DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

ACCENT TUTOR: A SPEECH RECOGNITION SYSTEM

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Sameer Koirala

Sushant Gurung

August, 2016

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by SAMEER KOIRALA and SUSHANT GURUNG entitled **"ACCENT TUTOR: A SPEECH RECOGNITION SYSTEM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

……

Sarbin Sayami

Assistant Professor

IOST, Tribhuvan University

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by SAMEER KOIRALA AND SUSHANT GURUNG entitled **"ACCENT TUTOR: A SPEECH RECOGNITION SYSTEM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…… Sarbin Sayami [Supervisor] Assistant Professor IOST, Tribhuvan University	…… Hitesh Karki Chief Academic Officer DWIT College
…….. Jagdish Bhatta [External Examiner] IOST, Tribhuvan University	…….. Rituraj Lamsal [Internal Examiner] Lecturer DWIT College

ACKNOWLEDGEMENT

We would like to thank Mr. Sarbin Sayami, Assistant Professor, IOST, TU and Mr. Ritu Raj Lamsal, Lecturer DWIT for supervising this project and giving assistance when faced with problems. Without their guidance and persistent help, this project would not have been possible.

We would like to thank Dr. Basanta Joshi, Assistant Professor, IOE, TU for providing technical help and providing deeper knowledge about the technology needed for this project. Without his help, we would be struggling to understand the theory needed and might have reached an incorrect conclusion.

We would like to thank Ms. Rojina Shrestha for assisting us during this project. She has been great motivator and helper, encouraging us to keep moving ahead on the project. We would also like to thank Mr. Hitesh Karki, Chief Academic Officer and Mr. Saroj Dhakal, Department Head of R&D of DWIT College for guiding us on various aspect of this project.

We would also like to thank all of the volunteers who helped us in collecting pronunciation records.

At last but not the least our sincere thanks goes to our parents and member of our family, who has always supported us and to all of our friends who directly or indirectly helped us to complete this project report.

Sameer Koirala
TU Exam Roll no: 1810/069

Sushant Gurung
TU Exam Roll no: 1821/069

STUDENT'S DECLARATION

We hereby declare that we are the only authors of this work and that no sources other than the listed here have been used in this work.

... ...
Sameer Koirala
Date: August, 2016

... ...
Sushant Gurung
Date: August, 2016

ABSTRACT

Accent Tutor is Automatic Speech Recognition System for Nepali words based on template matching using Mel-frequency Cepstral Coefficients for feature extraction and Dynamic Time Warping for feature matching. Two testing approaches; speaker dependent and speaker independent were used by taking the recording of two Nepali words 'नमस्ते' and 'धन्यवाद' from 9 volunteers.

The accuracy of 81.25 % and 62.5 % was obtained for speaker dependent and speaker independent speech recognition system respectively.

Keywords**:** Automatic Speech Recognition, Mel-frequency Cepstral Coefficients, Dynamic Time Warping, Speaker dependent, Speaker independent

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN:	Artificial Neural Network
ASR:	Automatic Speech Recognition
CAPT:	Computer Assisted Pronunciation Training
CSS:	Cascading Style Sheet
DCT:	Discrete Cosine Transform
DTW:	Dynamic Time Warping
FFT:	Fast Fourier Transform
GMM:	Gaussian Mixture Model
GSP:	Groovy Server Pages
HMM:	Hidden Markov Model
HTML:	Hyper Text Markup Language
LPC:	Linear Predictive Coding
MFCC:	Mel-frequency Cepstral Coefficients
MLP:	Multi-layer Perceptron
PCA:	Principal Component Analysis
SVM:	Support Vector Machine
WER:	Word Error Rate
URL:	Uniform Resource Locator

CHAPTER 1: INTRODUCTION

In present context if one wants to learn the proper pronunciation of a Nepali word then the person has to listen to other Nepali people pronouncing the word, practice pronouncing and get feedback on their pronunciation. While following above steps to learn the proper pronunciation of Nepali word, a learner must first find a proper tutor who has high knowledge about the pronunciation of Nepali words and has to maintain constant communication with the tutor.

If all of this process could be done with the help of automated program then it would be easier than above manual process and lot of time could be saved. Hence, the efficiency of the learning process can be increased. If ASR is designed carefully with the speech-enabled learning activities that give reliable feedback, then acceptable levels of recognition performance can be reached for pronunciation learning using ASR. (Neri, Cucchiarini, & Strik, 2003)

Accent Tutor is a web-based ASR application which allows people to learn proper pronunciation for Nepali words. It contains Nepali words list with their proper pronunciation through which the user will be able to learn the proper pronunciation of Nepali words.

1.1 Background

Speech carries information related to linguistic (e.g., message and language), speaker (e.g., emotional, regional, and physiological characteristics of the vocal apparatus), and environmental (e.g., where the speech was produced and transmitted). Among all speech tasks, ASR has been the focus of many researchers for several decades. Speech recognition is a process of converting speech signal to a sequence of words by means of an algorithm implemented as a computer program. Since the 1960s various research has been conducted to make computers able to record, interpret and understand human speech. Researchers learned that it is not an easy task to extract information of interest. Despite the

challenges researchers have made significant advances in technology that allows development of a speech-enabled application. (Adami, 2010; Gaikwad, Gawali, & Yannawar, 2010)

Various issues in designing ASR are:
Environment: - Type of noise, signal and noise ratio and working conditions
Transducer: - Microphone of recording devices
Channel: - Band amplitude, distortion, and echo
Speakers: - Speaker dependence and independence, sex, age and physical and psychical state
Speech styles: - Voice tone (quiet, normal, shouted), production (isolated words, continuous speech, spontaneous speech) and speed (slow, normal, fast)
Vocabulary: - Characteristics of available training data, specific or generic vocabulary (Anusuya & Katti, 2009)

Speech recognition system generally works in four stages: analysis, feature extraction, modeling, and testing. Different techniques like PCA, LPC, Cepstral analysis, Mel-frequency scale analysis, MFCC, Dynamic feature extractions, Cepstral mean subtraction, Integrated phoneme subspace method have been developed for feature extraction. For modeling techniques, the acoustic-phonetic approach, pattern recognition approach, template based approach, DTW, knowledge-based approaches, statistical-based approaches, learning based approaches, the artificial intelligence approach, and stochastic approach are used. Speech recognition engines use whole word matching and sub word matching techniques for matching a detected word to a known word. (Gaikwad, Gawali, & Yannawar, 2010)

1.2 Problem Statement

The listener can misinterpret the meaning of a word if a speaker doesn't use proper pronunciation during a conversation. Having proper pronunciation helps in conveying information properly during a day to day conversation. If ASR can be developed properly then it will assist many users in learning proper pronunciation. Many types of research on using ASR for learning pronunciation for different language like Bengali, Tamil, English etc. has been done but not much research have been done for the Nepali language. Many of the people (foreign and domestic) want to learn the proper pronunciation of Nepali words

but they are limited by lack of contact with Nepali words pronunciation expert. Accent Tutor will store proper pronunciation of Nepali words and assist people interested in learning the pronunciation of Nepali words.

1.3 Objectives

1.3.1 General objective:

To implement MFCC and DTW algorithms for developing ASR based on the isolated word template based approach that matches Nepali words pronunciation of user with stored proper pronunciation of Nepali words so that anyone can practice Nepali word pronunciation.

1.3.2 Specific objective:

a) To prepare Nepali words list with their respective proper pronunciation
b) To match user recorded pronunciation with stored pronunciation and determine accuracy

1.4 Scope

Accent Tutor can be used by people interested in learning to pronounce Nepali words properly. It can be used in Language Institute where the Nepali language is taught to foreigners. Also, this can be used in schools of Nepal where students can learn the proper pronunciation of Nepali words.

