Pronunciation Training on Isolated Kannada Words using "Kannada Kali" - A Cloud Based Smart Phone Application

Savitha Murthy*, Ankit Anand, Avinash Kumar
Ajay Cholin, Ankita Shetty, Aditya Bhat, Akshay Venkatesh
Lingaraj Kothiwale, Dinkar Sitaram†
PES University, Bangalore, India
*savithamurthy@pes.edu, †dinkars@pes.edu

Viraj Kumar
Indian Institute of Science, Bangalore, India
viraj.kumar.cs@gmail.com

***Abstract*—Automated feedback on pronunciation system on a smart phone is useful for a student trying to learn a new language at his or her own pace. The objective of our research is to implement a pronunciation training system with minimal language specific data. Our proposed system consists of an Android application as a front-end, and a pronunciation evaluation and mispronunciation detection framework as the back-end hosted on a cloud. We conduct our experiments on spoken isolated words in Kannada. Our pronunciation evaluation (for spoken word) implementation on the cloud involves training a classifier with features from Dynamic Time Warping (DTW) with Mel Frequency Cepstral Coefficients (MFCC) and Line Spectral Frequencies (LSF) and, without directly on LSF (without DTW). We study the performance of different machine learning algorithms for pronunciation rating. We propose a novel semi-supervised approach for detecting mispronounced segments of a word using Self Organizing Maps (SOM) that are also deployed on the cloud. Our implementation of SOM learns the features of an automatically segmented reference speech. The trained SOM is then used to determine the deviations in the learner's pronunciation. We evaluate our system on 1169 Kannada audio samples from students around 18 to 25 years of age. The Kannada words considered are taken from textbooks of first and second grade (considering learners as beginners who do not know Kannada) and include 2 to 5 syllable words. We report accuracy on binary classification and multi-class classification for different classifiers. The mispronounced segments detected using SOM correlate with the human ratings. Our approach of pronunciation evaluation and mispronunciation detection is based on minimal data and does not require a speech recognition system.**

***Index Terms*—pronunciation training, pronunciation evaluation, CAPT, Kannada, mispronunciation detection, SOM, cloud services**

I. Introduction

India is one among the many countries in the world that are experiencing rapid development. Many Indians moving temporarily or relocating permanently across the country either for pursuing education or due to job opportunities and requirements. With different local languages in India, communication issues are inevitably bound to happen. In today's digital age smart phones are very common and popular. Using this as an advantage we describe an Android application called "Kannada Kali". Users are evaluated based on their pronunciation of Kannada words using a cloud-based framework, that is connected to the front-end of the application. This application assists users to learn how to pronounce Kannada words by indicating the place of mispronunciations.

There are many advantages to using the cloud in the project. One of the main reasons being that, all of the processing happens on the instance not on the device hence, shaving off the load from it; significantly reducing the need to use processing power of the device. The cloud behaves as a central point for the data to be gathered for further processing; where analysis programs can be stored and updated periodically. This results in keeping the Android application simple thus increasing its reliability. With this kind of system, building and deploying more sophisticated analysis programs like addition of data processing pipelines can be easily done.

II. Related Work

Pronunciation evaluation and mispronunciation detection are the core of Computer Aided Pronunciation Training (CAPT) systems. Research work in CAPT make use of GMM (Gaussian Mixture Model)-HMM (Hidden Markov Model) [1]–[3] or Deep Neural Network (DNN) [4]–[7] models for obtaining phone posterior probabilities. Effective pronunciation training require native (L1) as well as non-native (L2) speech corpora. There are have been efforts to prevent the use of L2 data. Our study of literature gives an overview of the techniques used with both L1-L2 data and without L2 data for pronunciation training.

CAPT systems traditionally make use of an Automatic Speech Recognition (ASR) system that is trained on speech data from native (L1) and non-native (L2) speakers [6]–[12]. [9] have achieved pronunciation error detection at phoneme level employing the traditional GMM-HMM model with forced alignment while [4] demonstrate that a Deep Neural Network (DNN) based pronunciation evaluation system is more efficient as they make use of frame level posterior probabilities and do not require a decoding lattice. The work in [10], [11], [13] focuses on improving mispronunciation detection using DNN. To obtain more accurate information

978-1-5386-9441-1/18/$31.00 ©2018 IEEE
DOI 10.1109/CCEM.2018.00017

about mispronunciation, [13] used an LSTM embedded pronunciation vector.

Since gathering non-native speech is very difficult, there has been research work to achieve good results in pronunciation training without employing L2 data [1], [3], [14]–[17]. The research in [16] is maps the pronunciations between native and target languages. They train a multilingual DNN to model the articulatory features of non-native speech while [1] evaluate pronunciations of Indian English using native English and native Hindi speech using a GMM-HMM based acoustic model. [17] use the concept of anti-phones to detect substitutions in pronunciation and a filler model to detect insertions. [3] make use of a variant of Good of Pronunciation (GOP) scores called forced-aligned GOP (F-GOP) with logistic regression for evaluation. [18] make use of phonological features for CAPT and a multitask deep neural network model estimate HMM state probabilities. Speech attribute based decision tree was proposed to detect phonetic segmental mispronunciations and provide articulatory level feedback based on manner and place of articulation [19].

