Malaya Journal of Matematik, Vol. S, No. 1, 560-565, 2019

https://doi.org/10.26637/MJM0S01/0101

Automatic speech recognition system for utterances in Malayalam language

Rizwana Kallooravi Thandil[1]*, Muhamed Basheer K.P. [2] and Rababa Kareem Kollathodi[3]

Abstract

The paper is discussing the phoneme level interpretation of sound signal to form the words in Malayalam language. Natural language processing with advanced algorithms adopted here to ensure the accuracy and optimization of the processing. Analog to digital conversion of voice signal is taking numerous methodologies for a perfect processing. Training a new system with these techniques to test whether the given commands belongs to dataset or not.

Keywords

Fast Fourier Transform, pre-emphasis, Speech processing, MFCC,HMM, feature extraction, speech corpus.

AMS Subject Classification

90B50.

[1,2,3] *Department of Computer Science, Sullamussalam Science College, Areekode, Kerala-673639, India.*

*__Corresponding author__: [1] ktrizwana@gmail.com; [2]mbasheerkp@gmail.com [3]rababakareem@gmail.com

Article History: Received **24** January **2019**; Accepted **24** May **2019**

Contents

1. Introduction

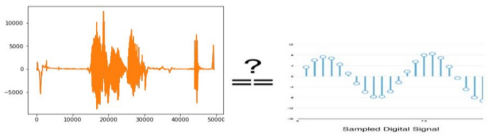
The Speech most basic and one of the natural forms of communication means of human beings. The speech is a form of transformation of sound signals through air. This paper presents an analysis of automatic speech recognition techniques and the advancements of speech recognition in Malayalam language. Speech is three times faster than typing any language particularly English which is very popularly used in entry than on mobile devices. Some works have been put forward on automatic speech recognition in Malayalam language. The speech recognition in Malayalam can be developed to use devices by giving voice commands in the local language and will be so much beneficial to all users and the blind people in particular.

The major challenges are differences in accent, noise and ambiguity which make it harder for engineering. Even though ample researches are being held in this area the accuracy in ASR being one of the major challenges calls for further research. Speech recognition software can be designed in three different ways; a) speaker-dependent b) speaker-independent and c) speaker-adaptive[9].

Recognition of continuous speech is a difficult complex activity and the complexity is attributed to the following properties of continuous speech. First, word boundaries are unclear in continuous speech. Second, co-articulatory effects are much stronger in continuous speech compared to isolated utterances[2].

2. Automatic Speech Recognition System

Speech signal corresponds to the raw audio signal of the words or message being spoken. Automatic speech recognition refers to determining the underlying meaning in the utterance. Speech recognition depends on extracting and modelling the speech dependent characteristics which can effectively distinguish one word from another[10]. The speech recognition system may be viewed as working in a four stages as shown in



3. Methodology

Feature Extraction

The feature extraction process is used to extract the features of the speech signal. This can be implemented by using one of the feature extraction techniques like Mel Frequency Cepstral Coefficients (MFCC). MFCC technique enables the speech features to be extracted for all the speech samples. All these extracted features are then fed to pattern trainer to train the system. The training is performed by Hidden Markov Model(HMM) which is used to create an HMM model for each word. After creating a model for each word a Viterbi decoding shall be used to select the one with maximum likelihood which is nothing but recognized word. Viterbi *decoding* algorithm is used for convolutional codes over noisy digital communication links.