1.5 Limitation

a) Sample of pronunciation cannot represent all of the Nepali accent tone
b) Some of the information of speech signal can be lost during pre-processing
c) Speaker normalization is not implemented
d) Accent tutor needs to be accessed from noise less environment.

1.6 Outline of Document

The report is organized as follows

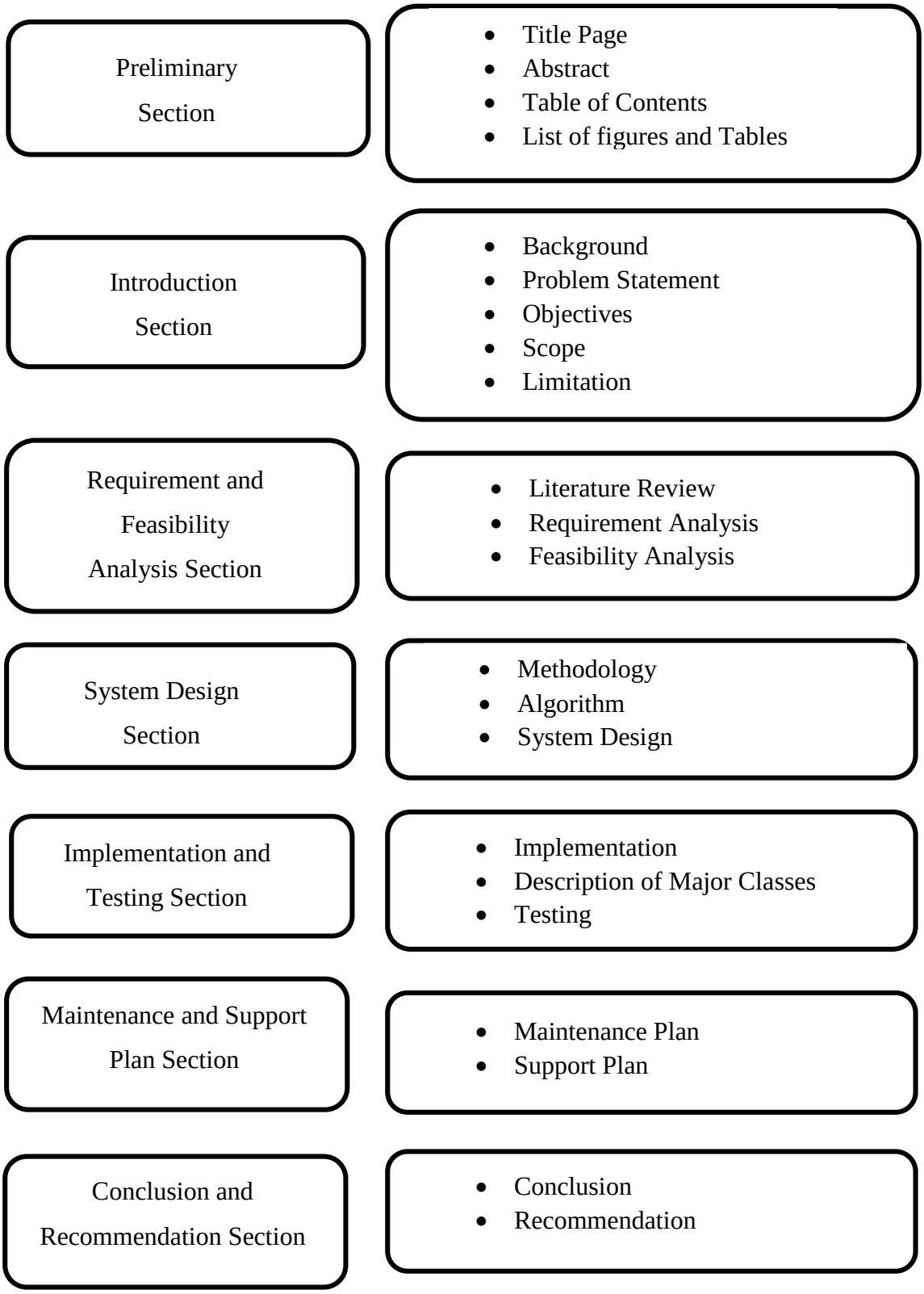

Figure 1- Outline of document

CHAPTER 2: REQUIREMENT AND FEASIBILITY ANALYSIS

2.1 Literature Review

2.1.1 ASR for learning pronunciation

ASR is criticized for the ability to recognize accented or mispronounced speech, and the ability to provide meaningful evaluation of pronunciation quality. Criteria such as recognition performance must be at an acceptable level and that the identification of speech errors must resemble that of native listeners in many cases are not met. But the researchers have found that this pessimism is not entirely justified.

Some of the problems occur because of little familiarity with ASR technology and with design matters within ASR-based CAPT. If ASR designed carefully with the speech-enabled learning activities that give reliable feedback, then acceptable levels of recognition performance can be reached for pronunciation learning using ASR. (Neri, Cucchiarini, & Strik, 2003)

From research done it was found that ASR method offers a great opportunity in teaching and learning pronunciation than regular instruction for English pronunciation. Learning through ASR is more useful in learning pronunciation than regular instruction. The educational environments in which ASR is used in the classroom are highly motivating environments for learning English pronunciation. Using ASR one learner can advance according to their own learning speed taking the character of the learner into consideration. (Elimat & AbuSeileek, 2014)

2.1.2 Different approaches and techniques for ASR

ASR can be implemented using different approaches. The acoustic-phonetic approach assumes that there exists finite, distinctive phonetic units (phonemes) in spoken language and uses these phonetic features for speech recognition. Pattern recognition approach uses

a well formulated mathematical framework and establishes consistent speech pattern representations, for reliable pattern comparison, from a set of labeled training samples via a formal training algorithm. DTW is an algorithm for measuring the similarity between two sequences which may vary in time or speed.

Vector Quantization uses compact codebooks for reference models and codebook searcher in place of a more costly evaluation function. Each vocabulary word is assigned a Vector Quantization codebook, based on training sequence of several repetitions of the word. Then the speech is evaluated by all codebooks and ASR chooses the word whose codebooks results in lowest distance measure.

Artificial intelligence approach uses both acoustic-phonetic approach and pattern recognition approach. Information regarding linguistic, phonetic and spectrogram is stored as experts' knowledge for speech recognition. A connectionist approach is an approach that uses ANN that attempts to mechanize the recognition procedure according to the way a person applies intelligence in visualizing, analyzing, and characterizing speech based on a set of measured acoustic features. (Anusuya & Katti, 2009)

From the comparison experiment done by Vimala C. and Radha V. for six algorithms of which one was template matching (DTW), two were statistical pattern matching (HMM, GMM) and three were machine learning techniques (ANN, SVM, and Decision Trees). MFCC feature vectors were used in the implementation of the six algorithms to recognize Tamil Speech. The experiment resulted in template matching and statistical pattern matching approaches consuming much time than machine learning techniques.

For test data recognition accuracy of 95.83%, 97.92%, 86.67%, 92.50%, 90.83%, 63.33% for DTW, HMM, GMM, MLP, SVM and Decision tree algorithms respectively was achieved. From the experiment, it was concluded that HMM and DTW followed by MLP and SVM give better recognition rate for the developed system. Also, the conclusion of machine learning approaches reducing time complexity compared with HMM and GMM techniques was reached. (C. & V., 2015)

2.1.3 Use of MFCC and DTW for isolated word recognition

We can have highly accurate speech recognition system if the system is trained using the user's voice who is going to use the system. Firstly, the user needs to record several

speeches of a word into the system and the system computes a statistical average of the multiple samples of the same word and stores the averaged sample as a template. Then when the word is spoken by a user, a feature is extracted using MFCC and a pair wise comparison of the feature vectors is carried out using DTW techniques.