Again, CAPT systems in spite of not using L2 data, still require sufficient L1 data (of the order of at least 40-50 hours) for training their systems. Comparison based approach try to eliminate the need for training a speech recognition system. Lee and Glass [8], [20] have implemented a comparison based pronunciation evaluation where they compare the spoken sentences of a teacher and student using various computations on Dynamic Time Warping (DTW) of the two utterances. They train a Support Vector Machines (SVM) classifier on phoneme based and word based features to evaluate a students speech. They experiment with different features including Mel Frequency Cepstral Coefficient (MFCC) and Gaussian posteriorgram [20] as well as Deep Belief Network (DBN) posteriorgrams [8] as input to the comparison framework.

Self Organizing Map (SOM) has been used in speech recognition for clustering or recognizing patterns. SOM as defined by Kohonen [21] is an effective tool to reduce high dimensional non-linear data to simple two dimensional grid while at the same time preserving the topological relationships. It produces a similarity graph of the input data. [22] proposed to modify the original SOM algorithm by incorporating the time dependent features of the input audio signal for phoneme recognition. The work in [23] describes a hybrid method based neural network algorithm for speech recognition by using Multi Layer Perceptron (MLP) along with SOM for Malay speech recognition. [24] involved SOM and Vector Quantization algorithms for variable-length and warped feature sequences. [25] have employed SOM for speaker clustering for speaker adaptive training and show a reduction in word error rate.

The rest of the paper is organized as follows - section III gives an overview of our approach, section IV describes the details of our method, section V mentions the experimental details and section VI discusses the results and is followed by the conclusion section.

III. Our Approach - Overview

We propose "Kannada Kali", a cloud-based pronunciation training framework with an Android front-end. The objective of our work is to assist L2 students learn and improve their pronunciation skills in a new language without training a language specific speech recognition system. We conduct our experiments for Kannada, a language of Karnataka - a state in south India. A Kannada speech corpus not being readily available, we explore alternatives for pronunciation training other than conventional speech recognition system. The speakers considered for our experiments are Indians with mother tongue other than Kannada.The focus of our experiments is to train beginners on formal Kannada pronunciation (our research do not address Kannada accents from different regions). Our work on pronunciation evaluation and mispronunciation detection in [26] focuses on the pedagogical aspects of pronunciation learning. In this paper, we describe the technical aspects of our approach.

In our research, we adapt a comparison based approach based on the work of Lee and Glass [8], [20]. They use a native corpus of 630 speakers (TIMIT) and non-native corpus of 100 speakers (CU-CHLOE) in their work. We study employs template speech (native) of 21 Kannada words by a single female speaker for comparing the non-native speech of 19 speakers. [27] have developed a cross-platform mobile application for pronunciation training where the back-end is hosted on a cloud. They make use of a traditional ASR system to detect the place of mispronunciations and display them on the mobile application. We also implement a cloud based pronunciation training system with an Android front-end but without making use of ASR. Our experiments are based on a scenario where there is no adequate L1 data to train an ASR.

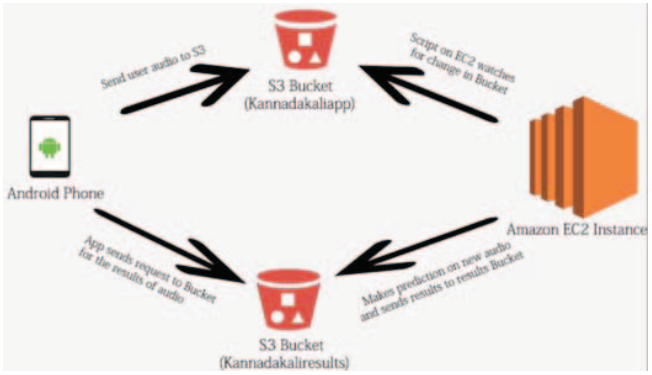


Fig. 1: Cloud Framework

We make use of Amazon Web Service (AWS) cloud services for storage and computation. AWS provides an extensive collection of services on-demand such as compute, storage, database and application services allowing us to deploy and maintain the application with ease. Automated scaling is another important factor for our choice of using AWS to host the application, allowing the application to meet variable demands through Elastic Load Balancing service, sharing the

increased load seamlessly. It can also shutdown idle instances when the demand decreases. Hence eliminating use of costly hardware and comprehensive administrative operations.

When a user records an audio sample, it is compressed to a suitable format and sent over to the cloud storage. To deliver reliable results there are certain requirements that need to be met. The communication must be secure and seamless so as to avoid any loss of data. The data is stored in the Cloud persistently. Another typical issue that arises when multiple users and the cloud are involved is synchronization, but with the setup we have i.e., using one instance per user we avoid having to handle synchronization. Scalability is another deliverable when the application is designed for multiple concurrent users. So, our design of one instance per user can be extended to match the number of users as we can assign more computing power to our instances, which can be set up as a trigger when an event (new user) occurs. AWS S3 provides security and query-in-place functionality which we can use in future to run powerful analytics on data at rest in S3 as our data set grows.

The Trained model is stored in a Cloud Instance and this allows us to reduce the on-device computation. The cloud instance used here is the AWS EC2 instance within which scripts run to watch the bucket and perform evaluation of user audio. The cloud instance is in a persistent state perpetually. The users audio file recorded on their devices is sent to the S3 Bucket. The instance is alerted of this addition using a script that is watching this bucket and the new file is then sent over to the instance for processing. The results of the pronunciation training framework are then stored back in the bucket. This rating is then written onto a file. The android application reads the result of the evaluation from this file and then displays it to the user as shown in Figure 1. The environment setup is defined as one instance (i.e., process) per each user; hence there is no issue of handling synchronization, user submits a response and the application waits for the query.