Sampling of signal

The first step in speech recognition is to feed the raw audio waves into a computer.Raw audio is represented as a simple 1D signal. The raw audio which is an analogue signal needs to be transformed in a form that is understood by the computer. To convert sound wave into numbers it is required to record the height of the wave at equally-spaced points.This process is called sampling. For speech recognition a typical sampling rate of 8KHz/16KHz is enough to cover frequency range of human speech.Each sample is quantised typically 8-bit/16-bit.The original sound wave can be reconstructed from the spaced out samples as long as the signal is sampled twice as the highest frequency to be recorded. A 1 second audio clip can be broken down into samples $a_1, a_2, ...a_n$.This can be represented as

1D vector=A=$[a_1, a_2, ...a_n]$

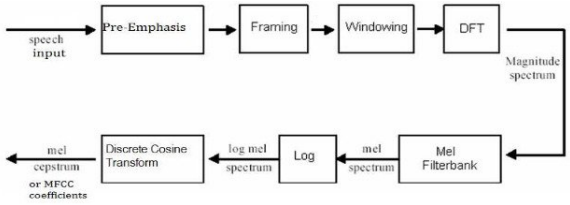


Fig.2 Computational Steps of MFCC

Mel Frequency Ceptral Coeffiients (MFCC)

This algorithm is used to convert the raw audio signal to some kind of parametric representation. MFCC is used to filter the raw audio and extract the unique features of speech samples. It represents the short term power spectrum of human speech. The MFCC technique typically uses two types of filters, namely, linearly spaced filters and logarithmically spaced filters.

Words are defined in almost all languages as the smallest linguistic units that can form a complete utterance by themselves. The minimal parts of words that deliver aspects of meaning to them are called morphemes. Depending on the means of communication, morphemes are spelled out via graphemes—symbols of writing such as letters or characters—or are realized through phonemes, the distinctive units of sound in spoken language.1 It is not always easy to decide and agree on the precise boundaries discriminating words from morphemes and from phrases[3].

Automatic Speech Recognition System usually represents words as sequence of "Phonemes" (units of sound). Phonemes are perceptually distinct units of sound that distinguish speech.

For a phoneme based approach the role of MFCC is to capture the phonetically important characteristics of speech. The signal is expressed in terms of Mel frequency scale. The Mel scale is mainly based on the study of observing the pitch or frequency perceived by the human. The scale is divided into the units mel. The Mel scale is normally a linear mapping below 1000 Hz and logarithmically spaced above 1000 Hz. Equation (3.1) is used to convert the normal frequency to the Mel scale. The formula used is

$$Mel = 2595log10(1 + f/700) \tag{3.1}$$

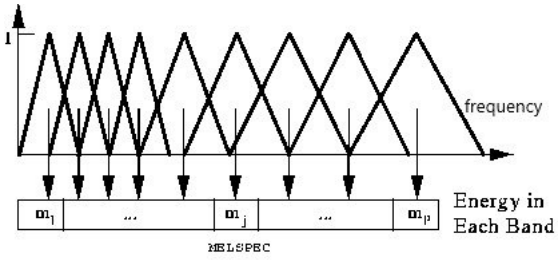
As shown in figure 1, MFCC is composed ofof six computational steps. Each step has its own functionalities and mathematical approaches as discussed briefly in the following:

1. Pre–emphasis

   Pre-emphasis is the first step in MFCC which will boost the amount of energy of signal at higher frequencies. This step processes the passing of signal through a filter which emphasizes higher frequency in the band of frequencies the magnitude of some higher frequencies with respect to magnitude of other lower frequencies in order to improve the overall SNR. This process will increase the energy of signal at higher frequency. [7]

2. Framing

   The process of segmenting the sampled speech samples into a small frames. The speech signal is divided into frames of N samples. Adjacent frames are being separated by $M(M < N)$. Typical values used are $M = 100$ and $N = 256$ (which is equivalent to ˜ 30 m sec windowing



The feature extraction process is implemented using Mel Frequency Cepstral Coefficients (MFCC) in which speech features are extracted for all the speech samples. Then all these features are given to pattern trainer for training and are trained by HMM to create HMM model for each word. Then Viterbi decoding will be used to select the one with maximum likelihood which is nothing but recognized word.