Then the total distance between the sequences which is the sum or the mean of the individual distance between feature vectors is calculated. From the experiment carried out it is seen that if the user's speech matches the template then the total distance between the sequences is zero. If the user's speech doesn't match the template then the total distance between the sequences is non-zero. This type of system is speaker dependent and recognition accuracy can be about ninety-five percent. This system was used to develop a speech recognition system for Tamil words. (Dharun & Karnan, 2012)

In research done by Md. Akkas Ali, Manwar Hossain and Mohammad Nuruzzaman Bhuiyan, the speech recognition techniques differ according to the pronunciation style for Bengali words recognition. Initially, it analyzed a set of speech and extracted features based on signal processing. They used methods MFCC, DTW, GMM and LPC for recognizing words in Bengali. The extracted signals was compared with the reference one and the recognition rate (%) and elapsed time (sec) was shown. Basically, they used four different models to calculate the perfection rate in Bengali words.

In model 1, they used MFCC for feature extraction and DTW for pattern matching. In model 2, they used LPC for feature extraction and DTW for pattern matching. In model 3, they used MFCC and GMM for feature extraction and Posterior Probability Function for pattern matching. In model 4, they used MFCC and Compress the result by LPC for feature extraction and DTW for pattern matching. From this speech recognition the accuracy rate for the English language was more than 95%. But in Bengali words they were successful to get highest of 84% for one hundred words using model 4. They also noted that speech detection and recognition systems were very dependent on the machines like different laptops, microphones and also on different environment. (Ali, Hassain, & Bhuiyan, 2013)

Another paper used template matching technique in which they used MFCC for feature extraction. But used DTW algorithm for comparison. DTW can be used as the pattern making for the word. It is an algorithm which similarities between signals that is

independent to each other in terms of time and speed. The combination of MFCC and DTW is shown in the paper through experiments.

Two experiments were performed, in one experiment the template or sample and the input audio was recorded through the same machine and another experiment when the machine is different. From the experiments it is seen 97.5% accuracy when the device is same but only 70.6% Accuracy when the devices are different. (Mishra, Shrawankar, & Thakare, 2013)

Another paper by Ms. Savitha and S Upadhya also used template matching approach for recognition where the speech is recorded and stored as a template. After then recognition by comparing the spoken utterance by a user with the pre-stored templates and select the best matching pattern. DTW algorithm is used for template matching between the templates. Single and Average template matching techniques, two different template matching were used that were developed to recognize the English digits. These algorithms were done for both speaker dependent and speaker independent and a comparison of accuracy was carried out.

For speaker-dependent, the templates were taken from the speaker. Again in single template matching techniques, it showed that the accuracy between the template and spoken utterance by the same speaker is 99.3 % accurate and for average template matching four utterances were taken from a single speaker and averaged to obtain a template and used the fifth utterance of the user as input and 100% accuracy was obtained.

For speaker-independent, the templates were taken from four different speakers. Again in single template matching techniques, the accuracy is 79.3 % accurate and for average template matching 93%. From the experiment, we can say that speaker dependent are much better than speaker independent here. And for accuracy average template matching can be used. (Ms Savitha & Upadhya, 2013)

2.2 Requirement Analysis

Table 1- Functional and non-functional requirements

Functional requirement	Non-functional requirement
Play proper pronunciation of Nepali words	Display List of words and after selecting the word pronounce it
Record user pronunciation of Nepali words	Each words will have button associated with it which will allow to record user pronunciation
Compare user pronunciation with stored standard pronunciation	Green notification will signify pronunciation match while red notification will signify pronunciation not matched

Table 1 describes the basic functionality of Accent Tutor which includes displaying a list of Nepali words and play the proper pronunciation of the words, record user pronunciation of the words as input and process it to compare the recorded pronunciation with stored template and provide feedback as output.

For ease of access when a user selects the word from the displayed list of Nepali words, the pronunciation of the word will be played. Then the user can click on the record button associated with the word to record their pronunciation of the word. Finally, Green notification signifies the pronunciation match and red notification signify not matched.

2.3 Feasibility Analysis

2.3.1 Technical feasibility

Accent Tutor is a web application that uses Grails Framework. It uses JavaScript, Groovy Server Pages (GSP) as front end and Groovy, Java as the back end. It requires a server, client, and internet connection to function properly.

It supports both Windows and Linux platform for its operation. All of the technology required by Accent Tutor are available and can be accessed freely, hence it was determined technically feasible.

2.3.2 Operational feasibility

Accent Tutor has a simple design and is easy to use. It uses two-tier architecture (i.e. Client and Server). It can be easily accessed from School having the internet connection and can be used to teach pronunciation of Nepali words to a student during Nepali class. Also, if the internet is available then it can be used by any user whenever they want to practice pronunciation of Nepali words. Hence, Accent Tutor was determined operationally feasible.

2.3.3 Schedule feasibility

Schedule of Accent Tutor is as following:

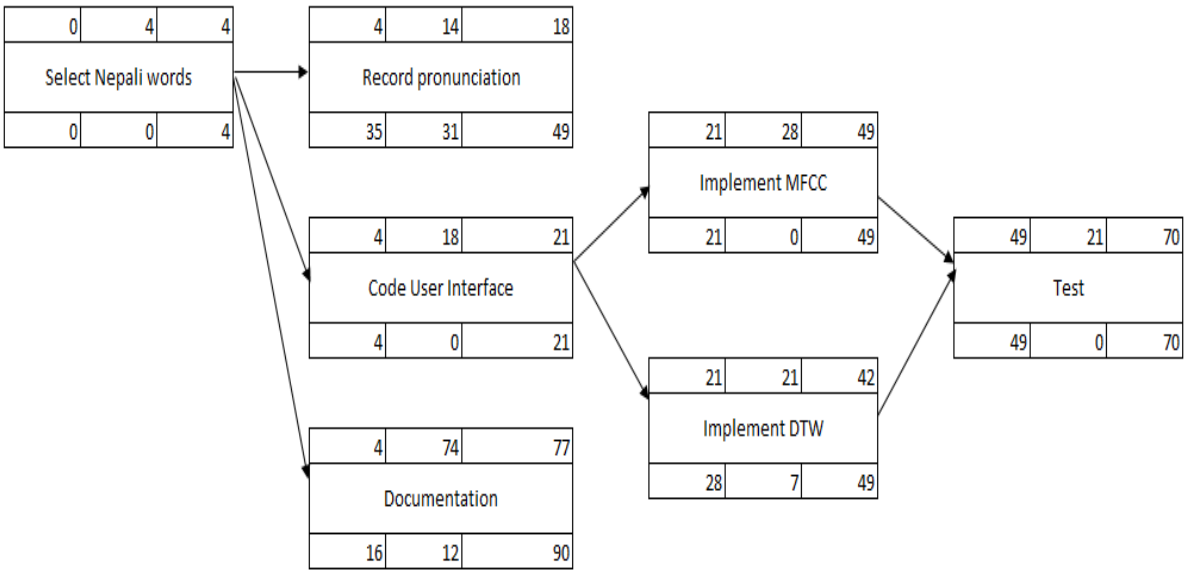

Figure 2- Activity network diagram of Accent Tutor

Figure 2 shows Activity Network Diagram of Accent Tutor. The early finish time for the project is 77 days and the late finish time for the project is 90 days. Three tasks “Record pronunciation”, “Implement DTW” and “Documentation” had the slack time of 31, 7 and 12 days respectively. The above we can see that Accent Tutor was completed in 13 weeks which is within 15 weeks of a semester. Hence, Accent Tutor project was determined to be feasible in terms of schedule.

CHAPTER 3: SYSTEM DESIGN

3.1 Methodology

Speech recognition system based on template matching has already been implemented for Bengali, Tamil and English words, using MFCC and DTW algorithm. The obtained result is satisfactory as the accuracy of DTW is high. Hence, Accent tutor also uses MFCC and DTW algorithm for speech recognition system based on template matching for Nepali words.