For our pronunciation training, we consider two aspects - assigning a rating to a spoken word which is **pronunciation evaluation** and detecting the segments of the word that have been mispronounced which is **mispronunciation detection**. Figure 2 depicts the overview of our evaluation and error detection framework. We first preprocess the speech audio to clip the silence regions and boost the amplitudes at higher frequencies (preemphasis). The preprocessed audio is then segmented in an unsupervised manner using Self Similarity Matrix (SSM) [20].

For pronunciation evaluation, we extract audio features (here we explore MFCC and LSF features) for each of the audio segments. We then perform DTW on these audio features and obtain computation values from the DTW cost matrix. These values are used for training a classifier and for classification of test speech. We explore different classifiers - Feed Forwards Neural Net, SVM, RandomForest, AdaBoost and GradientBoost.

For mispronunciation detection, we use MFCC features of the audio segments. Different SOM is trained for each segment. We then use the winner nodes obtained from SOM to determine the deviations of every segment. When the number of deviations for an audio segment crosses a threshold, the segment is identified as mispronounced. Our approach of determining the mispronounced segment does not employ L1 data (needed for forced alignment to determine pronunciation errors).

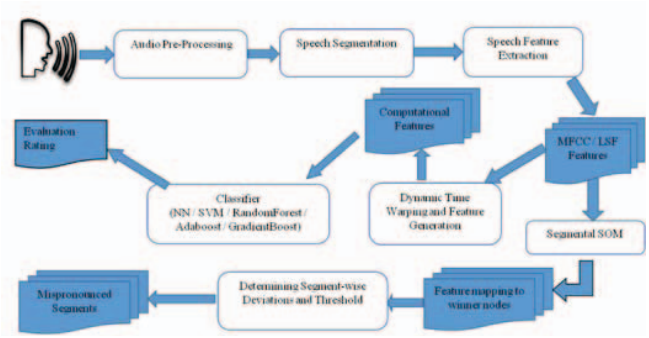


Fig. 2: Pronunciation Evaluation and Mispronunciation Detection

IV. Details of Method

A. Pronunciation Evaluation

We examine two methods for pronunciation evaluation of a spoken word. First one is a template based approach where, we obtain DTW of MFCC vectors of the template speech (spoken by a native Kannada speaker) and extract 14 computational features. We then train a classifier using these DTW features against ratings assigned by a Kannada teacher. We also implement unsupervised segmentation of the spoken words for finer granularity of comparison. We also investigate the DTW approach using LSF features instead of MFCCs. In the second approach, we extract 18 LSF values for the speech samples and directly train a classifier with these features against teacher ratings - no comparison is involved. We train different classifiers - Feed Forward Neural Network (NN), SVM, Random forest, AdaBoost and Gradient boost for pronunciation evaluation and report their accuracy.

Both our methods (specified in the previous section) require annotation of the spoken isolated Kannada word samples. The audio samples are rated by a Kannada teacher on a Likert scale of 1 to 5 (1 being the lowest and 5 being the highest rating). We use these ratings to train different classifiers for pronunciation evaluation. We describe the details of the two methods in this section results of which are given in section VI. section IV B mentions the details of unsupervised segmentation that is needed for the tasks in section IV A and section IV C. section IV C explains mispronunciation detection using SOM.

1) MFCC/LSF-DTW based Pronunciation Evaluation: We employ a template based approach where word samples from a native Kannada speaker, spoken with good clarity are recorded as templates. Audio samples for the same set of words are recorded from native and non-native speakers. The speech

samples are compared with the template using DTW. For fine granularity and better details, we segment the DTW cost matrix in an unsupervised manner (explained in section IV B) and compute several parameters from the segments of the cost matrix [20].

We extract MFCC features from the audio segments and perform DTW of template versus test audio to obtain a cost matrix. We map the boundaries of the template segments onto the DTW cost matrix and obtain the segments of the DTW cost matrix. We compute 14 different values [20] on each of these segments. We then train different models using these values for pronunciation evaluation. We also explore the use of LSF features for DTW instead of MFCCs.

2) Audio Feature Based Pronunciation Evaluation: We also investigate use of LSF of the audio files to train the classifiers instead of comparison. LSF are used to represent Linear Predictive coefficients (LPC). LSFs are more robust (less sensitive to quantization noise and exhibit filter stability) and useful for speech coding and analysis. They provide an accurate encoding of the speech at a low bit rate which could be helpful for processing inside various machine learning models. We train different models on LSFs obtained from the audio without the use of DTW. Here the LSF values extracted from the speech samples are directly used to train the classifier against human ratings.

B. Unsupervised Segmentation

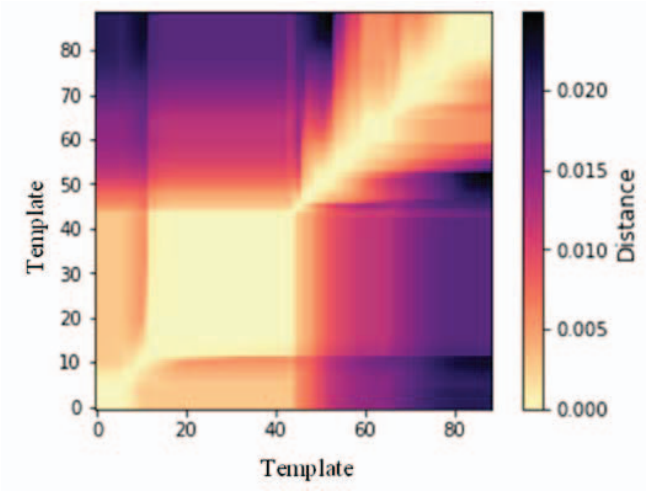


Fig. 3: SSM (template vs. template)