3. Hamming Windowing

   Each individual frame is windowed so as to minimize the signal discontinuities at the beginning and end of each frame. Hamming window is used as window and it integrates all the closest frequency lines. The Hamming window equation is given as: If the window is defined as W (n),$0 \leq n \leq N-1$ where 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:

$$Y(n) = X(n) * W(n) \tag{3.2}$$

$$W(n) = 0.54 - (-0.46)Cos(2n/N-1); 0 < n < N-1 \tag{3.3}$$

4. Fast Fourier Transform

   To convert each frame of N samples from time domain into frequency domain FFT is applied.

5. Mel Filter Bank Processing.

   The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale. The bank of filters according to Mel scale as shown in figure 4 is then performed.

```
from python_speech_features import mfcc
import scipy.io.wavfile as wav
import matplotlib.pyplot as plt
import sounddevice as sd
(rate,sig) = wav.read("wav/16K/train/makeCall/200100101.wav")
plt.plot(sig)
plt.show()
```

   Mel Filter Bank

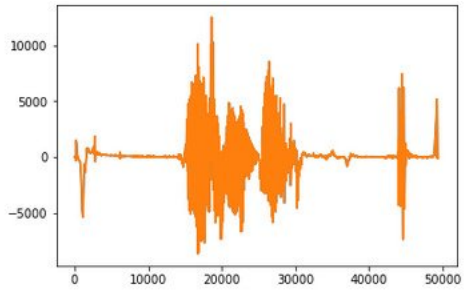
   The triangular filters will compute a weighted sum of filter spectral components so that the output of process approximates to a Mel scale. Each filter's magnitude frequency response is triangular in shape and equal to unity at the centre frequency and decrease linearly to zero at centre frequency of two adjacent filters. Then, each filter output is the sum of its filtered spectral components. The output is mel spectrum consists of output powers of these filters. Then its logarithm is taken and output is logmel spectrum.

6. Discrete Cosine Transform

   This is used to transform the log Mel spectrum into time domain using Discrete Cosine Transform (DCT). The output of this transformation is Mel Frequency Cepstrum Coefficients. The set of coefficient are referred to as acoustic vectors. As a result , each input utterance is transformed into a sequence of acoustic vector[7, 8].

Evaluation

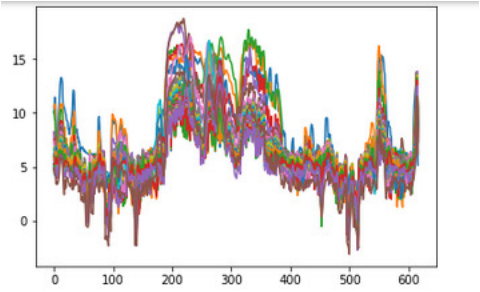
Following is a speech signal corresponding to the sentence "phone vilikkoo".




```
fbank_feat = logfbank(sig,rate)
print(fbank_feat)
plt.plot(fbank_feat)
plt.show()
```

After applying Mel Filter Bank Algorithm

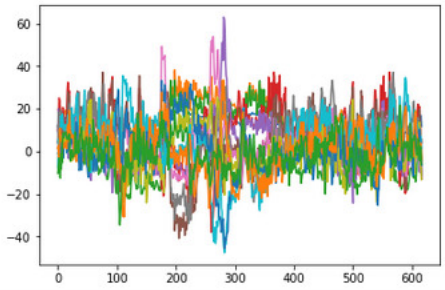
```
[[  9.97891524  10.53457194  10.17106598 ...,   5.24891636   4.82023894
    4.91000944]
 [ 11.45573593  10.82400358   9.40460547 ...,   5.29806525   4.65473475
    4.41053377]
 [ 11.33228295  10.74308033   9.36416145 ...,   5.32870205   4.70667916
    4.30079141]
 ...,
 [ 13.73197919  13.18153006  13.25055402 ...,  13.0168229   12.49503848
   11.50807314]
 [ 13.27929795  13.44698377  13.1506627  ...,  13.02117911  12.49983632
   11.51303473]
 [  5.16385969   7.4235737    7.16607461 ...,   7.78014394   7.43787245
    6.19004036]]
```




```
mfcc_feat=mfcc(sig,rate)
print(mfcc_feat)
plt.plot(mfcc_feat)
plt.show()
```