3.1.1 Data collection

Data used by Accent Tutor was collected by recording the pronunciation of volunteers. As the application needs the sound, the main problem was to get a good recorded sound. The sound must be noise free. So the problem was to determine the noise free environment for recording pronunciation and a good recording machine.

For recording, a recording studio was used where there was no external noise. The room was used to cut off the unwanted noises which may affect the application. Rather than using simple recorders for the data, the handheld digital audio recorder was used. The recorder records the sound and saves the files in .wav file which is supported by the application.

Two words were chosen, 'नमस्ते' and 'धन्यवाद'. There were 9 people that volunteered for recording and the sounds of the two words were recorded. Among the volunteers, there were 4 girls and 5 boys.

3.1.2 Data selection

Two different approaches are used for selecting the data: Speaker independent and Speaker dependent. In a speaker-independent approach, the sound templates of the different speaker are taken whereas in speaker dependent approach the sound templates of the same person are taken.

For speaker independent, 10 pronunciation of 'नमस्ते' and 'धन्यवाद' from 5 volunteers were used. For the speaker dependent 5 pronunciation of 'नमस्ते' and 'धन्यवाद' from 4 volunteers were used.

Table 2- Feature vector matrix of 'नमस्ते'

11.259876	-1.7335479	…	-0.13174616
11.639623	-0.4674986	…	-0.05172623
…			
…			
…			

Table 2 shows the sample of feature vector matrix of a Nepali word 'नमस्ते' computed from using MFCC algorithm.

Table 3- Feature Vector matrix of 'धन्यवाद'

11.847548	-0.1888611	...	-0.7067031
12.487079	-2.387002	...	-1.6215327
....			
....			
....			

Table 3 shows the sample of feature vector matrix of a Nepali word 'धन्यवाद' computed from using MFCC algorithm.

3.1.3 Data preprocessing

The recorded sounds were edited using the audacity application. It was used to cut off the unwanted part and also reduce some background noise. After that, the sounds were processed by applying a high filter to remove some of the background noise and also to produce some good sound files.

3.1.4 Validation of model

Word Error Rate and accuracy was computed for validation of model.

Word Error Rate (WER) = $\frac{Error\ Word\ Count}{Total\ Word\ Count}$

Accuracy = $(1 - WER) * 100\%$

3.2 Algorithm

Two algorithms were used:

3.2.1 Mel Frequency Cepstral Coefficients (MFCC)

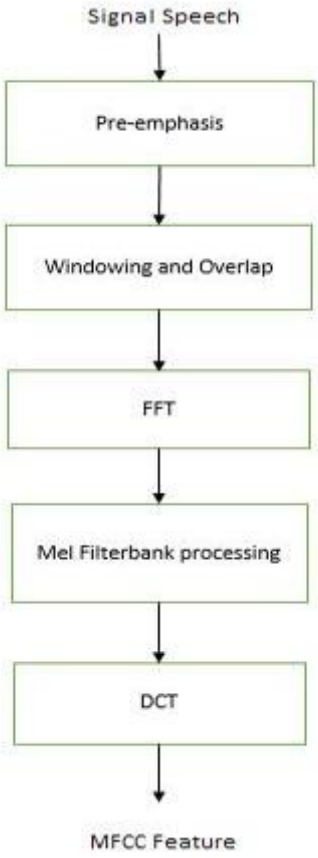

Figure 3- MFCC Extraction Algorithm

Figure 3 shows the steps of MFCC Extraction Algorithm which is described in detail below

Step 1: Pre-emphasis phase high pass filter is applied to the obtained signal that compensate suppression of high frequency part during the sound production.

If Y[n] = Output signal

X (n) = input signal, then result of pre-emphasis is shown below:

Equation 1- Pre-emphasis

$$Y[n] = (X[n] - 0.97 * X[n - 1])$$

Step 2: Input speech signal is segmented into frames of 20~30 ms with optional overlap of 1/3~1/2 of the frame size. Each frame is multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame.

If N = number of samples in each frame

Y[n] = Output signal

X (n) = input signal

W (n) = Hamming window, then the result of windowing signal is shown below:

Equation 2- Hamming windowing

$$y[n] = x[n] * h[n]$$

$$h[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right); 0 \leq n \leq N-1$$

Step 3: FFT is performed to obtain the magnitude frequency response (power spectrum) of each frame.

If x(w), H(w) and y(w) are the Fourier Transform of x(t), H(t) and y(t) respectively then the result of FFT is shown below:

Equation 3- FFT

$$y(w) = FFT[h(t) * x(t)] = H(w) * X(w)$$

Step 4: Mel filter bank processing helps to convert power spectrum to Mel spectrum.

Equation 4- Mel Filterbank

$$F(mel) = [maxFrequency * (\log 10[1 + f] * minFrequency)$$

Step 5: DCT is used to get the cepstrum coefficients.

Equation 5- DCT

$$x_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1.$$

3.2.2 Dynamic Time Warping (DTW)

DTW is a time series alignment algorithm. It determines the best possible alignment of two sequences of feature vectors by warping the time axis iteratively until an optimal match between the two sequences is found.

Suppose we have two time series A and B, of length n and m respectively, where

A = a1, a2,…, ai,…,an

B = b1, b2,…, bi,…,bn

We determine the distance between two series using the Euclidean distance computation:

Equation 6- Euclidean Distance

$$d(ai, bj) = (ai - bj)^2$$

Each matrix element of features vector (i, j) corresponds to the alignment between the points ai and bj. Then, accumulated distance is measured by:

$$D(i,j) = \min[D(i-1, j-1), D(i-1, j), D(i, j-1)] + d(i,j)$$

3.2.3 Thresholding

The threshold concept implemented is as follows:

Case 1: Distance almost 0; feature vectors match

Case 2: Distance > 0

Calculation of Average of distances with template

If Average > ¾ of calculated distances; feature vectors match.

Case 3: Feature vectors does not match otherwise.