We compare the template and the test audio after breaking them into segments for finer granularity. We initially obtain segments of template audio by using Self Similarity Matrix (SSM). [20] use MFCC features for segmentation. We perform a DTW of the template audio spectrogram (instead of MFCC) against itself to obtain SSM. Figure 3 shows the SSM of the word orange, pronounced in Kannada as /ki/ /tta/ /l̤e/
. We segment the SSM cost matrix such that each segment spans across a low distance followed by a high distance region about the diagonal (This very closely corresponds to syllables in Kannada). We map the boundaries of the template audio segments (Figure 4) for segmenting the DTW cost matrix when comparing the test against template audio as described in the following sub-section.

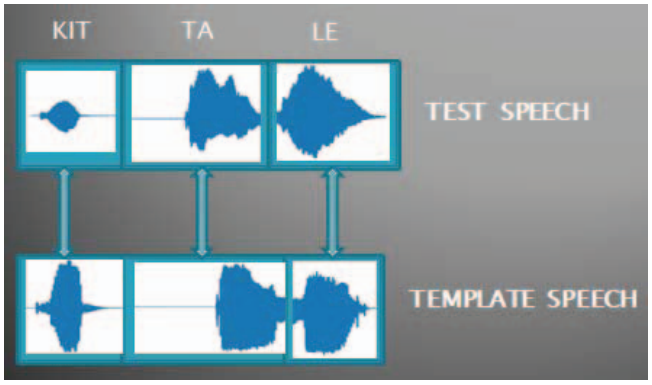


Fig. 4: Segment-wise comparison

C. Mispronunciation Detection

Since error detection is an integral part of CAPT systems and due to the lack of a readily available Kannada native speech corpus, we propose a novel approach of detecting mispronounced segments of a spoken word using SOM. We apply SOM to learn the pattern of the audio speech segments (since SOM are good in learning continuous data). We currently train different SOM for different speech segments of rating-5 audio. We determine the speech segment that is mispronounced by comparing the deviations of the template and test audio derived from the output of a SOM for the corresponding segment. Our system informs the learner regarding the exact segment he or she has mispronounced along with a pronunciation score. Our approach of mispronunciation detection requires minimal supervision.

We train a SOM on MFCC features obtained for each segment of the all audio samples with rating 5 of the training set as reference. Audio is segmented using unsupervised segmentation as described in section IV B. The test samples are also segmented using SSM. The predictions of the SOM on the test audio segments are then compared for deviations against the reference using a distance measure (we have currently used Manhattan distance) as given in equation 1. We determine the number of deviations for the winner nodes predicted by the SOM based on an empirically determined distance threshold. The audio segments that have the number of deviations beyond a certain count (determined by the maximum deviations for the reference) are identified as mispronounced.

$$Deviation(D) = |x - a| + |y - b| \quad (1)$$

where A(a,b) and B(x,y) be the two winner node locations.

V. Experiments

For our experiments, we implemented an Android front-end that connects to a pronunciation evaluation and mispronun-

ciation detection framework deployed on the cloud. Sample screens are shown in figure 5.

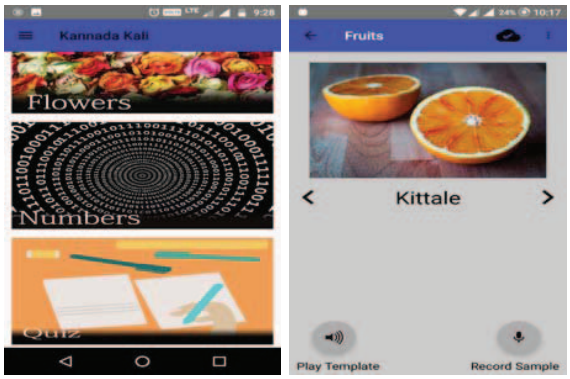


Fig. 5: Android App Screen Shot

A. Data

We collected single word speech samples based on a corpus of selected 3 to 4 words belonging to a variety of categories animals, birds, colors, flowers, fruits and numbers of the Kannada Language. The recordings are obtained for 21 unique Kannada words taken from textbooks of first and second grade Kannada textbooks. The pre-processing phase involved labeling of audio samples. Human labeling strategy was adopted. A native Kannada teacher was asked to rate the speech samples on a Likert scale of 1 to 5 (rating was based on how closely the quality of speech of the collected samples matched the audio template of native speaker). The speech samples were labeled accordingly. Post labeling, the audio samples were then processed to remove regions containing silence for meaningful analysis.

The data-set consists of 1169 spoken Kannada words. We obtained recordings from both native and non-native students with ages from 18 to 25 years. The audio samples were collected with the use of an Android Application developed as a part of this paper. The samples were obtained with JBL C100SI, Sony MDR-ex155 Noise cancellation earphones in environment with less background noise. The length of each of the audio samples is approximately 5 seconds and includes minute pronunciation variations and slight, in order to make the model robust to detect mispronunciations.