After applying MFCC Algorithm

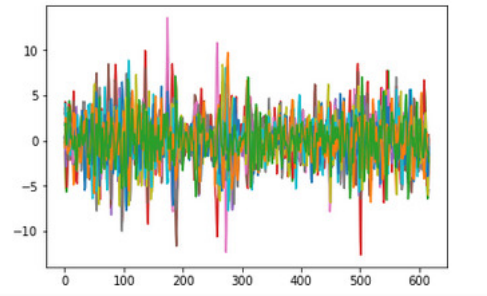
```
[[ 11.63800093  19.5883892   12.12007157 ...,  -0.9299861    1.56789163
  -10.33510269]
 [ 12.86059787  19.43498177  12.01193945 ...,  -0.42464516   2.70106031
   -6.86815176]
 [ 13.06206545  13.25253908  11.62713262 ...,  -0.15487113  -2.93420202
   -3.3016054 ]
 ...,
 [ 16.79890418   2.65988654  -7.00612406 ...,  -3.64094059   3.94835811
    2.13762253]
 [ 16.70081699   2.45704018  -7.30495928 ...,  -4.84228931   2.68248008
    0.84693726]
 [ 11.7343331   -4.37394458 -11.78899603 ...,  -6.47018089  -3.73351849
   -6.18763732]]
```



```
d_mfcc_feat = delta(mfcc_feat, 2)
print(d_mfcc_feat)
plt.plot(d_mfcc_feat)
plt.show()
```

After applying Delta

```
[[ 0.4070726  -1.28251077 -0.109401   ...,  0.20553984 -0.78710186
   1.75339455]
 [ 0.15372504 -3.44702445 -1.74155424 ...,  1.24431737  1.19554877
   1.92404777]
 [-0.52968495 -5.35169451 -4.07589131 ...,  1.65416177  2.54016123
  -0.17592227]
 ...,
 [-0.48991886 -4.78135409 -6.44343467 ..., -3.95586781 -0.85076277
  -1.30235087]
 [-1.24990343 -3.88760913 -5.05297641 ..., -3.64523752 -1.52648822
  -1.21029158]
 [-1.50956261 -2.0898647  -1.40497807 ..., -0.72863722 -2.17797518
  -2.36850943]]
```



4. Pattern Training using Hidden Markov Models (HMM)

The basic idea for speech recognition is to find most likely string of words given some acoustic input.

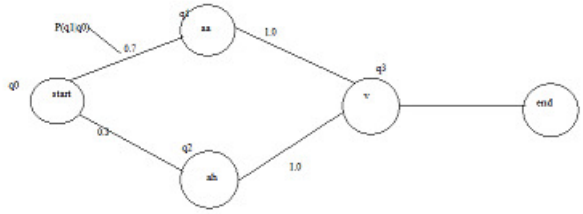
The likelihood of a word in a language can be found by $arg\,max_{w\in L}P(w/y)$ where w is the set of words and L is the language and y is the set of acoustic vectors received from the front end processor. There will be an n-dimensional vector in the front end that contains 10-20 ms of speech.

The basic equation for speech recognition is Bayes rule

$$arg\,max_{w\in L}P(w/y) = arg\,max_{w\in L}\frac{P(y/w)P(w)}{P(y)}$$

For a single speech input y will be a constant value and so is P(y).So the challenge is to find $arg\,max_{w\in L}P(y/w)P(w)$. Here $P(y/w)$ is the likelihood of the model that corresponds to the prior probability of the acoustics given the word uttered and $P(w)$ is the prior probability of the word string. Many acoustic vectors make up a word and we use the intermediate word representation which we call the phonemes.