3.3 System Design

3.3.1 Class diagram

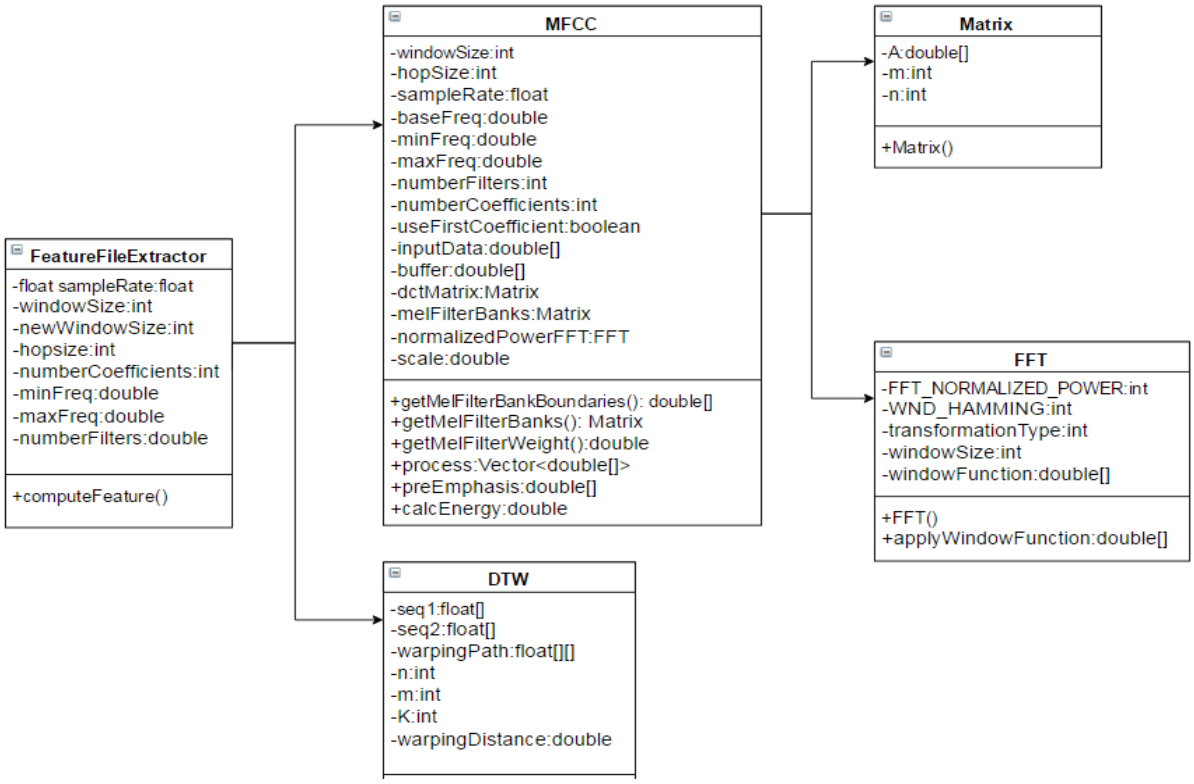

Figure 4- Class Diagram of Accent Tutor

Figure 4 explain the classes used in the Accent Tutor. There are five classes used in total, FeatureFileExtractor, DTW, MFCC, Matrix and FFT.

FeatureFileExtractor is the main class in the application. In this class, all the initializing process occurs, input sound is taken, the required classes are called for processing and the feedback is generated. This class simply calls another two classes: MFCC and DTW. MFCC class is used to extract feature it further uses to class one Matrix and FFT class.

3.3.2 State diagram

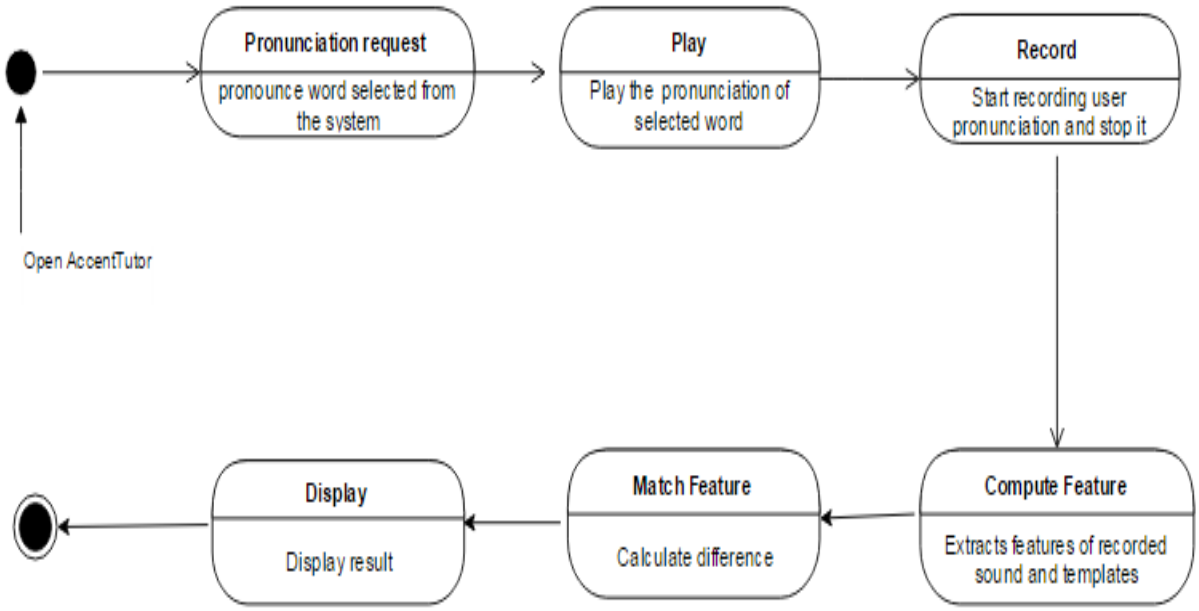

Figure 5- State Diagram of Accent Tutor

Figure 5 explains the different state of the system is shown. First, the user opens Accent Tutor then the system is in the state to receive pronunciation request. When a user selects a word to be pronounced, the system transits to the state where pronunciation of the selected word is pronounced by the system. When the user clicks record button then system transits to recording state where the user can record their sound and stop the recording.

After the recording state, the system transits to the state where important features of recorded sound are extracted. After feature extraction, the system goes to the state where the features of recorded sound and the templates are matched. Finally, the system transits to the state where feedback is displayed to the user.

3.3.3 Sequence diagram

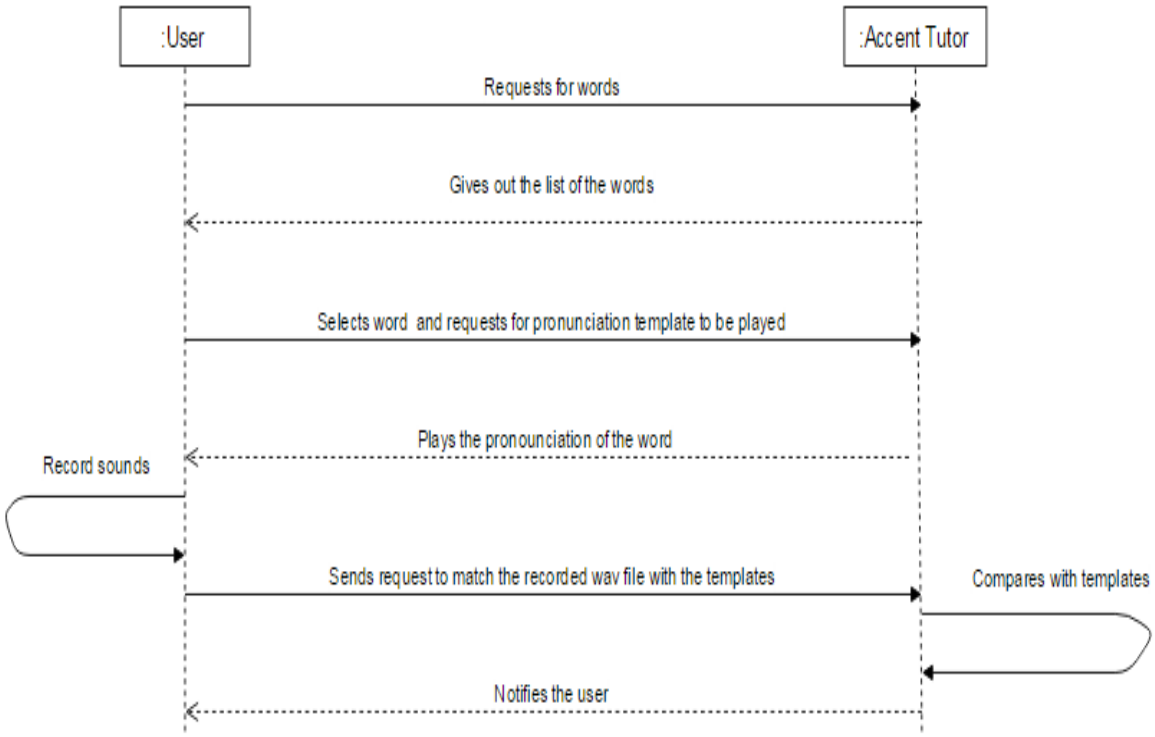

Figure 6- Sequence Diagram of Accent Tutor

Figure 6 explains the sequence of the Accent Tutor. Initially, a user opens a browser and requests for the list of words and the Accent Tutor in response gives out the list of all the words. Then the user selects a word of which pronunciation request is sent to the Accent Tutor. Then the Accent Tutor plays the requested word pronunciation after that user records a pronunciation of the word and sends it to compare with the templates in the Accent Tutor. After the comparison is done in the Accent Tutor it sends the feedback to the user.