For pronunciation evaluation. we divided the data-set into training and test sets having 80-20 split of the full data-set - with training set consisting of 955 samples and testing set having 214 samples. The data-set consists of 746 speech samples of rating 5, 243 speech samples of rating 4, 90 samples of rating 3, 50 samples of rating 2 and 40 samples of rating 1. 746 samples are of rating 5 and the other 423 samples comprises of human ratings 1,2,3 and 4 creating an imbalance. We adopted certain data-set balancing strategies. We up-sampled the minority classes (speech rated 1,2,3,and 4) against majority class (speech rating 5) and all the data-set samples were brought to 50:50 ratio for both minority and majority class. A random sampling of speech samples from minority class was done and new samples were generated to match the majority class. A simple random sampling on the data-set provided enough minority class samples to match 50:50 ratio of minority and majority class samples, thereby increasing a total sample count from 1169 to 1774. The up-sampled data-set was then used further training the models.

TABLE I: Classification Accuracies - 5 Class

Models	5 Class		
	DTW (MFCC) Features	DTW (LSF) Features	LSF
NN	42.96%	56.23%	57.52%
SVM	53.16%	77.08%	63.74%
RandomForest	75.48%	63.96%	**78.93%**
AdaBoost	74.24%	73.68%	70.42%
GradientBoost	**80.40%**	79.12%	78.35%

TABLE II: Classification Accuracies - 2 Class

Models	2 Class		
	DTW (MFCC) Features	DTW (LSF) Features	LSF
NN	84.4%	81.26%	73.95%
SVM	75.82%	91.21%	78.85%
RandomForest	92.38%	85.11%	89.98%
AdaBoost	**95.5%**	94.92%	90.32%
GradientBoost	93.62%	93.02%	**93.62%**

B. Pronunciation Evaluation

We perform both binary (2 Class) and multi-class (5 Class) classification for pronunciation evaluation. For binary classification, the human rated files are converted to a scale of 2 by grouping 1-3 as class 0 and 4-5 as class 1 whereas for multi-class classification, 5 classes correspond to the 5 scale rating provided by human evaluators. These values are one-hot encoded to be used by the classifier. A data-set is created consisting of features obtained from audio segments and prediction target as one-hot encoded ratings.

We train several machine learning classifiers - NN, SVM, Random Forest, AdaBoost and GradientBoost with the up-sampled data-set. The classifiers were trained separately on 14 DTW features (per segment) - with MFCC and LSF features separately and on 18 LSF features per speech sample. The accuracies of all the classifiers are obtained by 10-fold cross validation on the up-sampled data-set. The details of the classifiers are specified as follows:

1) Feed-forward Neural Network classifier: We train a feed-forward neural network with two hidden layers. The input layer of the neural net consists of 102 neurons (corresponding to 84 MFCC and 18 LSF features) with ReLU as an activation function, Ridge regularization using 0.02 as a penalization factor for weight updation. The first hidden layer consists of

51 neurons and second layer consists of 26 neurons. We use ReLU as an activation function, Ridge regularization using 0.02 as penalization factor for weight updation for both the layers. The output layer consists of 2 neuron with Sigmoid as an activation function for binary classification or 5 neurons for 5-class classification. We use additional Dropout regularization factor of 0.2 between different layers to prevent model from over-fitting. The neural network was trained for 100 epochs with batch size of 10 for weight updation.

2) Support Vector Machine Classifier: The SVM model used in the current implementation uses RBF kernel for non-linear decision boundary with an added penalty factor of 100 on the error for regularization to prevent over-fitting. SVMs are more suitable for binary classification, though here we also evaluate multi-class classification using SVM.

3) Random Forest: The Random forest classifiers is made using a pool of 1000 decision trees providing their decision criterion in all different parts of the data-set. The tree splitting criterion was taken as 'gini' rather than 'entropy' criterion as it performed best under hyper-parameter tuning using grid search by 10-fold cross validation.

4) AdaBoost: The boosted ensemble for AdaBoost is built using a collection of 10 Random forest classifiers with each ensemble of Random forest implemented with 1000 decision trees. Accuracy of predictions was optimized by decreasing the weight of each learner by a factor of 0.2 to prevent over-fitting over certain data-set instances

5) GradientBoost: The Gradient Boosting was implemented using several regression trees. A total of 100 different regression trees were used along with a logistic regression as a loss function and an tree splitting criterion of mean squared error. Over-fitting was prevented by decreasing the weight of each learners by a factor of 0.3 which as found by hyper-parameter tuning.

C. Mispronunciation Detection

For our experiments, we consider the samples of the word 'bekku (cat)' for which, the reference word in divided into three segments; 'mallige (jasmine)' for which, the reference word in divided into three segments and, 'kittaḷe (orange)' for which, the reference word is divided into four segments. We map the test audio samples onto the segments of the reference audio for segmentation. We then train different SOMs for different segments of each word. The SOMs are trained on all the audio segments that are rated 5 by the teacher. We trained SOMs on the MFCCs obtained from the audio segments for 1000 iterations. Dimensions of SOMs for the word 'bekku' are 4x4, for the word 'mallige' are 6x6 and for the word 'kittaḷe' are 6x6.