The word model for a particular phoneme in Malayalam language 'aa' can be modelled using finite automata as follows.



HMM being a sequence model incorporates the integration of local probability values over sequence. HMMs can be constructed to recognize the isolated words. If the vocabulary contains some n words then each word should be modelled by a distinct HMM. For each word in the vocabulary there should be a training set of K occurrences of each spoken word. Several observation sequences shall be considered.This is how we train the system using the HMM model. The HMM model models each word the words that are uttered.This can be used to assign a label or class to each unit in a sequence, thus mapping a sequence of observations to a sequence of labels. HMM is a probabilistic sequence model. The model compute the probability distribution of the labels or classes and choose the best label sequence over possible sequences of labels given a sequence of units of speech. It can be words, letters, phonemes, sentences etc. HMM involves a sophisticated mathematical framework for building automatic speech recognition systems. That is for each word in the vocabulary an HMM shall be built by estimating the model parameters that optimize the likelihood of the training observation vectors from the appropriate word.

The HMM model incorporates a powerful learning method which makes it appropriate for training purposes. It also involves decoding methods for the word sequences, good sequence handling capabilities and a rich mathematical framework with flexible topology. HMMs make the assumption that adjacent feature vectors are statistically independent and identically distributed which is incorrect for human speech process. HMM training algorithms are based on likelihood maximization, which assumes correctness of the models (which is known not to be true) and implies poor discrimination [4].

To define the sequence labelling of the speech, first it requires to introduce the Markov chain,also referred to as the observed Markov model. Markov chainsand hidden Markov models are both extensions of the finite automata.

We can build an HMM for each word using the associated training set. Let λw denote the HMM parameters associated with the word w, when presented with a sequence of observations sigma,choose the word with the most likely model, i.e., $W^* = \arg\max_{\{winW\}} \Pr(\delta \mid \lambda_w)$

There are basically three problems while building HM Model. They are:

(1)Find Pr (sigma | λ): the probability of the observations given the model.

(2)Find the most likely state trajectory given the model and observations.

(3)Adjust $\lambda = \{A, B, pi\}$ to maximize Pr (sigma | λ). [5]

HMMs are simple networks that can generate speech (sequences of cepstral vectors) using a number of states for each model and modeling the short-term spectra associated with each state with, usually, mixtures of multivariate Gaussian distributions (the state output distributions). The parameters of the model are the state transition probabilities and the means, variances and mixture weights that characterize the state output distributions [10]. This uses theory from statistics to (sort of) arrange the feature vectors into a Markov matrix (chains) that stores probabilities of state transitions. That is, if each of our code words were to represent some state, the HMM would follow the sequence of state changes and build a model that includes the probabilities of each state progressing to another state. HMM can be characterized by following when its observations are discrete:

1. N is number of states in given model, these states are hidden in model.
2. M is the number of distinct observation symbols correspond to the physical output of the certain model.
3. A is a state transition probability distribution defined by NxN matrix as shown below.

$$A = \{a_{ij}\}$$
$$a_{ij} = p\{q_{t+1} = j/q_t = i\}, 1 \leqslant i, j \leqslant N_n$$
$$\sum a_{ij} = 1, \qquad 1 \leqslant i, j \leqslant N_n$$

Where q_t occupies the current state. Transition probabilities should meet the stochastic limitations .

B is observational symbol probability distribution matrix (3.3) defined by NxM matrix equation comprises.

$$b_j(k) = p\{o_t = v_k \mid q_t = j\}, 1 \leqslant j \leqslant N, \quad 1 \leqslant k \leqslant M$$
$$\sum b_j(k) = 1, \qquad 1 \leqslant k \leqslant M$$

Where v_k represents the k^{th} observation symbol in the alphabet, and o_t the current parameter vector. It must follow the stochastic limitations

Π is an initial state distribution matrix defined by Nx1.

$$\Pi = \{\pi_i\}$$
$$\pi_i = p\{q_1 = i\} \qquad 1 \leqslant i \leqslant N$$

By defining the N, M, A, B, and π, HMM can give the observation sequence for entire model. as $\lambda = (A, B, \pi)$ which specify the complete parameter set of model[1] .