CHAPTER 4: IMPLEMENATION AND TESTING

4.1 Implementation

User can access the application through a browser and see the interface. Admin will login to the system by adding "/login" to the URL and configure template files and a sound file that consists of a pronunciation of the Nepali word. Template files need to have same name format with a number attached to it which starts from 1. The user can listen to the pronunciation of selected word by clicking "Play" button. User can record new sound and export it. After that user can compare the newly exported sound with the templates by clicking "Compare" button and selecting the downloaded file. The user can also use the sound that is already exported. Then the application gives the feedback to the user. All of this is shown in Appendix.

When a user clicks Compare button, the user is asked to choose the sound file of recorded pronunciation. Then the application passes the path of recorded pronunciation and the templates to the object of the FeatureFileExtractor class. It defines MFCC class object and DTW class object. MFCC class method returns the featured matrix of the templates used in the application and the recorded user pronunciation. Then the obtained feature vectors are passed to the method of DTW class where the comparison is done between recorded pronunciation and the templates. After comparison, the feedback is provided to the user informing about pronunciation matched or not matched.

4.1.1 Tools used

CASE tools:

a) Draw.io

Client side:

a) HTML
b) Twitter Bootstrap CSS
c) GSP

d) JavaScript

e) JQuery

Server side:

a) Groovy

b) Java

This section describes the technologies used in Accent tutor. Accent tutor is a web application that uses GRAILS framework. HTML, Twitter Bootstrap CSS, JavaScript, and JQuery are used to develop front-end and Java, Groovy, and MySQL are used to develop back-end. GSP is used for presentation technology. JavaScript and JQuery are implemented for the recording and exporting the sound of the user and also to show the result of the application in a dynamic way.

All the algorithms for the application are written in java classes and the groovy classes. Algorithms used in Accent Tutor is MFCC to extract features of the sound and DTW algorithm to find the similarities between two sounds. For implementation of MFCC library named CYBORG – Speech Alignment Engine is customized and used. For implementation of DTW, we coded the algorithm in Groovy and for displaying feedback to the user noty.js is used.

4.2 Description of Major Classes

The major classes in the application are:

4.2.1 FileExtractFeature

This is the main class which is run first after match request is send by the user.
Input: It takes the inputs that is the recorded sound and initialize parameters.
Process: It then calls all the other classes which are required i.e., MFCC and DTW
Output: It provide feedback to the user according to the result from DTW class.

4.2.2 MFCC

This class implements the MFCC algorithm to extract the features from sounds.
Input: This takes the audios and initializing parameters like window size, sample rate etc.

Process: It then uses the method, process() to process the taken input audio and process and extract the feature. It calls out two classes: Matrix and FFT. Matrix class is called to initialize a matrix that is returned as the output and FFT class for Fast Fourier transform.
Output: It then gives the feature in the two dimensional matrix.

4.2.3 DTW

This class implements the DTW algorithm which take feature from MFCC class of both the user's recorded sound and the sound templates.
Input: Two matrix one of the user's sound and another of the template. The matrix is given by MFCC class
Process: It then uses the method, compute() to start the comparing. It calculates the distance between two sounds and compares according to it.
Output: It then gives out the warping distance between two sounds.

4.3 Testing

During testing 2 words 'नमस्ते' and 'धन्यवाद' were taken as test words. For the testing purpose following two approaches were developed. Testing was conducted using two test cases

Test Case 1**:** Matching correct word pronunciation with the template
Test Case 2**:** Matching incorrect word pronunciation with the template

4.3.1 Testing speaker dependent approach

In this test approach, recordings of 4 users for 2 test words were taken. In total 5 recording of each user was recorded. Out of those 5 recordings, 4 recordings were used as the template and a recording was used for testing. Total of 8 recordings were used for testing purpose, which was divided equally into 2 test cases.

Table 4- Test result of 4 user for speaker dependent approach

	Correct Count	Incorrect Count	WER	Accuracy
'नमस्ते'	7	1	0.125	87.5 %
'धन्यवाद'	6	2	0.25	75 %

As shown in Table 4, for the test word 'नमस्ते' WER and accuracy were 0.125 and 87.5 % respectively and for the test word 'धन्यवाद' WER and accuracy were 0.25 and 75 %

respectively. Hence, accuracy of 81.25 % on average was obtained for speaker dependent approach.

4.3.2 Testing speaker independent approach

In this test approach, recordings of 5 users for 2 test words were taken. 2 recording of each user was recorded. From the total of 10 recordings, 4 recordings of each test word were used as the template. Each test case was tested with remaining 2 recordings.

Table 5- Test result of 6 user for speaker independent approach

	Correct Count	Incorrect Count	WER	Accuracy
'नमस्ते'	3	1	0.25	75 %
'धन्यवाद'	2	2	0.5	50 %

As shown in Table 5, for the test word 'नमस्ते' WER and accuracy were 0.25 and 75 % respectively and for the test word 'धन्यवाद' WER and accuracy were 0.5 and 50 % respectively. Hence, accuracy of 62.5 % on average was obtained for speaker dependent approach.

CHAPTER 5: MAINTENANCE AND SUPPORT PLAN

5.1 Maintenance Plan

Accent Tutor will implement corrective maintenance for resolving different bugs and errors that may occur when this project is made live. Perfective maintenance will be implemented for increasing efficiency of the Accent Tutor by optimizing various implementation methods. Preventive maintenance will be implemented to make sure that Accent Tutor will not be harmed by hackers and security mechanism will be added.

5.2 Support Plan

Accent Tutor will be presented to the respective authority of the Government for investment so that the Nepali language can be promoted and many can learn the proper pronunciation of Nepali words. For self-sustenance of the Accent Tutor, the community will be established which will collect accurate pronunciation of the Nepali words and increase the availability of Nepali words that user request.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Accent Tutor was successfully implemented using Grails framework. From the testing done for speaker dependent approach the accuracy obtained is 81.25% for Nepali words pronunciation and for speaker independent approach the accuracy obtained is 62.5% for Nepali words pronunciation. Hence, use of speech recognition system that implements MFCC and DTW algorithm and template matching approach for learning pronunciation is not recommended and further improvements need to be made to be used for learning pronunciation.

6.2 Recommendation

This project did not consider voice tone, age and psychical state of the speaker for speech recognition. Hence, Vocal tract normalization and talker normalization needs to be implemented for increasing accuracy of speaker independent speech recognition system.