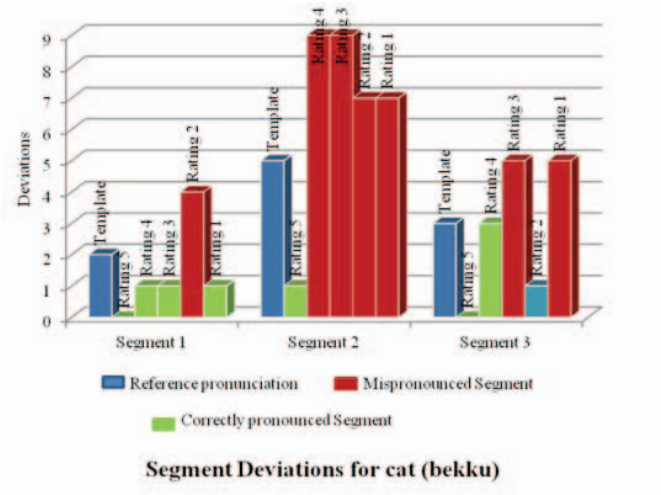
After training, the winner nodes of the template audio segments were used to determine the segment deviations of the other audio files in the training data-set using Manhattan distance. We empirically determined a threshold of 4.5 for 'bekku', 4.8 for 'mallige' and 4.7 for 'kittaḷe' (for each of the winner nodes). The number of deviations of the template segments is employed as a reference to determine the mispronounced segments of other audio files. The threshold for number of deviations for each of the audio segment with respect to template is set at three for orange and at two for cat and jasmine. The audio segments that have a deviation of more than the threshold are considered as mispronounced.

VI. Results and discussion

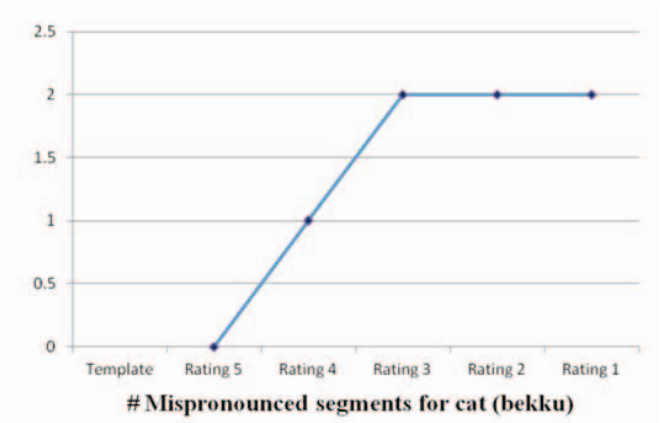
Tables I and II depict the cross validation accuracies obtained by running the models using different combination of features. The accuracies are obtained by performing a 10-fold cross validation by all the models.

A comparison between different models based on above obtained results suggests that by using MFCC DTW features the gradient boosting performs with 80% accuracy for 5-class classification and AdaBoost performs with 95% accuracy for 2-class classification. Since ensemble learners can overfit on the data easily hence the best choice would be artificial neural network with cross validation accuracy of 84% for binary classification. A comparison on using only LSF features leads us to choose Gradient boosting for both 5-class and 2-class decision, performing with 78% and 93% accuracy respectively in both categories. Among all the classifiers the Gradient boosting performs best for 5-class classification and Neural network and AdaBoost perform best for 2-class classification.

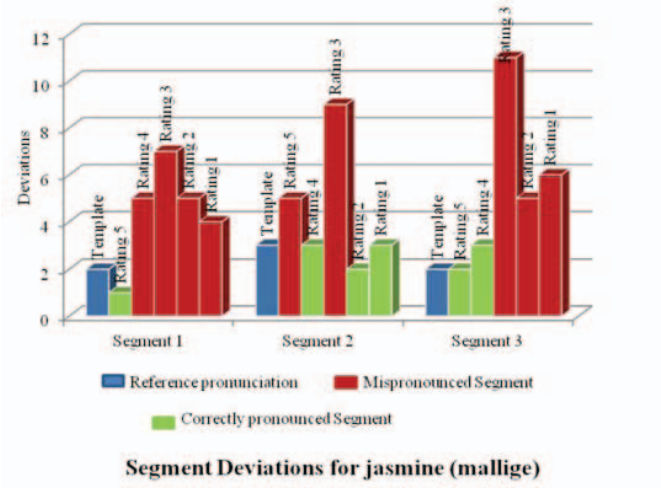
Figure 6 shows the segment deviations for the word 'bekku', 'mallige' and 'kittaḷe' which are divided into three, three and four segments respectively. Figure 6(a), Figure 6(c) and Figure 6(e) depict the correctly pronounced (marked as green) and mispronounced (marked as red) segments. This is determined based on the number of deviations of each segment with respect to the corresponding template (marked as blue) segment. Figure 6(b), Figure 6(d) and Figure 6(f) depict the total number of mispronounced segments of each audio file rated 5, 4, 3, 2 and 1 by the teacher. We can see the clear demarcation between rating 5 and audio files of other ratings - the number of mispronounced segments for rating 5 audio is less than the number of mispronounced segments of audio files with other rating. The number of mispronounced segments for rating 4 audio for all the three words is same as or just one greater than that of corresponding rating 5. The number of mispronounced segments of rating 2 and rating 1 audio are always more than that of corresponding rating 5 and rating 4 audio. We observe an anomaly in case of rating 3 audio (especially for 'mallige' - this may be due the noise present in the audio samples or due to insufficient samples. Also, rating 2 and rating 1 audio files have the same number of mispronounced segments. This may due to the fact that pronunciations in both the audio samples are very bad and human rating is subjective. Figure 7 shows the DTW distance values obtained for the audio samples of 'kittaḷe' using log-cosine distance. Here the distances for rating-4 and rating-3 audio are more than rating-2 and rating-1 audio samples. Hence DTW distances may be ineffective when it comes to determining segment wise deviations of a single word. SOM can be viewed as an effective tool to determine mispronunciations in an isolated word.