5. Pattern Matching and Decision Making

The forward backward estimation algorithm is used to train its parameters and to find log likelihood of voice sample. It is used to estimate the unidentified parameters of HMM. It is used to compute the maximum likelihoods and posterior mode estimate for the parameters for HMM in training process. The Viterbi algorithm takes model parameters and the observational vectors of the word as input and returns the value of matching with all particular word models.

Steps involved:

1. Input the speech signal
2. Feature analysis and vector quantisation (Using MFCC)
3. Train the system using HMM model
4. Perform the statistical probabilistic computation
5. Select the model with maximum probability

6. Conclusion

The Mel Frequency Cepstral Coefficient (MFCC) method is studied here for extracting the features of speech signal. The pre-processing and feature extraction stages of a pattern recognition system serves as an interface between the real world and a classifier operating on an idealised model of reality. Then HMM is used to train these features into the HMM parameters and used to find the log likelihood of entire speech samples. HMM is a dominant approach in most state-of-the-art speaker-independent, continuous speech recognition systems.

In recognition this likelihood is used to recognize the spoken word. Speech recognition systems till date are not a cent percentage accurate even though lots of research works are being going on in the field of speech recognition. The systems developed so far have limitations: The current systems are limited to number of vocabularies and future works are necessary towards expanding this vocabulary. ASR techniques can be extended to speech recognition in Malayalam language. Since Malayalam language has several accents and a dialect that varies from region to region it poses a great challenge in this area of research.

References

[1] Ms. Rupali S Chavan1, Dr. Ganesh. S Sable, *An Overview of Speech Recognition Using HMM*, IJCSMC,26(2013).

[2] Anuj Mohameda, K.N. Ramachandran Nairb, *HMM/ANN hybrid model for continuous Malayalam speech Recognition*, International Conference on Communication Technology and System Design,2011.

[3] Daniel M.Bikel and Imed Zitouni, *Multilingual Natural language processing applications From theory to practice*.

[4] Anuj Mohameda, K.N. Ramachandran Nairb, *HMM/ANN hybrid model for continuous Malayalam speech Recognition*, Science Direct, 30(2012), 616-622.

[5] Hemakumar G., *A Survey on Indian Languages*, International Journal of Information Science and Intelligent System,2(2013)

[6] Daniel Jurafsky, James H. Martin, *Hidden Markov Model Speech and Language Processing*, 2017.

[7] Ahmad A. M. Abushariah, Teddy S. Gunawan, Othman O.Khalifa, *English Digits Speech Recognition System Based on Hidden Markov Models*, International Conference on Computer and Communication Engineering (ICCCE) (2010).

[8] Anjali Bala, Abhijeet Kumar, Nidhika Birla, *Voice command recognition System Based on MFCC and DTW*, International Journal of Engineering Science and Technology,2(2010).

[9] Hemakumar G. and Punitha P., Large Vocabulary Speech Recognition: *Speaker Dependent and Speaker Independent*, Advances in Intelligent and Soft Computing,339(2015),73-80.

[10] Hemakumar G. and Punitha P., *Automatic Segmentation of Kannada Speech Signal into Syllables and Sub-words: Noised and Noiseless Signals*,International Journal of Scientific & Engineering Research, 5(2014),1707-1711.

⋆⋆⋆⋆⋆⋆⋆⋆⋆
ISSN(P):2319 – 3786
Malaya Journal of Matematik
ISSN(O):2321 – 5666
⋆⋆⋆⋆⋆⋆⋆⋆⋆