APPENDIX

Figure: Landing page of the Accent Tutor

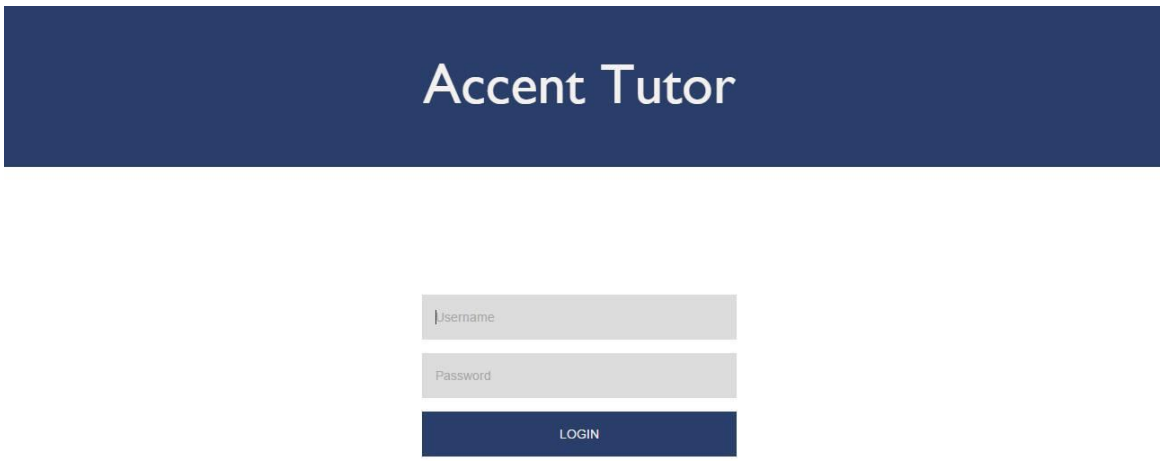

Figure: Login page

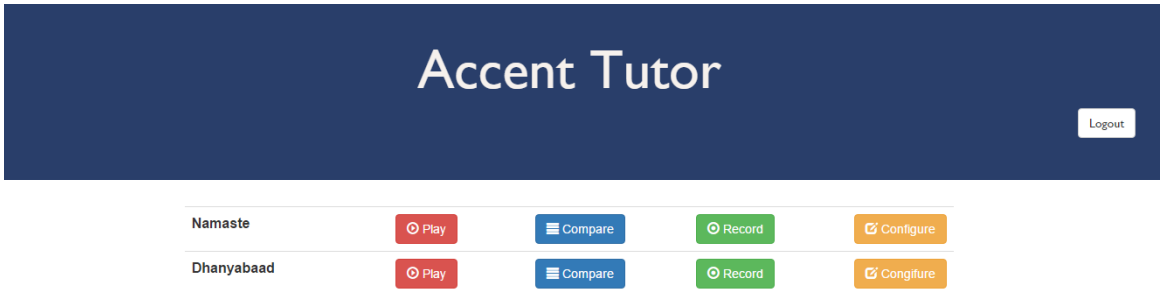

Figure: Admin dashboard

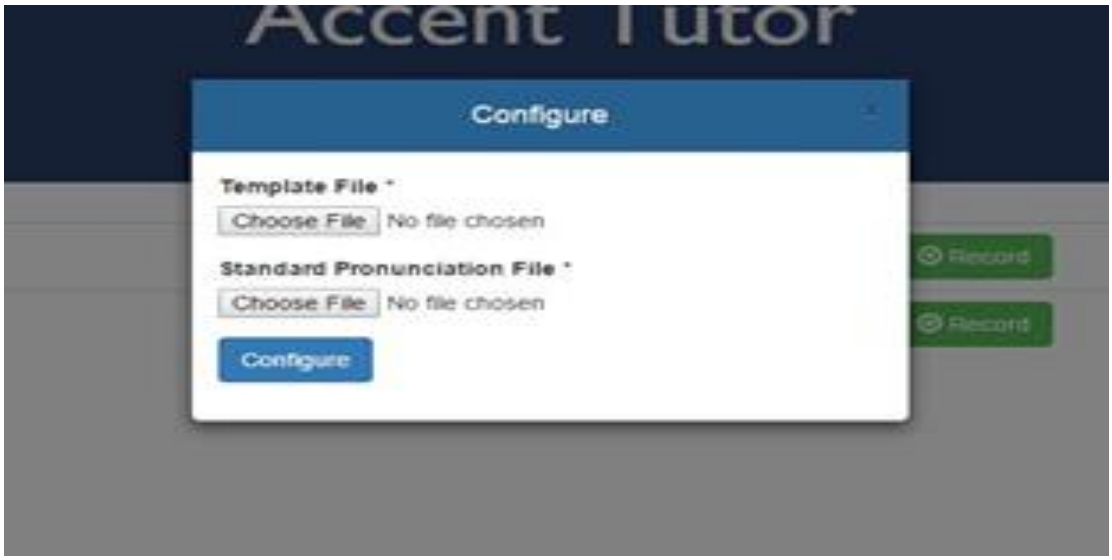

Figure: Popup after admin clicks “Configure” button

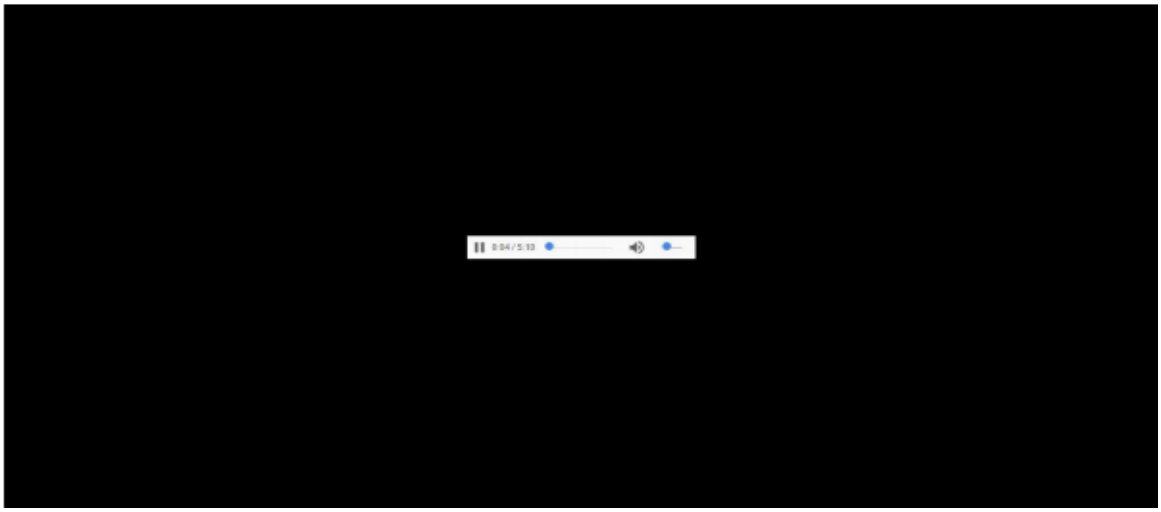

Figure: Playing the user selected word’s pronunciation

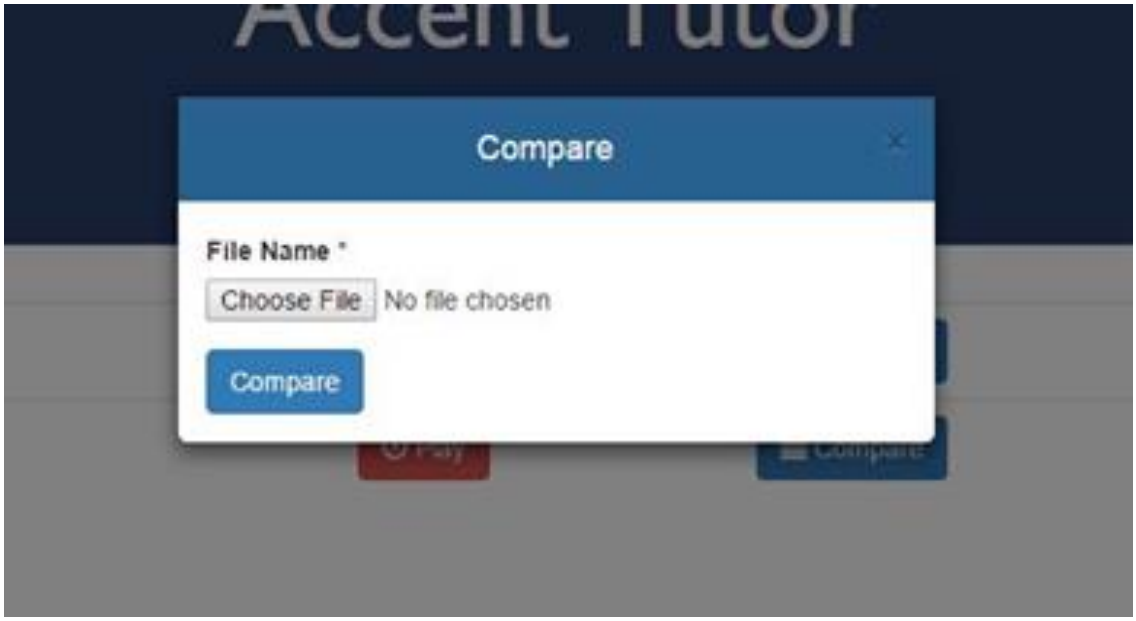

Figure: Popup after “Compare” button is clicked

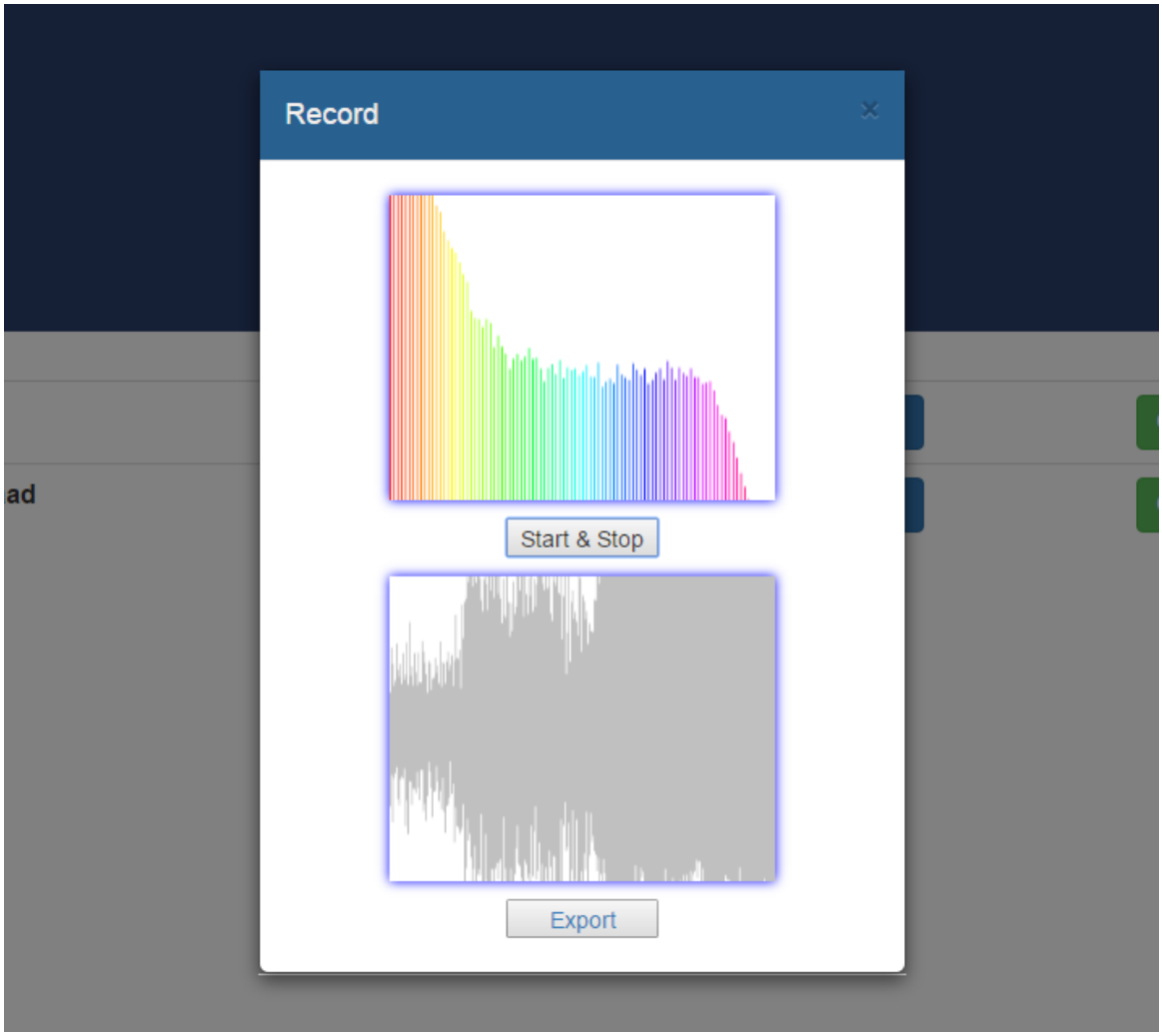

Figure: Popup after “Record” button is clicked

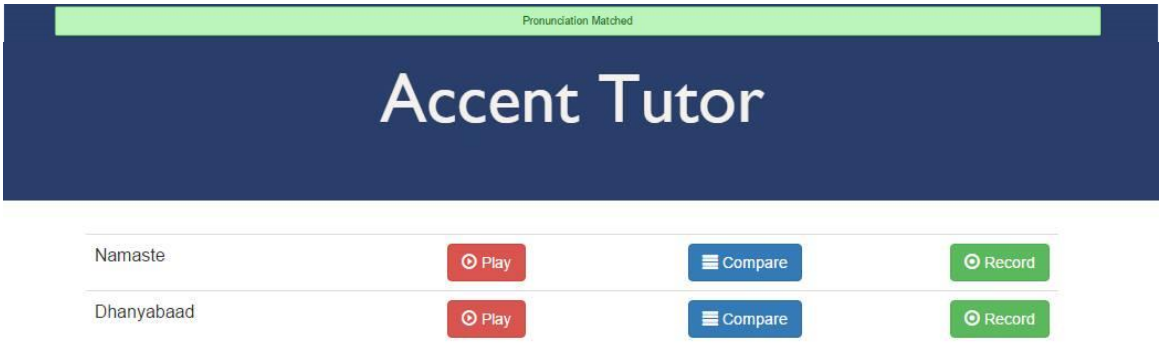

Figure: Feedback as “Pronunciation Matched”

Figure: Feedback as “Pronunciation not Matched”

REFERENCES

Adami, A. G. (2010). Automatic Speech Recognition: From the Beginning to the Portuguese Language. *The Int. Conf. on Computational Processing of Portuguese (PROPOR).* Rio Grande do Sul: Porto Alegre.

Ali, M., Hassain, M., & Bhuiyan, M. N. (2013). Automatic Speech Recognition Technique for Bangla Words. *International Journal of Advanced Science and Technology, 50*.

Anusuya, M. A., & Katti, S. K. (2009). Speech Recognition by Machine: A Review. *International Journal of Computer Science and Information Security, 6*, 181-205.