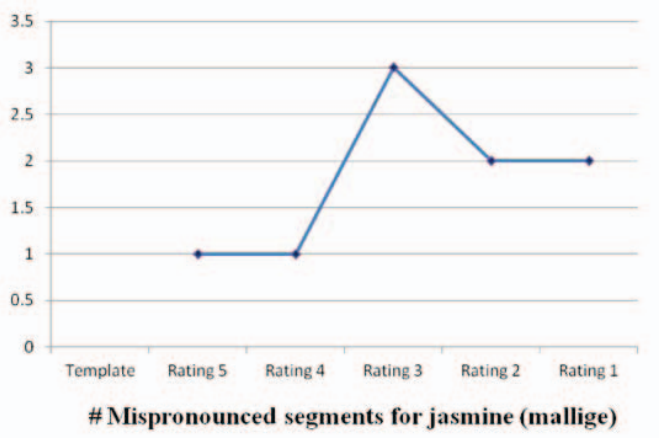
(a)



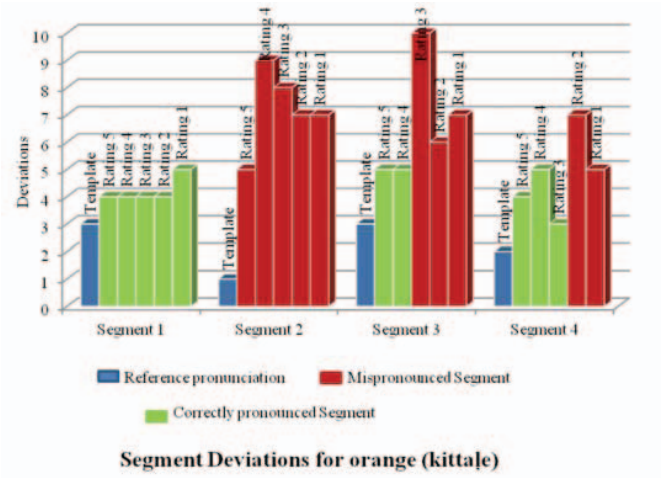
(b)



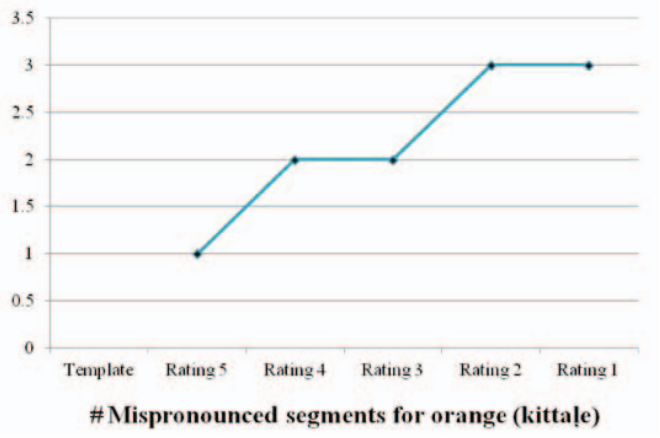
(c)



(d)



(e)



(f)

Fig. 6: Segment Deviations

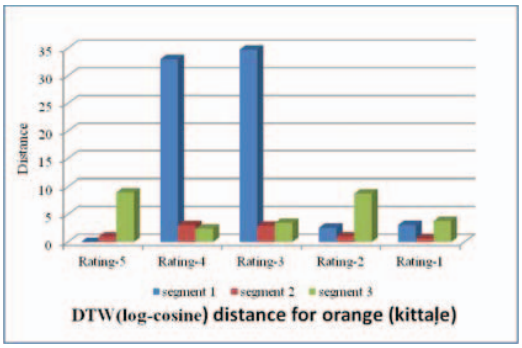


Fig. 7: Cosine Distance for orange

VII. CONCLUSIONS

We propose a pronunciation training framework deployed on the cloud to assist learners (beginners) hone their pronunciation skills at their own pace. Our initial results show that pronunciation training can be achieved through segmentation, supervised learning for pronunciation evaluation and semi-supervised learning for mispronunciation detection with minimal reference audio and human annotation. Currently we employ semi-supervised technique for mispronunciation detection as SOM is trained on 5-rating samples. Our method of pronunciation training does not require a speech recognition system trained on a huge language specific corpus. Other than human annotation (ratings) and canonical speech samples, our implementation does not employ any language specific data. We believe our approach is extendable to other languages. Our application is available for download at the link 'github.com/anandankit95/Kannada-Kali/tree/master/apk'

In future we plan to investigate combination of features to improve classification accuracy. We also plan to collect more data (through YouTube or crowd sourcing) to improve our results. We further plan to explore unsupervised learning in order to achieve a complete objective pronunciation evaluation.

REFERENCES

[1] C. Bhat, K. L. Srinivas, and P. Rao, "Pronunciation scoring for Indian English learners using a phone recognition system," *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia - IITM '10*, pp. 135–139, 2010.

[2] X. Li, J. Chen, M. Yao, D. Shen, and F. Lin, "English sentence pronunciation evaluation using rhythm and intonation," *2014 2nd International Conference on Systems and Informatics, ICSAI 2014*, no. Icsai, pp. 366–371, 2015.

[3] V. Laborde, T. Pellegrini, L. Fontan, J. Mauclair, H. Sahraoui, and J. Farinas, "Pronunciation assessment of Japanese learners of French with GOP scores and phonetic information," *Proceedings of the Annual Conference of ISCA, INTERSPEECH*, vol. 08-12-Sept, pp. 2686–2690, 2016.

