for our MULTISIMO corpus. We developed a user-friendly interface which improved the process of alignment of audio signals with transcripts to be very easy and quick. In this research work, we also improved the efficiency of the ProsodyLab tool by identifying the areas where the ProsodyLab tool was misaligning. The ProsodyLab Tool was failing to re-sample the audio files, which resulted in the misalignment. Hence that was fixed through the SOX tool, and with multiple HTK tool configuration changes, we improved the efficiency of the web interface. The results illustrate that our Forced Aligner web application's outputs are more accurate and efficient than that of the manual alignment results. The results also shows that our application has successfully aligned 475 words among 616 words accurately with the speech signals. Hence our tool is approximately 77.11% accurate in performing automated alignments. It is also evident from the manual analysis that the

misalignment is mainly due to the long pause, missing lexical transcripts for the audio, and noise in the speech signals. Using our application, to temporally align audio signals with the transcripts to find the start and end time of the utterances and phonemes, results accurate alignments with less than 2 ms of difference at most of the places. Forced Aligner web application has been used to perform forced alignment of MULTISIMO audio and transcript files. The complete alignment process took approximately 5 minutes for more than 1 hour of audio files. On performing, the same task manually would have taken many hours. Hence our automated web application has proven as an efficient, accurate, and cheap way of aligning audio files with their transcripts.

5.3 Final Remarks

Research is a never-ending process. There will be always scope for improvement. Our research work is mainly focused on the MULTISIMO corpus, but our web interface is flexible to align even the understudied and other language corpora. Future research can be carried out by building an acoustic model based on the Triphone approach using HTK or Sphinx or Kaldi. This approach considers phone context-dependency into consideration and improves the alignment efficiency of our application even more.

To conclude, our Automated Temporal Forced Alignment Web Application developed to align audio signals with the transcripts to find the utterance and phonemes start and end time can successfully replace the human annotations accurately.

Bibliography

[Baudouin de Courtenay, 1972] Baudouin de Courtenay, J. (1972). Selected writings of baudouin de courtenay. Bloomington: Indiana University Press.

[Burger et al., 2000] Burger, S., Weilhammer, K., Schiel, F., and Tillmann, H. G. (2000). Verbmobil data collection and annotation. In *Verbmobil: Foundations of speech-to-speech translation*, pages 537–549. Springer.

[DiCanio et al., 2013] DiCanio, C., Nam, H., Whalen, D. H., Timothy Bunnell, H., Amith, J. D., and García, R. C. (2013). Using automatic alignment to analyze endangered language data: Testing the viability of untrained alignment. *The Journal of the Acoustical Society of America*, 134(3):2235–2246.

[Goldman, 2011] Goldman, J.-P. (2011). *EasyAlign: an automatic phonetic alignment tool under Praat*. Interspeech'11, 12th Annual Conference of the International Speech Communication Association. Firenze (Italy). https://archive-ouverte.unige.ch/unige:18188.

[Gorman et al., 2011] Gorman, K., Howell, J., and Wagner, M. (2011). Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.

[Gorman et al., 2020] Gorman, K., Howell, J., and Wagner, M. (2011 (accessed September 3, 2020)). *Prosodylab English Dictionary File*. https://github.com/prosodylab/prosodylab.dictionaries/.

[Gowda, 2020] Gowda, K. B. (2020 (accessed September 3, 2020)). *Automated Forced Temporal Alignment of MULTISIMO Transcripts with Speech Signals*. https://github.com/kavyabgowda/ForcedAlignerWebApplication.

[Hatala, 2019] Hatala, Z. (2019). Practical speech recognition with htk. *arXiv preprint arXiv:1908.02119*.

[Hazen, 2006] Hazen, T. J. (2006). Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Ninth International Conference on Spoken Language Processing*.

[Johnson et al., 2018] Johnson, L. M., Di Paolo, M., and Bell, A. (2018). Forced alignment for understudied language varieties: Testing prosodylab-aligner with tongan data. University of Hawaii Press.

[Kazanina et al., 2018] Kazanina, N., Bowers, J. S., and Idsardi, W. (2018). Phonemes: Lexical access and beyond. *Psychonomic bulletin & review*, 25(2):560–585. Springer.

[Kempton, 2012] Kempton, T. (2012). *Machine-assisted phonemic analysis*. PhD thesis, University of Sheffield.