C., V., & V., R. (2015). Isolated Speech Recognition System For Tamil Language Using Statistical Pattern Matching And Machine Learning Techniques. *Journal of Engineering Science and Technology*, 617-632.

Dharun, V. S., & Karnan, M. (2012). Voice and Speech Recognition for Tamil Words and Numerals. *International Journal of Modern Engineering Research*, 3406-3414.

Elimat, A. K., & AbuSeileek, F. A. (2014). Automatic Speech Recognition technology as an effective means for teaching pronunciation. *The JALT CALL Journal*, 21-47.

Gaikwad, S. K., Gawali, B. W., & Yannawar, P. (2010). A Review on Speech Recognition Technique. *International Journal of Computer Applications, 10*, 16-24.

Mishra, N., Shrawankar, U., & Thakare, D. M. (2013). Automatic Speech Recognition Using Template Model for Man-Machine Interface. *Proceedings of the International Conference ICAET 2010.* Chennai, India: ICAET.

Ms Savitha, & Upadhya, S. (2013). Digit Recognizer Using Single and Average Template Matching Techniques. *International Journal of Emerging Technologies in Computational*, 357-362.

Neri, A., Cucchiarini, C., & Strik, W. (2003). Automatic Speech Recognition for second language learning: . *Proceedings of 15th International Conference of Phonetic Sciences*, (pp. 1157-2260). Barcelona.