[4] W. Hu, Y. Qian, and F. K. Soong, "A new DNN-based high quality pronunciation evaluation for computer-aided language learning (call)," *Proceedings of the Annual Conference of ISCA, INTERSPEECH*, no. August, pp. 1886–1890, 2013.

[5] X. G. Li, S. M. Li, L. R. Jiang, and S. B. Zhang, "A multi-parameter objective evaluation system for English sentence pronunciation," *Proceedings of the 2013 6th International Congress on Image and Signal Processing, CISP 2013*, vol. 3, no. Cisp, pp. 1292–1297, 2013.

[6] J. Lin, Y. Xie, and J. Zhang, "Automatic pronunciation evaluation of non-native Mandarin tone by using multi-level confidence measures," *Proceedings of the Annual Conference of ISCA, INTERSPEECH*, vol. 08-12-Sept, pp. 2666–2670, 2016.

[7] K. Kyriakopoulos, K. M. Knill, and M. J. F. Gales, "A deep learning approach to assessing non-native pronunciation of English using phone distances," no. September, pp. 1626–1630, 2018.

[8] A. Lee, Y. Zhang, and J. Glass, "Mispronunciation detection via dynamic time warping on deep believe network-based posteriorgrams," *ICASSP, IEEE*, pp. 8227–8231, 2013.

[9] A. Al Hindi, M. Alsulaiman, G. Muhammad, and S. Al-Kahtani, "Automatic pronunciation error detection of nonnative Arabic Speech," *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, vol. 2014, pp. 190–197, 2014.

[10] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166, 2015.

[11] Y. C. Hsu, M. H. Yang, H. T. Hung, and B. Chen, "Mispronunciation detection leveraging maximum performance criterion training of acoustic models and decision functions," *Proceedings of the Annual Conference of ISCA, INTERSPEECH*, vol. 08-12-Sept, no. 2, pp. 2646–2650, 2016.

[12] W. Li, N. F. Chen, S. M. Siniscalchi, and C. H. Lee, "Improving mispronunciation detection for non-native learners with multisource information and LSTM-based deep models," *Proceedings of the Annual Conference of ISCA, INTERSPEECH*, vol. 2017-Augus, pp. 2759–2763, 2017.

[13] K. Li and H. Meng, "Mispronunciation Detection and Diagnosis in L2 English Speech Using Multi - Distribution Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 1, pp. 193–207, 2017.

[14] A. Lee and J. Glass, "Mispronunciation Detection without Nonnative Training Data," vol. 1, pp. 643–647, 2015.

[15] A. Lee, N. F. Chen, and J. Glass, "Personalized Mispronunciation Detection And Diagnosis," *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6145–6149, 2016.

[16] R. Duan, T. Kawahara, M. Dantsuji, and J. Zhang, "Effective Articulatory Modeling For Pronunciation Error Detection Of L2 Learner Without Non-native Training Data," *ICASSP, IEEE.*, pp. 5815–5819, 2017.

[17] X. Qian, H. Meng, and F. Soong, "A two-pass framework of mispronunciation detection & diagnosis for computer-aided pronunciation training," *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2015*, no. December, pp. 384–387, 2016.

[18] V. Arora, A. Lahiri, and H. Reetz, "Phonological feature based mispronunciation detection and diagnosis using multi-Task DNNs and Active Learning," *Proceedings of the Annual Conference of ISCA, INTERSPEECH*, vol. 2017-Augus, pp. 1432–1436, 2017.

[19] W. Li, K. Li, S. M. Siniscalchi, N. F. Chen, and C.-h. Lee, "Detecting Mispronunciations of L2 Learners and Providing Corrective Feedback Using Knowledge - guided and Data - driven Decision Trees," pp. 3127–3131, 2016.

[20] A. Lee and J. Glass, "A Comparison Based Approach to Mispronunciation Detection," pp. 1–92, 2012.

[21] T. Kohonen, *Self Organizing Maps_Kohonen*, 3rd ed. Springer, 2001.

[22] J. Kangas, "Phoneme recognition using time-dependent versions of self-organizing maps," *ICASSP, IEEE*, no. July, pp. 101–104, 1991.

[23] G. Kia Eng and A. Manan Ahmad, "Malay Speech Recognition using Self-Organizing Map and Multilayer Perceptron," pp. 233–237, 2005.

[24] P. Somervuo and T. Kohonen, "Self-Organizing Maps and Learning Vector Quantization for Feature Sequences," pp. 151–159, 1999.

[25] H. Ahmed, M. Elaraby, A. M. Mousa, M. Elhosiny, S. Abdou, and M. Rashwan, *An unsupervised speaker clustering technique based on SOM and I-vectors for speech recognition systems*, 2017, vol. 2017.

[26] S. Murthy, A. Anand, A. Kumar, A. Shetty, A. Cholin, A. Bhat, A. Venkatesh, L. Kothiwale, V. Kumar, and D. Sitaram, "Kannada Kali : A Smartphone Application for Evaluating Spoken Kannada Words and Detecting Mispronunciations using Self Organizing Maps," in *Technology for Education (T4E) 2018*, 2018.

[27] P. Liu, K.-w. Yuen, W.-k. Leung, and H. M. Meng, "mENUNCIATE : Development Of A Computer-Aided Pronunciation Training System On A Cross-platform Framework For Mobile , Speech-Enabled Application Development," *Chinese Spoken Language Processing (ISCSLP)*, pp. 170–173, 2012.