[Koutsombogera and Vogel, 2018] Koutsombogera, M. and Vogel, C. (2018). Modeling collaborative multimodal behavior in group dialogues: The multisimo corpus. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

[Koutsombogera and Vogel, 2020a] Koutsombogera, M. and Vogel, C. (2018 (accessed September 3, 2020)a). *MULTISIMO AUDIO Corpus*. https://multisimo.scss.tcd.ie/AUDIO/.

[Koutsombogera and Vogel, 2020b] Koutsombogera, M. and Vogel, C. (2018 (accessed September 3, 2020)b). *MULTISIMO Transcripts*. https://multisimo.scss.tcd.ie/ANNOTATIONS/.

[Lenzo, 2020] Lenzo, K. (accessed September 3, 2020). *CMU Pronunciation Dictionary*. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[Ma, 2012] Ma, X. (2012). Ldc forced aligner. In *LREC*, pages 3405–3408.

[Milne, 2015] Milne, P. M. (2015). Improving the accuracy of forced alignment through model selection and dictionary restriction.

[Mohamed et al., 2009] Mohamed, A.-r., Dahl, G., and Hinton, G. (2009). Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, volume 1, page 39. Vancouver, Canada.

[Schillingmann et al., 2018] Schillingmann, L., Ernst, J., Keite, V., Wrede, B., Meyer, A. S., and Belke, E. (2018). Aligntool: The automatic temporal alignment of spoken utterances in german, dutch, and british english for psycholinguistic purposes. *Behavior research methods*, 50(2):466–489. Springer.

[Strunk et al., 2014] Strunk, J., Schiel, F., Seifart, F., et al. (2014). Untrained forced alignment of transcriptions and audio for language documentation corpora using webmaus. In *LREC*, pages 3940–3947.

[Styler, 2013] Styler, W. (2013). Using praat for linguistic research. *University of Colorado at Boulder Phonetics Lab*.

[Thomas et al., 1998] Thomas, I., Zukerman, I., and Raskutti, B. (1998). Extracting phoneme pronunciation information from corpora. In *New Methods in Language Processing and Computational Natural Language Learning*.

[Young et al., 2002] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al. (2002). The htk book. *Cambridge university engineering department*, 3(175):12.

Appendix A

Using our Forced Aligner Web Application for corpus other than MULTISIMO corpus

Our web application can be used to align audio and transcripts of corpus other than the MULTISIMO corpus as well. Since we have not hosted our web application yet, the code available at [Gowda, 2020] can be downloaded to a local instance and after installing the pre dependencies, run app.py in the root directory to launch our web application. The following actions need to be performed based on the corpus for which the alignment needs to be carried out.

A.0.1 Corpus consisting of English Language

For corpus including the English Language audios and transcripts, run our web application on a local instance. Upload the audio (.wav) file and click on the re-sample button. Then choose a clean transcript (.lab) file of the audio previously uploaded. In the same window choose the config.yaml and eng.dict file whose sample is already available in [Gowda, 2020]. Change the config.yaml and eng.dict file to match your corpus and click on the Align button. The TextGrid output file downloaded after the completion of the alignment process can be viewed in Praat or ELAN tool, for further analysis on the corpus at the word and phoneme level.

A.0.2 Corpus consisting of a language other than the English Language

Since our web application does not support the training of our own Acoustic HMM model of a different language other than English, it can be done through ProsodyLab [Gorman et al., 2011]. Execute the ProsodyLab tool by placing audio and transcripts of clean audio of more than 1 hour in a directory inside the root directory of the tool. ProsodyLab's training own model functionality generates the HMM model for the new language for which data has been provided. This model then needs to be uploaded to our web application along with config and dictionary files on which training is done to get quick, easy, and accurate results.

Appendix B

Abbrevations

API	Application Programming Interface
CLI	Command Line Interface
CMU	Carnegie Mellon University
CUED	Cambridge University Engineering Department
HMM	Hidden Markov Model
HTK	Hidden Markov Model ToolKit
MFA	Montreal Forced Aligner
MFCC	Mel Frequency Cepstral Coefficient
MUAS	Munich AUtomatic Segmentation System
MULTISIMO	MULTI-modal and MULTI-party Social Interactions Modelling
OOV	Out-of-the-Vocabulary
P2FA	Penn Phonetics Lab Forced Aligner
STT	Speech-to-Text
WSGI	Web Interface GatewayInterface